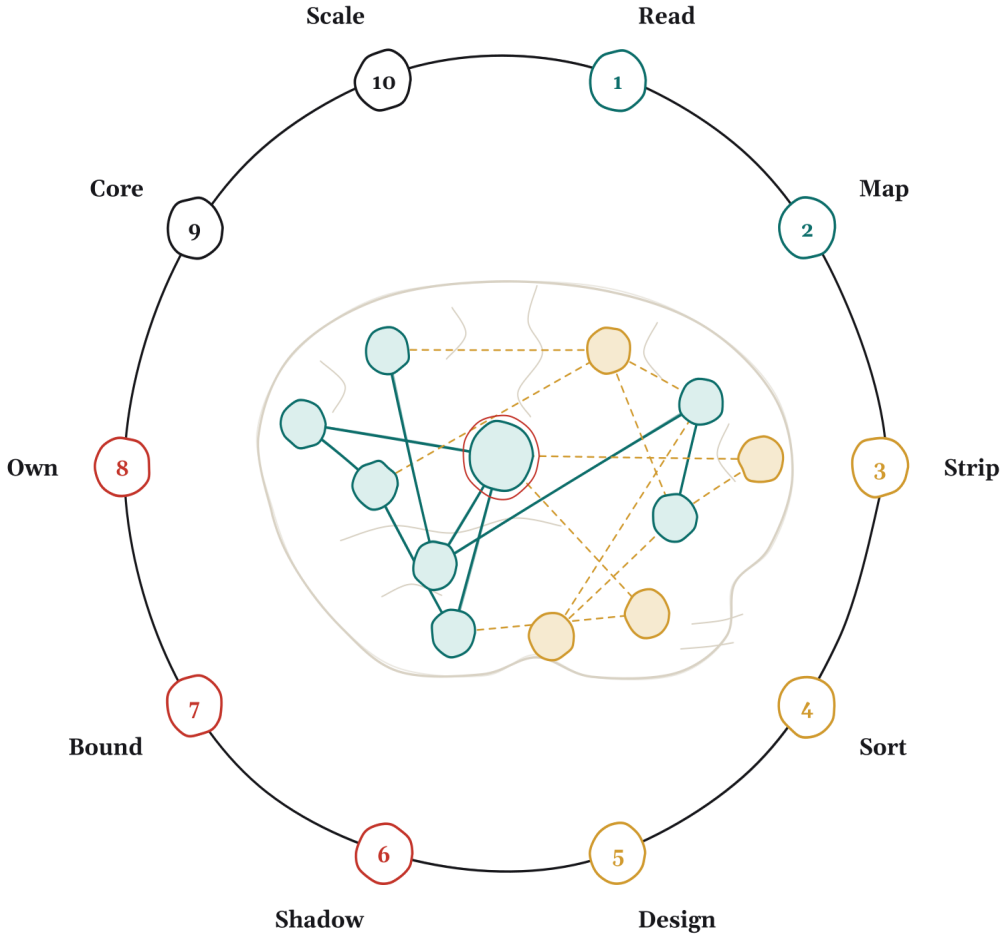
The New Business Engineer

From the Laws of Thought to Closed Loops of Action

The Complete Discipline for Embedding AI Into the Enterprise Core



> Read the distribution. Write the rules.
> Close the loop.

Gennaro Cuofano

The New Business Engineer

From the Laws of Thought to Closed Loops of Action: The Complete Discipline for Embedding AI Into the Enterprise Core

The New Business Engineer
From the Laws of Thought to Closed Loops of Action: The Complete Discipline for Embedding AI Into the Enterprise Core

The Business Engineer Library · The Embedding Series, collected edition

Gennaro Cuofano
businessengineer.ai

First published 2026. Fourth edition. This volume brings together The Business Engineering Body of Knowledge, The Work Loop Canvas, The Embedding Lifecycle and The Work Loop Library, with the Mental Models Library, ordered so that the discipline cascades from its foundations. The composite insurer and manufacturer in this book are illustrative. Their figures are authored to teach the method, set within published ranges, and not drawn from any company. Sources are listed in Notes and Sources.

Figures drawn for this volume in the Library's ink register.

Contents

The Second Hard Problem 10

Part One: Foundations **13**

I. The Laws of Thought 14

II. The Limits of Rules 17

III. Strange Loops and Levels 21

IV. The Neural Turn 25

V. Neurosymbolic AI 28

VI. Closed Loops 32

VII. Why This Matters for the Enterprise 36

Part Two: The Core **39**

VIII. What Business Engineering Is 40

IX. Reading the Distribution 44

X. How Distributions Are Made 47

XI. Abstraction 50

XII. Compression 53

XIII. Mental Models 56

XIV. Testing and Judgement 59

XV. The Closed Loop of Business Engineering 62

XVI. VTDF: Reading the Business Model 66

XVII. Where the Work Loop Plugs In 70

Part Three: Reading the Business **73**

XVIII. The Shape of This Body of Knowledge 74

XIX. Distributions 78

XX. Value and the Map 85

XXI. Work and Process 92

Part Four: Designing the Loop 100

XXII. Rules and Decisions 101

XXIII. The Executor 108

XXIV. Control and Acceptance 115

Part Five: Owning the Result 124

XXV. People and Accountability 125

XXVI. Economics 133

XXVII. Governance 139

XXVIII. How the Discipline Cascades 145

Part Six: A Century of Designing Work 149

XXIX. The Line 150

XXX. The Divisions 153

XXXI. The Stop Cord 155

XXXII. Obliterate, Then Encode 158

XXXIII. The Business Model on One Page 161

XXXIV. The Bots 163

XXXV. From Business Model to Business Engineering 165

XXXVI. What Each Era Left Behind 168

Part Seven: The Inside-Out Problem 171

XXXVII. Eleven Days 172

XXXVIII. Outside-In, Inside-Out 175

XXXIX. Why AI Stalls at the Edge of the Core 177

XL. The Unit Is the Work Loop 180

XLI. Shaped Like the Organization's Brain 182

Part Eight: The Work Loop Canvas 185

XLII. Outcome 186
XLIII. Compensations 189
XLIV. Rules 192
XLV. Context 195
XLVI. Actions 197
XLVII. Decisions 199
XLVIII. Junctions 202
XLIX. Exceptions 204
L. Handoffs 207
LI. Standard 209
LII. People 212
LIII. Economics 214
LIV. The Contested Decision 217
LV. The Work Loop Map 219
LVI. The Scale Gates 221
LVII. Running the Session 224
LVIII. What the Canvas Leaves Out 227

Part Nine: The Embedding Lifecycle 229

LIX. The Middle 230
LX. What Embedding Means 233
LXI. The Executor That Decides 237
LXII. What the Executor Is Made Of 240
LXIII. Lifecycles Before This One 246
LXIV. The Lifecycle at a Glance 251
LXV. Why the Lifecycle Has This Shape 255
LXVI. Before Step One 260
LXVII. Step 1: Read 263

LXVIII. Step 2: Map 269
LXIX. Step 3: Strip 275
LXX. Step 4: Sort 281
LXXI. Step 5: Design 288
LXXII. Step 6: Shadow 294
LXXIII. Step 7: Bound 299
LXXIV. Step 8: Own 304
LXXV. Step 9: Core 310
LXXVI. Step 10: Scale 316
LXXVII. How Lifecycles Break 322
LXXVIII. Adapting the Lifecycle 329
LXXIX. Running Many Loops at Once 334
LXXX. Questions Leaders Ask 338
LXXXI. What Finishing Is Worth 343

Part Ten: The Work Loop Library **347**
LXXXII. A Library of Work 348
LXXXIII. The Template Kit 352
LXXXIV. The Lead-to-Cash Chain 366
LXXXV. The Loops of Lead to Cash 369
LXXXVI. Collections and Cash Application 381
LXXXVII. The Procure-to-Pay Chain 392
LXXXVIII. The Loops of Procure to Pay 395
LXXXIX. Purchase Requisition and Approval 407
XC. The Hire-to-Retire Chain 419
XCI. The Loops of Hire to Retire 422
XCII. Candidate Screening 435
XCIII. The Record-to-Report Chain 447

XCIV. The Loops of Record to Report 450

XCV. Account Reconciliation 461

XCVI. The Issue-to-Resolution Chain 473

XCVII. The Loops of Issue to Resolution 476

XCVIII. Support Ticket Resolution 487

XCIX. The Order-to-Delivery Chain 499

C. The Loops of Order to Delivery 502

CI. Shipment Exception Handling 512

CII. What Thirty-Nine Loops Have in Common 524

Part Eleven: The Mental Models Library 530

CIII. How the Models Library Works 531

CIV. Models for Every Loop 533

CV. Heavy-Tailed Loops 543

CVI. High-Volume Routine Loops 551

CVII. Loops That Decide About People 559

CVIII. Regulated and Audited Loops 567

CIX. Customer-Facing Loops 575

CX. Loops That Depend on Outside Parties 582

CXI. Loops Whose Steps Are Generated 589

CXII. Loops in a Chain 597

Part Twelve: Across the Discipline 604

CXIII. Notation 605

CXIV. The Measures 610

CXV. The Competencies 613

CXVI. The Evidence Base, as Read in 2026 617

The Cycle Closes 620

Epilogue: The New Business Engineer 622

Appendix A: The Lifecycle on One Page 624

Appendix B: Who Signs What 625

Appendix C: Entry and Exit Checklists 626

Appendix D: Foundational Reading 628

Appendix E: Areas Against Steps 629

Appendix F: The Blank Canvas 630

Appendix G: The Insurer's Claims Loop, Filled 632

Appendix H: The Manufacturer's Invoice Loop in Numbers 635

Appendix I: One Loop Through Every Area 639

Appendix J: The Thirty-Nine Reference Loops 640

Appendix K: The Kit as Files 645

Appendix L: The Lifecycle and Governance Frameworks 646

Appendix M: A Facilitator's Script 647

Appendix N: Reading Map 648

Appendix O: The Thirty-Nine Loops by Type 649

The Five Moves 652

The Meters 653

The Mental Models 654

Glossary 657

Notes and Sources 662

The Thread 675

Also in This Library 676

The Second Hard Problem

In the 2000s, the hard problem for most companies was the business model. The web had made new ways of creating and capturing value possible, and old ones fragile. Companies needed a way to see their whole model, argue about it and change it. They got one: a single page with nine blocks, which became one of the most used management tools of its time.

The hard problem has moved. It is now inside the company. AI creates value by doing part of the work itself: reading the invoice, applying the rule, deciding the case, writing the record. To let it do that, a company has to understand its own work far more precisely than it ever needed to. That means its loops, its rules, its decisions, its limits, its costs and who answers for each. Most companies do not. Their rules are scattered across policies, systems and people's heads, and their processes are layered with workarounds for constraints that disappeared years ago. Some of their most important decisions were never agreed at all.

That is why so many AI programmes stall between a pilot that works and a process the business relies on. The models are not the bottleneck. The bottleneck is the company's knowledge of itself, and the absence of a discipline for turning that knowledge into designs that people sign.

This book sets out that discipline. It calls the practice business engineering, and the person who practises it the business engineer. It is built from four volumes, each of which can be read alone, brought together here and connected into one account.

Why it starts with the laws of thought

The machines now entering companies are the latest step in a very old project: writing down how thinking works, so that something other than a person can do it. That project has had two traditions. One captured thought in explicit rules and symbols, from Boole's algebra to the expert systems of the 1980s. The other let machines learn from examples, from the first artificial neurons to today's language models. Each succeeded where the other failed. The most capable systems now combine them, and researchers call the combination neurosymbolic AI.

A company that embeds AI into its core is building such a combination, whether it knows it or not. The model brings the learned half. The company

has to bring the other half: its rules, its decisions, its limits, its standards and the loops that check results against the world. **That half is the subject of this book.** So the book begins with the foundations, and everything else follows from them.

How the book is built

Part One, Foundations, traces the two traditions from the laws of thought, through Gödel and the strange loops of *Gödel, Escher, Bach*, to neurosymbolic AI. It adds the science of closed loops, which connects any of it to real action. Part Two, The Core, sets out the method of business engineering. It covers reading distributions, how distributions are made, abstraction, compression, mental models, testing and judgement, and the closed loop that joins them. It also shows how the method reads a business from the outside.

Parts Three to Five are the body of knowledge. They define every concept, measure and instrument of the discipline once, across nine areas, and end with how the rest of the discipline cascades from them.

The cascade comes next. Part Six looks back over a century of designing work, from the moving assembly line to software robots, and finds the idea each era left behind. Part Seven explains why AI is an inside-out technology and why it stalls at the edge of the core. Part Eight introduces the instrument: the Work Loop Canvas, which maps one unit of work on one page in twelve connected nodes, and the Work Loop Map, which places all of a company′s loops. Its worked example is a composite insurer′s claims loop.

Part Nine describes the sequence: ten steps that take a loop from the first reading of value to an executor running in the core, with what is signed at each step and by whom, and why each step works. Part Ten is the Work Loop Library: thirty-nine reference loops that almost every company runs, across six chains, with six of them taken through the full lifecycle and every template filled in. Part Eleven is the Mental Models Library: the models a practitioner needs, organised by the type of loop in which each matters most. Part Twelve gathers what runs across the whole discipline: the notation, the exact measures, the competencies of practitioners and the state of the evidence.

The book can be read straight through. It is also built to be consulted. The appendices hold the lifecycle on one page, the signatures step by step, the checklists, the foundational reading, the catalogue of thirty-nine loops, the loops sorted by type, and the glossary.

How the drawings work

After almost every section, a small drawing turns its idea into an everyday object: a thermostat, a sieve, a canal lock, a kite and its string. Each carries a caption that says what maps to what. Read them as a second path through the book. A reader who looks only at the drawings should still follow the argument.

Who it is for

It is for the people who must get AI into the core of a company and answer for the result. That includes executives who sponsor the change, owners of the processes being changed, and finance, risk and operations leaders who must agree to it. Above all, it is for the practitioners who will carry the work from step to step. No technical background is assumed. Every term is explained before it is used.

A composite manufacturer of industrial pumps and valves provides most of the examples, from the body of knowledge through the reference processes. The canvas chapters use a composite insurer. Both companies and their figures are illustrative. They are set within published ranges, which the notes identify, so that the arithmetic teaches something real.

PART ONE

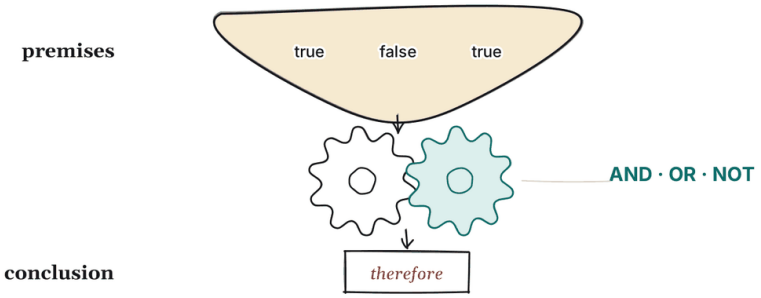
Foundations

I

The Laws of Thought

In 1854, a self-taught mathematician in Cork named George Boole published a book with an ambitious title: *An Investigation of the Laws of Thought*. His aim was to show that reasoning itself could be written as algebra. Statements could be treated as variables that were either true or false, and combined with operations such as "and", "or" and "not". If the rules of combination were right, correct conclusions would follow mechanically from correct premises, with no judgement required.

Boole was continuing a very old line of work. Aristotle had catalogued the valid forms of argument more than two thousand years earlier. In the seventeenth century, Gottfried Leibniz imagined a universal language of concepts and a "calculus of reasoning". With them, he thought, disputes could be settled by calculation instead of argument: "let us calculate". Boole made a large part of that dream concrete.

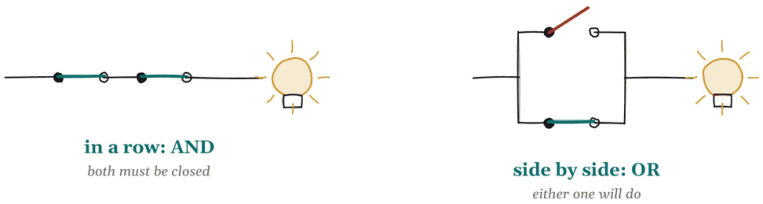


Like a mill: *pour in true premises, turn the handle of the rules, and correct conclusions come out, with no judgement needed along the way.*

From algebra to machines

Boole's algebra waited eighty years for its most consequential use. In 1937, a graduate student at the Massachusetts Institute of Technology named Claude Shannon noticed that electrical relays behaved like Boole's variables. A relay was open or closed, and relays wired in series or in parallel computed "and" and "or". His master's thesis, *A Symbolic Analysis of Relay and Switching Circuits*, showed that any logical expression could be built as a circuit. Every digital computer since rests on that observation. Boole's laws of thought became the laws of the machine.

For the next several decades, the dominant idea of intelligence in computing followed directly. To make a machine think, write down the symbols and the rules, and let it calculate. In 1956, a workshop at Dartmouth College launched the field of artificial intelligence on largely this premise. Its proposal stated that every feature of intelligence could in principle be described so precisely that a machine could simulate it.

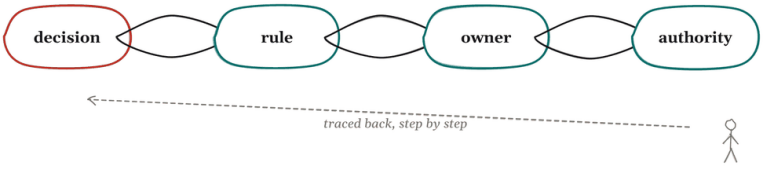


Like a hallway light with two switches: *wire them one after the other and the bulb needs both; wire them side by side and either will do.*

What the symbolic tradition got right

The symbolic tradition gave us things no business could now do without. Rules written explicitly can be read, checked, audited and changed. A rule can say exactly why a decision was taken. Two people reading the same rule should reach the same answer. Logic guarantees that if the premises hold, the conclusion holds. Accounting systems, tax engines, eligibility checks, approval matrices and every workflow configured in enterprise software are this tradition at work.

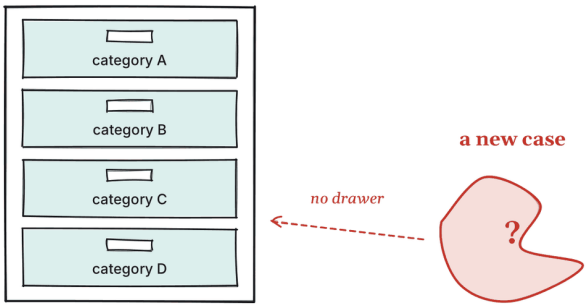
Its idea of reliability is the one auditors, regulators and boards still use. A decision is trustworthy if it can be traced to a rule, the rule to an owner, and the owner to an authority. **The registers, standards and gates of this book are the symbolic tradition applied to a company's own work.**



Like a chain of custody: *every decision links to a rule, every rule to an owner, every owner to an authority, and anyone can follow the chain back.*

What it assumed

The symbolic tradition made a quiet assumption: that the world could be described completely in advance. Every case would fit a category, every category would have a rule, and every rule would be consistent with the others. In mathematics, that assumption was about to be tested to destruction. In business, it had never been true at all.



Like a filing cabinet built in advance: *it works perfectly until a case arrives that fits no drawer.*

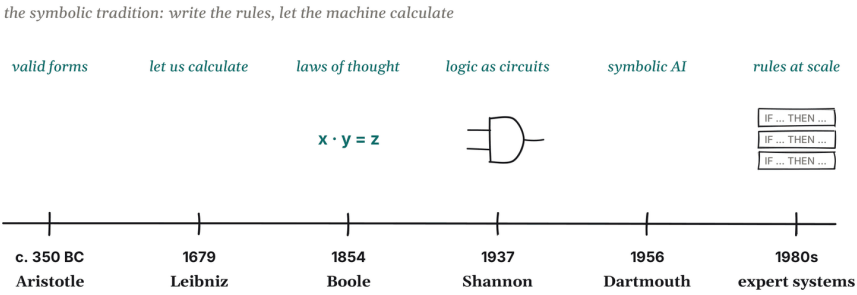


FIGURE 1 The laws of thought. From Aristotle's syllogisms to Boole's algebra to Shannon's relay circuits: reasoning written as rules, then built into machines.

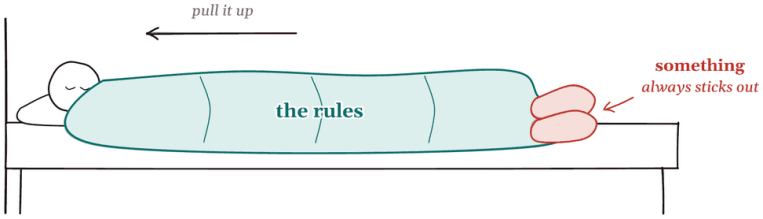
II

The Limits of Rules

At the start of the twentieth century, the mathematician David Hilbert set out a programme to put all of mathematics on a complete and consistent set of rules. Every true statement would be provable from the axioms, and no contradiction could ever be derived. It was the symbolic dream at its most ambitious.

In 1931, a twenty-five-year-old logician named Kurt Gödel showed that it could not be done. Any consistent formal system powerful enough to express ordinary arithmetic contains true statements that the system cannot prove. Adding the missing statement as a new rule does not help: the larger system has its own unprovable truths. Five years later, Alan Turing showed a related limit for machines. There is no general procedure that can decide, for every program, whether it will ever finish running.

Gödel's result is about mathematics, and it should not be stretched into a slogan about everything. But it carries a lesson that applies by analogy to any attempt to capture a complex domain in rules. A rule system is a map. However detailed it becomes, it can be complete only for a world simpler than the one it describes.



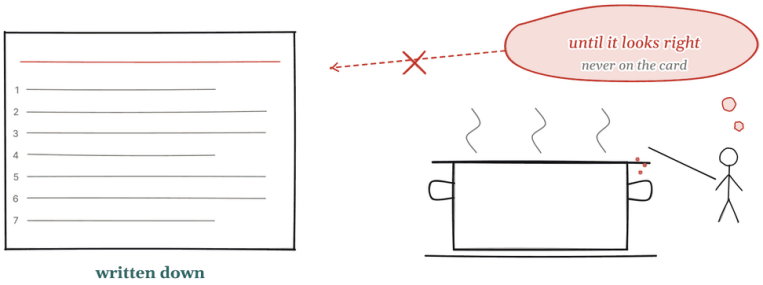
Like a blanket that is too short: *pull it up to cover the shoulders and the feet stick out, so every rulebook leaves some truth uncovered.*

The same limit, in business

Companies met a practical version of this limit long before they met AI. In the 1970s and 1980s, expert systems tried to capture specialist knowledge as thousands of if-then rules. MYCIN diagnosed blood infections at Stanford. XCON configured computer orders at Digital Equipment Corporation, and for a time saved it millions of dollars a year. These systems worked well inside

their domains and failed abruptly at the edges. They also became very expensive to maintain as rules multiplied and began to conflict.

Edward Feigenbaum, one of the field's founders, named the central difficulty the knowledge acquisition bottleneck. Experts could not simply state what they knew. Much of their knowledge was tacit. The philosopher Michael Polanyi had put it in a single sentence in 1966: "we can know more than we can tell".



Like a family recipe card: *it lists every step, but the cook's sense of when it looks right never made it onto the card.*

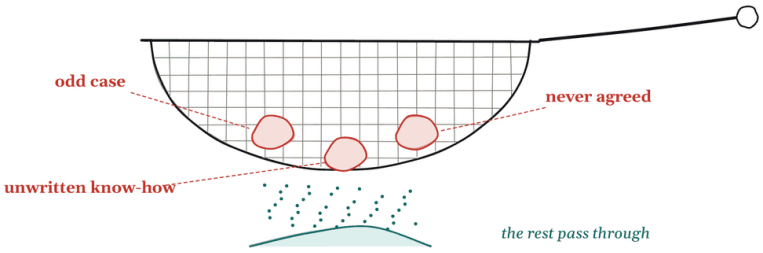
Three limits a business engineer meets

Every embedding programme meets three forms of the same limit.

The first is the tail. However many rules a loop has, some cases fall outside all of them: the new supplier with an unusual tax treatment, the claim that fits two categories, the customer whose history changes the answer. Rules handle the body of a distribution well and the tail badly, because the tail is where the world is richer than the map.

The second is tacit knowledge. Many of the rules a loop really follows live in experienced people's heads, and they cannot all be written down. Some can be drawn out by asking, by walking real cases and by watching what people do. Others show up only as judgement.

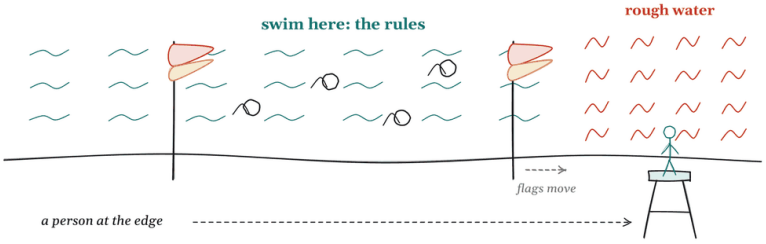
The third is the contested decision. Some questions cannot be settled inside the rule system at all, because the organisation never agreed on the answer. No amount of further rule-writing resolves them. Someone has to step outside the system and decide. In Gödel's terms, the system has to be extended from outside, and only a person with authority can do that.



Like a kitchen sieve: *most cases pass straight through the rules, but three kinds stay behind, the odd case, the unwritten know-how and the question nobody settled.*

What follows for design

The limits of rules are not an argument against rules. They are an argument for knowing where rules stop. A well-designed loop writes down every rule it can and sorts every decision. It also says plainly what happens at the edges: which cases go to people, who decides what was never agreed, and how the rules are extended when the world shows them a new case. Much of this book is about drawing that boundary well.



Like a beach with flags: *swim freely between them, a lifeguard is posted where the safe water ends, and the flags move when the sea changes.*

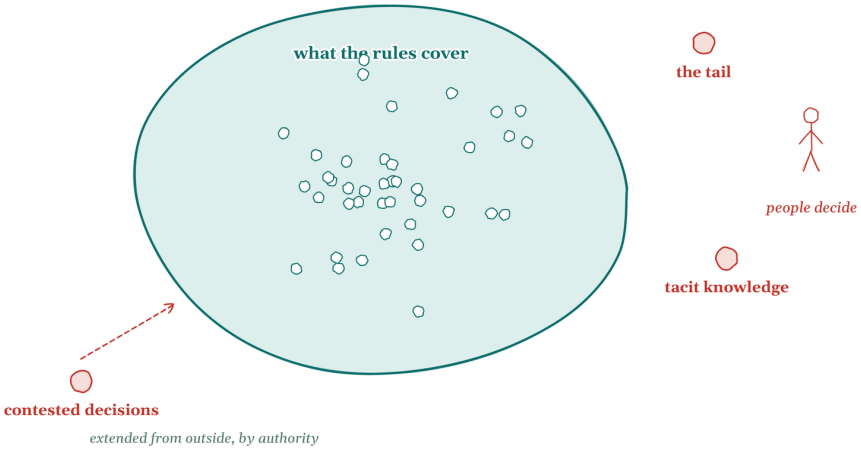
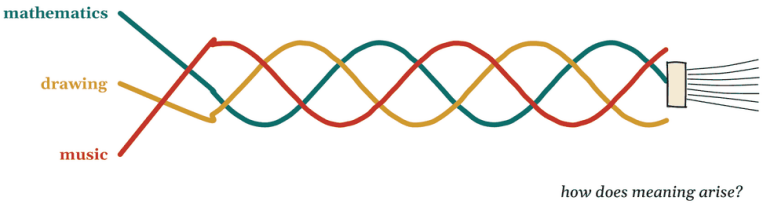


FIGURE 2 The limits of rules. A rule system covers the body of cases well; the tail, tacit knowledge and contested decisions lie outside it, where people must decide and the rules must be extended from outside.

III

Strange Loops and Levels

In 1979, the cognitive scientist Douglas Hofstadter published *Gödel, Escher, Bach: An Eternal Golden Braid*. It won the Pulitzer Prize and became one of the most influential books ever written about minds and machines. It braids together the mathematics of Gödel, the drawings of M. C. Escher and the music of Johann Sebastian Bach. Through them it asks how meaning and selves can arise from systems of meaningless symbols. Four of its ideas are directly useful to anyone designing work for machines.

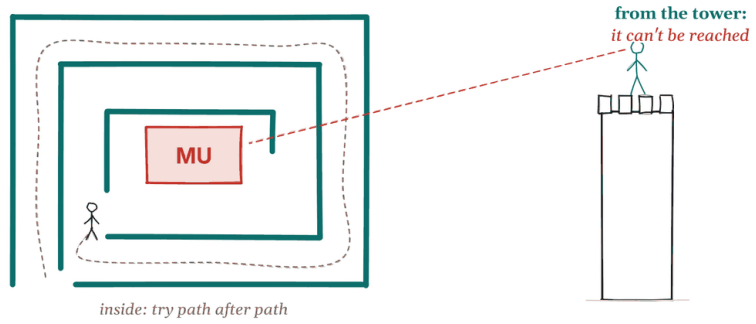


Like a three-strand braid: *mathematics, drawing and music woven into one rope that asks how meaning can grow out of symbols that mean nothing alone.*

Formal systems and meaning

Hofstadter opens with a simple puzzle. Begin with the string MI. Apply a few mechanical rules: add a U after a final I, double everything after the M, replace III with U, drop UU. Can you produce MU? Working inside the rules, people try string after string. The answer comes only from stepping outside the rules and noticing a property of them: the number of Is is never divisible by three, so MU is unreachable.

The lesson is about the difference between working inside a system and reasoning about it. A rule-following executor works inside. Knowing whether the rules can ever produce the result you want requires a view from outside. In a company, that outside view belongs to the people who design and own the loop.

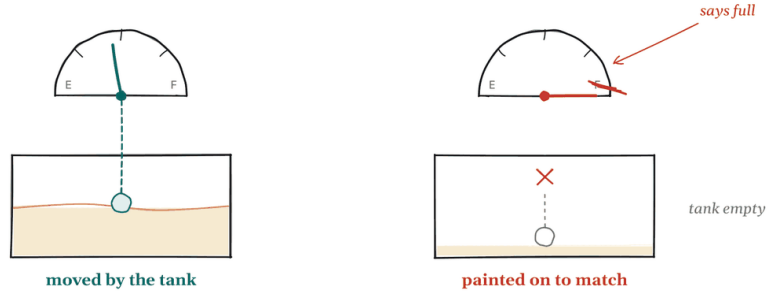


Like a hedge maze: *walkers inside try path after path, while someone on the tower sees at once that the centre can never be reached.*

Isomorphism: how symbols come to mean something

A string of symbols means nothing by itself. It acquires meaning when its structure mirrors the structure of something else, so that operations on the symbols track changes in the world. Hofstadter calls this isomorphism. Numbers mean quantities because arithmetic mirrors counting. A ledger means a company's finances because its entries mirror real transactions.

For embedding, this is the central question. A rule register, a decision register or a standard means something only if it mirrors how the work actually runs. A register copied from the procedure manual mirrors the work as it was designed, not as it is. That is why the discipline insists on walking real cases and measuring real baselines. The symbols must be anchored in the world, or they are only a formal game.

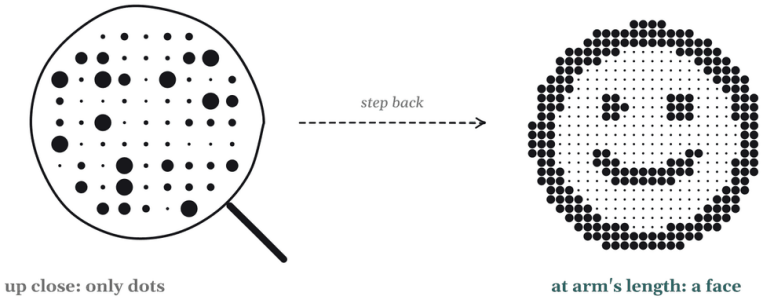


Like a fuel gauge: *the needle means something only because a float in the tank moves it; a needle painted on to match the manual tells you nothing.*

Levels

Hofstadter shows that the same system can be described at many levels, and that each level has its own useful laws. An ant colony behaves intelligently although no single ant does. A brain thinks although no single neuron thinks. Explanation at the wrong level is not wrong, but it is useless.

Businesses are the same. A case can be described as keystrokes, as process steps, as decisions, as a loop, or as a line in the financial model. The skill of abstraction, which Part Two describes, is largely the skill of choosing the level at which a problem becomes tractable.

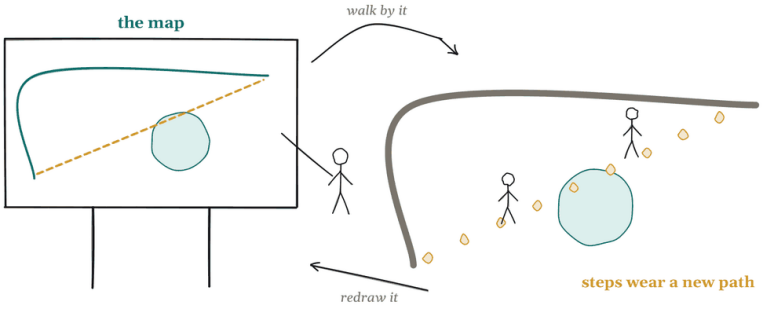


Like a newspaper photo: *under a magnifier it is only dots, at arm's length a face; the skill is picking the distance at which the problem makes sense.*

Strange loops

The book's central image is the strange loop. Moving through the levels of a system, you find yourself back where you started. A system that represents itself becomes able to change itself. Hofstadter argued that this kind of self-reference is the root of what we call a self. Escher's drawings of hands drawing each other and staircases that climb forever are visual versions of it.

A company that embeds AI well develops a strange loop of its own. It builds a model of its work: the map, the registers, the standard. It acts through that model, and the action changes the work. The changed work then changes the model, through the exceptions, the drift and the map review. The organisation comes to represent itself, and the representation becomes part of how it runs. **The method of business engineering is the discipline of keeping that loop honest:** anchored in real cases, revised when the world changes, and owned by people who answer for it.



Like a park map on a signboard: *walkers follow it, their steps wear a new shortcut, and the keeper redraws the map to match the ground.*

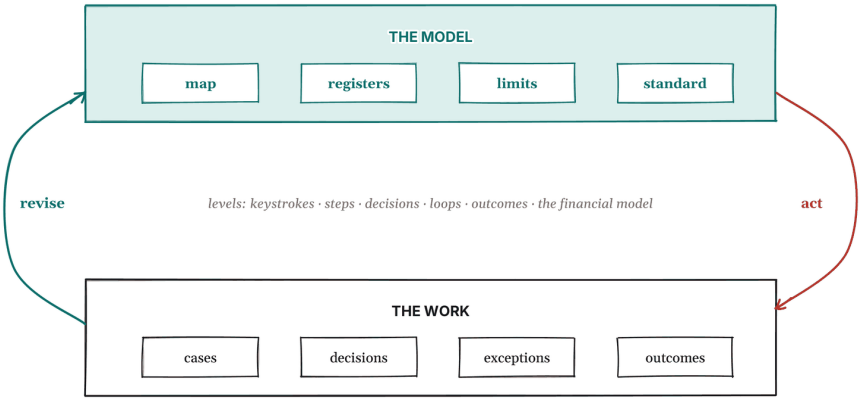


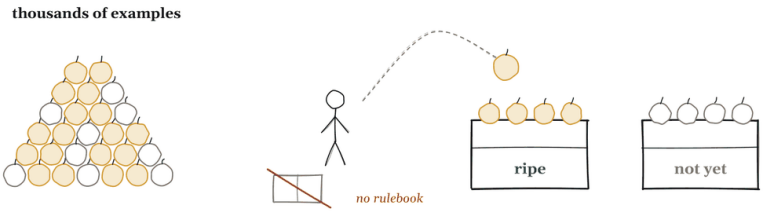
FIGURE 3 Strange loops and levels. The company builds a model of its work, acts through it, and the action changes the work; the changed work changes the model. Kept honest, the loop is how the company learns.

IV

The Neural Turn

Alongside the symbolic tradition ran another one, almost as old. In 1943, the neurophysiologist Warren McCulloch and the logician Walter Pitts proposed a simplified model of a neuron. It took inputs, weighed them and fired if the total passed a threshold. They showed that networks of such units could compute logical functions. In 1958, Frank Rosenblatt built the perceptron, a machine that learned to classify images by adjusting its weights from examples rather than following written rules.

The idea was radical. Instead of telling the machine the rules, show it examples and let it find the regularities itself. It went through long winters. In 1969, Marvin Minsky and Seymour Papert showed the limits of simple perceptrons, and interest collapsed for more than a decade. It revived in 1986, when David Rumelhart, Geoffrey Hinton and Ronald Williams popularised a method for training networks with many layers. It revived again, decisively, after 2012, when deep networks trained on large datasets with fast hardware began to outperform every other approach to images, speech and text.

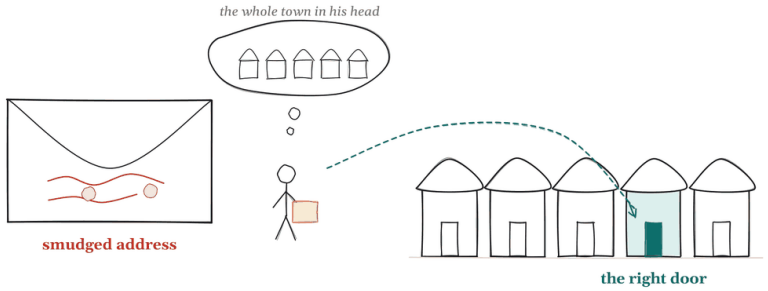


Like an apprentice at a fruit stall: *nobody hands her a rulebook; she watches thousands of apples sorted until she can tell a ripe one herself.*

Learning a distribution

A neural network does not store rules. It stores a compressed representation of the patterns in the data it was trained on. A language model, trained to predict the next word across a large share of the world's text, ends up encoding a great deal about grammar, facts, styles of reasoning and how situations usually unfold. Researchers have shown formally that good prediction and good compression are two sides of the same thing. In a precise sense, a language model is a compression of a distribution.

That explains both its strengths and its weaknesses. Its strength is that it handles the messy, unstructured, never-quite-the-same inputs that defeated rule systems. It can read an invoice in a layout it has never seen, understand an email written in anger, and summarise a contract nobody templated. It covers ground the symbolic tradition could not.

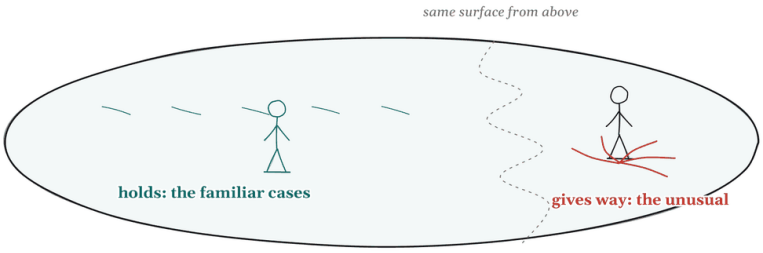


Like a seasoned postman: *years of rounds have packed the whole town into his head, so even a smudged, misspelled address still reaches the right door.*

What it does not give you

Its weaknesses mirror the strengths of rules. A neural model offers no guarantee that its output follows any particular rule. It can be confidently wrong, especially on cases unlike those it learned from. Its competence is uneven in ways that are hard to predict. One study of consultants called this a "jagged frontier": quality rose sharply on tasks inside it and fell on tasks just outside, where people trusted answers they should have checked. And a model cannot, by itself, say why it decided as it did in terms an auditor would accept.

Put simply, a neural model is excellent at the body of a distribution it has seen and unreliable at the tail it has not. That is the opposite trade to rules. Rules are precise but brittle at the edges, and learned models are flexible but unaccountable. **Neither is enough on its own for work a business must answer for.**



Like a frozen lake: *the ice holds firm over most of it, but its edge is jagged and the thin patches look exactly like the thick ones.*

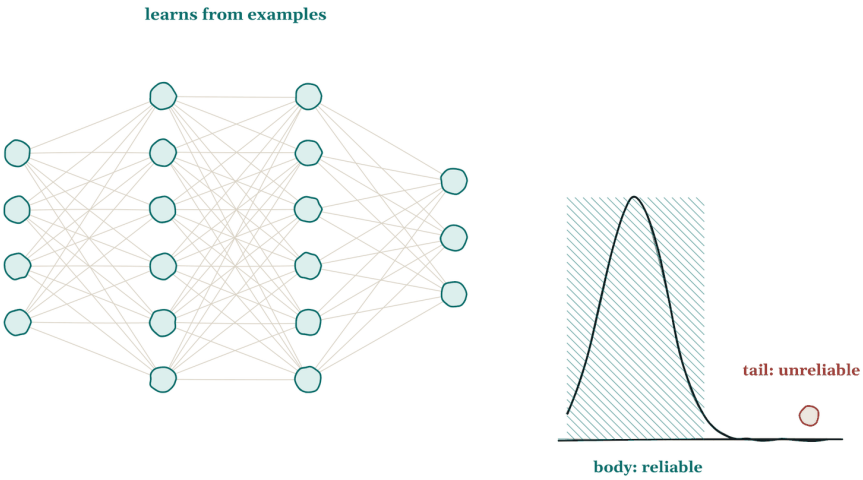
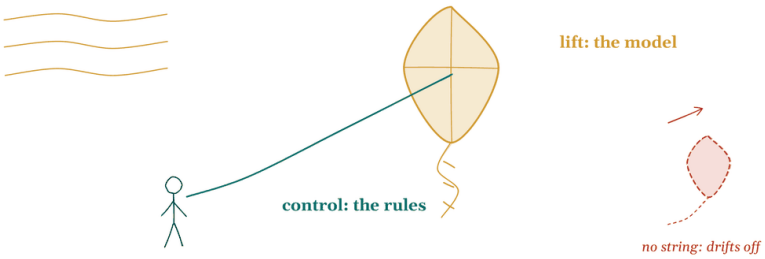


FIGURE 4 The neural turn. Instead of writing rules, show the machine examples and let it compress the patterns. A learned model is strong on the body of the distribution it saw and unreliable on the tail it did not.

V

Neurosymbolic AI

The obvious response to two traditions with complementary strengths is to combine them. Researchers have argued for this for decades, and in the 2020s it became the leading direction for systems that must be both capable and reliable. The approach is called neurosymbolic AI.

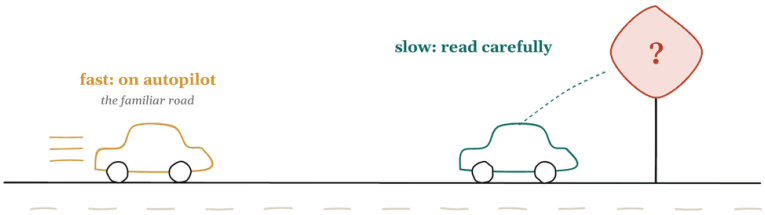


Like a kite and its string: *the kite gives lift from the wind, the string gives control, and on its own either one just falls or drifts away.*

Two systems

A helpful analogy comes from psychology. In *Thinking, Fast and Slow*, published in 2011, Daniel Kahneman described two modes of human thought. System 1 is fast, automatic and associative: recognising a face, reading a mood, sensing that a number looks wrong. System 2 is slow, deliberate and rule-following: checking a calculation, applying a policy, constructing an argument. People rely on System 1 for most of what they do and call on System 2 when the stakes or the doubts are high.

The analogy is loose, but useful. A learned model behaves somewhat like System 1: fast pattern recognition over rich, unstructured input. Symbolic components behave like System 2: explicit rules, checks, logic and records. A capable system needs both, and the design question is how they hand work to each other.



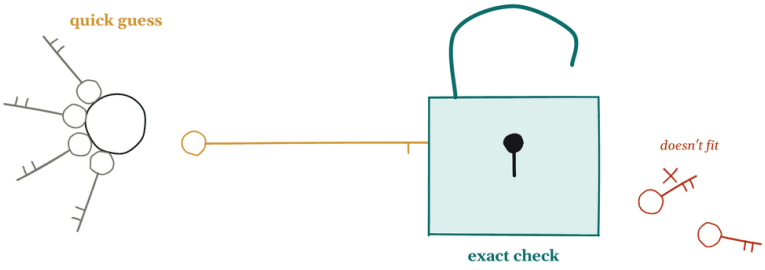
Like driving a familiar road: *hands steer on autopilot until an odd sign appears, then the driver slows down and reads it carefully.*

Ways of combining them

Researchers have catalogued several ways to join the two halves. Henry Kautz, in a widely cited 2020 lecture, set out a taxonomy that ranges from symbolic systems calling neural components to neural systems calling symbolic tools. Artur d'Avila Garcez and Luís Lamb described neurosymbolic AI as the field's "third wave", after symbolic AI and deep learning. Gary Marcus argued that robust AI would need explicit knowledge and reasoning alongside learning.

There are now striking demonstrations. In 2024, DeepMind's AlphaGeometry solved olympiad geometry problems at close to gold-medal level. A language model proposed promising constructions, and a symbolic engine checked and completed the proofs. Neither half could do it alone.

In enterprise systems, the most common combination is simpler and already everywhere. A language model reads and interprets. It then calls tools that apply rules exactly, write to systems of record and check results. A grader verifies the output against a written standard. People decide where the stakes demand it.



Like a locksmith with a ring of keys: *a quick hand guesses which key might fit, and the lock itself tests each guess exactly.*

The enterprise loop as a neurosymbolic system

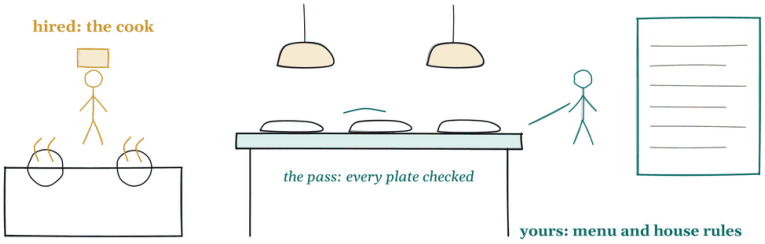
Seen this way, the parts of an embedded loop fall into place.

The neural half is the model: it reads context, interprets cases, proposes decisions and drafts outputs. It covers the variety of the real world that rules cannot.

The symbolic half is what the business writes down. It includes the rule register and decision register, which say what must be followed and who may decide what. It includes the bounded limits, which say where the model may act alone. The standard and its rule checks say what counts as a correct result. The junctions and contested-decision log say where people decide, and what was decided when the rules ran out.

The connection between them is the platform and its logs: routing, tools, permissions, the switch-off and the record of every case.

This is why a business cannot outsource the symbolic half to a vendor. A model can be bought. A company's own rules, decisions, limits and standards cannot, because they are facts about the company. Writing them down is the work of this discipline. A company that embeds a model without them has built half a neurosymbolic system and is relying on the half that offers no guarantees.



Like a restaurant kitchen: *you can hire the cook, but the menu, the house rules and the pass where every plate is checked must be your own.*

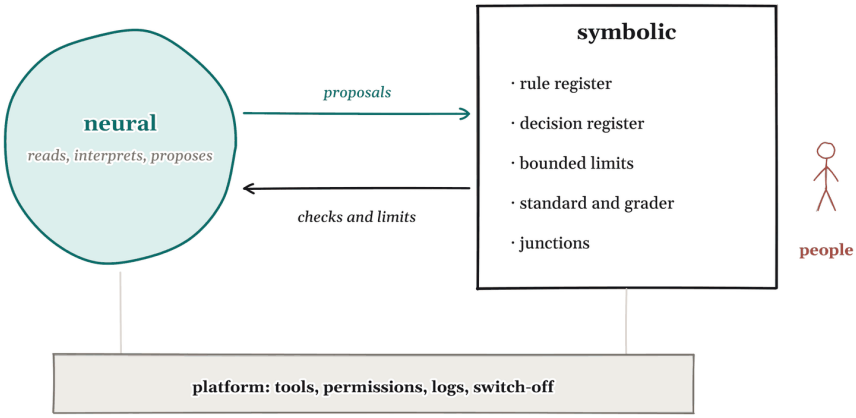
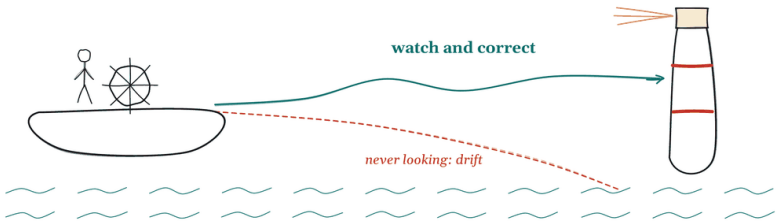


FIGURE 5 Neurosymbolic AI in the enterprise. The model reads and proposes; the symbolic layer of rules, limits, standard and junctions checks, constrains and records; people decide at the edges. The platform connects them.

VI

Closed Loops

Rules and learned models are ways of deciding. Neither, by itself, connects a decision to the world. That connection is the subject of a third body of thought: cybernetics, the science of control and communication in systems that act and sense the results of their action.

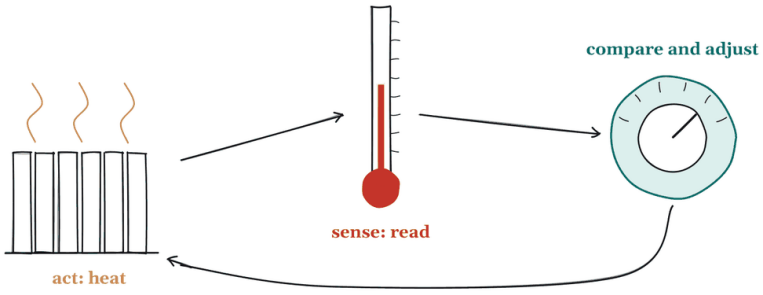


Like a helmsman: *choosing a heading is not enough; he keeps watching the bow against the lighthouse and correcting the wheel as the boat drifts.*

Feedback

In 1948, the mathematician Norbert Wiener published *Cybernetics*. Its central idea was feedback. A system acts, senses the effect of its action, compares it with a goal, and adjusts. A thermostat does this, and so does a pilot. So does any organisation that measures results and changes course. Without feedback, a system can only hope its actions had the intended effect. With it, the system can correct itself.

An embedded loop is a feedback system. The executor acts on cases, and the grader senses whether outcomes meet the standard. The operating review compares results with targets, and people adjust rules, limits and context. Remove the grader or the review, and the loop is running open: acting without knowing whether its actions are right.

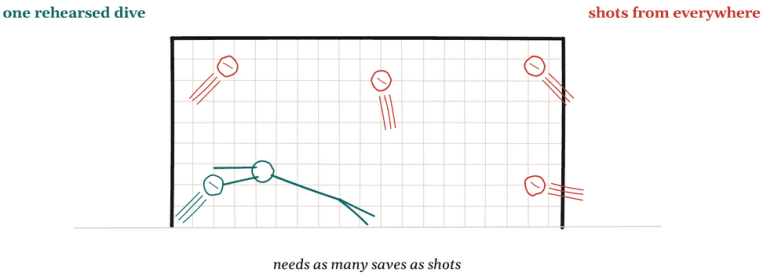


Like a thermostat: *it heats, reads the room, compares the reading with the setting and adjusts; take away the thermometer and the heater runs blind.*

Requisite variety

In 1956, the cybernetician W. Ross Ashby stated what he called the law of requisite variety. A controller can only control a system if it can produce at least as many distinct responses as the disturbances it must handle. Put plainly, only variety can absorb variety.

This law explains much of the design of embedded work. A fixed rule set has limited variety, so it handles the body of a distribution and fails on the tail. A learned model adds a great deal of variety, which is why it handles messy inputs. People, at junctions and in the exception queue, add the variety of judgement for cases nobody anticipated. A loop that tries to handle a heavy-tailed distribution with rules alone violates Ashby's law, and so does a loop that removes the people from its tail.



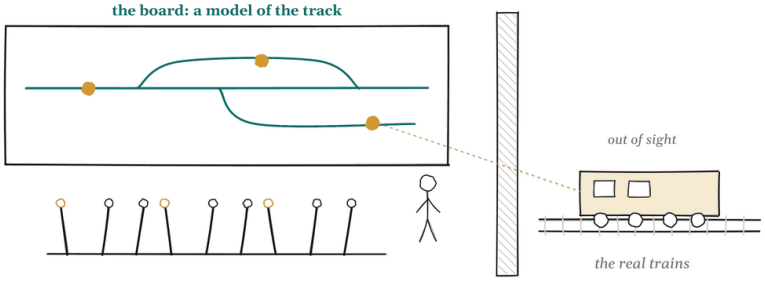
Like a goalkeeper facing shots from every angle: *one rehearsed dive covers one corner, and only as many kinds of save as kinds of shot keeps the goal safe.*

The good regulator

In 1970, Roger Conant and Ashby proved a theorem whose title states its result: "Every good regulator of a system must be a model of that system". To

control something well, you must represent it. A controller that does not model the system it controls can only succeed by luck.

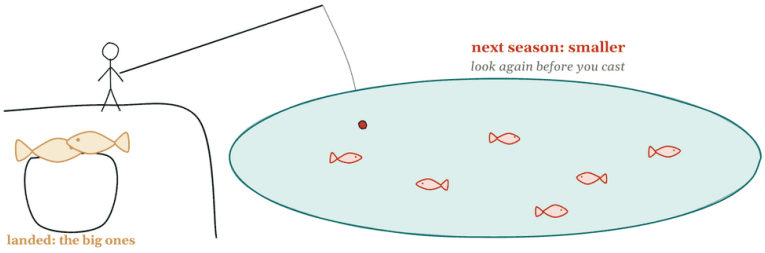
This is the strongest argument for the discipline in this book. A company that wants to control an embedded loop must hold a model of that loop. That means its distribution, its rules, its decisions, its limits, its costs and the way it fails. The map, the canvas, the registers and the standard are that model. **They are not documentation produced for its own sake. They are the regulator.**



Like a railway signal box: *the signaller never sees the trains, only a board that mirrors the track, so without a faithful board safe control is luck.*

Loops that change the world they read

One more property matters. Closed loops in business change the distribution they read. A collections loop that chases invoices faster changes how customers pay. A screening loop that contacts candidates sooner changes who applies. The investor George Soros called this reflexivity in markets: participants′ views change the reality those views are about. Embedded loops are reflexive in the same way. This is why the method never finishes. Read the distribution, act, read it again, because the acting changed it.



Like fishing a lake: *keep landing the big fish and next season the catch is smaller, because the fishing changed the lake you are reading.*

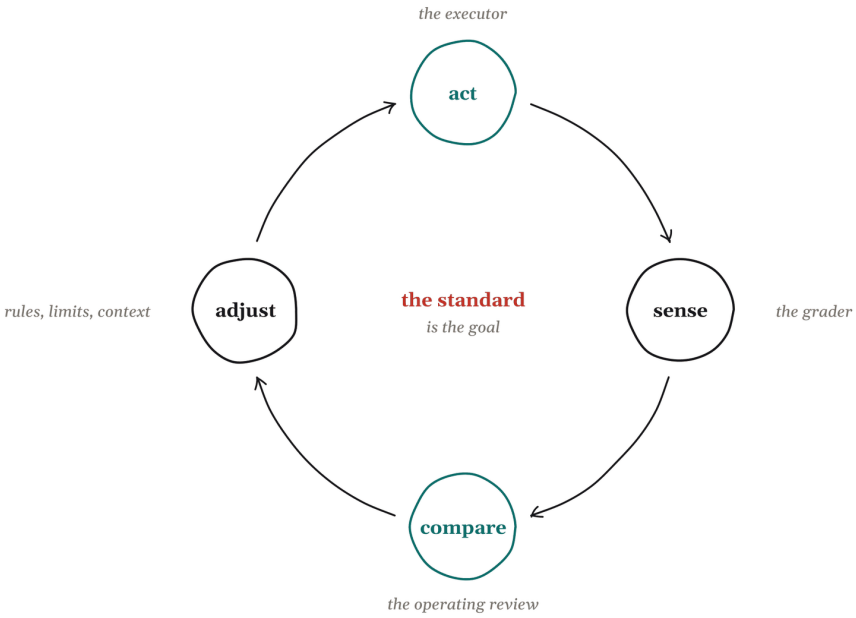
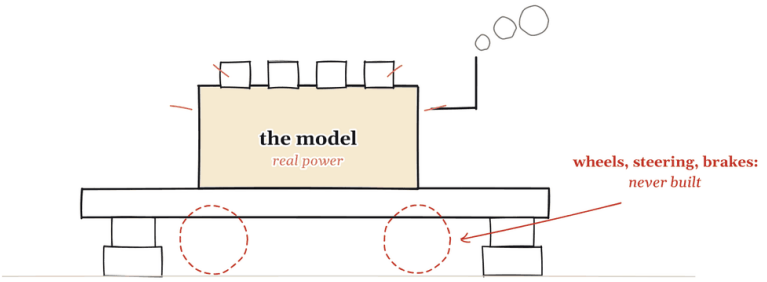


FIGURE 6 Closed loops. Act, sense the result, compare with the goal, adjust. In an embedded loop the executor acts, the grader senses, the operating review compares, and people adjust the rules, limits and context.

VII

Why This Matters for the Enterprise

The previous chapters can now be drawn together into a single argument. It explains why AI stalls in so many companies, and what has to be built for it not to.



Like a powerful engine bolted onto a cart with no wheels: *the power is real, but nothing was built for it to move, so it stalls.*

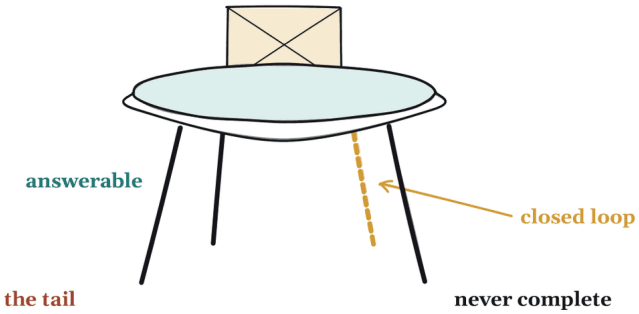
Four claims

First, an executor that decides is a learned model, and learned models are strong on the body of a distribution and unreliable on its tail. They come with no guarantees and no reasons an auditor would accept.

Second, a business's work must be answerable. Decisions must trace to rules, rules to owners, owners to authority. That requires a symbolic layer: rules, decisions, limits and standards written down and signed.

Third, no rule system is complete. Some cases fall outside every rule, some knowledge stays tacit, and some decisions were never agreed. Those edges need people, placed deliberately, and a way to extend the rules when the world shows them something new.

Fourth, control requires a model and a closed loop. To run an embedded loop well, a company must hold a model of it, grade every outcome, feed the results back, and revise the model as the loop changes the world it reads.

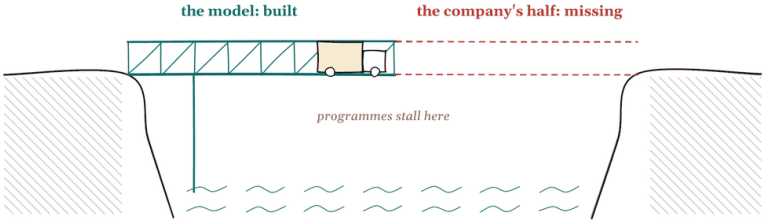


Like a four-legged stool: *the tail, answerable work, incomplete rules and a closed loop each hold a corner, and the seat tips if any leg is missing.*

The conclusion

Together, these claims say that embedding AI into a company's core means building a neurosymbolic, closed-loop system around each unit of work. The model provides the neural half. The company must provide everything else: the symbolic layer, the human junctions at the edges, and the feedback that keeps the whole thing anchored in reality.

Most companies have the first half and little of the rest. Their rules are scattered and contradictory, and their decisions are unsorted. Their edges are undesigned, and their loops run open. That is the gap in which programmes stall. It is not a technology gap. It is a gap in the company's knowledge of itself, written down well enough to regulate an executor that decides.



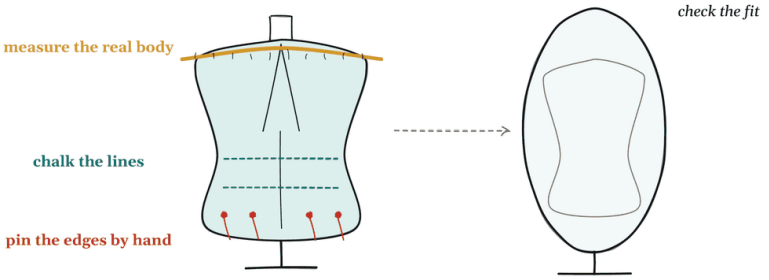
Like a bridge built from one bank only: *the model's half reaches out over the river, and the company's half, where programmes stall, was never built.*

What the business engineer builds

The business engineer is the person who builds the missing half. They read the real distribution of a loop's cases to know where the body ends and the tail begins. They abstract and compress the work into rules, decisions and limits, which become the symbolic layer. They place people where rules run out. They

write the standard and build the feedback that grades every outcome. And they keep the strange loop honest, revising the company's model of itself as the work changes.

The rest of this book describes how. Part Two sets out the core method. Parts Three to Five define the nine areas of knowledge in which the method is applied. Parts Six to Eleven show how the rest of the discipline cascades from them: the instrument, the sequence, the record and the models.



Like a tailor fitting a suit bought off the rack: *measure the real body, chalk the lines, pin the edges by hand and keep checking the fit in the mirror.*

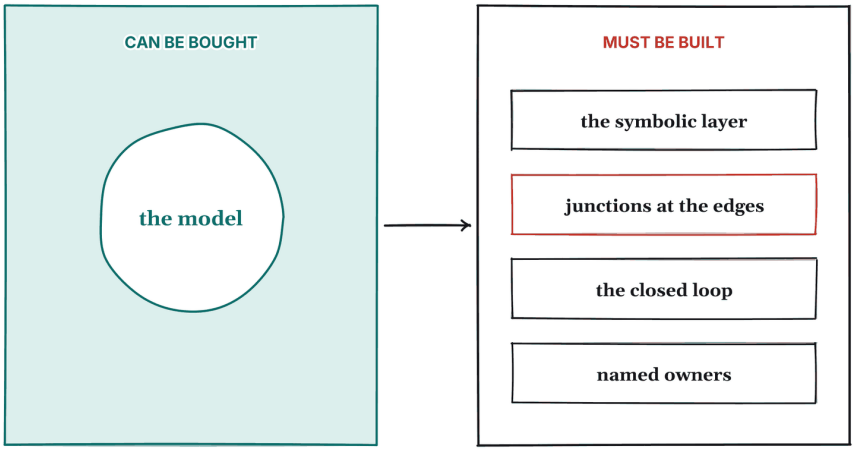


FIGURE 7 Why this matters. The model is the half a company can buy. The symbolic layer, the human junctions at the edges and the closed loop are the half it must build, because they are facts about the company.

PART TWO

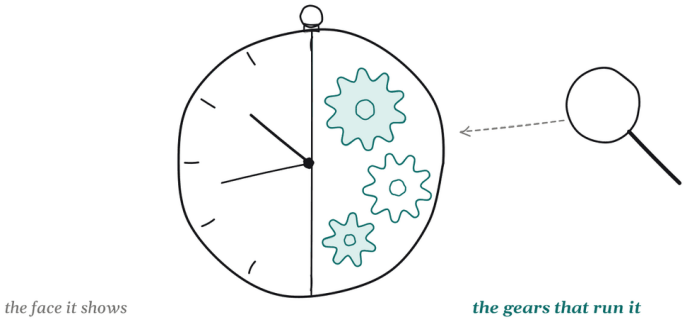
The Core

VIII

What Business Engineering Is

Business engineering is the practice of understanding how a business actually works, well enough to redesign it. It treats a company as a system: a set of loops that turn inputs into outcomes, governed by rules, decisions, people and money. It reads that system from evidence rather than from organisation charts, and it changes it on purpose.

The word "engineering" is chosen deliberately. Engineers work from the properties of the materials in front of them, not from how they wish those materials behaved. They measure before they design, test before they rely on anything, and write down why a design is the way it is. Business engineering applies that temperament to companies. The materials are cases, rules, decisions, people, systems and cash flows. The designs are the loops that produce the company's outcomes.



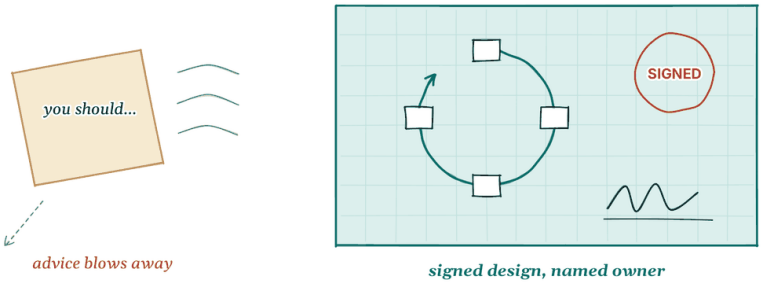
Like a watchmaker who opens the case: *the face only shows the time, so you read how the gears really turn before redesigning anything.*

What it is not

It is not strategy, though it informs strategy. Strategy decides where a company competes and how it wins. Business engineering works one level down, on how the work runs, and it often finds that strategy rests on untrue assumptions about the work.

It is not software engineering, though it works closely with it. Software engineering builds systems. Business engineering decides what those systems should be allowed to do inside the business, and how anyone would know they did it well.

It is not consulting in its traditional form, though consultants can practise it. Its outputs are not recommendations. They are signed designs, measured baselines and loops running in the core, owned by named people inside the company.

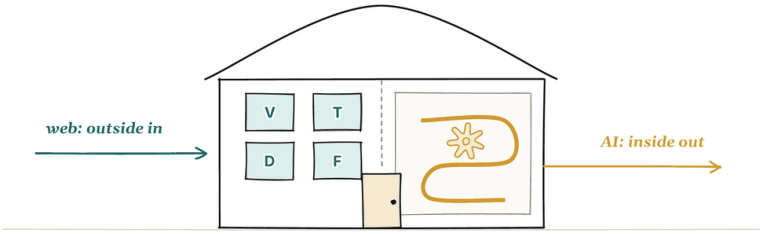


Like a signed blueprint rather than a note of advice: *the advice blows away, while the signed design stays and a named person owns it.*

Outside in and inside out

Business engineering reads a company from two directions. From the outside, it reads the business model through four connected models: value, technology, distribution and financials, abbreviated VTDF. It asks what value the business creates and for whom, and what technology makes that value possible. It asks how the value reaches customers, how the money flows, and how the four reinforce or undermine each other. That view grew up in the platform era, when technology and distribution became the model itself, and outcomes followed power laws.

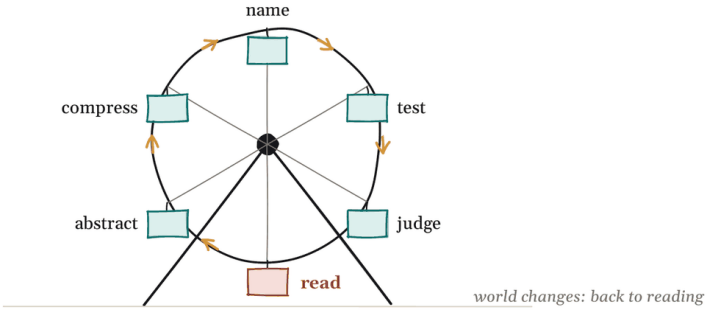
From the inside, it reads the work: the loops, their rules, decisions and costs. The web changed companies from the outside in, through how they reached customers. AI changes them from the inside out, through the work itself. The embedding era makes the inside view as important as the outside one. The two connect wherever a loop's outcome serves a value proposition, its costs roll into the financial model, and its handoffs meet customers.



Like a house: *the web remodelled the front from the street, AI rewires the plumbing from inside, and both meet at the front door.*

The method

At the centre of the discipline is a loop of six moves. Read the real distribution of cases. Abstract the few kinds of case behind the many. Compress the mechanism into the fewest words that still predict. Name it, so it becomes a mental model people can use. Test it against the real world. Build judgement from the results, and return to the distribution when the world changes. The chapters that follow take the moves one at a time, and the last chapter of this part shows how they close into a loop that acts on the world.



Like a Ferris wheel: *six cars come round in turn, read, abstract, compress, name, test, judge, and the wheel returns to reading when the world changes.*

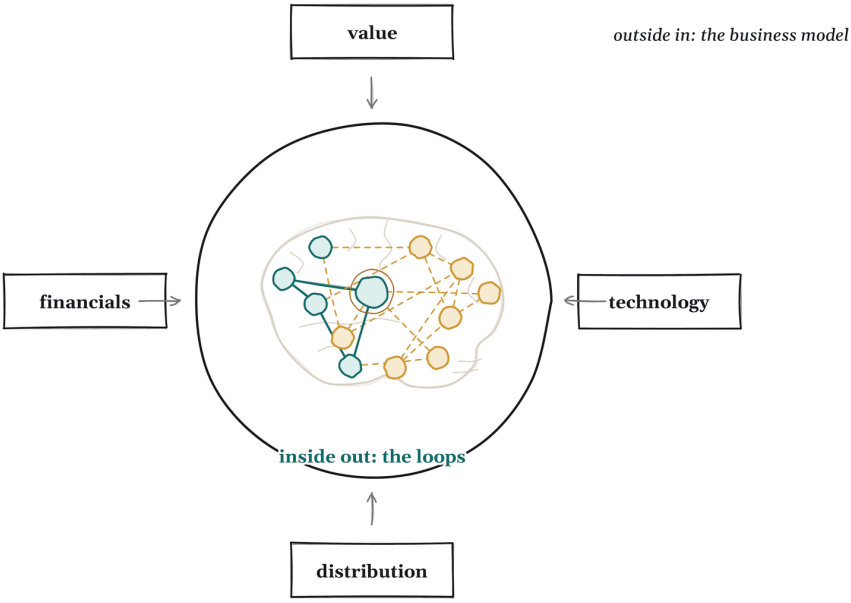
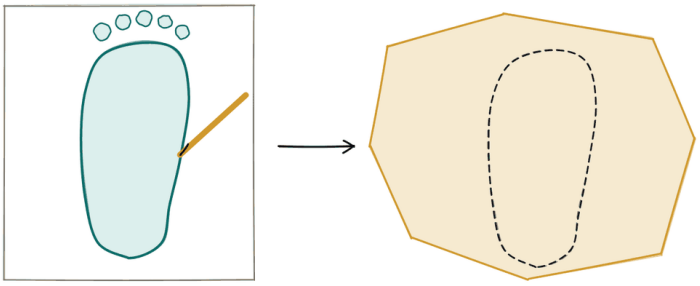


FIGURE 8 Business engineering reads a company from two directions: from the outside, through value, technology, distribution and financial models; from the inside, loop by loop.

IX

Reading the Distribution

The method begins with the real world, before any abstraction. The first question about any piece of work is how its cases are spread. That shape decides more of the design than any other single fact.



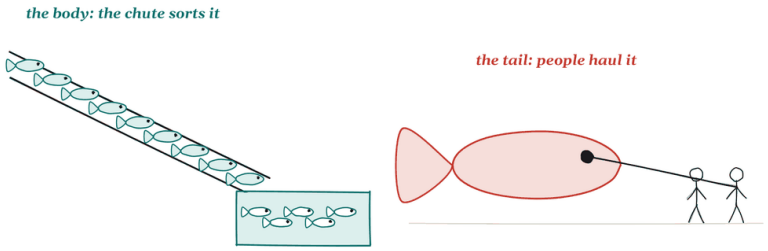
Like a shoemaker who traces the foot before cutting any leather: *the shape of the cases is taken first, and it decides the design.*

Two worlds

Some work produces cases that cluster around a typical value. The heights of adults, the hours on a timesheet and the time to process a routine address change behave like this. Statisticians call the shape normal. In these domains averages mean something, a sample tells you about the whole, and rules work well.

Other work produces cases where a few are enormous and most are small. City sizes, book sales, customer revenue, insurance losses and invoice values behave like this. The usual name for the shape is a power law, and the general term is heavy-tailed. In these domains averages mislead, a sample can miss the cases that matter most, and the largest cases can be larger than all the others put together. The essayist Nassim Taleb called the two worlds Mediocristan and Extremistan.

Most business processes are a mixture: a routine body of similar cases and a thin, heavy tail. That split is the most useful fact in embedding, because a learned model and a set of rules can usually handle the body, while people must hold the tail.

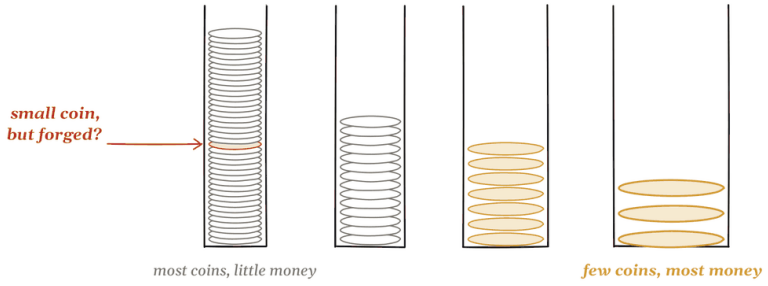


Like a fishing catch: *a sorting chute handles the many small fish of the body, while the rare giant of the tail needs people to haul it in.*

How to read one

Reading a distribution is simple arithmetic and rarely done. Pull a few months of cases. Sort them by the dimension that matters, usually value, sometimes difficulty or risk. Group them into bands and compare, for each band, its share of cases with its share of value. Where the two shares are similar, the work is close to normal. Where a small band carries a large share of value, the tail is heavy.

Then ask the second question: is the distribution the same for value, for difficulty and for risk? Often it is not. A large invoice from a long-standing supplier is high in value and low in difficulty. A small first invoice from a new supplier with new bank details is low in value and high in risk. A good reading names the body and the tail on each dimension that matters.

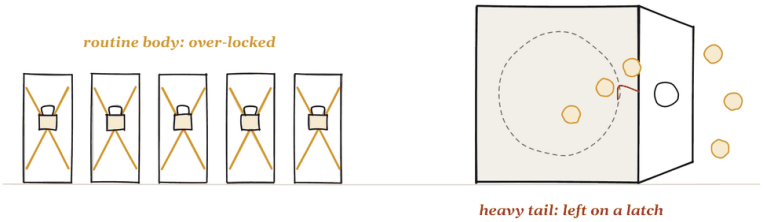


Like sorting a cash box: *compare each tube's share of coins with its share of the money, then check separately for the small forged coin.*

Why it comes first

Every later choice depends on the reading. It decides which loop to start with, what the executor may do alone, where junctions go, how long to test, and how to report results. A design made without it treats every case the same. That

design either over-controls the body and wastes money, or under-controls the tail and loses it.



Like locking a house with one standard lock: *the broom cupboards get chains they never needed, while the vault is left on a flimsy latch.*

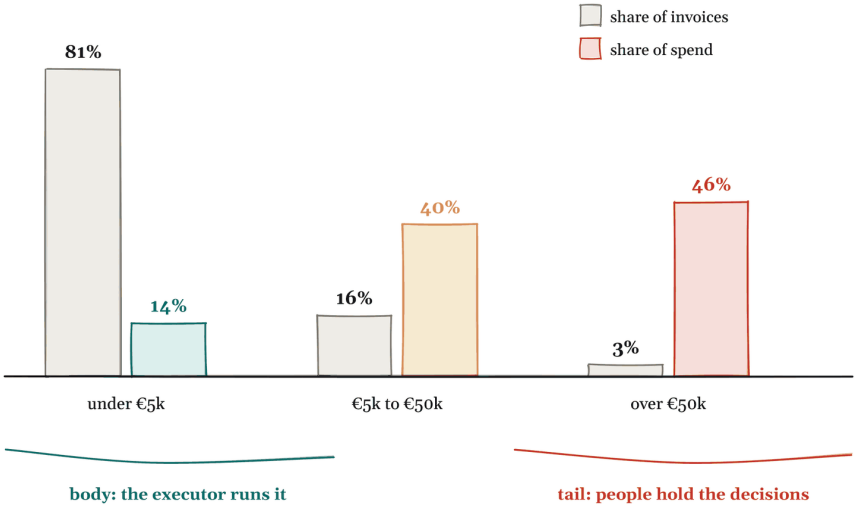
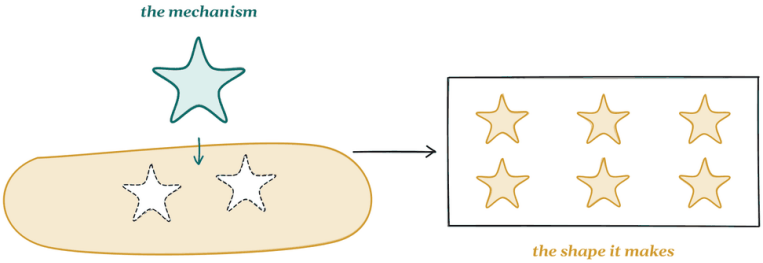


FIGURE 9 Reading the distribution. Share of cases against share of value, by band: a routine body that carries little of the value and a thin tail that carries almost half.

X

How Distributions Are Made

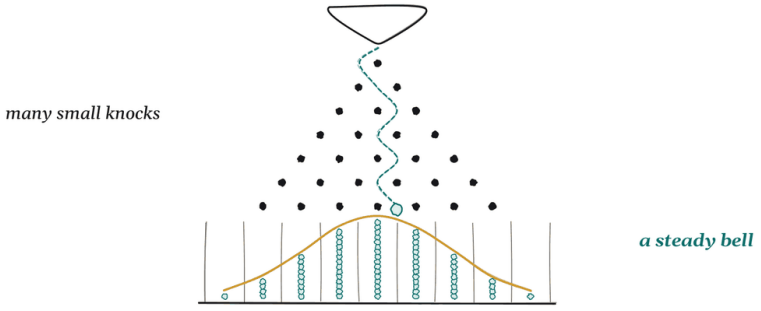
Distributions are not just found. They are produced by the mechanisms that generate the cases, and a business engineer who understands the mechanism can predict the shape, and sometimes change it.



Like a cookie cutter: *the shape of every cookie comes from the cutter, so whoever knows the cutter can predict the cookies, and change them.*

Independent causes make normal shapes

When many small, independent influences add together, the result tends towards a normal distribution. The time a routine form takes depends on many small things, none dominant, and the times cluster. Work like this is stable and predictable, and its average is a good guide.



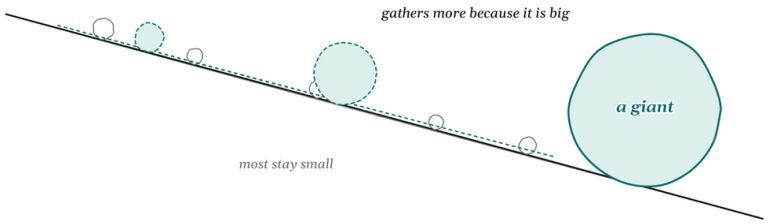
Like a bean machine: *each ball is knocked left or right by many small pegs, none dominant, and together they pile into a steady bell.*

Compounding makes heavy tails

When outcomes feed on themselves, the result is a heavy tail. A customer who buys more gets better terms and buys more again. A product that sells is

stocked more widely and sells more. A website with many links attracts more links. The sociologist Robert Merton called this the Matthew effect, after the line "to every one that hath shall be given". Albert-László Barabási and Réka Albert showed in 1999 how this preferential attachment produces power laws in networks. Wherever there is compounding, there will be a few giants and a long tail of small cases.

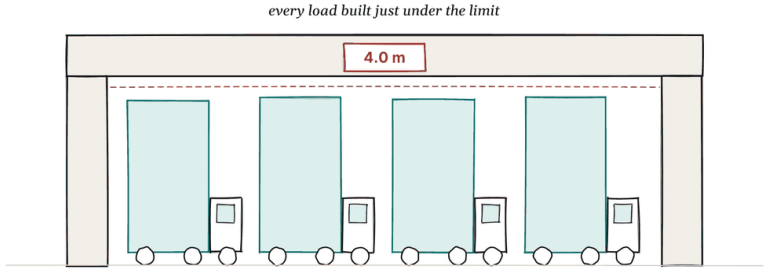
In the platform era, business models were built deliberately on such loops. Network effects, data advantages and distribution flywheels were engineered to compound. Business engineering grew up reading these loops from the outside. It reads them from the inside too: a company's largest customers, suppliers and contracts are heavy-tailed because the relationships compound.



Like snowballs rolling downhill: *each one gathers more snow because it is already big, so a few become giants and most stay small.*

Rules and thresholds reshape distributions

Distributions inside a company are also shaped by its own rules. An approval limit at €5,000 creates a cluster of invoices just below it. A policy that routes every claim over a threshold to a senior adjuster creates a queue at that threshold. A target creates bunching near the target. Many odd shapes in a baseline are the fingerprints of old rules. That is useful, because it shows where the rules are steering the work.

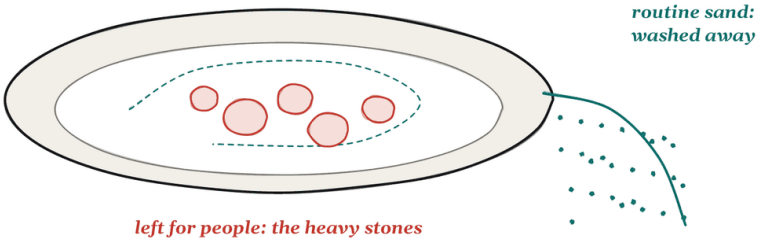


Like lorries under a low bridge: *every load is built just under the posted height, so the bunching is the fingerprint of the rule, not of the cargo.*

Embedded loops make new distributions

Finally, embedding changes distributions. When an executor clears the routine body quickly, the work that remains for people is almost all tail. People's time redistributes towards the hardest cases. Customers and suppliers adapt to faster responses, and the shape of what arrives shifts. This is why the method re-reads the distribution after a loop runs, and why a map review a year later rarely looks like the first.

Understanding how a distribution is made turns reading into prediction. If the cases come from compounding relationships, expect a heavy tail and design for it. If they come from many small independent causes, expect a stable body and trust its average. If they are shaped by a rule, ask whether the rule still serves its purpose.



Like panning for gold: *once the water carries off the fine routine sand, almost everything left in the pan for people is heavy stones.*

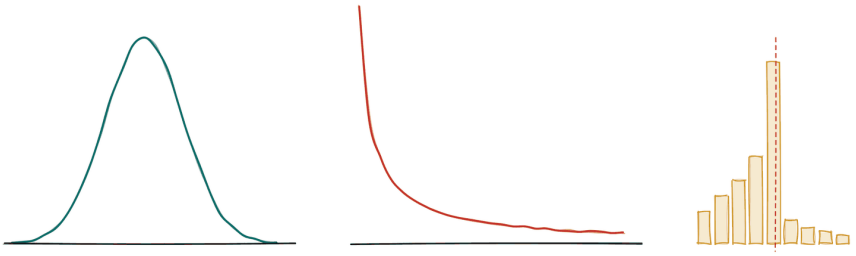
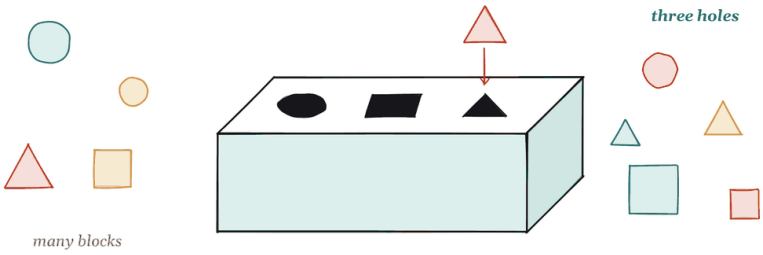


FIGURE 10 How distributions are made. Many small independent causes add up to a normal shape; compounding, where the large grows faster, produces a heavy tail; rules and thresholds leave their own fingerprints.

XI

Abstraction

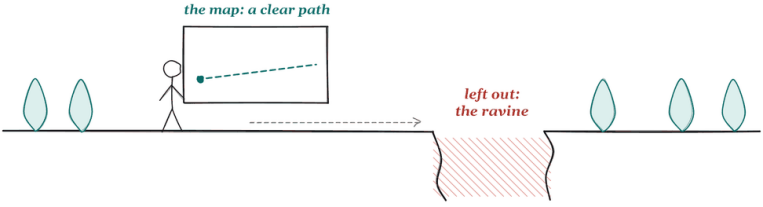
Abstraction is the act of seeing the few kinds of case behind the many. It deliberately loses detail so that the differences that matter become visible.



Like a child's shape sorter: *blocks of every colour and size, yet only three holes, because the shape is the difference that matters.*

The map is not the territory

In 1933, the philosopher Alfred Korzybski put the central warning of abstraction into a phrase that has lasted: "the map is not the territory". Every abstraction leaves things out. That is its purpose, and also its danger. A good abstraction leaves out what does not matter for the decision at hand and keeps what does. A bad one leaves out exactly the thing that will cause trouble.



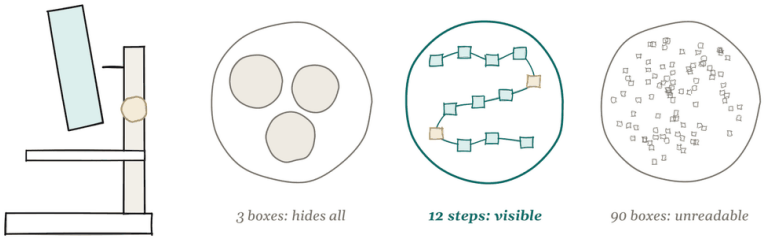
Like a trail map that rightly leaves out every tree, and wrongly leaves out the ravine: *an abstraction is judged by what its omissions cost.*

Choosing the level

The same work can be described at many levels. There are keystrokes and screens, steps and handoffs, decisions and rules, loops and outcomes, and finally lines in the financial model. Each level has its own useful regularities, and a problem that is hopeless at one level can be simple at another. The time an invoice takes is invisible at the level of keystrokes and obvious at the level of

waits between steps. Whether a loop is worth embedding is invisible at the level of steps and obvious at the level of outcomes and economics.

Much of a business engineer's skill is choosing the right level for the question. Process maps that go wrong usually go wrong on level. Three boxes hide everything, and ninety boxes cannot be read. Between eight and twenty steps per loop tends to be the level at which workarounds, waits and decisions become visible.

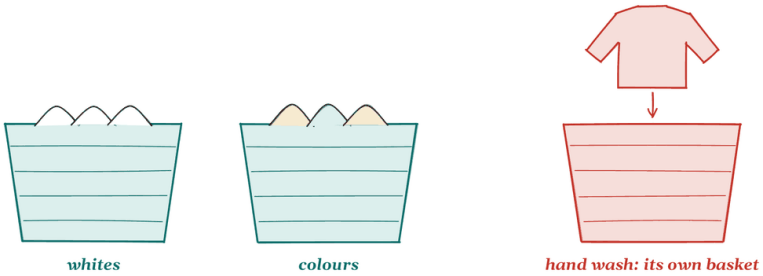


Like a microscope's focus: *three blurred blobs hide everything, ninety specks cannot be read, and in between a dozen steps come sharp.*

Categories that predict

The most useful abstractions are categories whose members behave alike. At the manufacturer, thousands of invoices became three kinds. Small, order-backed invoices from known suppliers were the body. Mid-sized invoices with occasional mismatches came next. Large or unusual invoices needed judgement. Each category predicted how the invoice should be handled. A category that does not predict behaviour is only a label.

Abstraction is where many AI programmes first go wrong. A pilot is built on the cases that were easy to collect. Those cases are abstracted into "invoices", and the tail is quietly left out of the category. The executor then meets the tail in production as a surprise. A good abstraction names the tail as a category of its own from the start.



Like sorting laundry: *each basket predicts its wash, and the delicate jumper needs a basket of its own from the start, or it shrinks.*

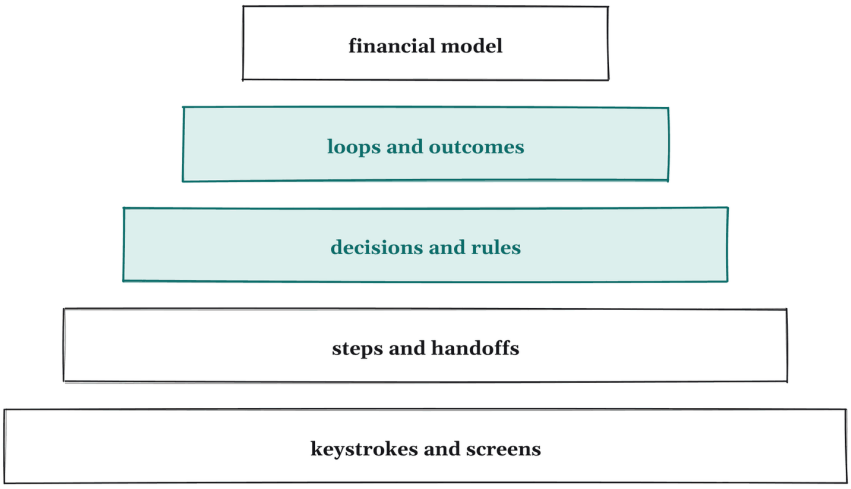
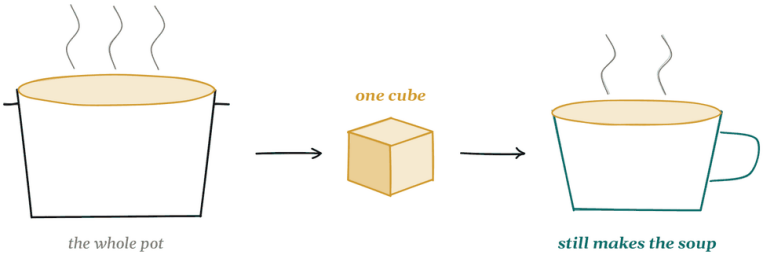


FIGURE 11 Abstraction. The same work at different levels, from keystrokes to the financial model. Each level has its own useful regularities; choosing the level is most of the skill.

XII

Compression

Compression is stating a mechanism in the fewest words that still predict something. If abstraction finds the categories, compression finds the rule that connects them.

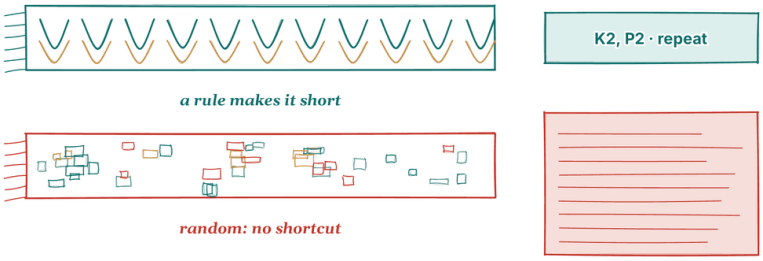


Like a stock cube: *a whole pot of soup reduced to one small cube that still makes the soup again when you add water.*

Prediction and compression

There is a deep link between compressing and understanding. In information theory, data that follows a pattern can be described more briefly than random data, and finding the shortest description means finding the pattern. Mathematicians formalised the idea in the 1960s as the shortest program that reproduces a given output. Statisticians turned it into a principle for choosing models: prefer the model that explains the data in the fewest total bits.

Language models are an instance of the same idea. Trained to predict the next word, they become extraordinarily good compressors of text. Researchers have shown that the two tasks are equivalent. The model is a compression of the distribution it learned from.

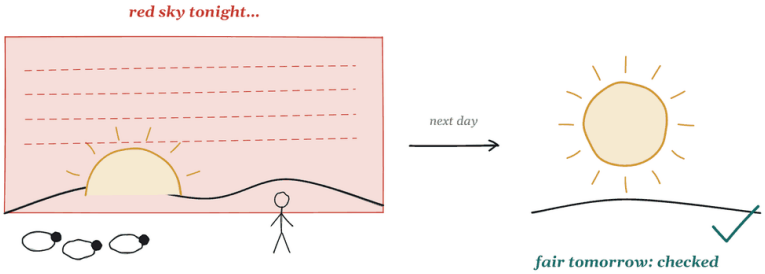


Like a knitting pattern: *a patterned scarf fits on one short line because it follows a rule, while a random one can only be described stitch by stitch.*

Human compression

A business engineer compresses too, but in words that people can use and check. "Most of the elapsed time is waiting between teams" compresses a sixteen-day process into a mechanism. So does "the executor can take the body; people must hold the tail", and so does "a contested decision cannot be engineered away". Each can be tested against new cases and found right or wrong. Each tells someone what to do.

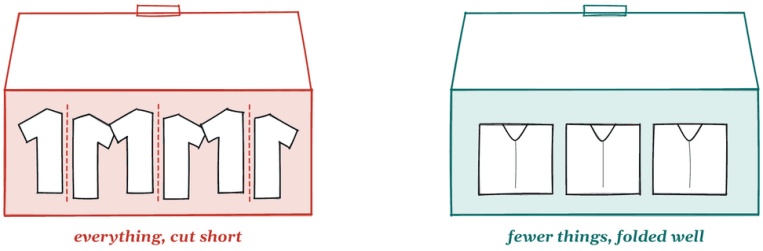
The test of a good compression is that it predicts something you did not already know. "Invoices are complicated" compresses nothing. "Missing goods receipts cause most of the remaining exceptions" predicts that fixing the receiving loop will clear the queue, and it can be checked.



Like the saying 'red sky at night, shepherd's delight': *a few words that predict tomorrow, which tomorrow can prove right or wrong.*

Fewer ideas, given room

Compression is not the same as shortening. A compressed explanation drops the ideas that do not carry weight and gives room to the ones that do. It does not keep every idea and delete the reasoning that connects them. A slide full of fragments is not compressed. It is merely truncated. The discipline of compression is choosing the few mechanisms that matter and stating each clearly enough to be tested.



Like packing a case: *leave out what carries no weight and fold the rest well, rather than cutting every shirt in half to make it all fit.*

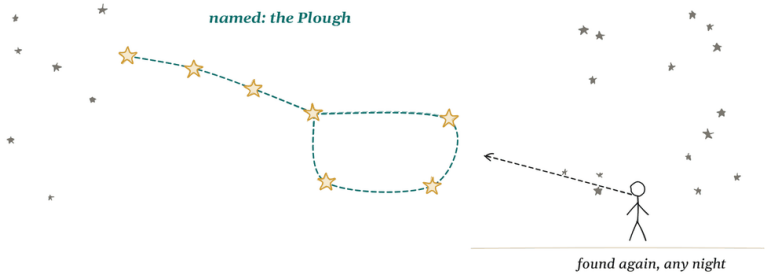


FIGURE 12 Compression. Many cases go in; a short statement that predicts comes out. The test of a good compression is that it can be checked against new cases and found right or wrong.

XIII

Mental Models

A compression that proves useful is given a name, and becomes a mental model. Names matter because they let people find the same pattern again, in a new place, quickly.

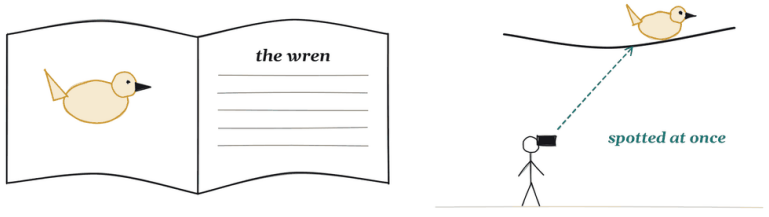


Like a constellation: *once a few stars are joined and named the Plough, anyone can find the same pattern again, quickly, on another night.*

Why naming works

A team that has the words "contested decision" will spot one in a meeting. A team without them will argue about the case in front of it and never notice that the argument has happened twenty times before. Naming turns a one-off insight into a tool that travels. It is also how a discipline becomes teachable: a vocabulary of named patterns that newcomers can learn and practitioners can share.

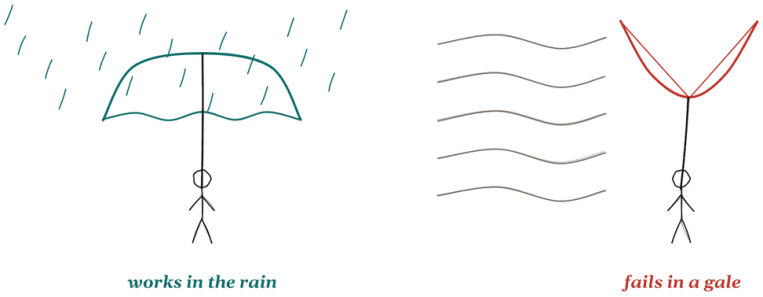
Most entries in this book are named models of this kind. They include the body and the tail, the compensation, the strip ratio and the junction. They also include the four classes of decision, the switch-off, cost per accepted outcome and the Scale Gates. Each compresses a mechanism and gives it a handle.



Like a birdwatcher's field guide: *with the name in hand, the walker spots the bird at once instead of arguing about yet another brown one.*

All models are wrong

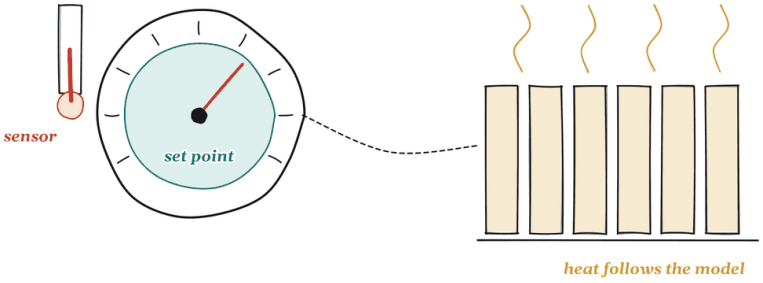
The statistician George Box wrote that all models are wrong, but some are useful. A mental model is a deliberate simplification, and it will fail somewhere. The discipline is to know where. The body-and-tail model fails when a small case is risky. The four decision classes fail when the organisation changes its mind. Good practitioners hold their models firmly enough to act on and loosely enough to revise.



Like an umbrella: *a simple tool that keeps you dry in rain and turns inside out in a gale, so you carry it and watch the wind.*

Models as regulators

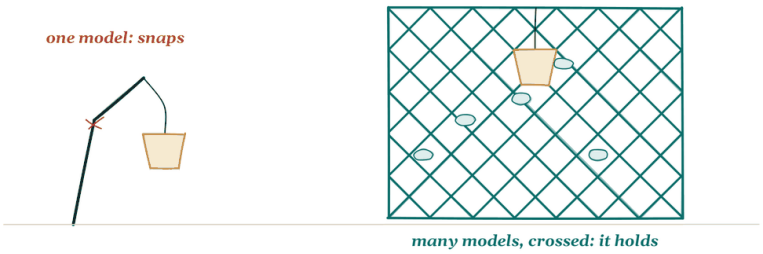
There is a stronger reason models matter in embedding. Conant and Ashby's good regulator theorem says that controlling a system well requires a model of it. In an embedded loop, the company's models do the regulating. The distribution reading, the registers, the limits, the standard and the economics are the means by which people control an executor that decides. Weak models mean weak control. That is why this book treats a register or a standard as part of the machine, not as paperwork about it.



Like a thermostat: *it keeps the room right only because it carries a model of the room, a set point and a sensor; weak model, weak control.*

A latticework

No single model is enough. The investor Charlie Munger described good judgement as a latticework of models from many disciplines, each correcting the others' blind spots. The models in this book are drawn from statistics, logic, cybernetics, psychology, accounting and operations. Their value comes from being used together. Reading a loop through only one of them, whether cost or risk or the technology, produces the one-sided designs that stall.



Like a garden trellis: *a single stick bends and snaps under the load, while many crossed laths hold it because each covers the others' weak spots.*

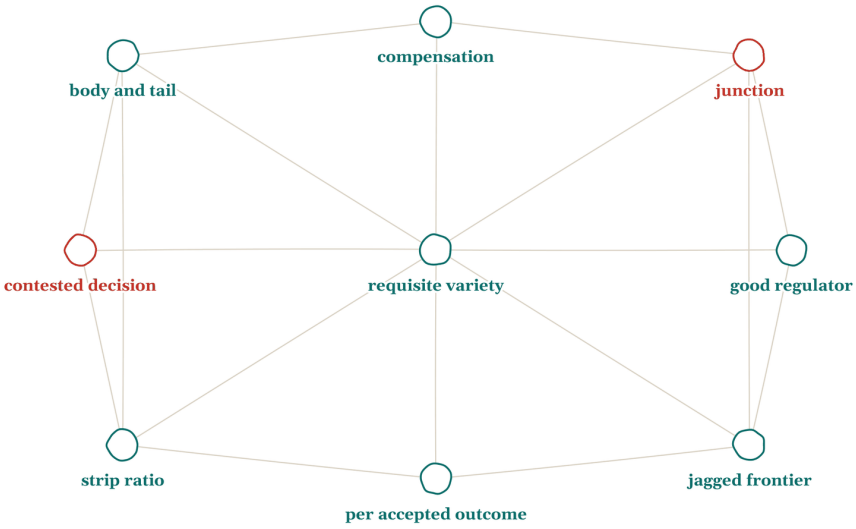
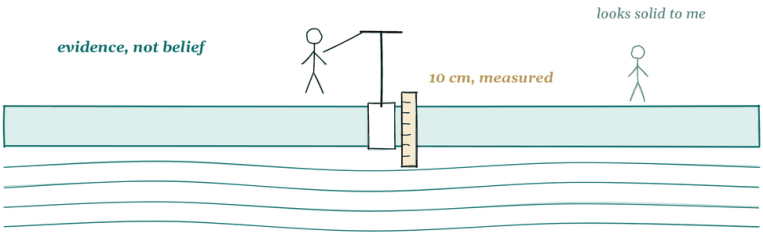


FIGURE 13 Mental models as a latticework. Named compressions from many fields, used together, each correcting the others' blind spots.

XIV

Testing and Judgement

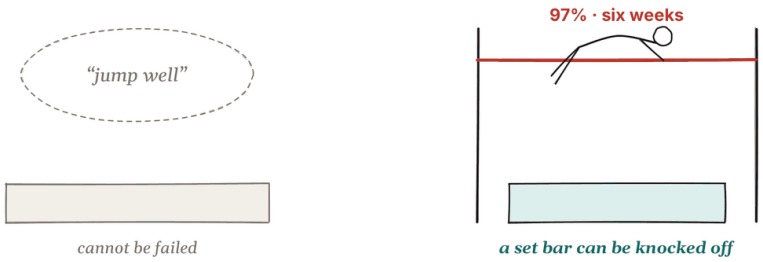
Every model is a claim about the world, and claims must be tested. Testing replaces belief with evidence. Over many cycles, it builds the judgement that lets a practitioner act well when evidence is thin.



Like drilling the ice before walking on the lake: *belief says it looks solid, the drill measures how thick it really is.*

Falsifiable claims

The philosopher Karl Popper argued that a claim is scientific only if some observation could prove it false. The same standard works for business models. "The executor will improve the invoice process" cannot be proven false. "The executor will match the clerks' results on at least 97% of invoices under €5,000 over six weeks" can. Shadow running, gate reviews and measured economics are all ways of putting falsifiable claims to the test before relying on them.

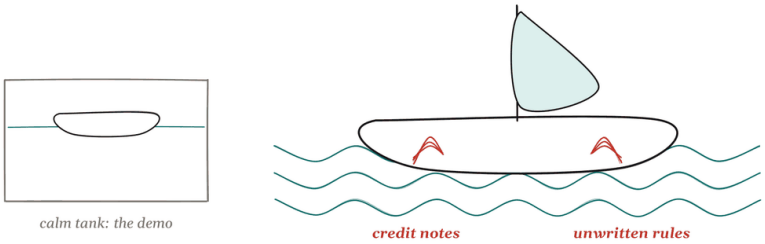


Like a high-jump bar: *'jump well' can never be failed, but a bar set at a stated height can be knocked off, so clearing it means something.*

Testing on real cases

Tests must use real cases, including the tail. A demonstration on prepared cases tests the demonstration. Shadow running on the live stream of cases

tests the loop. It reveals what the abstraction left out: credit notes that look like invoices, remittances in unfamiliar formats, rules the clerks apply that nobody wrote down. Every one of those findings improves the model.

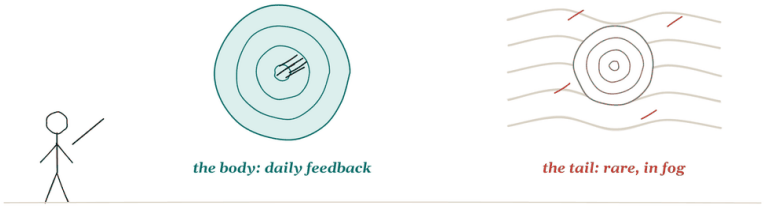


Like a boat's sea trial: *a calm test tank proves only the tank, while the open sea finds the leaks nobody knew were there.*

Where judgement can be trusted

Testing builds judgement: a practitioner's sense of what will work, earned from being right and wrong many times. Research by Daniel Kahneman and Gary Klein suggests that expert intuition is reliable only in environments that are regular and give fast, clear feedback. It is unreliable where cases are rare, varied and slow to show whether a decision was right.

The body of most processes is the first kind of environment, and the tail is often the second. The same expert who can be trusted to say "that invoice is fine" in seconds should be asked to show evidence on a large disputed one. The same holds for an executor. High agreement on the body says little about its judgement on the tail.

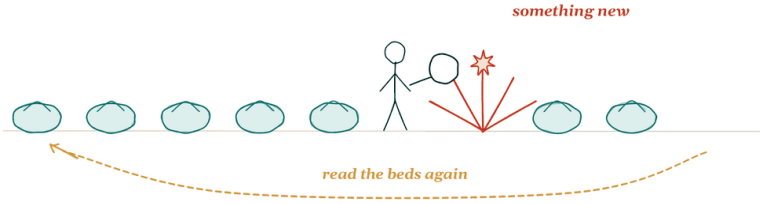


Like a darts player: *trust the throw at the board practised on every day, not at a far board in the fog that is rarely seen.*

Judgement notices change

Judgement does one more thing that tests cannot. It notices when the world has changed: a new kind of case, a shift in the distribution, a supplier that

redesigned its invoices. That noticing sends the method back to its first move. The practitioner reads the distribution again, and the cycle restarts.



Like a gardener who spots a plant nobody sowed: *noticing something new sends the work back to reading the beds again from the start.*

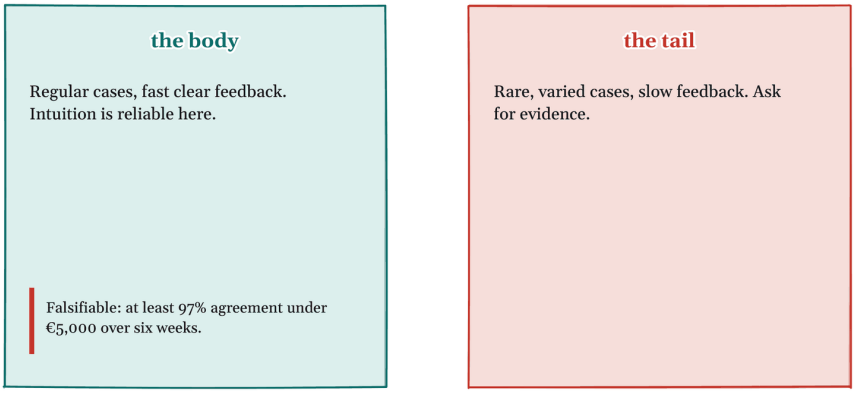
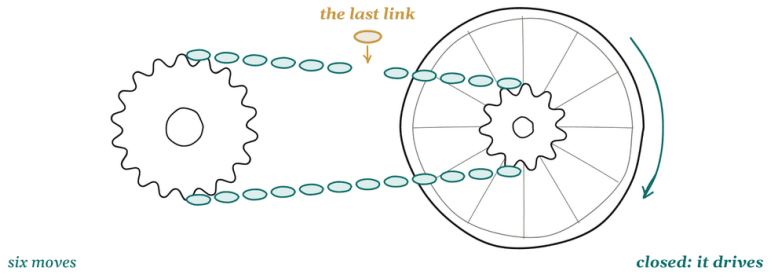


FIGURE 14 Testing and judgement. Falsifiable claims are put to real cases; intuition is trusted where feedback is fast and regular, in the body, and asked for evidence where it is not, in the tail.

XV

The Closed Loop of Business Engineering

The six moves of the method now close into a loop, and the loop connects to real action. This is what turns business engineering from a way of understanding a company into a way of running one.

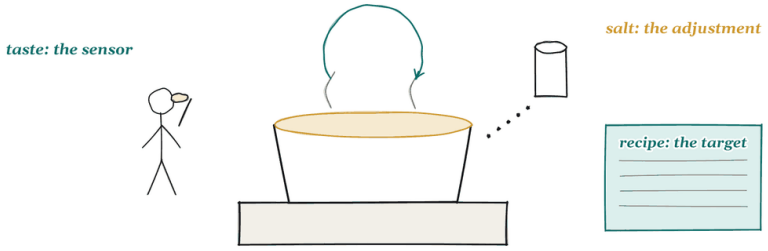


Like joining the last link of a bicycle chain: *the six links of the method only drive the wheel once the loop is closed.*

From understanding to control

Reading the distribution, abstracting, compressing and naming produce a model of the work. Testing anchors the model in reality, and judgement keeps it current. In cybernetic terms, that model is the regulator. Each loop's registers, limits, junctions and standard are how the company controls the executor that does the work. The grading of every outcome is the sensor. The operating review is the comparator, and the decisions taken there are the adjustment.

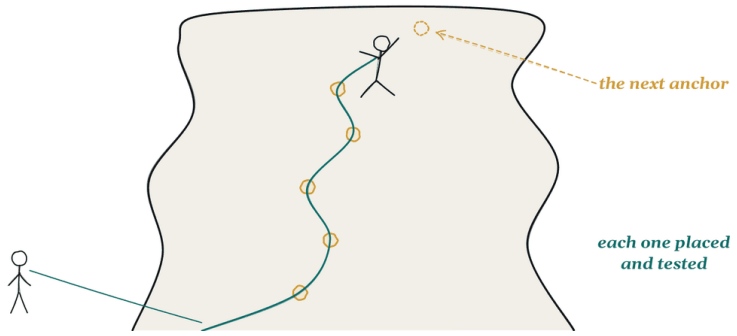
A loop built this way is closed. It acts on cases, senses whether the outcomes meet the standard, compares them with targets and adjusts. A loop without grading, without review or without a named owner is open. It acts and hopes.



Like a cook who tastes the soup: *act, taste, compare with the recipe, adjust the salt; a cook who never tastes simply acts and hopes.*

From one loop to many

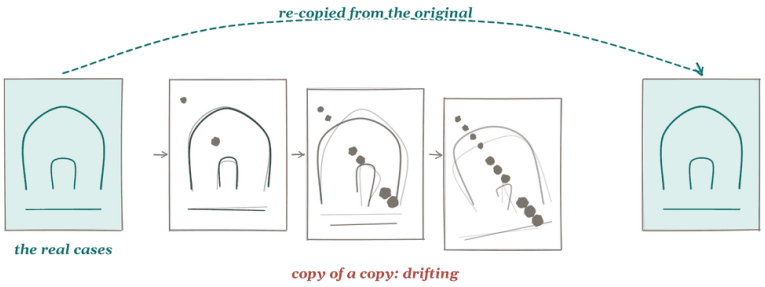
The same structure repeats at a larger scale. Each loop in the core feeds the map review: what its exceptions say about upstream loops, where people's time now sits, how readiness has changed. The map review chooses the next loop to read. The company's model of itself grows one loop at a time, and each cycle starts from a better reading than the last.



Like a climber placing anchors: *each one placed and tested makes the next pitch safer, so the route grows one secure step at a time.*

The strange loop, kept honest

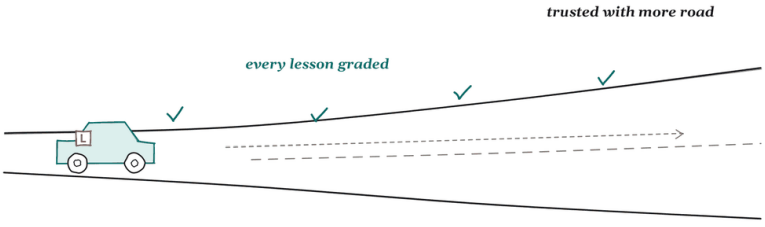
Here the strange loop from Part One becomes practical. The company builds a model of its work and acts through it, and the action changes the work. The changed work then changes the model. This reflexivity is not a flaw. It is how a company learns. But it can drift into fiction if the model stops being anchored in real cases. Registers can be copied from manuals rather than walked from cases. Standards can grade activity rather than outcomes. Reviews can stop reading the tail. The disciplines of the method are what keep the loop honest: read the real distribution, measure rather than estimate, test on real cases, and get decisions signed.



Like a photocopy of a photocopy: *each copy of the model drifts further from life unless the next one is taken again from the real original.*

Real-world action and interaction

Closed loops are what make AI useful in the world rather than in a demo. An executor that acts inside a closed loop can be trusted with more over time, because every action is graded and every error is fed back. One that acts in an open loop cannot, however capable it is. The core of business engineering, in the end, is this: build models of the work good enough to regulate it, close the loop between action and evidence, and keep the model honest as the world changes. The nine areas that follow are where that core is applied.



Like a learner driver: *because every lesson is graded and every mistake corrected, the learner is trusted with busier roads over time.*

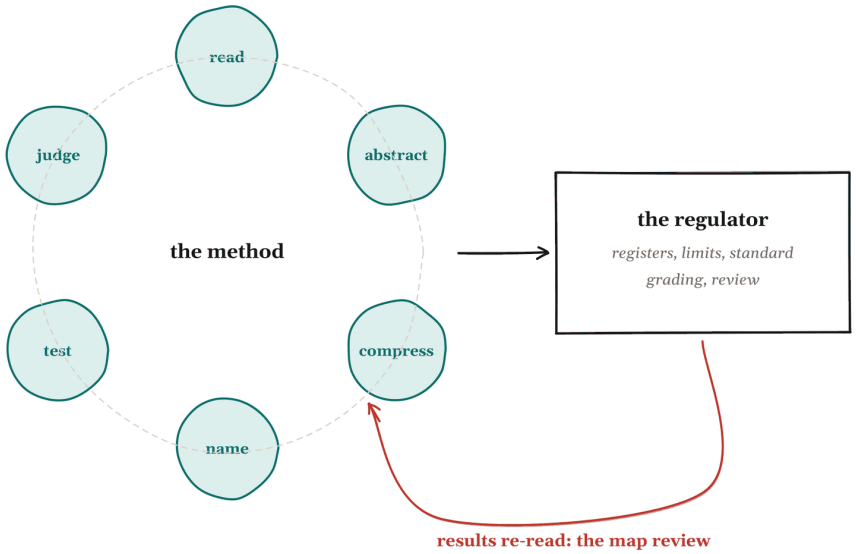


FIGURE 15 The closed loop of business engineering. The method builds the model that regulates each loop; each loop's results feed the map review; the map review chooses the next loop to read.

XVI

VTDF: Reading the Business Model

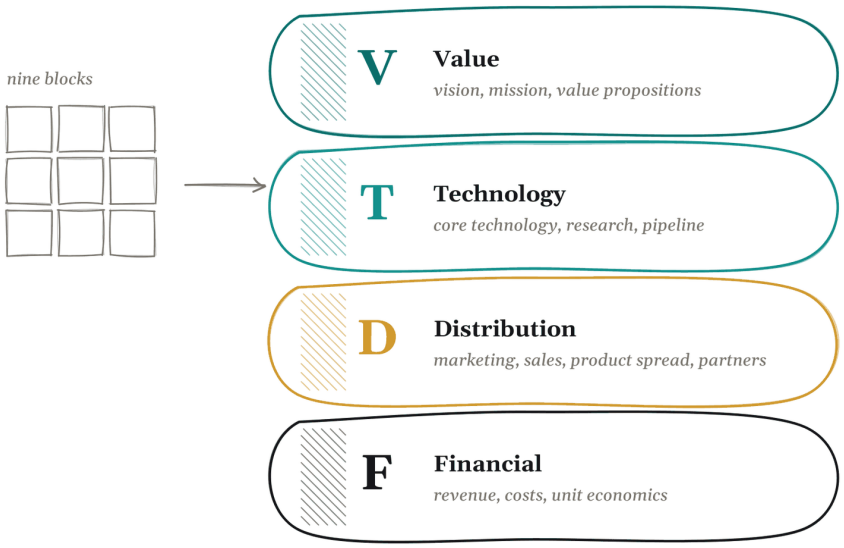
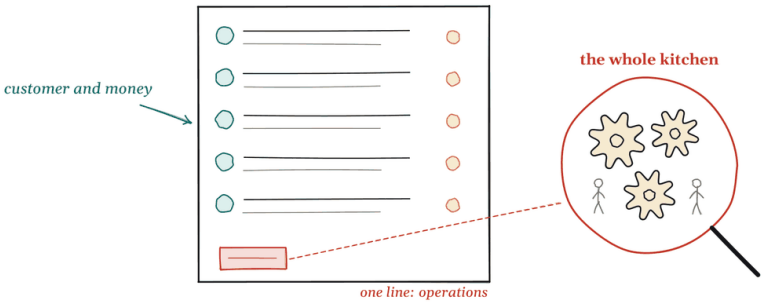


FIGURE 16 From Business Model to Business Engineering

The best-known product of the business engineering loop is VTDF, a way of reading a business model through four connected models. It was built because the business model canvas, for all its strengths, left two things in supporting roles that had become central.

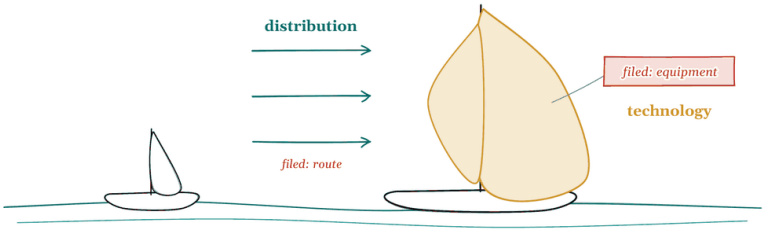
Look at where the Business Model Canvas spends its space. Five of the nine blocks face the customer and the money. Three cover operations: key activities, key resources and key partners. Everything the firm does to produce its product sits in one box, often holding a single sticky note that says "claims handling" or "underwriting".



Like a restaurant menu: *pages on dishes and prices for the diner, while the whole kitchen behind them is squeezed into one small line at the bottom.*

Others saw that gap. In 2017 Andrew Campbell, Mikel Gutierrez and Mark Lancelott published the *Operating Model Canvas*, which unfolds the operational side into value delivery chains, organization, locations, information, suppliers and a management system. It is a good tool for designing how people and systems deliver a strategy. It also assumes, reasonably for its time, that people do the work by following procedures and software does it by following instructions.

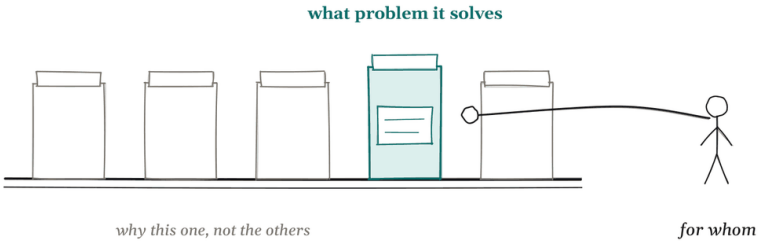
There was a second gap, and it mattered more for the businesses that grew fastest after 2010. Search engines, marketplaces, app stores and software platforms won through product engineering and distribution as much as through their offer. A canvas that treated technology as a key resource and distribution as a channel missed where the advantage sat.



Like a sailing race scored by cargo: *the old form listed the sails as equipment and the wind as a route, yet those two decided who won.*

VTDF reads a business through four models instead.

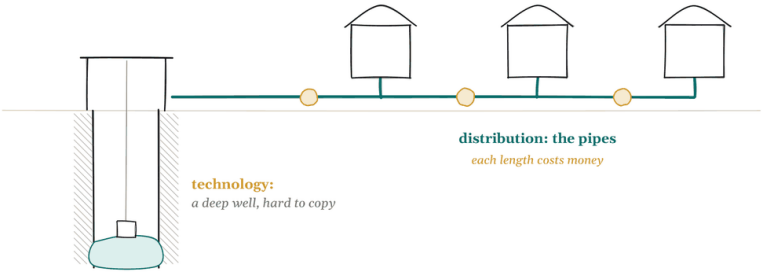
The **Value model** covers the vision, the mission and the value proposition for each kind of customer. It asks what problem the business solves, for whom, and why they would choose it.



Like a shopper reaching past a full shelf for one jar: *the value model says what problem it solves, for whom, and why they pick it.*

The **Technology model** covers the core technology, the research spend and the pipeline of innovation. It asks what the business can build that others cannot easily copy.

The **Distribution model** covers how the business reaches customers: marketing and sales, product features that spread it, partnerships and deals. It asks how the business grows, and what that growth costs.

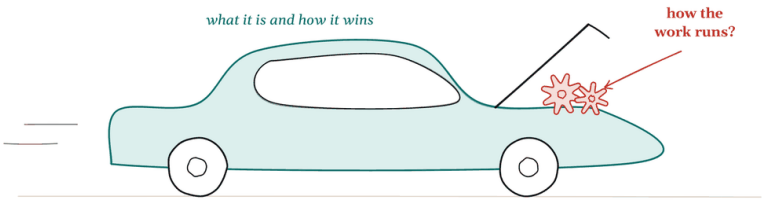


Like a deep well and its pipes: *the well is what rivals cannot easily dig, and the pipes are how water reaches homes, each at a cost.*

The **Financial model** covers revenue streams, cost structure, unit economics and the path to profit. It asks whether the business works as arithmetic.

VTDF puts technology and distribution on the same level as value and money, and it reads the four together, because in a modern business they move together. Like every product of the loop, it is a compression. It keeps what drives a business model and leaves out the rest.

VTDF still describes what the business is and how it wins. It says much less about how the work inside it runs, and almost nothing about how that work changes once agents join. That question needs its own instrument, one level down.



Like a car seen from the kerb: *you can tell what it is, how it wins and what it costs, but how it runs needs the bonnet lifted.*

XVII

Where the Work Loop Plugs In

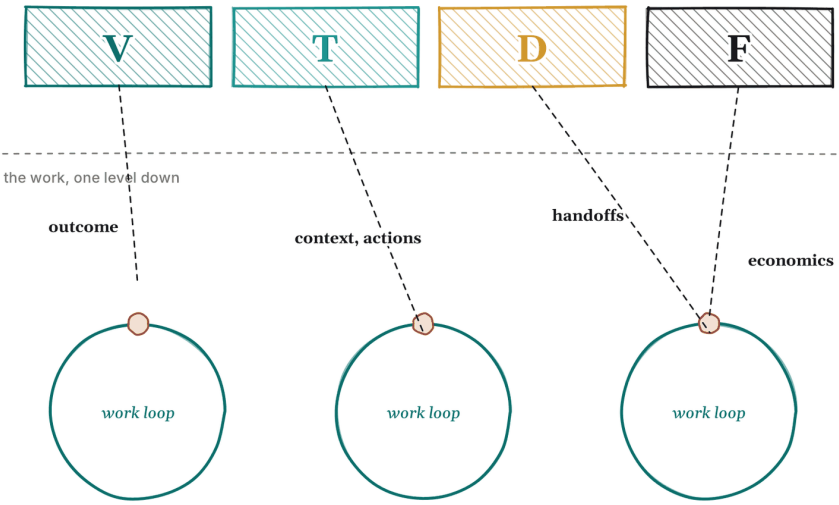
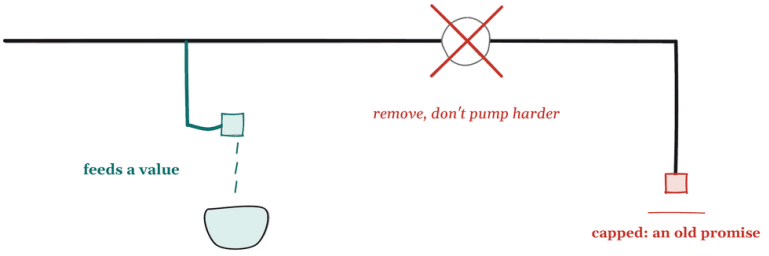


FIGURE 17 Where the Work Loop Plugs In

The Work Loop Canvas sits underneath VTDF. The two connect at four points, and each connection is a check a team can run.

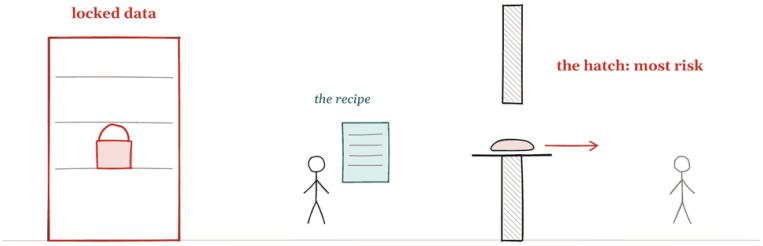
The first connection is value. Each loop's outcome should trace to a value proposition in the Value model. If a loop serves no value proposition, it is a candidate for removal, not automation. Many back-office loops fail this test when it is first applied. They exist to serve other internal loops, which exist to serve a customer promise that has since changed.



Like old plumbing traced tap by tap: *a pipe feeding a working tap stays, and one ending at a capped stub is removed, not given a bigger pump.*

The second connection is technology. The Technology model sets what the firm's executors can read and change. In canvas terms, it fills the context and actions nodes. A loop that needs data the firm cannot reach, or needs to write to a system nobody can safely open, is blocked at the technology level. No amount of process design will fix it.

The third connection is distribution. The Distribution model meets the loop at its handoffs, wherever a loop passes work to a customer, a partner or a channel. These handoffs carry the most risk, because a mistake there reaches someone outside the firm.

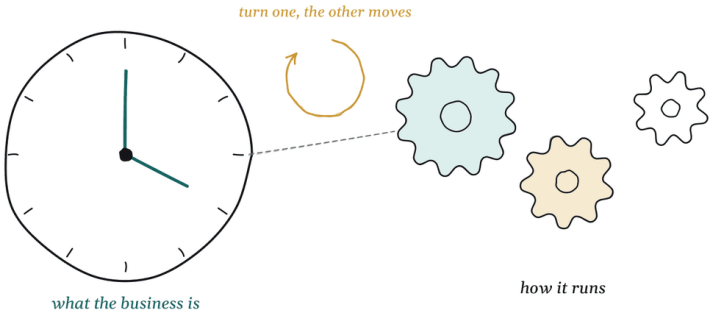


Like a kitchen with a locked pantry and a serving hatch: *no recipe helps when the pantry is locked, and the hatch is where a mistake reaches the diner.*

The fourth connection is money. The Financial model receives each loop's economics. Cost per accepted outcome, added up across all the loops, is the firm's cost structure seen from the inside. When the numbers do not add up, either the loops are badly measured or the cost structure includes work no loop accounts for.

So the two instruments work as a pair. **VTDF says what the business is. The work loops say how it runs.** A change in one should show up in the other. A new value proposition should create or change a loop. A loop that gets much cheaper should change the Financial model, and perhaps the price.

The method says what to look for. This part says why the core, rather than the market, is where AI now has to work, and why the unit of design has to change.



Like a clock's face and its gears: *the face says what the business is, the gears say how it runs, and turning one moves the other.*

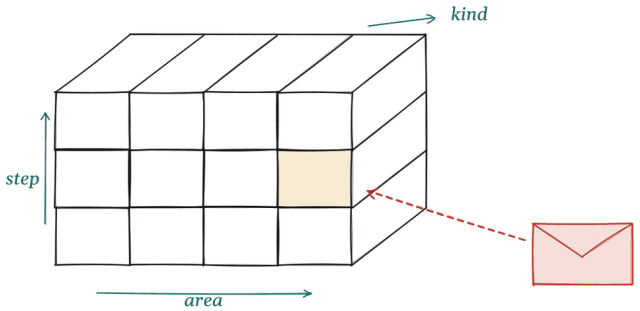
PART THREE

Reading the Business

XVIII

The Shape of This Body of Knowledge

A body of knowledge needs an architecture, so that a reader can find any concept and see how it relates to the others. This one has three dimensions.



Like a post-office sorting cabinet: *every concept lands in one pigeonhole found by three coordinates, its area, its step in the lifecycle and its kind.*

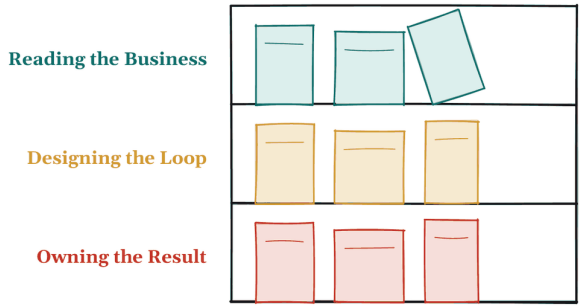
Nine areas

The first dimension is the kind of knowledge. There are nine areas, grouped into three parts.

Part Three, Reading the Business, holds three areas. Distributions covers how cases spread and what that implies. Value and the Map covers which loops matter and in what order. Work and Process covers how work actually runs.

Part Four, Designing the Loop, holds three more. Rules and Decisions covers what the loop must follow and who decides what. The Executor covers what does the work and what it is made of. Control and Acceptance covers how anyone knows the work is good and how it is stopped when it is not.

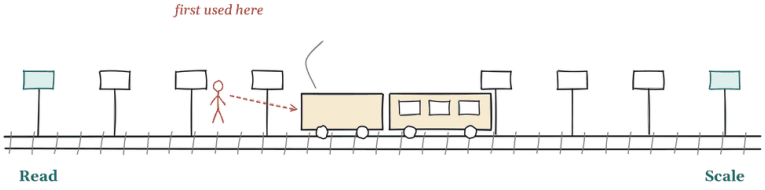
Part Five, Owning the Result, holds the last three. People and Accountability covers who answers for what. Economics covers what the work costs and what it is worth. Governance covers how decisions are recorded and how they meet the company's obligations.



Like a bookcase with three shelves of three volumes: *reading the business on top, designing the loop in the middle, owning the result below.*

Ten steps

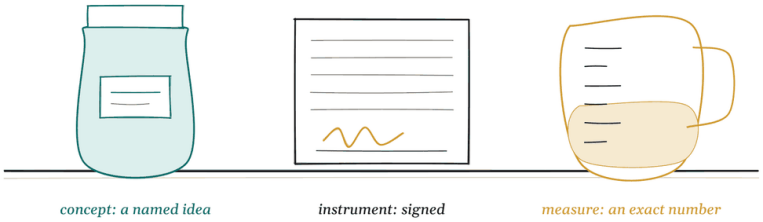
The second dimension is time. Every concept has a place on the ten-step embedding lifecycle: Read, Map, Strip, Sort, Design, Shadow, Bound, Own, Core and Scale. Each area opens with a table showing where its concepts are first used. Appendix E gives the full matrix of areas against steps.



Like a railway line with ten stations from Read to Scale: *every concept has a stop where it first boards the train.*

Three kinds of content

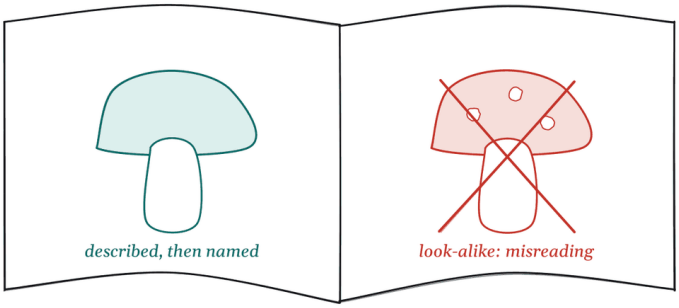
The third dimension is the kind of content. Each area contains concepts, which are ideas with names; instruments, which are documents or drawings a team fills in and signs; and measures, which are numbers with exact definitions. The entries say which is which. Part Twelve gathers the notation for the instruments and the formulas for the measures, so they are defined in one place.



Like a kitchen shelf: *a labelled jar is a concept, a signed recipe card is an instrument, and a marked measuring jug is a measure.*

How entries are written

Every entry follows the same discipline, though not the same template. It defines the concept in plain words before naming it. It says why the concept matters, usually by describing what goes wrong without it. It shows how the concept is used, often with the composite manufacturer. And it names the common misreadings, because in a young discipline the misreadings spread faster than the definitions.



Like a mushroom field guide: *each entry describes the thing plainly, then names it, then warns about the look-alike that people mistake for it.*

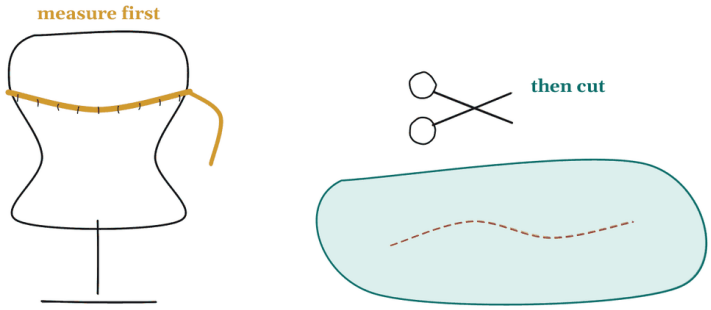


FIGURE 18 The shape of this body of knowledge. Nine areas of knowledge read against the ten steps of the embedding lifecycle; filled marks show where an area is central, open marks where it contributes.

XIX

Distributions

This area covers how the cases a business handles are spread, and what that spread implies for design. It comes first because every later decision depends on it. Which loop to start with, what an executor may do alone, where people must decide, how long to test and how to report results all turn on the shape of the cases.



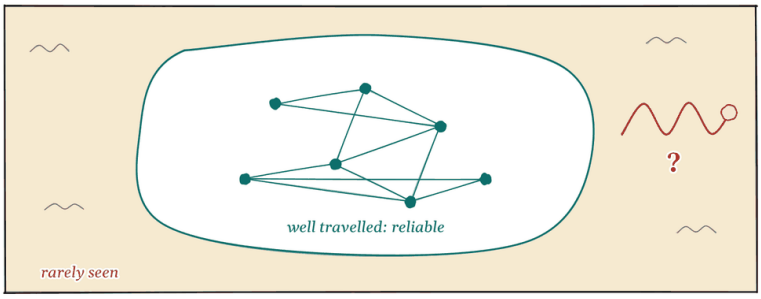
Like a tailor who measures before cutting: *the shape of the cases decides where to start, what runs alone, where people decide and how long to test.*

Where it comes from

This area applies the first move of the method. Its foundations are statistics and the science of complex systems: normal and heavy-tailed shapes, and the mechanisms that produce them. In AI terms, a learned model is a compression of the distribution it was trained on. So the shape of a loop's distribution predicts where the model will be reliable and where it will not.

Concept	Kind	First used at
Case	Concept	Read
Distribution	Concept	Read
Normal and heavy-tailed shapes	Concept	Read
Body and tail	Concept	Read
Share of cases against share of value	Measure	Read
Segmenting by value, difficulty and risk	Concept	Read, Shadow
Distribution shift	Concept	Core, Scale

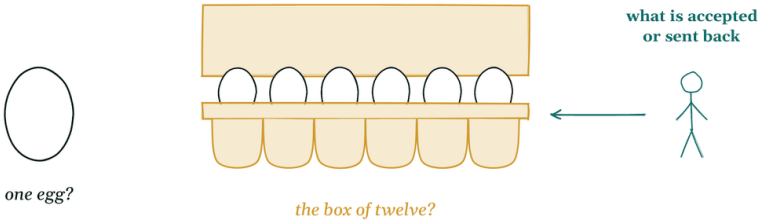
Concept	Kind	First used at
Where intuition can be trusted	Concept	Shadow, Core



Like an old map drawn from travellers' reports: *sharp where many people went, blank where few did, just as a model is reliable only where its training cases were dense.*

Case

A case is one unit of work that a loop handles from start to finish: one invoice, one claim, one support ticket, one job application. Defining the case is the first act of reading a loop, and it is less obvious than it sounds. Is the case one invoice, or one supplier's monthly batch? Is it one ticket, or one customer's problem across three tickets? The right answer is the unit whose outcome a receiver can accept or reject. Getting it wrong distorts every count that follows.



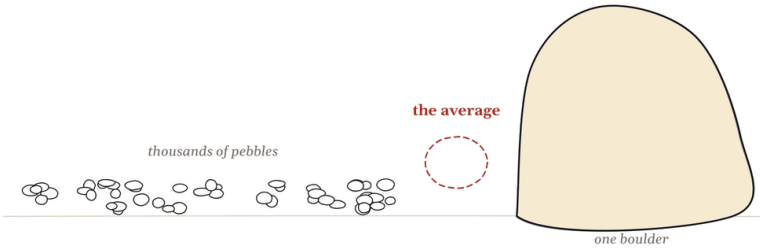
Like a grocer choosing the unit of sale, one egg or the box: *the right case is the unit a receiver can accept or send back.*

Distribution

The distribution of a loop is how its cases spread along some dimension: their value, their difficulty, their risk, or the time they take. It is read by counting cases in bands and comparing the bands. A distribution answers a simple

question that most companies cannot answer about their own processes: what do our cases actually look like?

A common misreading is to treat the average case as representative. In many loops no actual case looks like the average. A loop with thousands of tiny invoices and a few enormous ones has an average invoice that hardly anyone ever sees.



Like the average stone on a beach of pebbles and one boulder: *it describes no stone that is actually there, just as the average case may not exist.*

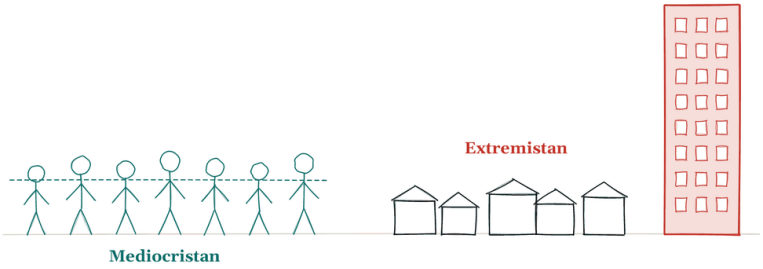
Normal and heavy-tailed shapes

Distributions come in two broad shapes, and most real loops are a mixture.

In a roughly normal distribution, cases cluster around a typical value and extreme cases are rare and not very extreme. Weekly hours on a timesheet behave like this, and so do routine address changes. In these loops, averages are meaningful, rules work well, and a sample tells you a lot about the whole.

In a heavy-tailed distribution, a small share of cases carries most of the value or the risk, and the largest cases can be enormous relative to the typical one. Invoice values, insurance claims, customer revenue and contract sizes often behave like this. The commonest heavy-tailed shape is called a power law. In these loops, averages mislead, a sample can miss the cases that matter most, and the design has to treat the tail differently from the body. The essayist Nassim Taleb popularised a pair of names for these two worlds: "Mediocristan" for the first and "Extremistan" for the second.

The practical test is simple. Rank the cases by value and ask what share of the total the top few percent carry. If the top 5% carry 10% of the value, the loop is close to normal. If they carry half, it is heavy-tailed.



Like people's heights against the buildings of a city: *heights cluster near a typical value, while one tower can dwarf a whole street of houses.*

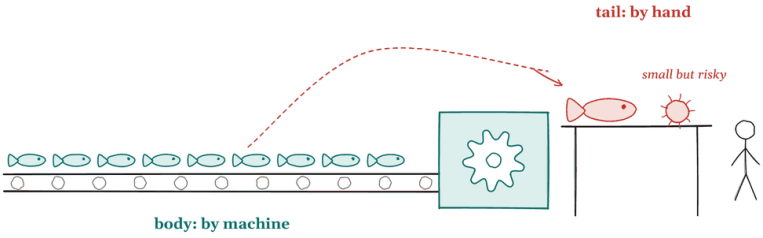
Body and tail

Most real loops have two parts. The body is the large share of routine cases that resemble each other. The tail is the small share of cases that carry most of the value, the risk or the difficulty.

The distinction is the most useful single idea in embedding. An executor that decides can usually run the body, because its cases repeat and feedback is fast. People should usually hold the tail, because the cost of error is high and each case is different. Many design choices follow from it. Junctions sit mostly in the tail. Shadow results are reported separately for each part, and so are the economics. And the first bounded scope usually covers the body.

At the manufacturer, 81% of supplier invoices were under €5,000 and carried 14% of the spend. That was the body. The 3% over €50,000 carried 46% of the spend. That was the tail.

A common misreading is to draw the line between body and tail by volume alone. A case can be small in value and still belong in the tail because it is risky, such as a first invoice from a new supplier with new bank details.

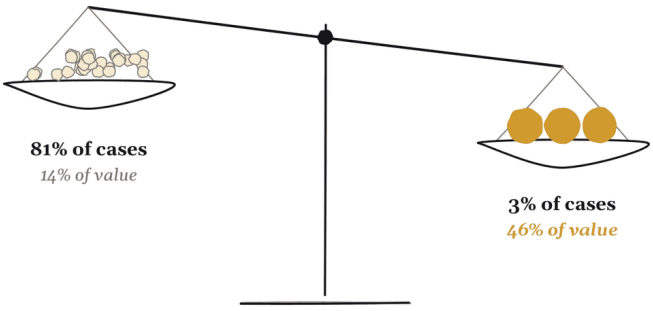


Like a fish market: *the many ordinary fish run through the machine, while big catches and small poisonous ones go to a person's table.*

Share of cases against share of value

The basic measure of a distribution compares, band by band, the share of cases with the share of value. It is drawn as paired bars, one for each share, for each band. Where the bars are similar in height, the loop is close to normal. Where the value bar dwarfs the case bar in the top band, the tail is heavy.

The measure is easy to compute from a few months of data. Its definition is given in Part Twelve. It should be read at Step 1, before a loop is chosen, and again at Step 10, after the body has been embedded.

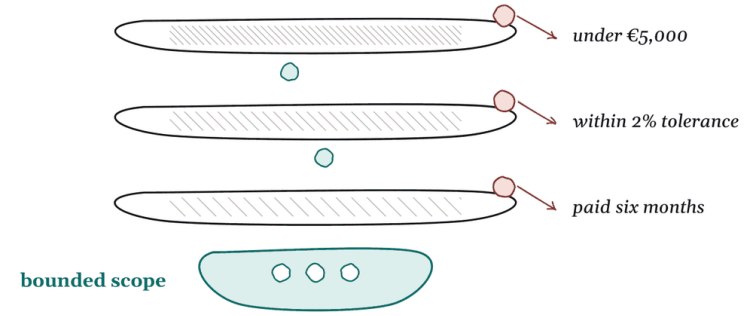


Like a set of scales with a heap of copper coins on one pan and three gold coins on the other: *the few gold coins outweigh the heap.*

Segmenting by value, difficulty and risk

A loop has more than one distribution. Its cases spread differently by value, by difficulty and by risk, and the three do not always line up. A large invoice from a long-standing supplier with a perfect match is high in value but low in difficulty. A small invoice for a new supplier with an unusual tax treatment is low in value but high in difficulty and risk.

Good designs read all three. Bounded scope is usually defined on several dimensions at once, as at the manufacturer: under €5,000, within a 2% tolerance, from a supplier paid for at least six months. Each condition cuts a different distribution.

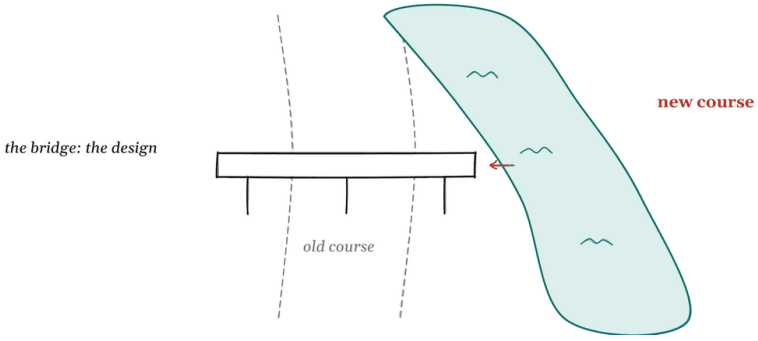


Like three stacked sieves with different meshes: *value, difficulty and risk each catch different cases, and only those passing all three enter the bounded scope.*

Distribution shift

Distributions change. A new product line brings new kinds of case. A regulation changes which cases are risky. A large customer changes how it orders. A shift can quietly move cases from the body to the tail, where the executor should not be deciding alone.

Shifts are caught in two ways. The first is grading every outcome and watching the share that meets the standard, which falls when the executor meets cases it handles badly. The second is re-reading the distribution itself at each map review. Both are needed. Grading catches the effect, and re-reading explains the cause.



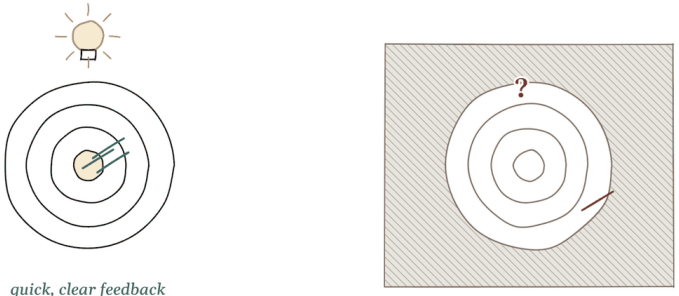
Like a river that changes course after a storm: *the bridge still stands, but the water now runs past its end, and only re-reading the river shows why.*

Where intuition can be trusted

Practitioners develop a feel for which cases are routine and which are not, and that feel guides many small decisions. Research by Daniel Kahneman and Gary

Klein suggests that expert intuition is reliable only where the environment is regular and gives quick, clear feedback. It is unreliable where cases are rare, varied and slow to reveal whether a decision was right.

The body of most loops is the first kind of environment, and the tail is often the second. So the same expert who can be trusted to say "that invoice is fine" in seconds should be asked to show evidence on a large disputed one. The same logic applies to the executor. A high agreement rate on the body says little about its judgement on the tail.

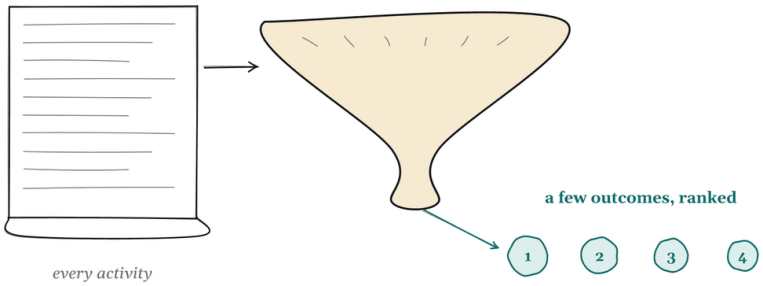


Like a darts player who sees every throw land in a lit room: *skill is trustworthy where feedback is quick and clear, not in the dark.*

XX

Value and the Map

This area covers how a company decides which work matters, which loop to embed first, and how its loops connect. It turns a company's long list of activities into a small number of outcomes that can be compared, ordered and owned.



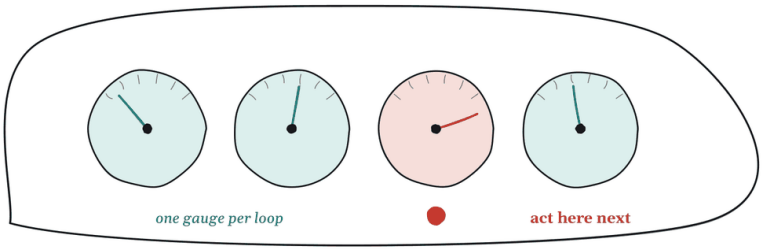
Like a funnel: *the long list of a company's activities goes in, and a few outcomes come out that can be compared, ranked and owned.*

Where it comes from

This area applies abstraction at the level of the whole company. It reads the business from the inside, loop by loop, and connects that view to the business model read from the outside. In cybernetic terms, the map is the company's model of its own work at the highest level: the regulator for deciding where to act next.

Concept	Kind	First used at
Work loop	Concept	Read
Outcome	Concept	Read
Value to the core	Measure	Read
Readiness	Measure	Read
Work Loop Map	Instrument	Read
The four regions of the map	Concept	Read
The first loop	Concept	Read
Chains and handoffs between loops	Concept	Read, Scale

Concept	Kind	First used at
The map review	Instrument	Scale
The inside-out complement to the business model	Concept	Read



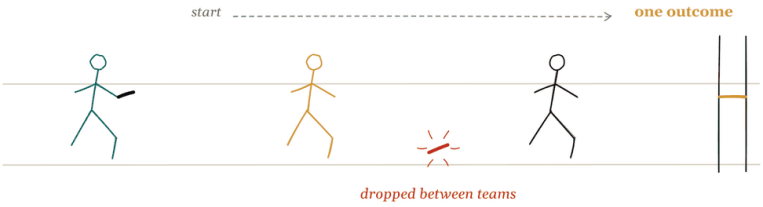
Like a car's dashboard: *built from readings taken inside the engine, it is the driver's model of the machine and tells them where to act next.*

Work loop

A work loop is the unit of work that produces one outcome end to end, with everything needed to produce it well. That means its rules, context, actions, decisions, junctions, exceptions, handoffs, standard, people and economics. It is smaller than a department and larger than a task. It usually crosses teams.

The loop is the unit of embedding. AI is not embedded into "finance" or "customer service". It is embedded into a loop, such as supplier invoice approval or support ticket resolution, one at a time.

A common misreading is to equate a loop with a team's process. A team's process is often a fragment of a loop, and a loop often includes steps no single team owns. That gap is where accountability gets lost.

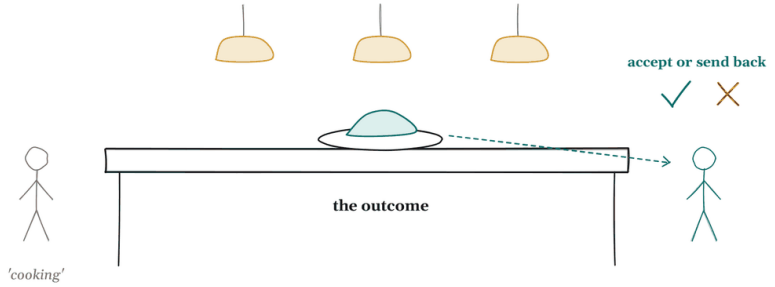


Like a relay race: *the loop is the whole race to the finish, run across several teams, and the baton is usually dropped at a handoff.*

Outcome

The outcome is the result a loop produces, stated so that someone outside the team can tell an accepted result from a rejected one. "An approved supplier invoice, matched and posted for payment on time" is an outcome. "Process invoices" is an activity.

Stating the outcome is harder than it sounds and more useful than it looks. It names the receiver of the work. It also exposes loops whose teams disagree about what they produce, and it gives the standard something to grade.

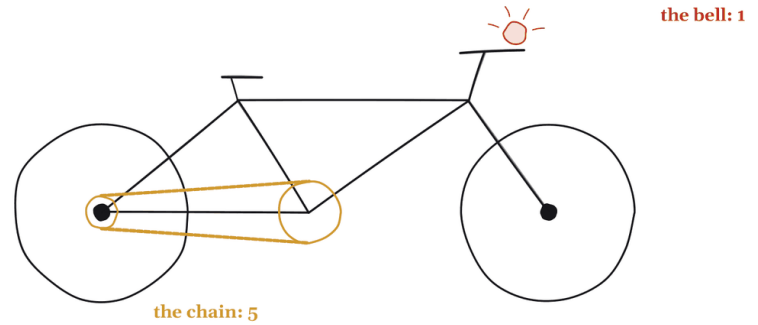


Like a dish placed on the kitchen pass: *the waiter can take it or send it back, whereas 'cooking' is only an activity nobody can accept.*

Value to the core

Value to the core scores how much a loop matters to the business, from one to five. It combines volume, cost, effect on customers, effect on cash and exposure to risk. The score is a judgement, not a calculation, made by the people who know the work and argued over in the room.

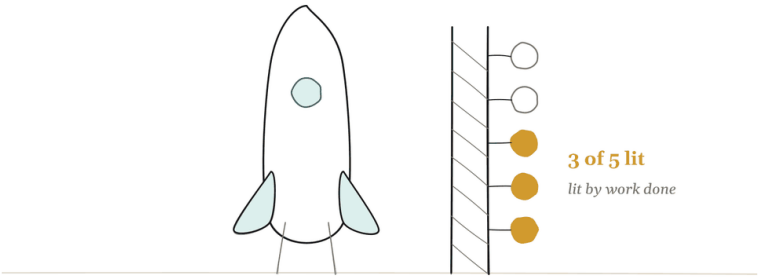
A common misreading is to score value by how impressive the AI would look. Some of the highest-value loops are dull. Supplier invoices, goods receipts and payment runs rarely appear in demos and often carry more value than the customer chatbot.



Like a bicycle's chain and its bell: *the shiny bell draws the eye, but the dull chain carries the ride, and value is scored on the chain.*

Readiness

Readiness scores how close a loop is to being embedded, also from one to five. It asks how well the loop's rules are known, how measurable its outcome is, how much of its context is already in systems, how willing its owner is, and how many unsettled disagreements it contains. Readiness rises as work is done, both on the loop itself and on its neighbours.



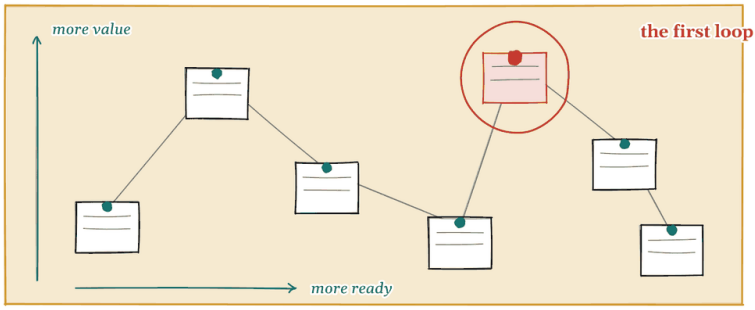
Like a rocket on the launch pad with a row of check lamps: *readiness is how many are lit, and each one is lit by work done.*

Work Loop Map

The Work Loop Map places every loop of a company on two axes: value to the core and readiness. Lines between loops show where one hands work to another. The map is the instrument of Step 1 and Step 10. It is drawn in a working session with the heads of the main functions, and signed by the executive sponsor with the first loop marked.

The map's main use is to make choices visible. Leadership teams rarely compare their processes side by side. The map forces the comparison, and the

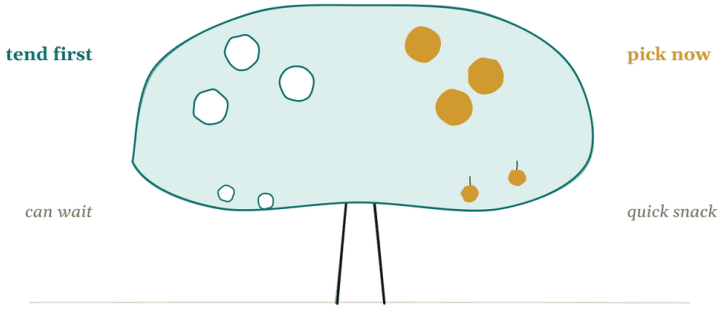
argument over scores is often the first honest conversation about which work matters.



Like a corkboard with every case pinned up and strung together: *once all the loops hang side by side, the room has to compare them and pick one.*

The four regions of the map

The map divides into four regions. High value and high readiness is where to start. High value and low readiness is where to map before building: these loops matter, but their rules or data are not ready, and the lifecycle's early steps are the work to do first. Low value and high readiness holds quick wins, which teach little and save little. Low value and low readiness can wait.

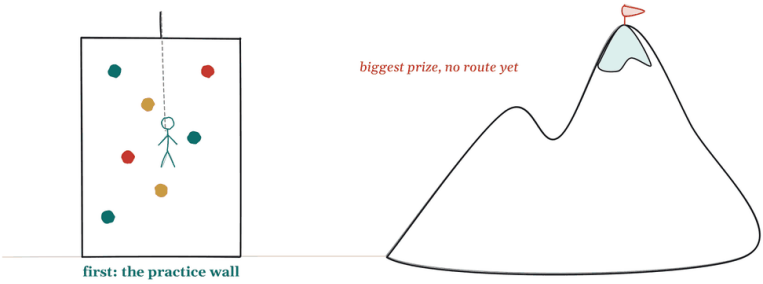


Like an orchard tree: *pick the high ripe fruit now, tend the high unripe fruit first, take low ripe ones as snacks, and leave the rest.*

The first loop

The first loop has a job beyond its own savings: it teaches the company the lifecycle. It should have repeating cases, so evidence accumulates quickly. Feedback should be fast, its outcome measurable, its owner willing, and its distribution should have a body and a tail. The most valuable loop on the map is often not the right first loop, because its readiness is low.

At the manufacturer, order fulfilment scored highest on value, but its rules were scattered across three plants. Supplier invoice approval scored four on value and three on readiness, and it went first.

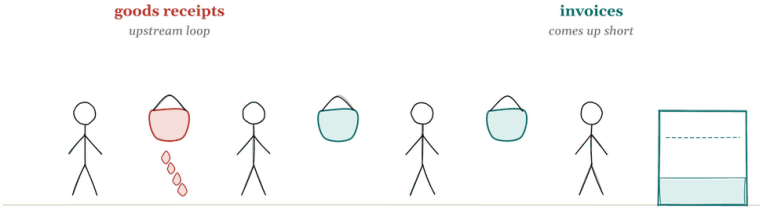


Like learning to climb on the practice wall before the summit: *the first loop is chosen to teach the method fast, not because it is the biggest prize.*

Chains and handoffs between loops

Loops connect into chains, sequences in which one loop's outcome is the next loop's input. Common chains include lead to cash, procure to pay, hire to retire, record to report, issue to resolution and order to delivery. The map draws the handoffs between loops as lines.

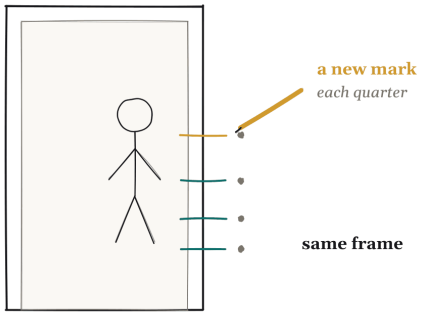
Chains matter because a loop's performance depends on its neighbours. Once the manufacturer's invoice loop ran in the core, 38% of its remaining exceptions came from missing goods receipts, a problem in the loop upstream. Loops in the same chain often can only scale together.



Like a bucket brigade: *each loop passes its output to the next, so a leaking bucket upstream leaves the next loop short however well it works.*

The map review

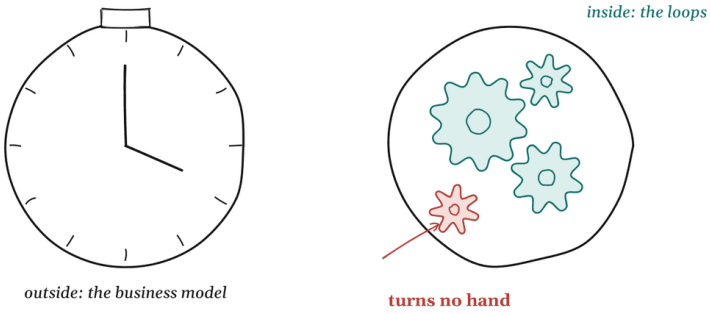
The map review is the quarterly update of the map, signed by the sponsor at Step 10. It updates readiness, records the state of each lifecycle in progress, reads the edges of embedded loops for the next candidates, and chooses what to read next. It is the same instrument that started the lifecycle, one cycle later.



Like height marks on a doorframe: *the same frame, read again each quarter, records how far each loop has grown and where to look next.*

The inside-out complement to the business model

The map is the inside-out complement to the business model. A business model describes what a company is and how it wins, read from the outside: its value, technology, distribution and financial models. The map describes how its work runs, read from the inside. Each loop's outcome should trace to a value the business creates. Each loop's economics should roll up into the financial model. A loop whose outcome traces to nothing the business values is a candidate for removal, not for embedding.

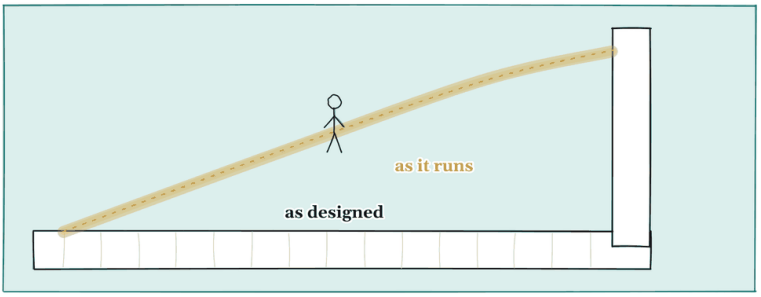


Like a watch's face and its movement: *the face is read from outside, the gears run inside, and a gear that turns no hand should come out.*

XXI

Work and Process

This area covers how work actually runs inside a loop: its steps, its waits, its handoffs, its workarounds and its costs. It is the knowledge most companies believe they have and most do not, because their process documents describe how work was designed rather than how it runs.



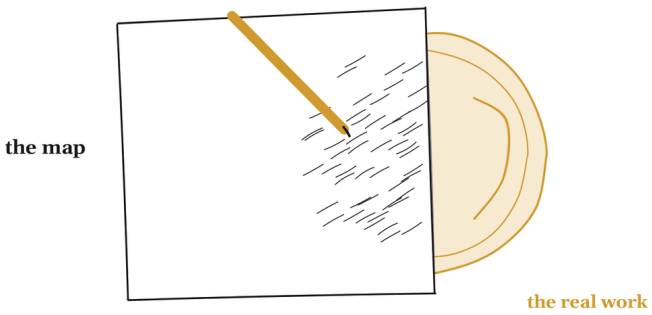
Like the paved path across a lawn and the worn track beside it:
documents show how the work was designed, the worn track shows how it runs.

Where it comes from

This area anchors the symbols in the world. A process map is only meaningful if it mirrors how the work really runs, which is why it is drawn with operators and walked with real cases. Its compensations are the accumulated rules of an older executor, kept long after the constraints that justified them.

Concept	Kind	First used at
Process map	Instrument	Map
Walk-through	Concept	Map
Step, wait and handoff	Concept	Map
Flow efficiency	Measure	Map, Core
Baseline	Instrument	Map
Compensation	Concept	Strip
Strip ratio	Measure	Strip
Topology and accretion	Concept	Strip
Rework	Concept	Map, Strip

Concept	Kind	First used at
The map and the log	Concept	Map
Generated steps	Concept	Design

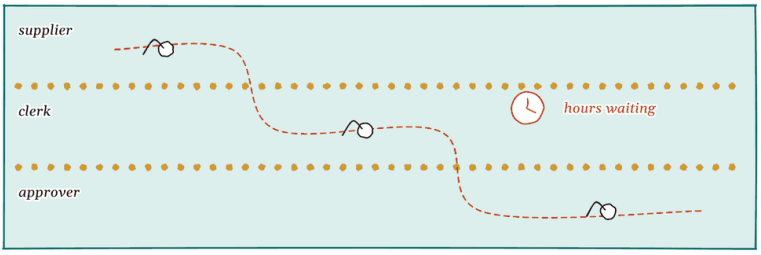


Like a pencil rubbing over a coin: *a process map means something only when it is pressed directly from the real work, with the people doing it.*

Process map

The process map records how a loop's work runs today. It is drawn as swimlanes: one row per role, team or outside party, with steps running left to right in the order they happen. Each step records the system or channel it uses, the minutes of hands-on work it takes, and the hours the case waits before the next step begins. Workarounds and rework are marked.

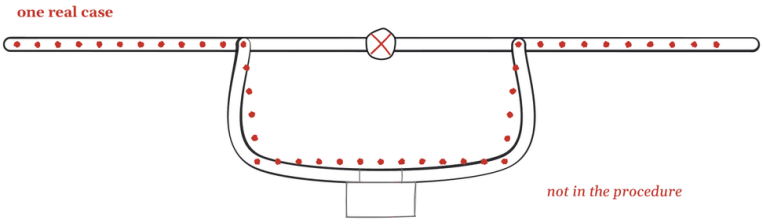
The map is drawn with the people who do the work, not with those who manage it. Managers describe the process as designed. Operators describe it as it runs. For one loop, eight to twenty steps is usually the right grain.



Like a swimming pool's lanes: *each role swims its own lane, the case moves left to right, and every crossing and pause is marked.*

Walk-through

A walk-through follows real cases through the map, from arrival to outcome, including at least one that went wrong. It is the quickest way to find steps the official process omits. At the manufacturer, the walk-through revealed that half the invoices still arrived as PDFs that a clerk printed and scanned, and that budget owners approved on paper every Friday. Neither appeared in the procedure.

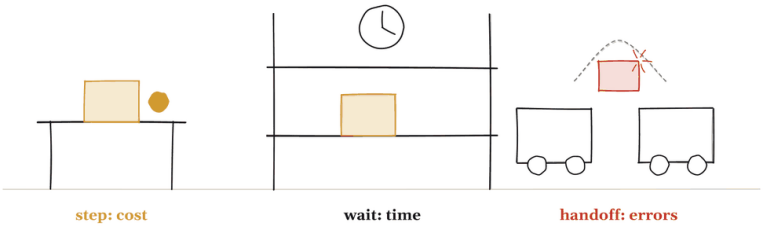


Like dye dropped into a plumbing system: *follow one real case through the pipes and it shows the bypass no drawing ever recorded.*

Step, wait and handoff

A step is a piece of work someone does. A wait is the time a case spends between steps, untouched. A handoff is a point where the case moves from one lane to another.

These three are the grammar of a process map. Steps are where cost sits. Waits are where elapsed time sits, usually most of it. Handoffs are where errors and delays gather, because each one requires the receiver to understand what the sender meant.

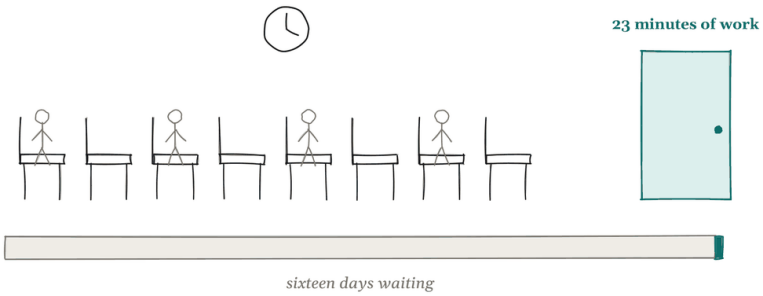


Like a parcel's journey: *a few minutes on the sorting bench, days on a shelf, and a knock each time it changes vans.*

Flow efficiency

Flow efficiency is the hands-on time a case receives divided by the total elapsed time from arrival to outcome. In most office processes it is very low, often under 5%. At the manufacturer, twenty-three minutes of work took sixteen days, a flow efficiency well under 1%.

The measure matters because it shows where improvement is possible. An executor that does each step faster leaves the waits untouched. A loop with 1% flow efficiency can only become much faster by removing waits, which means removing handoffs, batch steps and queues.

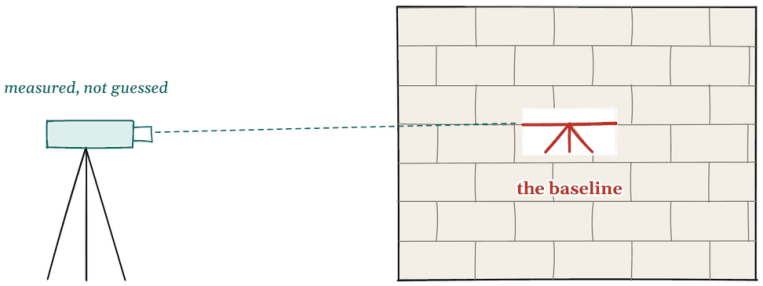


Like a clinic visit: *twenty-three minutes with the doctor after sixteen days in the waiting room, so a quicker doctor barely changes the total.*

Baseline

The baseline is the measured performance of a loop before it changes. It records five figures on a real sample of cases: volume, fully loaded cost per case, elapsed time, the share of cases raising an exception, and the share right the first time. It is signed by the process owner at Step 2.

The baseline is the reference for every later claim. It must be measured, not estimated. The usual period is several weeks, and the sample should include the tail.



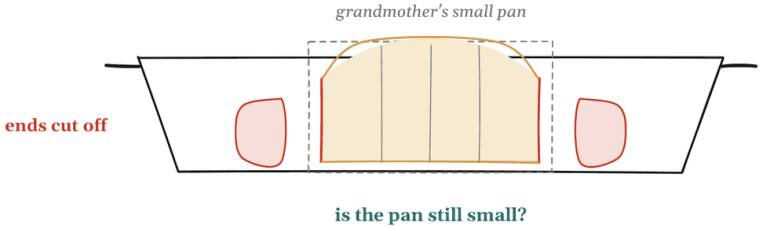
Like a surveyor's benchmark carved into a wall: *every later height is measured from it, so it has to be measured carefully, not guessed.*

Compensation

A compensation is a step that exists only to work around a constraint. A batch step exists because a system once ran overnight. A handoff exists because two teams once sat apart. A second review exists because errors were once frequent. A printed form exists because approvals were once on paper.

Compensations accumulate over years and become indistinguishable from real work. People are trained in them, controls rely on them, and improvement projects polish them. Attach an executor to a process full of compensations and it will run the workarounds faster, while every wait between them stays.

For each step, ask which constraint it compensates for and whether that constraint still applies. If it is gone, strike the step. If it still applies, keep the step and record the constraint, because a junction or the standard may meet it better. The register of these answers is signed at Step 3.

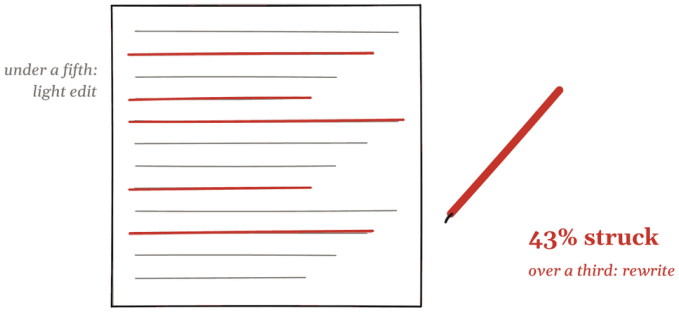


Like cutting the ends off a roast because grandmother's pan was small: *for each step, ask whether its constraint still applies, and strike it if not.*

Strip ratio

The strip ratio is the share of a map's steps struck as compensations. Under about a fifth, the process is mostly real structure, and embedding is largely a

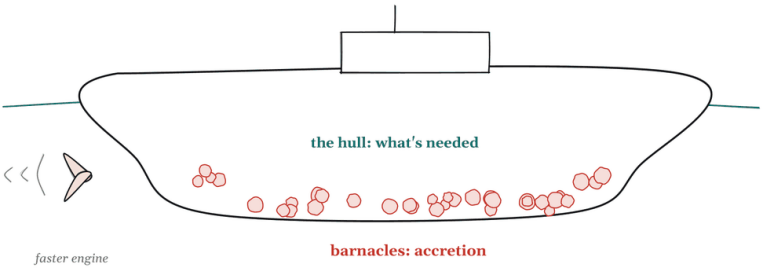
matter of letting an executor run what remains. Over about a third, the process is mostly accumulated workaround, and embedding is a redesign. The manufacturer's invoice loop had a strip ratio of 43%.



Like a manuscript marked with a red pen: *the share of lines struck tells you whether it needs a light edit or a rewrite.*

Topology and accretion

The steps an outcome genuinely requires are the loop's topology. The layer of compensations around them is accretion. The distinction gives the strip ratio its meaning, and it carries a warning. Automating a topology produces a faster loop. Automating an accretion produces a faster version of a loop that should not exist in its current form.

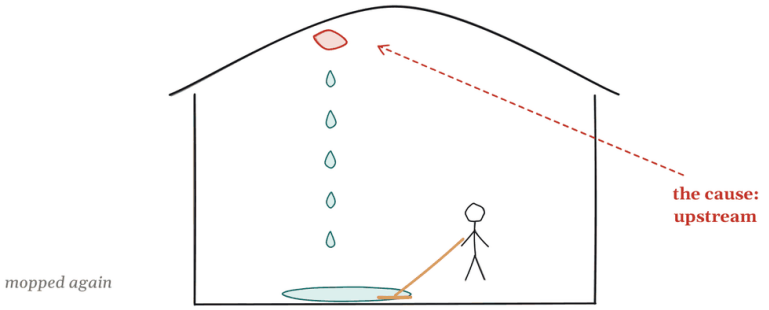


Like a ship's hull crusted with barnacles: *the hull is what the voyage needs, the barnacles are accretion, and a bigger engine just drags them faster.*

Rework

Rework is any step that redoes something that should have been right the first time: chasing a missing document, correcting a coding error, re-requesting information already supplied. It is marked on the map separately from compensations. A compensation is a step designed around a constraint.

Rework is a step caused by an error upstream. Rework usually points to a problem in a neighbouring loop or in the context the loop receives.

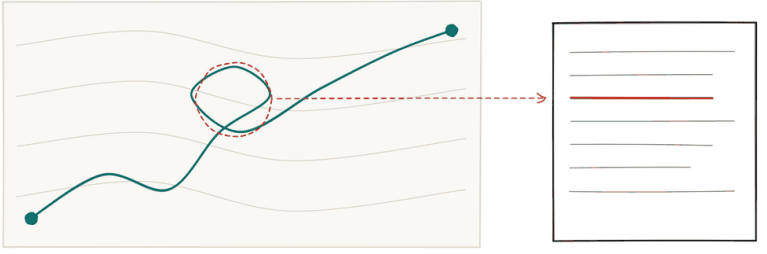


Like mopping the same floor again and again under a leaking roof: *rework is the mop, and the cause usually sits upstream in another loop.*

The map and the log

A process map records what people say happens. Most company systems also record what actually happened, as event logs: time-stamped records of each step each case went through. Process mining is the analysis of those logs to reconstruct the real flow of work. It was developed by Wil van der Aalst and others from the late 1990s.

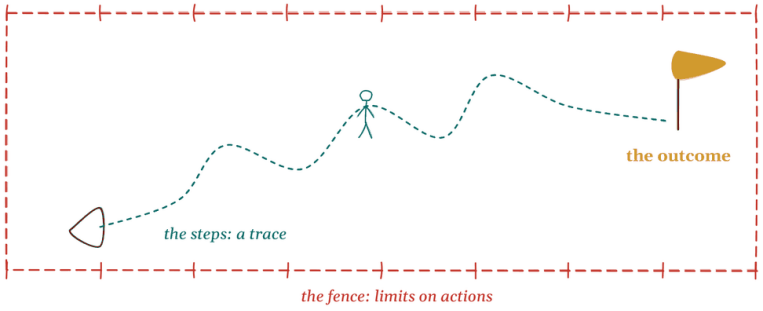
The two views complement each other. The log shows variants, loops and waits that nobody mentions in a workshop. The map explains why they happen, and marks which steps are compensations. Where logs exist, the map should be checked against them. Where they disagree, the log is usually right about what happened, and the people are usually right about why.



Like a hiker's GPS track beside their trail diary: *the track shows where they actually went, the diary explains why they turned back.*

Generated steps

In some loops the executor plans its own steps for each case: researching a proposal, investigating an incident, drafting an analysis. There is no fixed sequence to map in advance. In these loops the design moves up a level. The outcome, the rules, the limits on actions, the junctions and the standard become the design, and the steps become a trace of what the executor did. Mapping the current process still matters, because it gives the baseline and shows what the work achieves today.



Like an orienteering race: *you set the finish, the boundary and the rules rather than the route, and the route is recorded afterwards as a trace.*

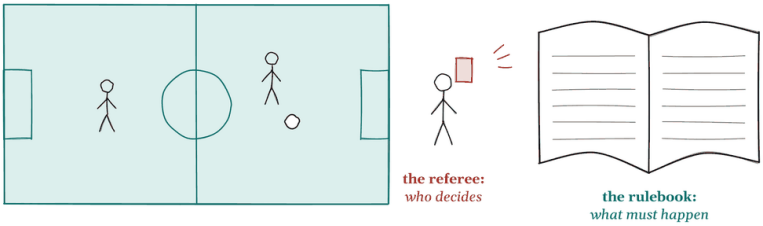
PART FOUR

Designing the Loop

XXII

Rules and Decisions

This area covers what a loop must follow and who decides what inside it. It is where the company's unwritten agreements are written down, and where most of the important findings of an embedding programme appear.

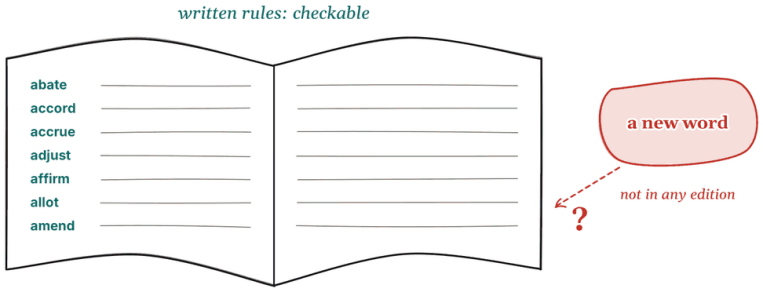


Like a football match: *the rulebook says what every player must follow, the referee says who decides, and writing both down shows where nobody agreed.*

Where it comes from

This area builds the symbolic half of the neurosymbolic loop. Its foundations are the laws of thought and their limits: rules written explicitly can be checked and audited, and no rule set is complete. The four classes of decision say where rules suffice, where judgement within limits suffices, where people must decide, and where the organisation itself must first decide.

Concept	Kind	First used at
Rule and rule source	Concept	Sort
Rule register	Instrument	Sort
Contradiction	Concept	Sort
Decision	Concept	Sort
The four classes of decision	Concept	Sort
Decision register	Instrument	Sort
Contested decision	Concept	Sort
Contested-decision log	Instrument	Sort
Decision rights	Concept	Sort, Own
Separating decisions from process	Concept	Sort, Design

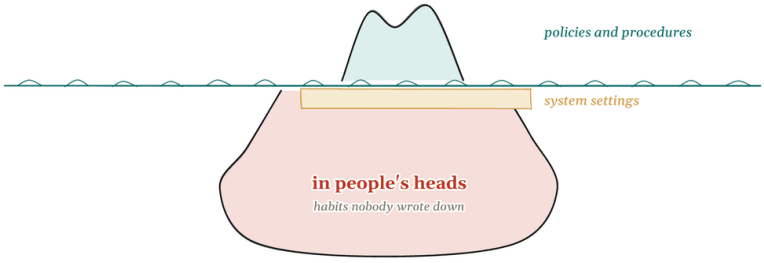


Like a dictionary: *every word written in it can be looked up and checked, yet new words keep arriving that no edition contains.*

Rule and rule source

A rule is a statement of what must or must not happen in a case: an approval limit, a tolerance, an eligibility condition, a required check. Rules live in three places. Some are documented in policies and procedures. Some are configured in systems, as tolerances, validations and workflows. And some exist only in the heads of experienced staff, as habits nobody wrote down.

The third source is the one most often missed and the one that matters most to an executor. A clerk who knows that one supplier always invoices freight separately is applying a rule. If nobody writes it down, the executor cannot apply it.

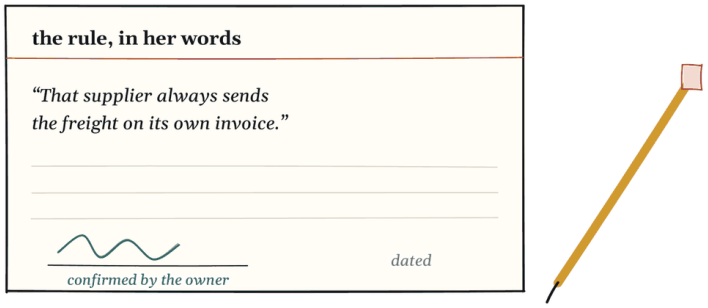


Like an iceberg: *a few rules show above water in policies, some sit at the waterline in system settings, and most hide below in people's heads.*

Rule register

The rule register collects every rule of a loop in one place, with its source, its owner and the date it was confirmed. Rules from people's heads are written in the words of the person who holds them, then confirmed by the owner. The

register is signed at Step 4 and becomes the main source of the executor's instructions.

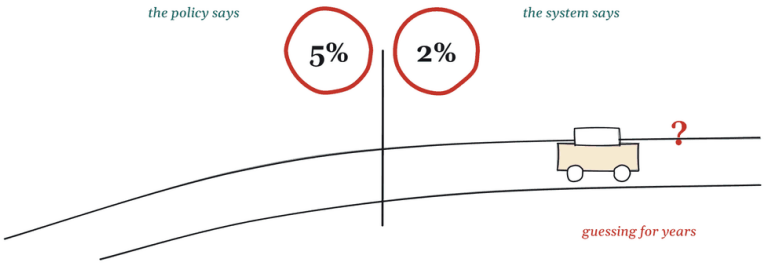


Like writing down a family recipe: *the cook's habit goes on the card in her own words, then it is checked, signed and dated so anyone can follow it.*

Contradiction

A contradiction is a pair of rules that cannot both be followed: the policy says one thing, the system enforces another, or two experienced staff apply different rules to the same case. At the manufacturer, the policy allowed a 5% price difference before approval was needed, while the ERP blocked anything over 2%. Clerks had been releasing the difference by hand for years.

Contradictions are findings, not noise. Each one is a place where the current process has been guessing, and where an executor would have to guess. Each needs an owner to decide which rule stands.

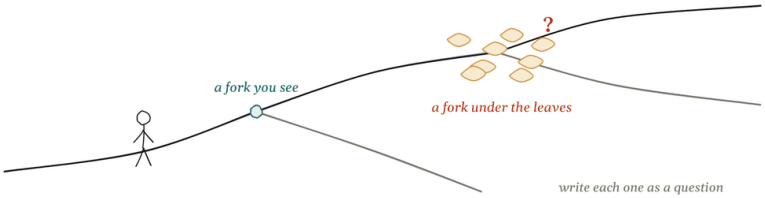


Like two speed signs on the same road that disagree: *drivers have been guessing for years, and an owner must decide which sign stands.*

Decision

A decision is any point in a loop where the work could go more than one way. Is this a duplicate? Does it match within tolerance? Which cost centre? Approve or hold? Decisions are best written as questions. Listing them is

harder than it seems, because in a process built for people, many decisions are buried inside steps and taken without anyone noticing.



Like a woodland path with forks: *some forks are plain to see, others lie under the leaves, and walkers turn there without noticing a choice.*

The four classes of decision

Every decision in a loop falls into one of four classes.

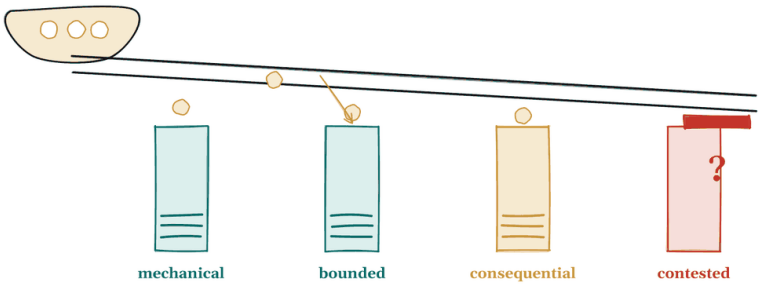
A mechanical decision follows a written rule completely. The executor can take it alone.

A bounded decision needs judgement, but within a written limit: approve a price difference up to 2% on orders under €5,000. The executor can take it alone, inside the limit, graded against a standard.

A consequential decision carries a high cost of error, or leads to an action that cannot be undone. A person always decides, with the executor preparing the evidence.

A contested decision is one the organisation has never agreed on. It cannot be designed; it must be settled.

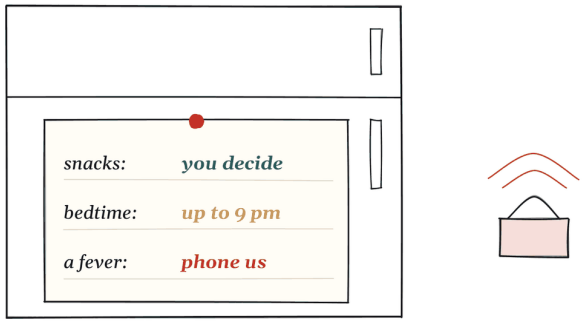
The class depends on the state of the rules as well as the stakes. A decision is mechanical only if its rule is written and uncontradicted, and bounded only if someone has signed the limit. A common misreading is to class decisions by comfort: everything becomes consequential because nobody wants to let go, or everything becomes mechanical because a pilot looked good.



Like a coin sorter with four tubes: *each decision drops by its kind, and the fourth tube is capped until the company settles what the coin is worth.*

Decision register

The decision register lists every decision in a loop, as a question, with its class, the reason for the class, and the limit if it is bounded. It is signed at Step 4. It is the single most useful document for anyone trying to understand what an executor in the loop is allowed to do.



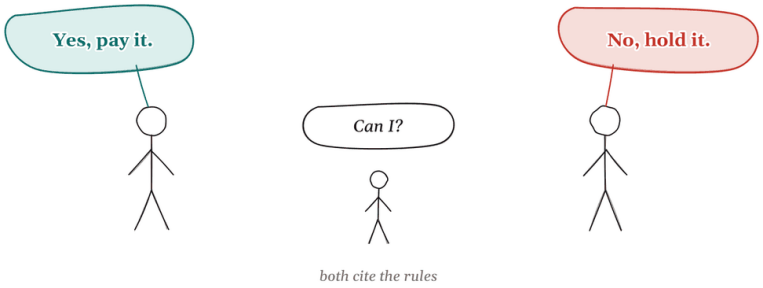
Like the note left for a babysitter: *what they may decide alone, what they may decide up to a limit, and when they must phone the parents.*

Contested decision

A contested decision is a decision the organisation has never actually agreed on. Two senior people would answer it differently, and both would cite policy. It has usually been handled for years by escalation, by habit, or by not looking. Pay a disputed invoice to protect supply? Grant a goodwill refund outside policy? Accept a late claim from a loyal customer?

No design can resolve a contested decision. An executor cannot be given a rule the company does not have. Someone with authority must decide, write it down and sign it. Contested decisions are found by asking, in the room, which

decisions two senior people would answer differently. They are the most important findings of Step 4, because each one would otherwise surface as an incident the first time an executor met it.

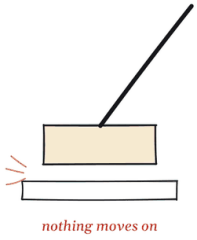


Like a child asking two parents the same question and hearing yes and no: *there is no rule to give the babysitter until the parents settle it.*

Contested-decision log

The contested-decision log records each contested decision: the question, the positions held, who settled it, the answer and the date. The lifecycle does not pass from Sort to Design while any entry is open. At the manufacturer, the log had one entry, on disputed invoices from single-source suppliers. The chief financial officer and the chief operating officer settled it in a thirty-minute meeting.

question	positions	ruling	date
pay a disputed invoice?	*pay / hold*	SETTLED	*signed*
a late claim?	*accept / refuse*	OPEN	



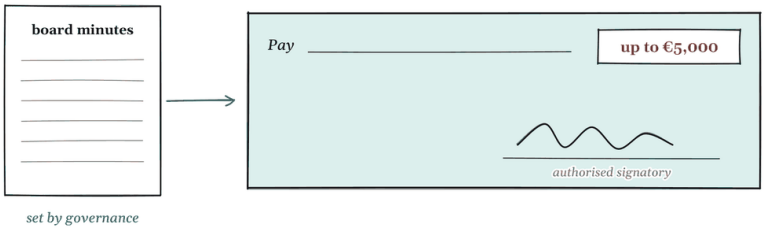
Like a court docket: *each dispute is entered with both positions, the ruling, who gave it and the date, and the list cannot move on while one stays open.*

Decision rights

Decision rights say who may take which decision, up to which limit. In most companies they are partly written, in approval matrices and delegations of authority, and partly assumed. Embedding forces them into the open. Each bounded decision needs a signed limit, which is a decision right delegated to

the executor. Each consequential decision needs a named person, who holds the decision right at a junction.

A common misreading is to treat decision rights as a matter for the executor's configuration. They are a matter for the company's governance. The configuration should reflect what the company has decided, not substitute for it.



Like signing limits on a company chequebook: *the board decides who may sign and up to what amount, and the printer only prints what was decided.*

Separating decisions from process

In 2015, the Object Management Group published DMN, a standard for modelling decisions separately from the processes that use them. Decision tables were written down once and called from many places. The idea was sound, but adoption stayed in pockets, partly because no executor could act on a decision table without a person translating it into steps.

The executor that decides changes that. Rules and decisions written down and sorted can be given to it directly. Separating decisions from process, the old idea behind DMN, becomes the practical core of Step 4.

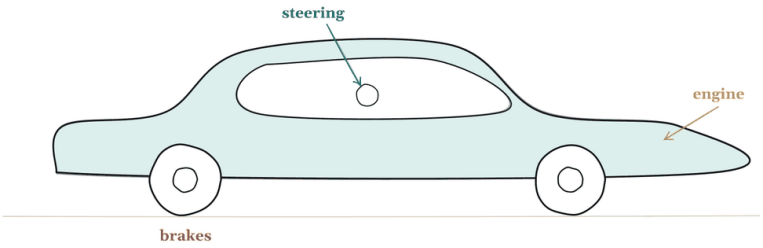


Like a price list pinned on the shop wall: *written once and read by every till, instead of each cashier carrying the prices in their head.*

XXIII

The Executor

This area covers what does the work in an embedded loop: what an executor that decides is, what it is made of, what it can read and change, and how the business keeps control of each part. It is written for business readers. It explains each part well enough to make the decisions that part requires, and no further.



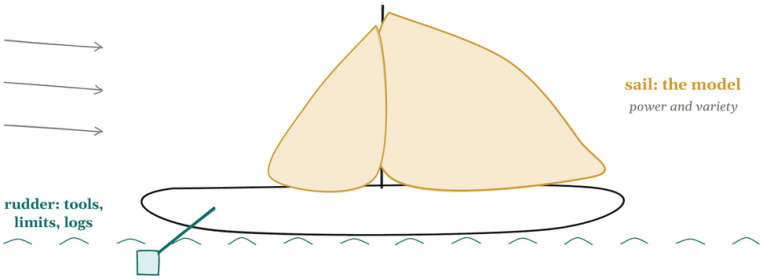
Like a car explained to a new owner: *enough about engine, steering and brakes to make each decision about driving it, and no further.*

Where it comes from

This area covers the neural half of the loop and the machinery that connects it to the symbolic half. A language model brings the variety rules lack. Tools, permissions, logs and the platform turn its proposals into accountable actions.

Concept	Kind	First used at
Executor	Concept	Read
Instruction-follower and decider	Concept	Read
Model	Concept	Design
Instructions	Concept	Design
Context and keepers	Concept	Design
Actions and reversibility	Concept	Design, Bound
Permissions	Concept	Bound
Platform	Concept	Design, Bound
The jagged frontier	Concept	Shadow
Pilot	Concept	Shadow

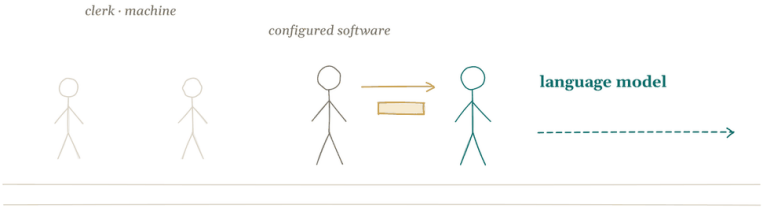
Concept	Kind	First used at
Build, buy or rent	Concept	Design



Like a sailing boat: *the wind in the sail brings the power and the variety, and the rudder, sheets and log turn it into a course someone answers for.*

Executor

The executor is whatever does the work in a loop. For most of the history of business it was a person, a machine following a sequence, or software applying configured rules. In an embedded loop it is, for much of the work, a language model with access to the company's systems, acting within limits the business has set.

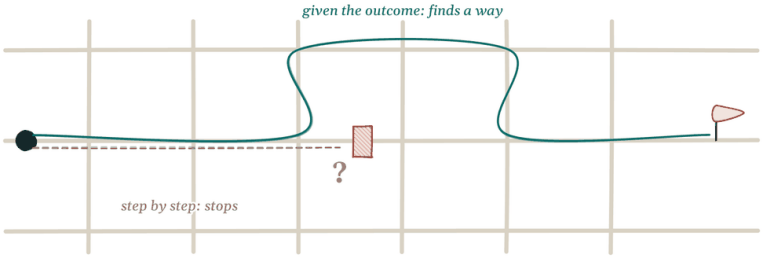


Like a relay race: *the baton of the work passed from people to machines to configured software, and is now handed to a language model.*

Instruction-follower and decider

An instruction-follower needs the work specified step by step in advance, and hands to a person wherever the specification runs out. A decider is given an outcome, rules and limits, and works out the steps itself. It can read a document it has never seen, apply a rule written in plain language, and reach a decision.

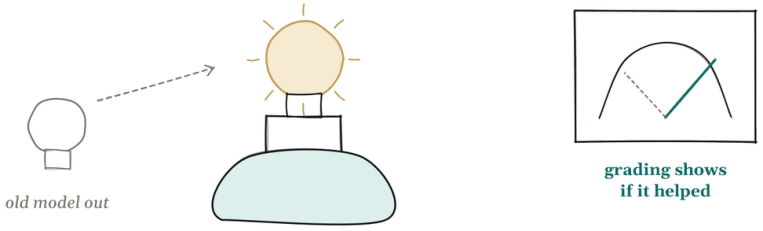
The difference changes how work is designed. For a decider, the important things to specify are no longer the steps. They are what counts as done, what it may not do, where it must stop and ask, and how its results will be graded.



Like a satnav versus a local driver: *the satnav stalls when its route is blocked, while the driver given the address and no-go streets finds a way.*

Model

The model is the language model at the centre of the executor, trained on large amounts of text to read, reason and write. Companies usually buy access to a model from a provider, or run an openly available model on their own infrastructure. Models improve quickly and are replaced often. A loop should be designed so that its model can be changed, and its grading should show whether the change helped.



Like a bulb in a standard socket: *the lamp is built so the bulb can be swapped, and a light meter shows whether the new one is better.*

Instructions

The instructions tell the model its job: the outcome, the rules, the limits, the junctions and what to do with exceptions. In a well-designed loop they come straight from the signed registers. A loop whose rules were never gathered can only give its executor vague instructions, and vague instructions produce inconsistent decisions.

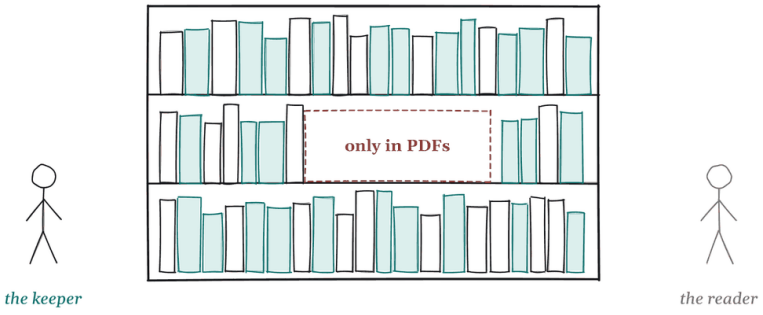


Like sheet music: *a signed score gets the same tune every time, while a note saying play something nice gets a different tune each night.*

Context and keepers

Context is the information the executor must read to act well on a case: the order, the receipt, the contract, the customer's history. It lives in the company's systems, and the executor reaches it through connections someone must build and maintain. Each source has a keeper, the person responsible for keeping it accurate.

A gap in context is a decision made blind. At the manufacturer, contract prices for about a third of suppliers existed only in PDF agreements. Procurement loaded the prices for the top two hundred suppliers into the ERP before shadow running.



Like a library with a keeper for each shelf: *the reader can only act well on books that are there, and a missing shelf means deciding blind.*

Actions and reversibility

Actions are the changes the executor may make: posting a record, placing a hold, sending a message, releasing a payment. Each is described by the system it touches and one question: can it be undone, and at what cost? Reversibility

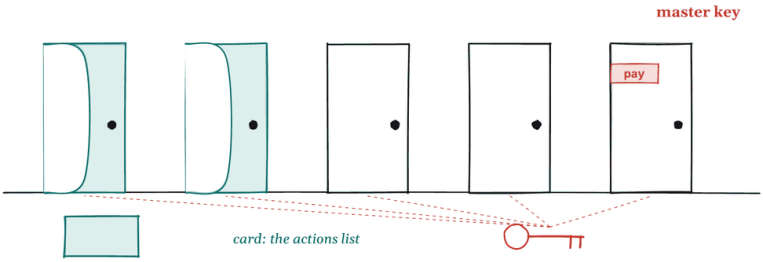
is the main input to where junctions go. Reading is cheap. Posting can be reversed. Releasing a payment usually cannot.



Like reading, writing in pencil and posting a letter: *reading costs nothing, pencil can be rubbed out, and a letter in the postbox cannot be called back.*

Permissions

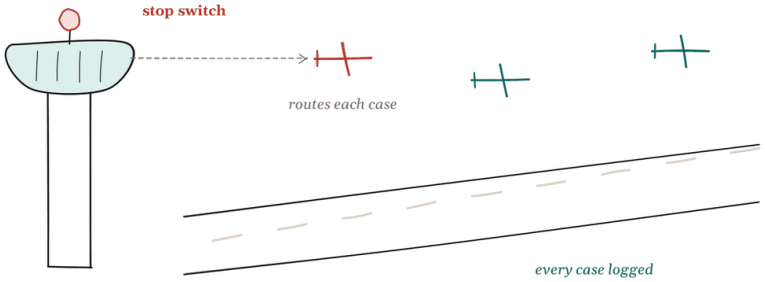
Permissions are the rights an executor holds in each system. They should match its list of actions exactly. Permissions granted broadly for a pilot "to get things working" are the commonest source of executors doing things nobody intended. Cutting permissions back to the actions list is part of Step 7.



Like a hotel key card coded for the rooms on its list: *a master key handed over just to get things working opens every door, payments included.*

Platform

The platform is the layer that runs the loop around the model. It routes each case, applies limits, sends consequential cases to junctions and exceptions to the queue, records every case, and holds the switch that stops the loop. Whether it is built or bought, the business should require three things of it: a complete log of every case, a switch the named person can use, and access for the grader to every result.

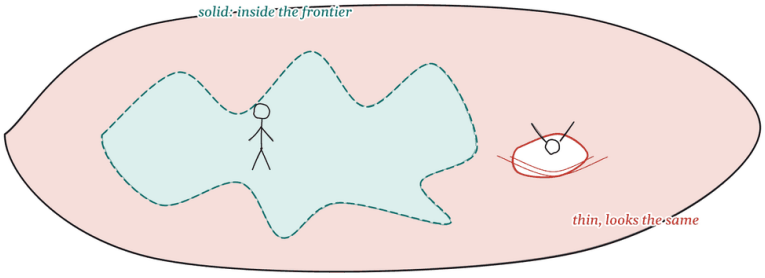


Like an airport control tower: *it routes every plane, sends the unusual ones to a person, logs every movement, and can stop the runway.*

The jagged frontier

Executors are not uniformly capable. A 2023 study of consultants using AI, by Fabrizio Dell'Acqua and colleagues, described a "jagged technological frontier". On tasks inside the frontier, AI help raised quality substantially. On tasks just outside it, people using AI did worse than those without it, because they trusted answers they should have checked. The frontier is uneven, and it is not visible in advance.

That is why shadow running reads agreement case by case and band by band, rather than trusting a demonstration. It is also why the tail stays with people until the evidence says otherwise.

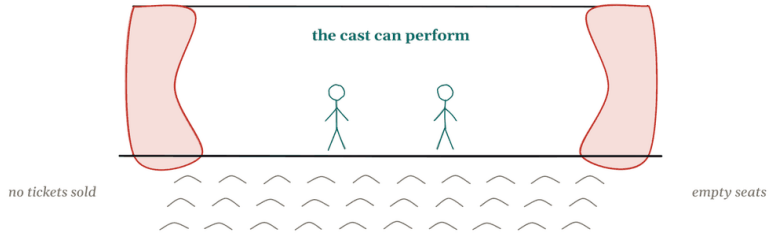


Like a frozen lake with uneven ice: *solid in places, thin just beyond, and the edge cannot be seen, so you test each patch before you trust it.*

Pilot

A pilot is a test of whether an executor can do a task, on a prepared set of cases. It usually builds the model and instructions well, connects some context, and leaves out real actions, the platform's switch and log, and the grader. A pilot is valuable evidence and a small part of the work. In the

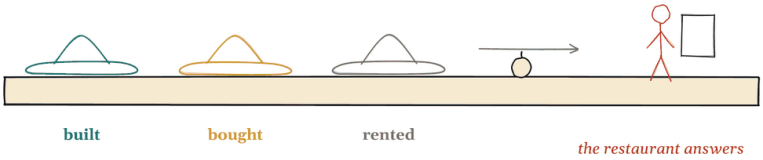
lifecycle it belongs inside Step 6, as the executor in shadow, after Steps 1 to 5 have been done.



Like a dress rehearsal: *real proof the cast can perform, but with empty seats, no tickets sold and no reviewers, so the real run is still ahead.*

Build, buy or rent

Every part of the executor can be built, bought or rented, and the choice changes over time. The choice does not change the business's responsibilities. Whoever supplies the executor, the company must sign its rules, sort its decisions, place its junctions, write its standard and measure its economics. A vendor-run loop needs contract terms that give the company the log, the switch, access for its grader, and notice before the model changes.

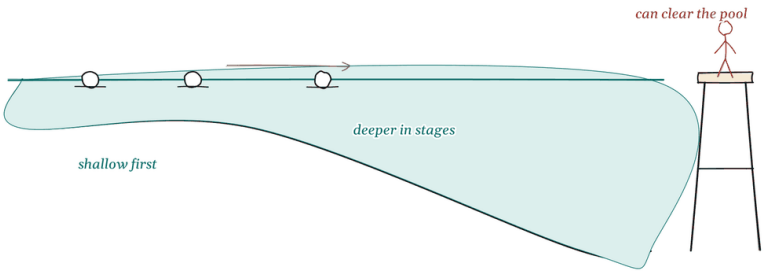


Like a restaurant's kitchen pass: *dishes cooked in-house, bought in or catered all cross the same pass, and the restaurant answers for every plate.*

XXIV

Control and Acceptance

This area covers how a company knows an embedded loop is doing good work, where people decide, how trust is earned in stages, and how the loop is stopped when it should be. It holds the concepts that make an executor that decides safe to rely on.



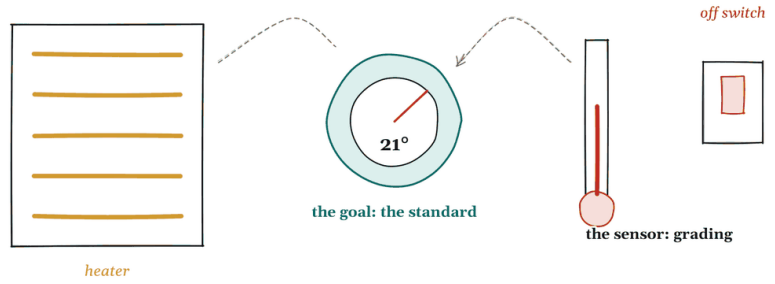
Like a lifeguard at a pool: *watching that the swimming goes well, standing by the deep end, letting swimmers in deeper by stages, and able to clear the water.*

Where it comes from

This area is cybernetics applied. The standard is the goal, grading is the sensor, and junctions add the variety of human judgement at the tail. The switch-off is the means of opening the loop safely. Without this area, an embedded loop runs open: acting without knowing whether its actions are right.

Concept	Kind	First used at
Junction	Concept, instrument	Design
Switch-off	Instrument	Design, Bound, Core
Exception queue	Instrument	Design
Standard	Instrument	Design
Grading	Concept	Design, Shadow
Shadow running	Concept	Shadow
Agreement, body and tail	Measure	Shadow
Bounded scope	Concept	Bound
Scale Gates	Concept	Design to Core

Concept	Kind	First used at
Drift	Concept	Core
Override rate	Measure	Core
Controls that survive and controls that invert	Concept	Design, Bound
The ironies of automation	Concept	Own, Core



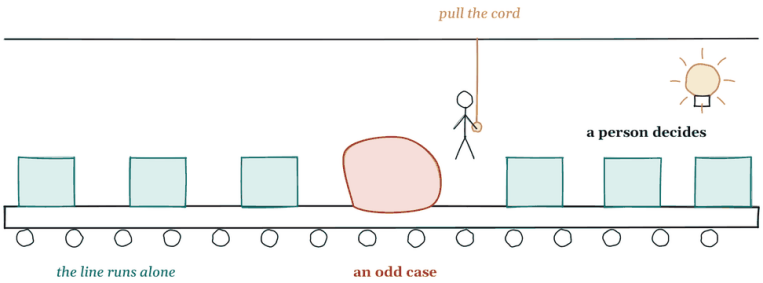
Like a thermostat: *the setting is the standard, the thermometer is the grading, a hand can override it, and the switch opens the loop safely.*

Junction

A junction is a point where a person decides instead of the executor. It is defined by three things: who decides, what evidence the executor puts in front of them, and how a wrong decision is reversed. Junctions are placed by the reversibility and cost of the action, not by how comfortable anyone feels. Every consequential decision and every irreversible action needs one.

The ancestor of the junction is the cord on Toyota's assembly lines that any worker could pull to stop the line when something looked wrong. The routine runs on its own, an abnormal case stops it, and a person decides.

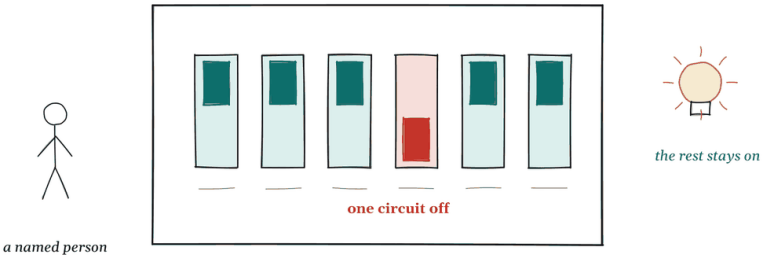
A common misreading is to treat any human approval as a junction. An approval with no designed evidence, no clear holder and no way to reverse the decision is a rubber stamp.



Like the cord on Toyota's assembly line: *the line runs on its own until an odd case appears, a worker pulls the cord, and a person decides.*

Switch-off

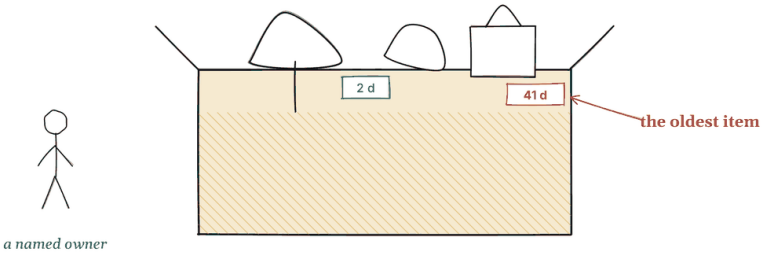
The switch-off is the means by which a named person stops the loop, and returns its cases to people, within a stated time and without losing any. It is designed at Step 5, tested before the first live case at Step 7, and tested on a schedule in Core. Using it on part of the scope for a few days, when drift or an incident calls for it, is normal operation, not failure.



Like a fuse box with labelled switches: *a named person turns one circuit off for a few days, the rest of the house stays lit, and nothing is lost.*

Exception queue

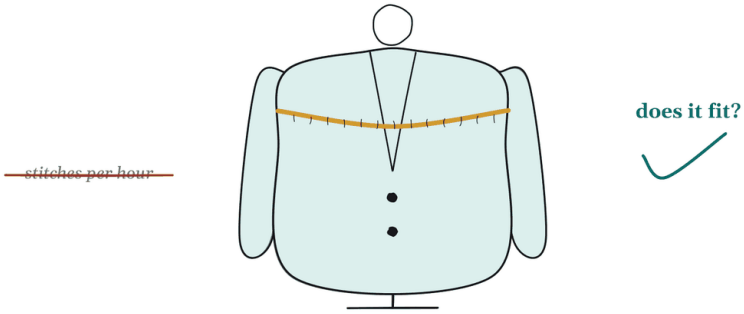
The exception queue is the single place where cases outside every rule go. It has a named owner, a time limit, and a weekly review that turns repeated exceptions into new rules. The age of its oldest case is one of the most revealing measures of a loop's health.



Like a station's lost-and-found box: *one place for things that fit nowhere, one owner, a weekly sort, and the oldest tag says how healthy it is.*

Standard

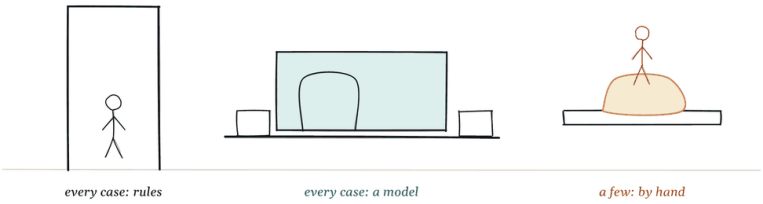
The standard is the written definition of an accepted outcome, with the method of grading it. Its criteria describe the outcome, not the activity. "Right supplier, amount, tax and cost centre, no duplicate, ready within terms" is a standard. "Invoices processed per hour" is not. Customer-facing loops include the customer's view in the criteria, through complaints, repeat contacts and reopened cases.



Like a tailor's fitting: *the suit is judged by whether it fits the customer, not by how many stitches were sewn per hour.*

Grading

Grading applies the standard to outcomes. The usual pattern combines three layers. Rules check every case for what can be checked mechanically. A second model grades every case against the criteria. And a small human audit, weighted towards the tail, calibrates the model grader. Grading every outcome, not a sample, is what makes it possible to run an executor that decides, because there is no procedure to audit instead.



Like airport security: *an arch checks every passenger, a scanner checks every bag, and an officer opens a few by hand to keep the scanner honest.*

Shadow running

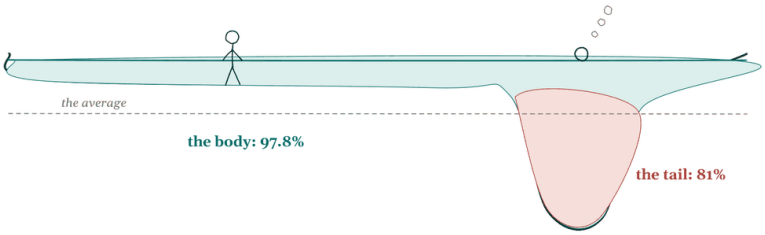
Shadow running puts the executor beside the current process on real cases, reaching its own results without acting on anything, so they can be compared with the people's results. It runs long enough for the tail to arrive in useful numbers, typically four to eight weeks. Every disagreement is classified. The executor may have been wrong, or the person may have been wrong. Both may be defensible, or the standard may be unclear.



Like a new clock hung beside the old one for weeks: *it keeps time, nobody sets the trains by it, and every difference is written down.*

Agreement, body and tail

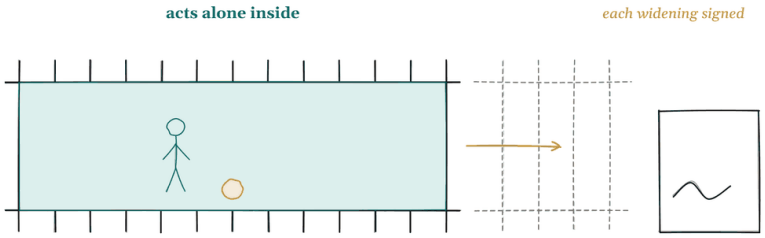
Agreement is the share of cases where the executor's result matches the accepted result. It is always reported separately for the body and the tail, because a single overall figure hides the difference that matters. At the manufacturer, agreement was 97.8% on invoices under €5,000 and 81% on invoices over €50,000.



Like a river whose average depth is one metre: *the wide shallows hide the deep channel, so you read the shallows and the channel separately.*

Bounded scope

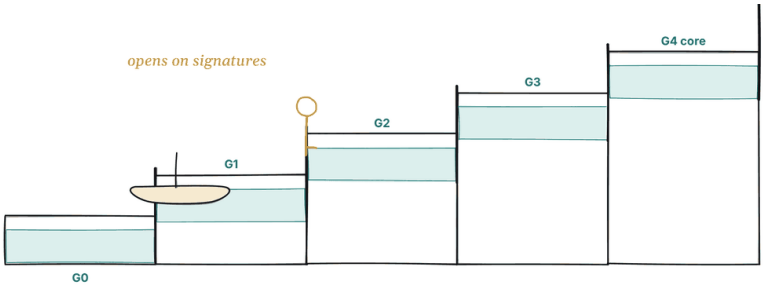
Bounded scope is the written set of conditions under which the executor may act alone: values, tolerances, counterparties, case types. It is drawn from shadow evidence and signed by the control owner at Step 7. Every widening is recorded and agreed. A common misreading is to set the scope by ambition, because the pilot handled large cases well in a demonstration.



Like a playground fence: *the child plays alone inside it, the fence follows what was seen to be safe, and moving it out is agreed and signed.*

Scale Gates

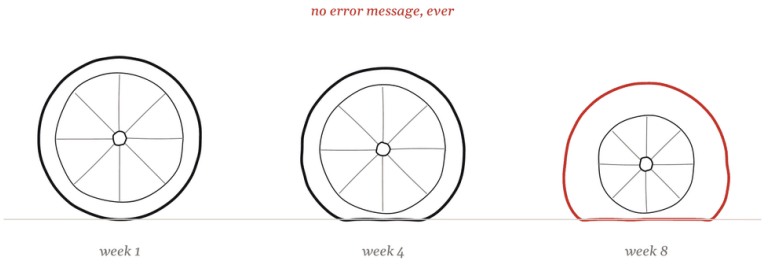
The Scale Gates are five stages a loop passes on its way to the core, each opened by particular signatures. At G0, Mapped, every node of the loop's canvas is drafted. At G1, Shadow, the outcome, compensations, rules, context and standard are signed. At G2, Bounded, the actions, decisions, junctions and exceptions are signed too. At G3, Owned, the people and economics are also signed. At G4, Core, all twelve are signed. Scale follows signatures, not confidence.



Like a flight of canal locks: *the boat rises one chamber at a time, and each gate opens when its keys are turned, not when the crew feels ready.*

Drift

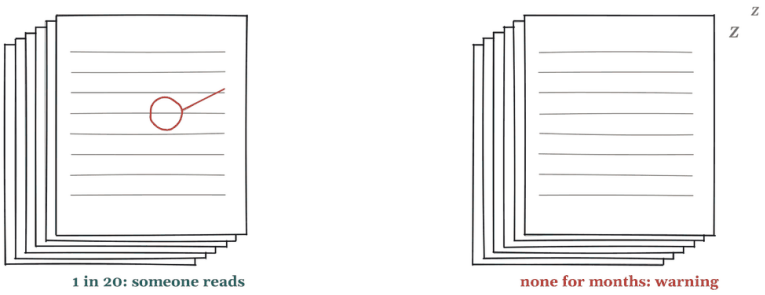
Drift is a gradual change in the quality of an executor's outcomes, caused by changes in its inputs, its model or the people around it. It produces no error messages. It shows as a slow fall in the share of outcomes meeting the standard, often in one segment first. It is caught by grading every outcome, by a weekly human sample that checks the grader, and by reading the results body and tail separately each month.



Like a slow puncture: *no bang and no warning light, just a tyre a little softer each week, caught only by checking the pressure regularly.*

Override rate

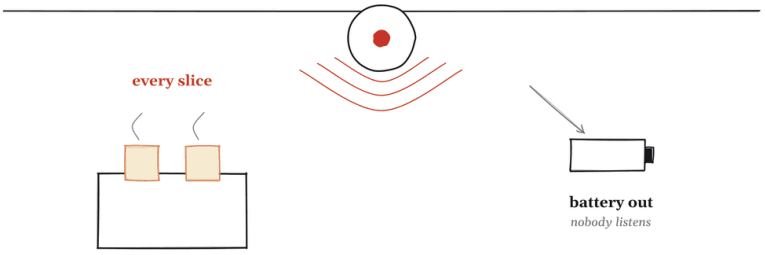
The override rate is how often junction holders overrule the evidence the executor puts in front of them. A rate near zero for months is a warning that the junction has become a rubber stamp. A rate that is above zero and explained is healthy. At the manufacturer, the controller overrode about one large invoice in twenty, usually because of information the executor could not see.



Like a proofreader: *one correction in twenty pages shows someone is reading, while months without a single mark are the warning.*

Controls that survive and controls that invert

Most controls in a process were designed for people, and they fare differently with an executor that decides. Some survive unchanged, such as the call-back on changed bank details. Some invert. Segregation of duties, which separates the person who records from the person who approves, becomes a separation between the executor and the standard it is graded against, with junctions at the consequential points. A four-eyes review becomes a grader plus a junction. Sampling becomes grading every outcome. And some controls become dangerous if left unchanged. "Review everything", applied to an executor's thousands of decisions a day, guarantees that reviewers stop reading.



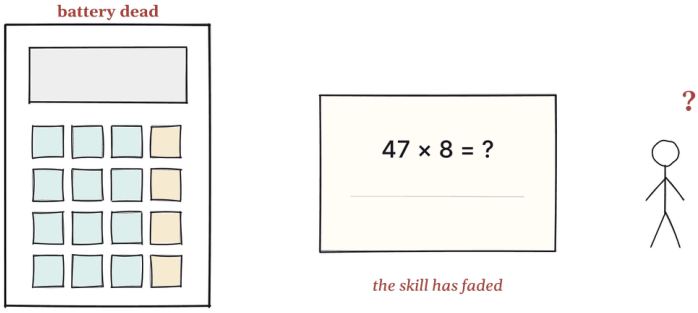
Like a smoke alarm that sounds at every slice of toast: *a control meant for rare events, applied to everything, teaches the house to take the battery out.*

The ironies of automation

In 1983, the engineer Lisanne Bainbridge described the "ironies of automation". The more a system is automated, the more crucial and the more difficult the human role becomes. People left to monitor automated systems lose the skills they need when the system fails, and they find it hard to stay alert to rare problems. Later research on automation bias found the same

pattern: people tend to accept a machine's recommendation even when the evidence suggests otherwise.

These findings shape three design choices: junction evidence that asks the holder to look before agreeing, override rates read as a health signal, and a deliberate plan for people to keep the skills the tail requires.



Like years of leaning on a calculator: *mental arithmetic quietly fades, and it is needed most on the one day the battery dies.*

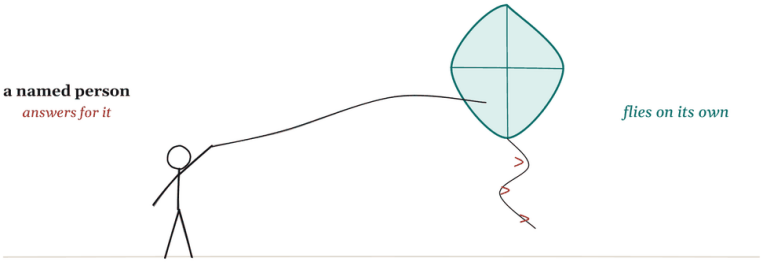
PART FIVE

Owning the Result

XXV

People and Accountability

This area covers who answers for an embedded loop, who decides at its junctions, whose work changes, and how people keep and build the judgement the work still requires. An executor cannot be accountable. Everything it does is answered for by a named person, and this area says which one.



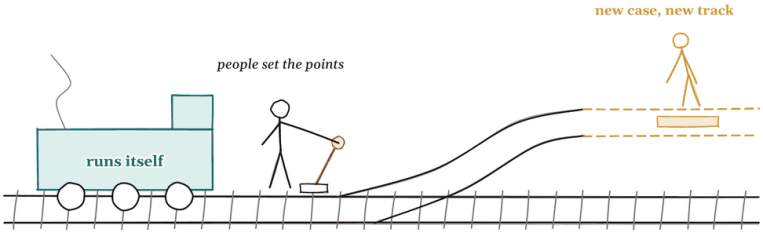
Like a kite on a string: *the kite flies on its own, but the person holding the string answers for where it goes.*

Where it comes from

This area keeps the strange loop honest. People design the symbolic layer, hold the junctions, extend the rules when the world shows them a new case, and answer for the results. An executor cannot be accountable; every decision it takes is answered for by a named person.

Concept	Kind	First used at
Executive sponsor	Role	Read
Process owner and loop owner	Role	Map, Design
Control owner	Role	Bound
Junction holder	Role	Design, Own
Exception owner	Role	Design
Grader	Role	Design, Shadow
Adoption owner	Role	Own
Business engineer	Role	Read to Scale
Signature	Concept	All steps

Concept	Kind	First used at
Loop charter	Instrument	Own
Role change plan	Instrument	Own, Scale
Learning the tail	Concept	Own
Loop roles and the RACI chart	Concept	Own



Like a railway: *the train runs itself, but people lay the track, set the points, and extend the line when a new town appears.*

Executive sponsor

The executive sponsor is the senior person who commissions embedding across a set of loops and can settle disagreements between departments. The sponsor signs the Work Loop Map, the contested decisions that need their authority, the map review and the role change plan. A sponsor who cannot settle a contested decision is not senior enough for the loops in question.

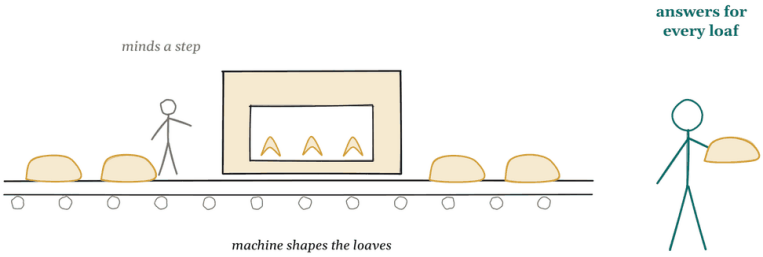


Like a tug-of-war referee: *two departments pull against each other, and only someone senior enough can blow the whistle and settle it.*

Process owner and loop owner

The process owner is responsible for how a process runs today. During Design, the process owner usually becomes the loop owner, who answers for the loop's outcome, including the parts the executor produces. The change of title marks

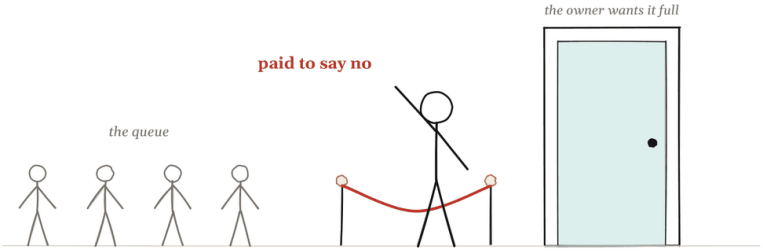
a change of responsibility. A process owner looks after steps. A loop owner answers for results.



Like a bakery line: *one person minds each step of the oven, but the head baker answers for every loaf that leaves, including those the machine shaped.*

Control owner

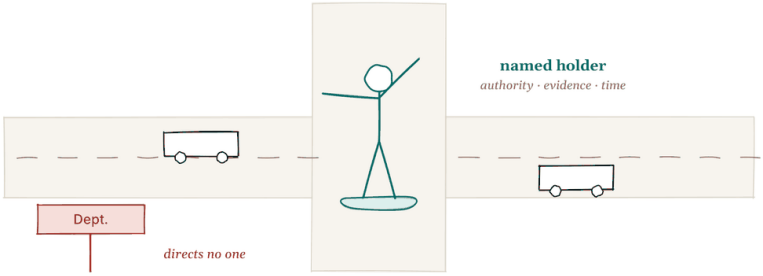
The control owner is responsible for the company's controls over a process, such as the financial controller or the head of risk for that area. The control owner signs the gate review at Bound and agrees each later widening of scope. The role is deliberately separate from the loop owner, who naturally wants the loop to run. The person who signs the limits should be the one whose job is to say no.



Like a club doorman: *the owner wants the room full, so the person who decides who gets in is the one whose job is to say no.*

Junction holder

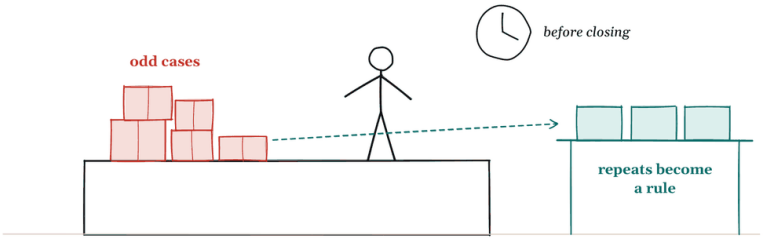
A junction holder is the named person who decides at a junction. Holders need the authority to decide, the evidence to decide well, and the time to do it. A junction whose holder is listed by department, not by name, has no holder.



Like a busy crossroads: *a named officer with authority directs the traffic, while a sign that only names a department directs no one.*

Exception owner

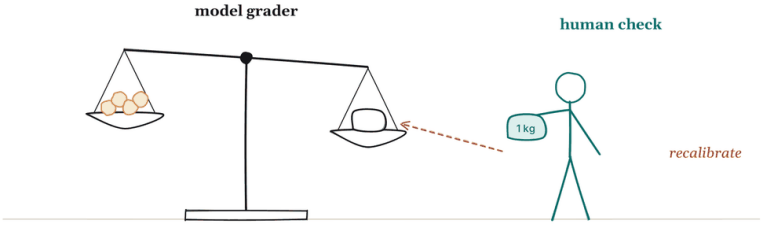
The exception owner works the exception queue, keeps it within its time limit, and runs the weekly review that turns repeated exceptions into rules.



Like a returns counter: *one clerk clears the odd parcels before closing time, and a parcel that keeps coming back gets its own shelf.*

Grader

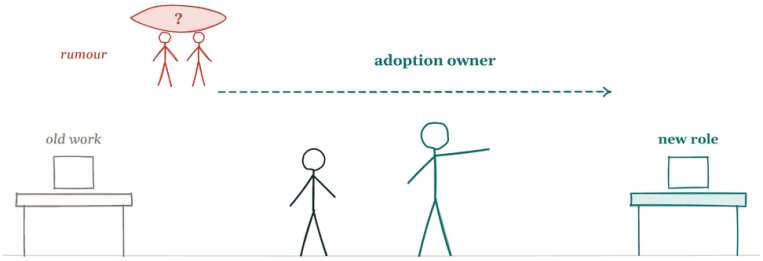
The grader is the person who keeps the standard and its grading calibrated: reviewing the human audit, comparing it with the model grader, and adjusting the criteria when the business changes. The grader signs the shadow report with the loop owner.



Like a weights inspector at a market stall: *she checks the stall's scale against her own reference weight and adjusts it when the two disagree.*

Adoption owner

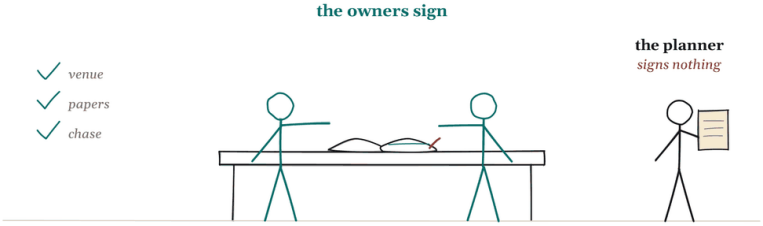
The adoption owner is responsible for the people whose work changes: agreeing the changes openly, helping people into new roles, and designing how new staff will learn the work. In most programmes this role is missing, and the change to people's work is left to rumour.



Like an office move: *someone has to show each person to their new desk, or the move happens by rumour in the corridor.*

Business engineer

The business engineer runs the lifecycle for a loop. They book the sessions, prepare the documents, chase the signatures and know what each gate requires. They sign nothing. Owners and control holders sign. The business engineer makes sure there is something worth signing. In companies running many loops, business engineers form a small function that owns the method, not the loops.

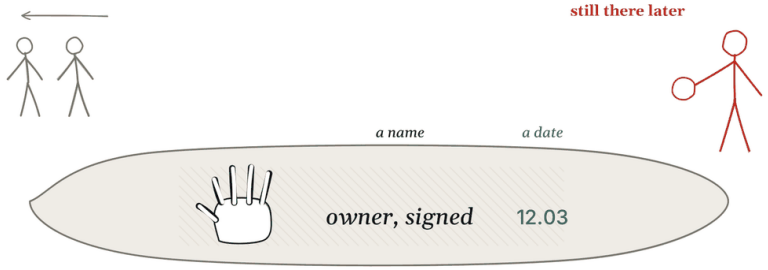


Like a wedding planner: *books the venue, prepares the papers and chases everyone, but never signs the register; the couple do.*

Signature

A signature is a named person's dated, recorded acceptance of a document as true or as decided. It does not need ink. It needs a name, a date and a record that survives the meeting. Signatures turn agreement into accountability.

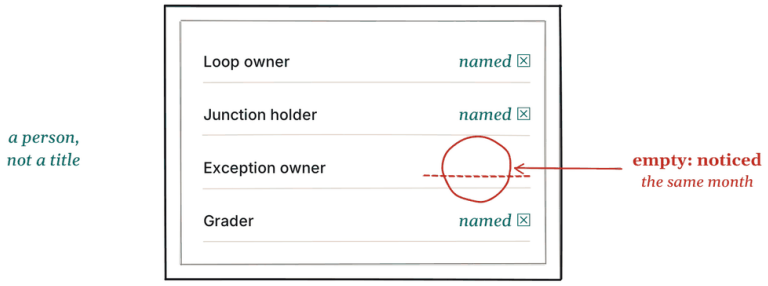
When an auditor, a regulator or a new manager asks why a loop behaves as it does, the signatures are the answer.



Like initials pressed into wet cement: *a name and a date that stay long after everyone has left the meeting, ready for whoever asks later.*

Loop charter

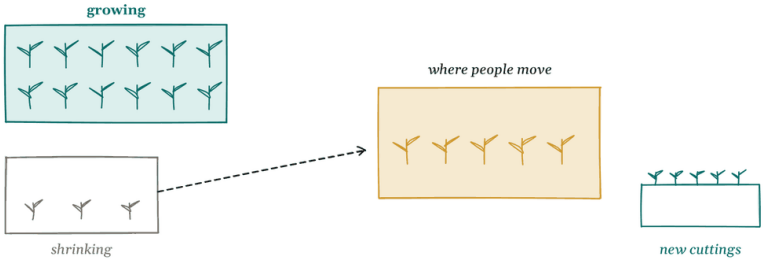
The loop charter names every role a loop needs with a person, not a title. It records the agreed changes to people's work and the learning path for new staff. It is signed by the loop owner at Step 8 and checked at every operating review, so that a role left empty by someone's move is noticed the same month.



Like a theatre cast list on the stage door: *every part carries an actor's name, so a blank slot is noticed before the next show.*

Role change plan

The role change plan is the map-wide view of how work is changing across all embedded loops. It records which roles are growing, which are shrinking, where people are moving, and how expertise will be rebuilt. It is signed by the sponsor at Step 10, because it commits the company, not just one team.

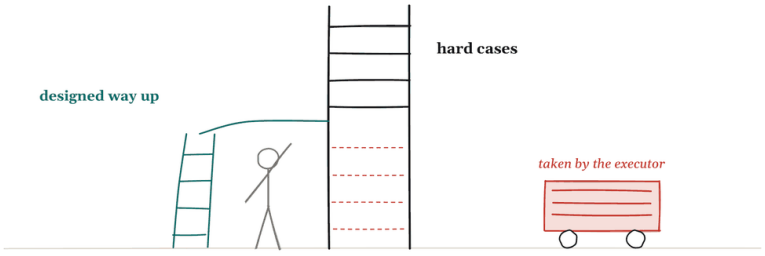


Like a replanting plan for a whole garden: *it shows which beds grow, which shrink, where plants move and where new cuttings are raised.*

Learning the tail

In most processes, people learned their judgement by handling simple cases until they were ready for hard ones. When an executor takes the simple cases, that path disappears. A company that does not replace it will, within a few years, have no one able to hold its junctions.

Research points both ways. A large study of customer-support agents by Erik Brynjolfsson, Danielle Li and Lindsey Raymond found that an AI assistant raised productivity by about 14% on average, and most for the least experienced, who learned faster with it. But that was an assistant alongside the work. Where the executor takes the routine cases entirely, the learning has to be designed. Common designs are rotating new staff through the exception queue, pairing them with junction holders, and having them review graded cases.

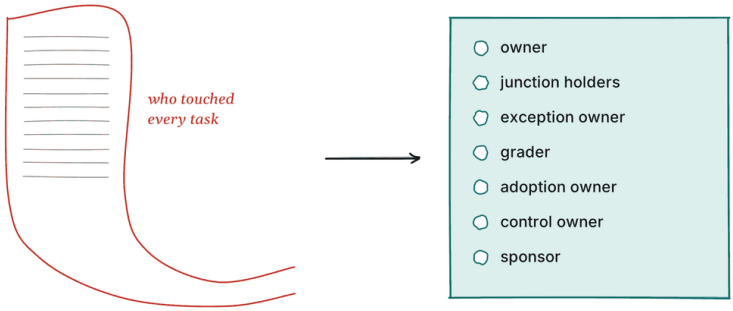


Like a ladder whose bottom rungs are taken away: *no one can climb to the hard top rungs unless a new way up is built.*

Loop roles and the RACI chart

Many companies assign responsibility per task with a RACI chart, listing who is responsible, accountable, consulted and informed. With an executor doing

many of the tasks, per-task charts become long and misleading. Embedded loops replace them with a short set of loop roles: owner, junction holders, exception owner, grader, adoption owner, control owner and sponsor. Each is a named person with a defined decision.



Like swapping a receipt as long as your arm for a team sheet: *seven named roles, each with its own decision.*

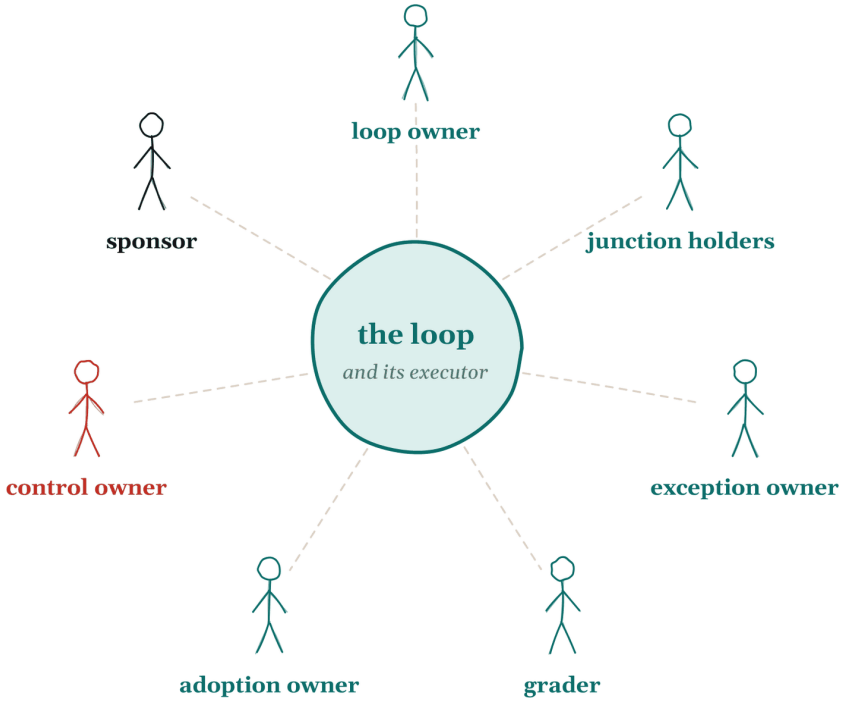
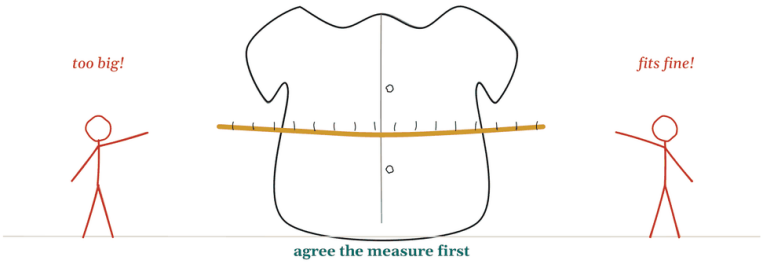


FIGURE 19 People and accountability. Every loop has named people around it: the owner of its outcome, the holders of its junctions, the owner of its exceptions, its grader, the owner of adoption, the control owner who signs its limits and the sponsor above them all.

XXVI

Economics

This area covers what an embedded loop costs, what it saves and when, and how those numbers are made credible. Most disputes about the value of AI are disputes about measurement. This area defines the measurements so the disputes can be settled.

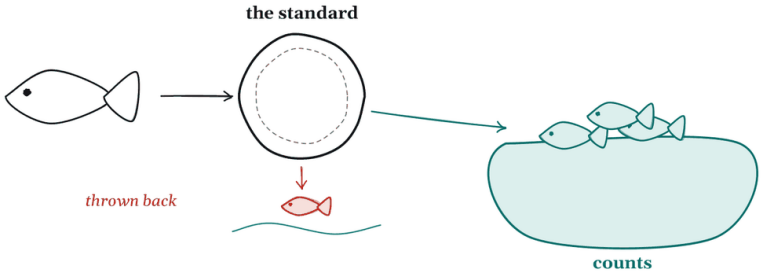


Like a tape measure in an argument about fit: *agree first how the jacket is measured, and the dispute about value settles itself.*

Where it comes from

This area measures whether the closed loop is worth running. Its foundations are accounting and measurement theory, including the warning that a measure turned into a target stops measuring. Cost per accepted outcome ties the economics to the standard, so that only results which passed the grader count.

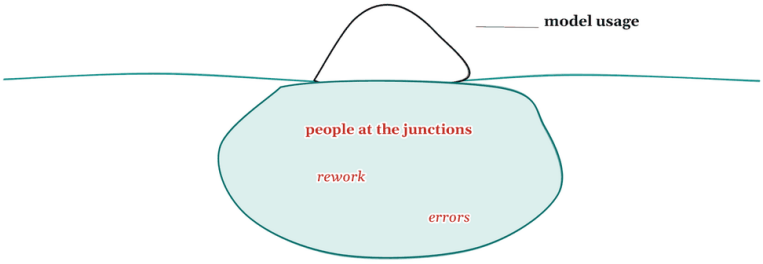
Concept	Kind	First used at
Fully loaded cost	Concept	Map
Baseline	Instrument	Map
Cost per accepted outcome	Measure	Own
Body and tail economics	Measure	Own
Economics sheet	Instrument	Own
The value curve	Concept	Read, Own
The capacity to embed	Concept	Scale
Measures that mislead	Concept	All steps
Pricing by outcome	Concept	Design



Like a fishing gauge: *only fish that pass the measuring ring count toward the catch, so the economics count only results that met the standard.*

Fully loaded cost

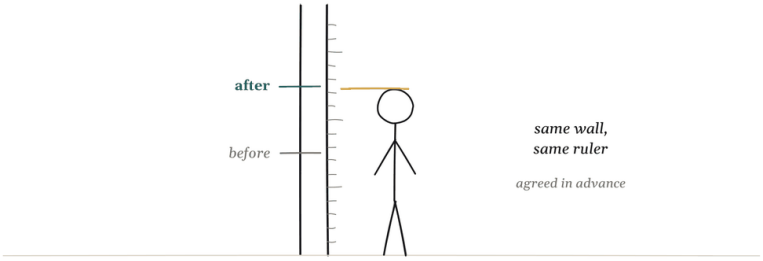
Fully loaded cost counts everything a loop consumes. That includes people's time at its real cost, systems and platforms, model usage, grading and audit, rework, and the cost of errors that reach customers or suppliers. Partial counts flatter whichever side leaves the most out. The commonest omission in embedded loops is the people still working inside them, at the junctions and in the exception queue.



Like an iceberg: *the model bill shows above the water, while people's time at the junctions, rework and errors sit below it.*

Baseline

The baseline is the loop's measured performance before it changes, defined in the Work and Process area. Its economic role is to be the reference against which every saving is claimed. It is measured the same way as the later economics, on a comparable period, and finance agrees the method in advance.

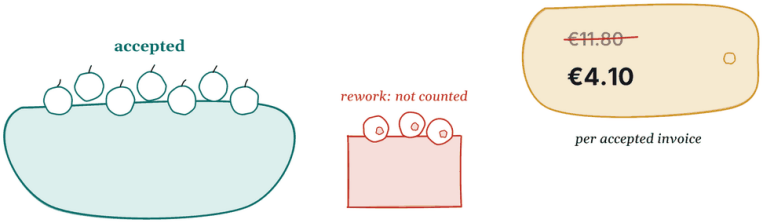


Like the pencil marks on a doorframe: *the child is measured the same way each time, against the same wall, so the growth claimed is real.*

Cost per accepted outcome

Cost per accepted outcome is the fully loaded cost of a loop over a period, divided by the outcomes in that period that met the standard. It is the central economic measure of an embedded loop. Dividing by accepted outcomes, not by cases processed, means that a loop which processes cases quickly but produces rework is not counted as cheap.

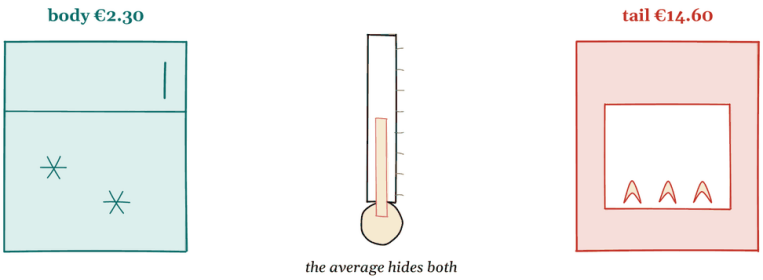
At the manufacturer, the baseline was €11.80 per invoice. After embedding, it was €4.10 per accepted invoice, measured over eight weeks across a quarter-end. At 7,400 invoices a month, the difference was worth about €680,000 a year.



Like paying a picker per good apple, not per apple picked: *divide the cost by the ones that passed, so fast but bruised picking is not cheap.*

Body and tail economics

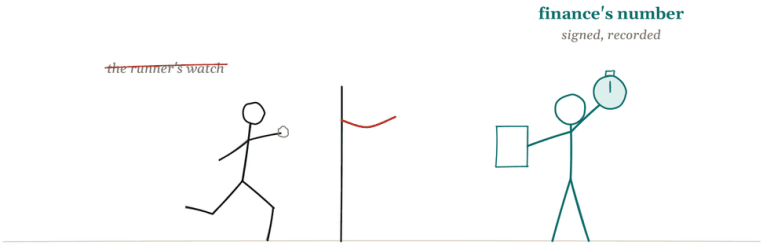
Cost per accepted outcome is reported separately for the body and the tail. A loop can cut the cost of routine cases sharply and leave the tail almost unchanged, because people still decide there. The average hides both facts. At the manufacturer, body invoices cost €2.30 each and tail invoices €14.60.



Like reading one thermometer between a freezer and an oven: *the average looks comfortable and describes neither, so body and tail are reported apart.*

Economics sheet

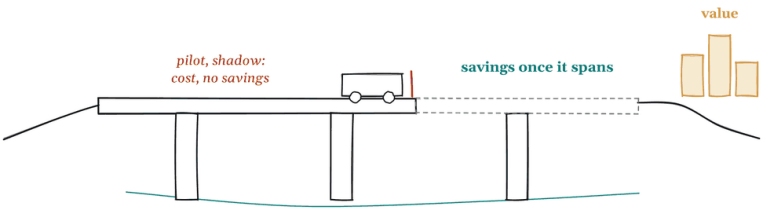
The economics sheet records the cost lines, the accepted outcomes, the cost per accepted outcome for the body and the tail, the comparison with the baseline, and the period. It is signed by finance, usually the planning and analysis lead, at Step 8. It is finance's number, not the programme's.



Like the official timekeeper at a race: *the runner's own watch may read faster, but the recorded time is the timekeeper's, as the economics sheet is finance's.*

The value curve

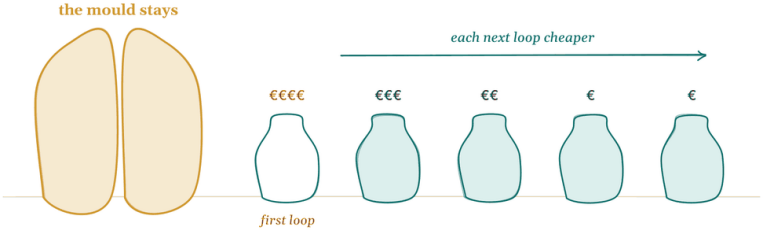
The value of an embedding programme arrives late. Pilots and shadow running produce evidence but no savings, because people still do all the work. Savings begin at Bound, on the body of the distribution, and grow at Core. A programme that stops before Bound has spent most of its cost and captured almost none of its value. This shape explains why so many surveys find little return from AI, and why finishing matters more than starting.



Like building a bridge: *the money goes into the piers first, and no one crosses until the span reaches the far bank, so stopping halfway earns nothing.*

The capacity to embed

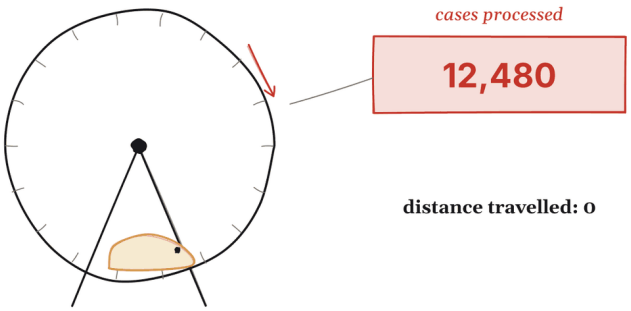
The first loop through the lifecycle is the most expensive, because the company is learning the steps. Later loops are cheaper and faster. The lasting asset is not any single loop's savings. It is the company's capacity to move loop after loop through the lifecycle, measured by the falling time and cost per loop.



Like a potter's mould: *the first pot pays for the mould, every later pot comes quicker and cheaper, and the mould is what you keep.*

Measures that mislead

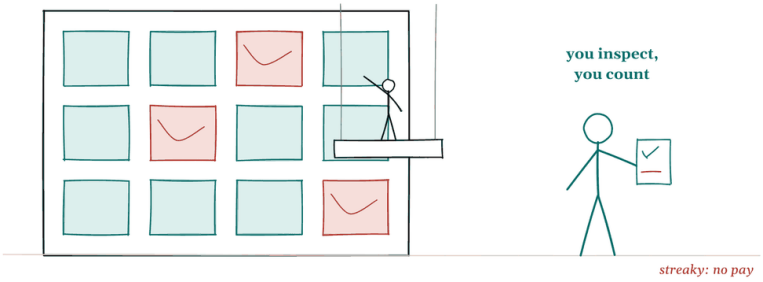
Several popular measures rise whether or not the business is better off: cases processed by AI, agents deployed, steps automated, hours saved estimated from task timings. They count activity, not accepted outcomes. The economist Charles Goodhart observed that a measure that becomes a target tends to stop measuring what it was meant to. Activity counts are especially prone to this, because they are easy to raise. They should be kept out of embedding reports.



Like a hamster wheel with a mileometer: *the count climbs fast while the hamster goes nowhere, so activity counts stay out of the report.*

Pricing by outcome

Some vendors price their executors by the outcome rather than by licence or usage: per resolved ticket, per processed invoice. Outcome pricing aligns the vendor with the loop, but only if "outcome" means an outcome that met the company's standard, graded by the company. A per-outcome price on outcomes the vendor counts itself is a usage price under another name.

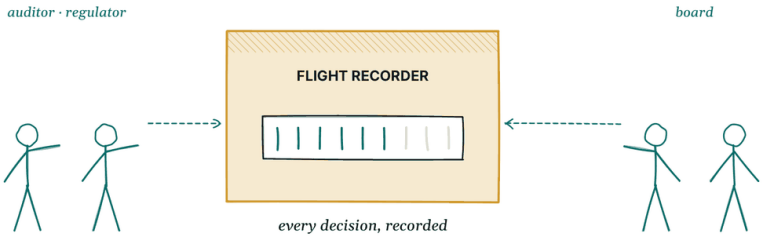


Like paying a window cleaner per clean window: *fair only if you, not the cleaner, inspect which windows are clean.*

XXVII

Governance

This area covers how the decisions made in embedding are recorded, reviewed and connected to the company's obligations. It is the area that turns good practice into evidence an auditor, a regulator or a board can rely on.

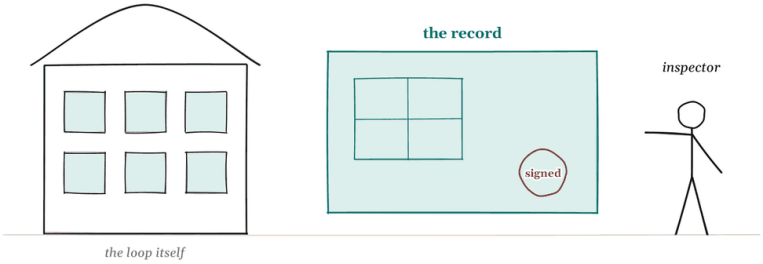


Like a flight recorder: *each decision is written down as it happens, so an auditor, a regulator or a board can rely on it later.*

Where it comes from

This area records the symbolic layer so that people outside the loop can rely on it. Signatures turn agreement into accountability. The record set is the model of the loop that auditors, regulators and boards read.

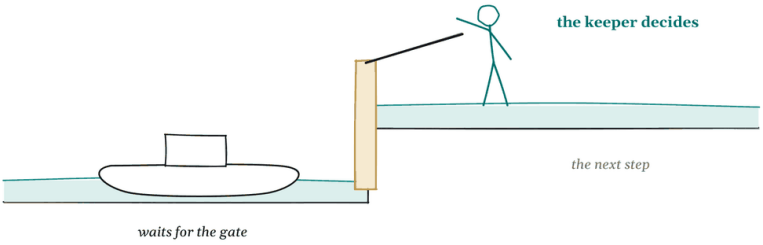
Concept	Kind	First used at
Gate	Concept	All steps
Gate review record	Instrument	Bound
Operating review	Instrument	Core
The record set	Concept	All steps
Scope change	Concept	Core
Governance frameworks	Concept	Bound, Core
High-risk loops	Concept	Read, Sort, Bound
Liability for what the executor says	Concept	Design



Like a building's stamped drawings: *inspectors read the signed plans to learn how the building works, without taking down a wall.*

Gate

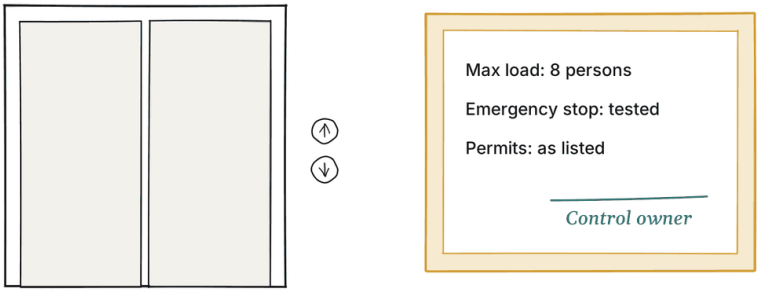
A gate is the point at the end of a lifecycle step where a named person decides whether the work may move on. It has written criteria and a signer. The idea comes from stage-gate reviews in product development, described by Robert Cooper in 1990, where gates protected companies from spending heavily on projects that should have been stopped. Gates in embedding protect against something different: an executor acting before the business has decided what it may do.



Like a canal lock: *the boat rises to the next level only when the keeper, having checked the water, opens the gate.*

Gate review record

The gate review record documents the decision at Bound. It records the limits the executor may act within, the evidence that supported them, the permissions granted, the switch-off test and any audit notes. It is signed by the control owner. It is the document an auditor asks for first.

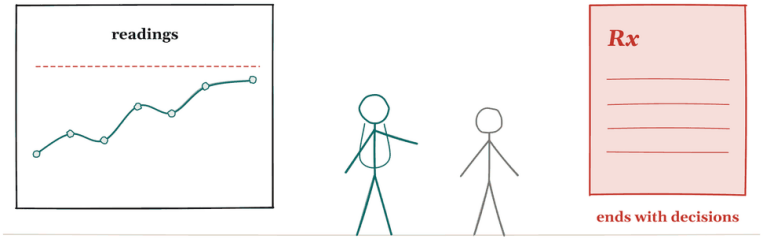


the first page an auditor asks for

Like the inspection certificate inside a lift: *it states the load limit and the last emergency-stop test, signed by the inspector, and it is checked first.*

Operating review

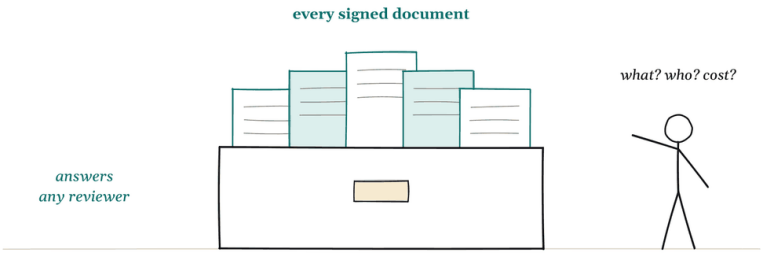
The operating review is the monthly review of a loop in the core, held by the loop owner. It reads the share of outcomes meeting the standard for the body and the tail, the cost per accepted outcome, the exception queue, the junction override rate and any drift. It ends with recorded decisions. A review that ends without decisions is a status meeting.



Like a doctor's check-up: *readings are taken every month, and the visit only counts if it ends with something prescribed.*

The record set

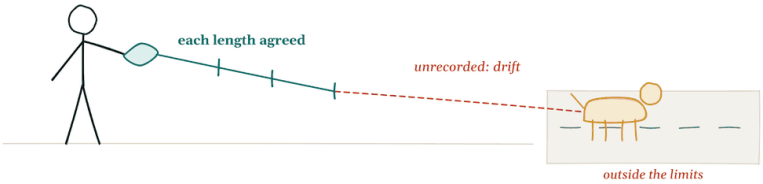
The record set is the full collection of signed documents for a loop. It includes the map, baseline, strip register, rule and decision registers, contested-decision log, canvas, junction register and standard. It also includes the shadow report, gate review record, loop charter, economics sheet, operating reviews and switch-off plan. Together they answer the questions any reviewer will ask: what the loop does, under which rules, who decides what, how it is graded, what it costs, and who answers for it.



Like a house's deeds box: *the survey, plans, permits and receipts kept together answer any question a buyer could ask about the house.*

Scope change

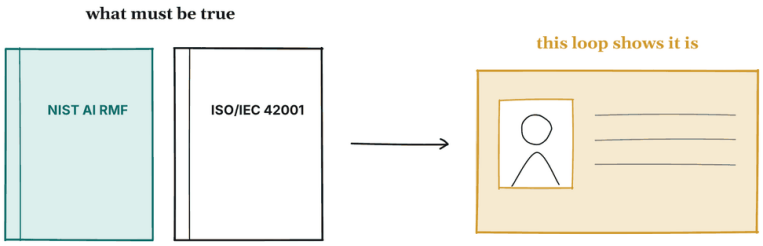
A scope change is any widening of what the executor may do alone: a new value band, a new kind of case, a new counterparty. Each is a small gate, recorded and agreed by the control owner. Unrecorded scope changes are the commonest way a loop drifts outside the limits that were signed.



Like an extending dog lead: *each extra length is clicked out on purpose, and a lead left unlocked lets the dog drift into the road.*

Governance frameworks

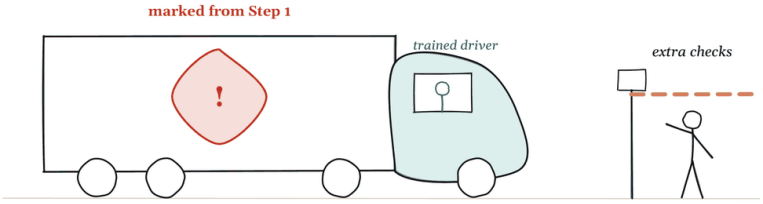
Several frameworks set out what good AI governance requires. The NIST AI Risk Management Framework, published in January 2023, organises the work into four functions: govern, map, measure and manage. ISO/IEC 42001, published in 2023, is a certifiable management-system standard for AI. Both describe what must be true. The record set is how a particular loop shows that it is.



Like the rules of the road and a driving licence: *the frameworks say what a safe driver must do, and the record set shows this one passed.*

High-risk loops

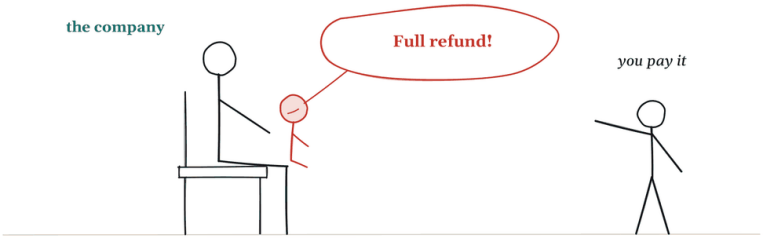
Some loops fall within laws on automated decisions. The European Union's AI Act, adopted in 2024, treats certain uses as high risk, such as decisions on employment, creditworthiness and access to essential services. It requires risk management, logging, human oversight by competent people and monitoring by deployers. Loops in these areas need legal or compliance input at Sort and at the Bound review. Their junction registers, switch-off plans and charters serve as legal evidence. A loop's risk category should be noted on the map at Step 1.



Like a lorry carrying hazardous goods: *it wears its warning placard from the start, and gets extra checks and a trained driver on the way.*

Liability for what the executor says

A company is generally held to what its executor tells customers and suppliers. In 2024, a Canadian tribunal held Air Canada responsible for incorrect refund information given by the chatbot on its website. Loops in which the executor makes promises, such as refunds, delivery dates and exceptions to policy, need those promises inside signed limits, with junctions where they exceed them.



Like a ventriloquist and his dummy: *whatever the dummy promises, the audience holds the ventriloquist to it, so promises stay inside signed limits.*

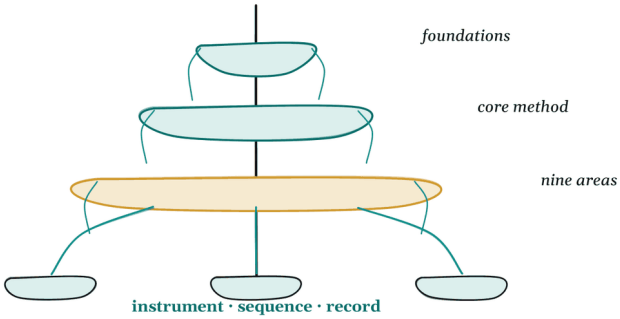


FIGURE 20 Governance. The record set of one loop: every signed document from the map to the operating reviews, each with its signer, shelved in lifecycle order.

XXVIII

How the Discipline Cascades

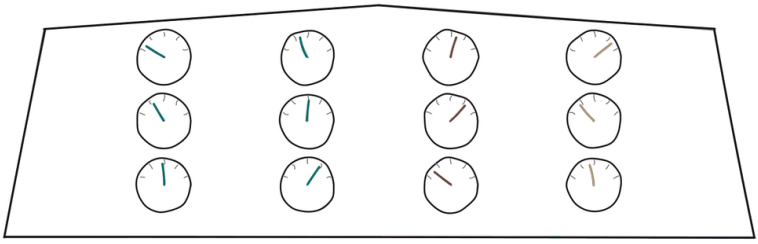
Everything else in the discipline follows from the first five parts. The foundations explain why an embedded loop must be a neurosymbolic, closed-loop system. The core method explains how a practitioner builds the model that regulates it. The nine areas define the knowledge that model is made of. From there the discipline cascades into three working forms.



Like a tiered fountain: *water poured in at the top fills each basin in turn, and the bottom basin spills into three working spouts.*

The instrument

The Work Loop Canvas is the nine areas compressed onto one page for one loop. Its twelve nodes are the areas made concrete. The outcome and economics come from value and economics, and the compensations from work and process. The rules and decisions come from the symbolic layer. Context and actions come from the executor, and junctions, exceptions, the standard and handoffs from control. The people come from accountability. The Work Loop Map is the same knowledge at the level of the company.

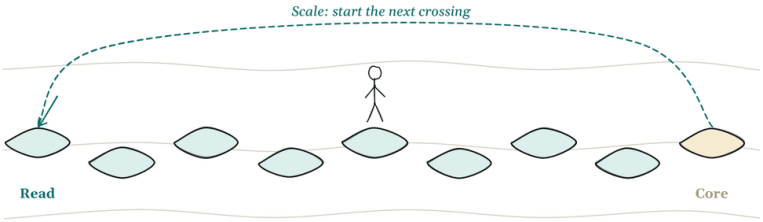


twelve dials, one loop, one panel

Like a car's dashboard: *everything under the bonnet shows up as one dial each, so the whole car is read from one panel.*

The sequence

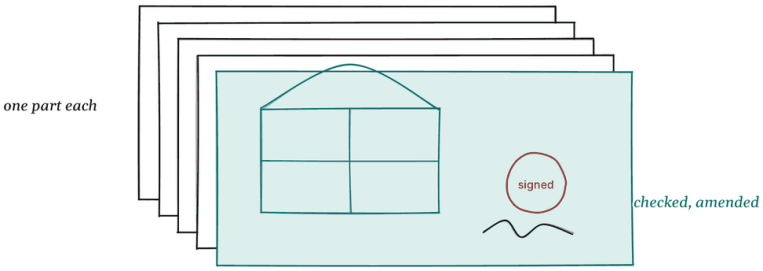
The embedding lifecycle is the core method run in time, one loop at a time. Read is the distribution and the map. Map, Strip and Sort are abstraction and compression applied to real work, producing the symbolic layer. Design closes the loop on paper. Shadow and Bound are testing: falsifiable claims about agreement and limits, put to real cases. Own measures the economics and names the people. Core runs the closed loop. Scale re-reads the distribution and starts again.



Like stepping stones across a river: *one loop crosses stone by stone from Read to Core, and at Scale you turn back to start the next crossing.*

The record

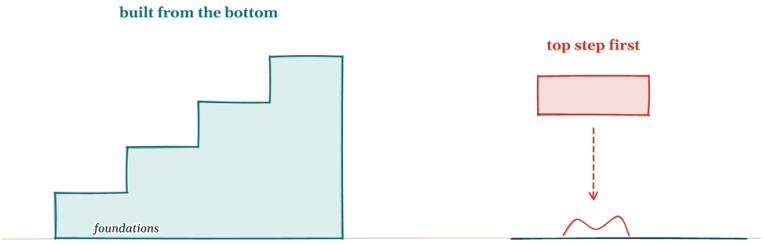
The template kit is the symbolic layer written down in a form people sign. Each template records one part of the model that regulates a loop. These include the map, baseline, registers, contested-decision log, junction register and standard. They also include the shadow report, gate review record, charter, economics sheet, operating review and switch-off plan. Together they are the record set: the company's model of its own work, auditable and revisable.



Like a building's set of signed drawings: *each sheet records one part of the house, and together they can be checked and amended.*

Why the order matters

The cascade runs one way. A company that adopts the templates without the areas will fill in forms it does not understand. A company that adopts the areas without the core method will define terms without reading its own distribution. And a company that adopts the method without the foundations will not know why the symbolic layer, the human junctions and the closed loop are not optional. Starting from the foundations is slower for a week and faster for every loop after.



Like a concrete staircase: *each step rests on the one below, so laying the top step first leaves it with nothing to stand on.*

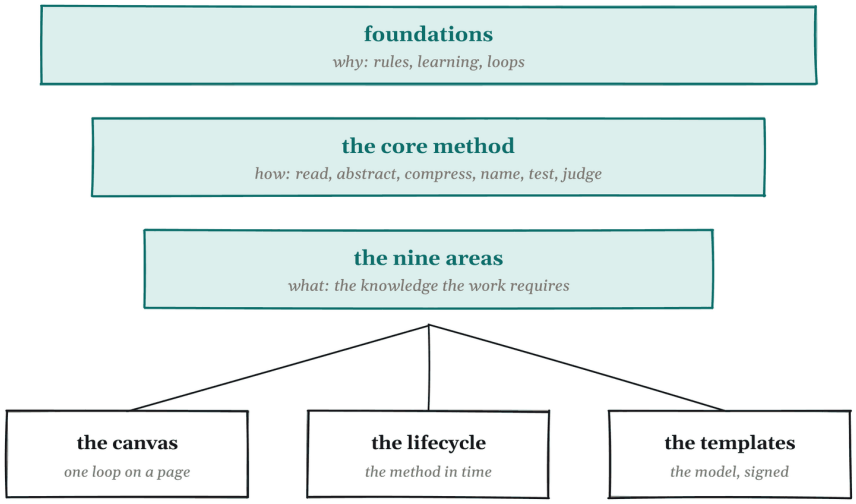


FIGURE 21 How the discipline cascades. The foundations explain why; the core method explains how; the nine areas define the knowledge; the canvas, the lifecycle and the templates follow.

PART SIX

A Century of Designing Work

XXIX

The Line

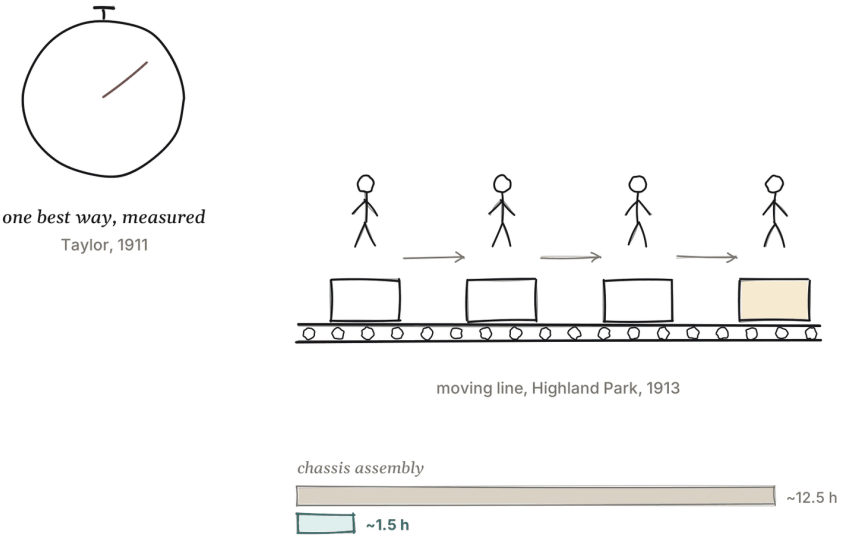
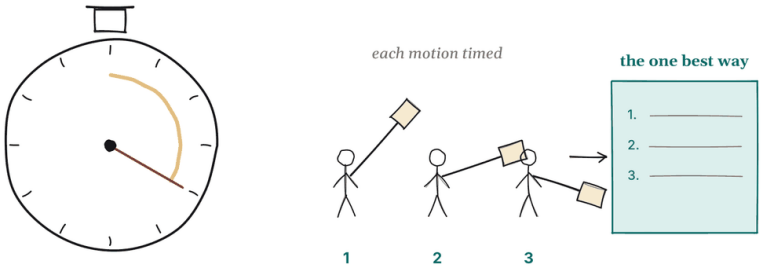


FIGURE 22 1911–1913: The Line

In 1911 Frederick Winslow Taylor published *The Principles of Scientific Management*. His claim was simple. Every task has one best way, and management's job is to find it, measure it and teach it. Taylor timed workers with a stopwatch, broke each job into motions and wrote down the fastest sequence. Work became something that could be studied and written as a standard.

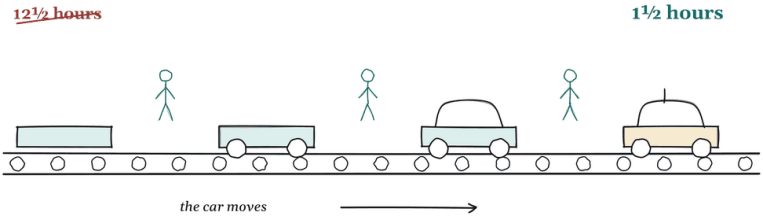
Taylor's methods were resented by many of the people they were applied to, and with reason. They treated the worker as a part to be tuned. But the underlying idea survived every backlash: a task can be described precisely enough that someone else can perform it to the same result. Every standard operating procedure written since descends from it.



Like a stopwatch held over a shoveller: *time every motion, keep the fastest sequence, and write it down so anyone can repeat it.*

Two years later Henry Ford put the idea in motion. In 1913 his Highland Park plant in Michigan introduced a moving assembly line for the Model T. Instead of workers moving around a stationary car, the car moved past stationary workers. Assembling a chassis had taken about twelve and a half hours. Within a few months it took about an hour and a half.

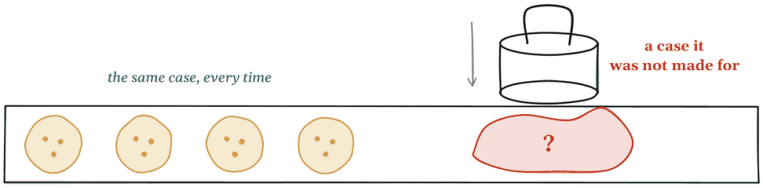
The line turned work into a designed flow. Each station received the car, did one thing and passed it on. The designer's job shifted from describing a task to arranging a sequence of tasks so the whole moved at a steady pace. **The handoff became the basic unit of industrial work**, and most process maps are still drawn that way, as boxes joined by arrows.



Like Ford's moving line: *the car rolls past workers who stay put, each does one thing and passes it on, and twelve hours become ninety minutes.*

Ford also discovered the cost of the design. The line was efficient because nobody on it needed to think. Turnover at Highland Park was so high that in 1914 Ford doubled pay to five dollars a day to keep people on it. The flow worked only as long as every case was the same case.

Two parts of the Work Loop Canvas come from this era. The standard, a written description of a good result, is Taylor's. The handoff, the point where work passes from one place to the next, is Ford's.



Like a biscuit cutter on a tray: *every biscuit comes out the same, until a lump of dough arrives that the cutter was never shaped for.*

XXX

The Divisions

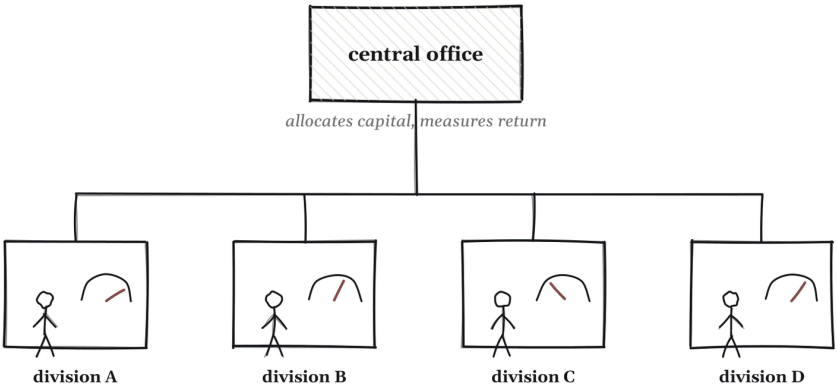
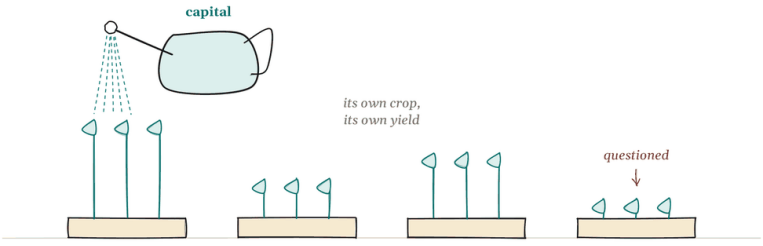


FIGURE 23 1920s: The Divisions

Ford's company ran on one product and one will. When the market wanted variety, it struggled to provide it. Alfred Sloan built something different at General Motors.

In the 1920s Sloan organized the company into divisions, each running its own operations for its own segment of the market. A central office did not run the divisions. It allocated capital among them and measured each one on its return. Divisions that earned their capital got more. Divisions that did not were questioned.

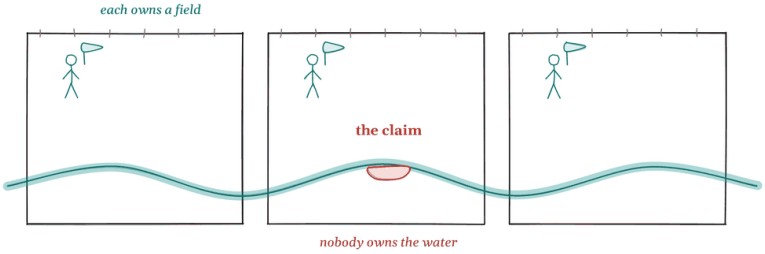


Like a gardener with one watering can and four beds: *each bed grows its own crop, and the water goes to the beds that yield the most.*

The historian Alfred Chandler later described this as the multidivisional form, and it became the standard shape of the large company for most of the century. Its real innovation was not the organization chart. It was accountability that could be read on a page. Each division had a named head and a number. A manager owned an outcome and answered for its economics.

That pairing of an owner with a measured result is the second inheritance a modern loop depends on. A loop without a named owner drifts, because nobody answers when it produces bad results. A loop without measured economics cannot be compared with anything, so nobody can say whether it is better than what it replaced.

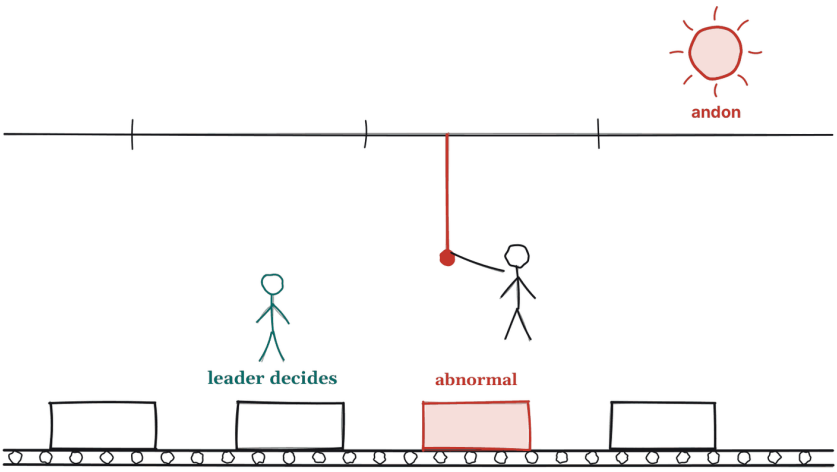
Sloan's model also carried a weakness the canvas has to correct. Accountability ran down the hierarchy, from the central office to division heads to their managers. Work, however, runs across the hierarchy, from one team to the next. A claim passes through intake, assessment and settlement, each in a different department with a different head. Everyone owns their piece. Nobody owns the claim. The work loop puts the owner where the work is, on the outcome rather than on the department.



Like a stream crossing three fenced farms: *each farmer answers for his own field, but nobody answers for the water that runs through them all.*

XXXI

The Stop Cord

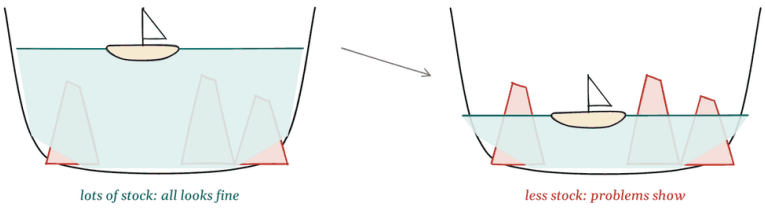


from the Toyoda loom that stopped when a thread broke

FIGURE 24 1950s–1970s: The Stop Cord

After the Second World War Toyota faced a problem Ford never had. It needed to build many models in small volumes, with little capital and little space. Ford's line, built for one model at enormous scale, did not fit.

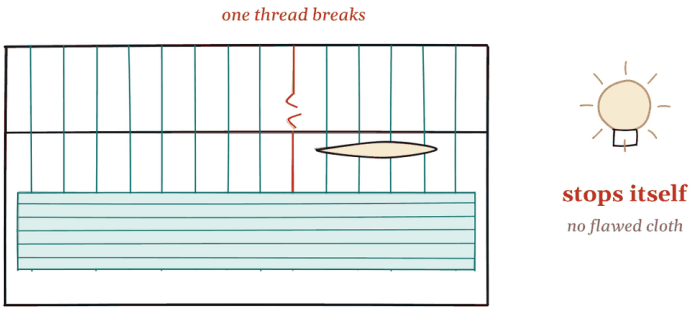
Taiichi Ohno and his colleagues built what became the Toyota Production System around two ideas. The first was just-in-time: make only what the next step needs, when it needs it. Inventory between steps was treated as waste, because it hid problems and tied up capital.



Like a lake that hides its rocks: *high water, the stock piled between steps, floats the boat over every problem; lower it and the rocks appear.*

The second idea was jidoka, sometimes translated as automation with a human touch. It came from Sakichi Toyoda's automatic loom, which stopped itself when a thread broke instead of weaving defective cloth. Toyota applied the same rule to people. Any worker who saw a problem could pull a cord running along the line, called the andon. A light came on, a team leader came over, and together they decided what to do. If they could not fix it quickly, the line stopped.

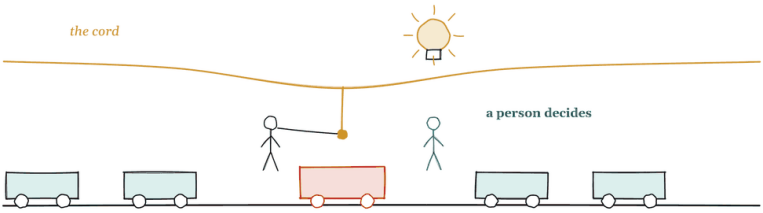
Western managers who visited were often shocked. Stopping the line was the worst thing that could happen in a Ford-style plant. At Toyota it was the point. A line that stops on every abnormality forces problems into the open, where they get solved instead of passed downstream.



Like Toyoda's automatic loom: *when a single thread snaps, the machine stops itself instead of weaving metres of flawed cloth.*

The andon cord is a junction in everything but name. The routine work runs on its own. An abnormal case stops it. A person with authority sees the evidence and decides. Toyota designed this decades before anyone talked about agents, and the principle carries over almost exactly. An agent should run the routine cases, stop on the abnormal ones and hand them to a person who can decide.

Toyota left two parts of the canvas. The junction is the cord. The exception queue is what happens after the cord is pulled: a defined place for the cases the routine cannot handle, worked by a named person against a time limit.



Like the andon cord on Toyota's line: *routine cars roll on, an odd one gets the cord pulled, a lamp lights, and a team leader decides.*

XXXII

Obliterate, Then Encode

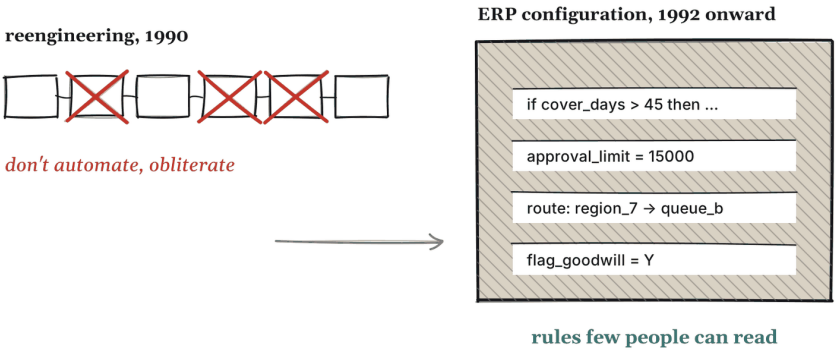
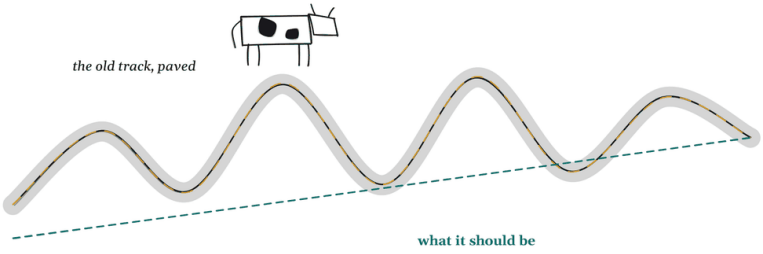


FIGURE 25 1990s: Obliterate, Then Encode

By 1990 large companies had computers everywhere and little productivity to show for them. Michael Hammer explained why in a Harvard Business Review article titled "Reengineering Work: Don't Automate, Obliterate." His argument was that companies had used technology to speed up old processes instead of questioning them. They had paved the cow paths.

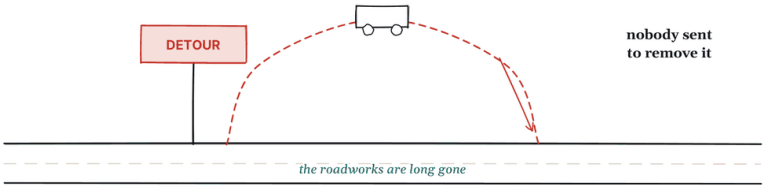
Hammer's examples were vivid. One large company had hundreds of people in accounts payable, matching purchase orders, receiving documents and invoices. A competitor did the same work with a fraction of the staff, because it had redesigned the process so that most of the matching was unnecessary. The lesson was that many steps existed only because of how work used to be done. They should be removed, not automated.



Like paving a cow path: *the tarmac makes the old winding track faster, but every bend stays, when the road should have been redrawn straight.*

Hammer and James Champy turned the argument into a book in 1993. Reengineering became one of the most popular management ideas of the decade, and then one of the most criticized. In practice it often failed. Projects were too large and too fast. The word became a polite label for layoffs, and employees learned to fear it.

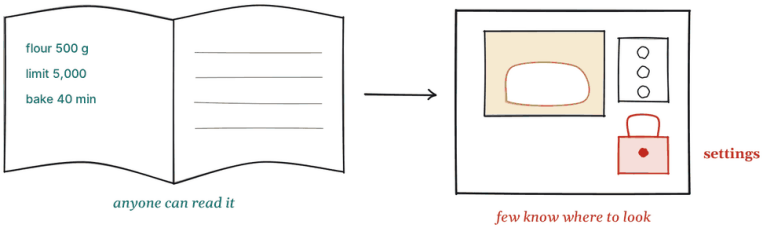
Reengineering had the right diagnosis and no executor. The redesigned processes still depended on people following new instructions and on systems that were hard to change. The diagnosis, though, is exactly the one the canvas starts from. Its second node, compensations, is Hammer's instruction in a sentence: strike every step that exists only to work around a constraint that no longer applies.



Like a detour sign left standing after the roadworks ended: *reengineering said take it down, but nobody was sent to do it, so cars still loop round.*

The same decade did something that made the next problem worse. SAP released R/3 in 1992, and enterprise resource planning spread through large companies. These systems wrote processes into software as configuration. Approval limits, routing rules and coverage conditions moved out of manuals and into settings screens.

That made processes more consistent. It also made their rules harder to see. A rule in a manual can be read by anyone. A rule in a configuration table can be read by the few people who know where to look. Decades later, when a company tries to tell an agent what the rules are, it often finds that nobody can list them. The canvas's rules node exists to bring them back into the open.



Like a recipe moved from the cookbook into a bread machine's program: *the loaves come out the same, but nobody can read the recipe any more.*

XXXIII

The Business Model on One Page

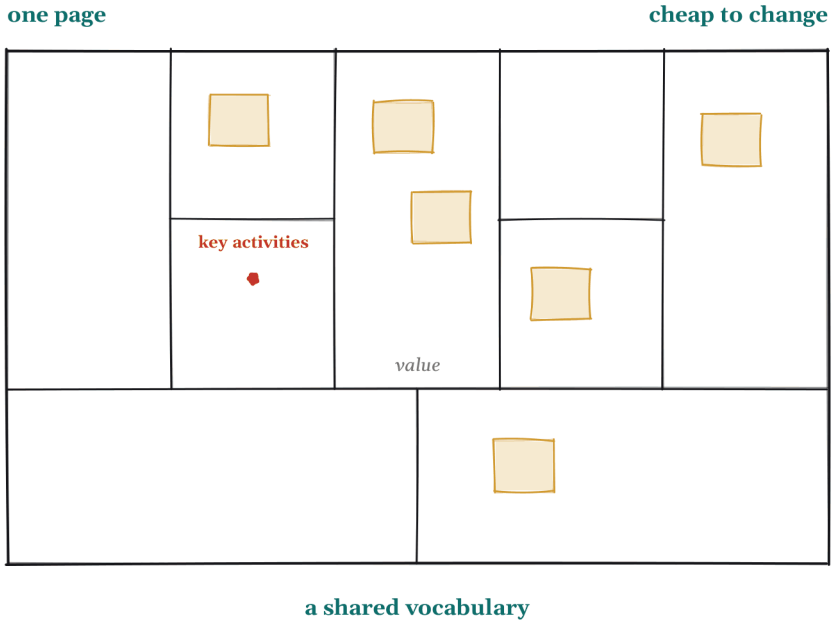
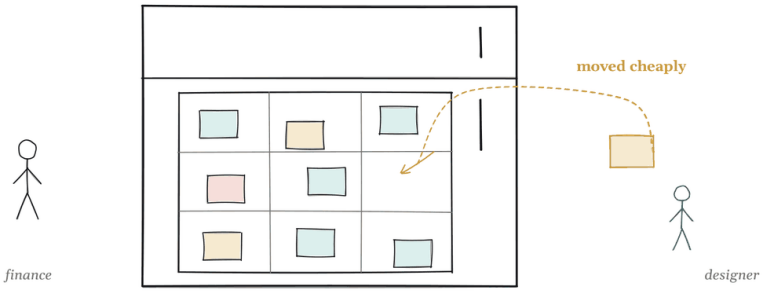


FIGURE 26 2000s: The Business Model on One Page

Alexander Osterwalder began his work on business models in a 2004 doctoral thesis. With Yves Pigneur he turned it into the Business Model Canvas, published in *Business Model Generation* in 2010. The canvas has nine blocks: customer segments, value propositions, channels, customer relationships, revenue streams, key resources, key activities, key partners and cost structure.

Its success had little to do with the blocks themselves. Most of them were already in strategy textbooks. It succeeded because of three properties. It fit on one page, so the whole model was visible at once. It gave a room of people a shared vocabulary, so a designer and a finance director could argue about the same box. And it worked with sticky notes, so a team could change its mind cheaply and in public.

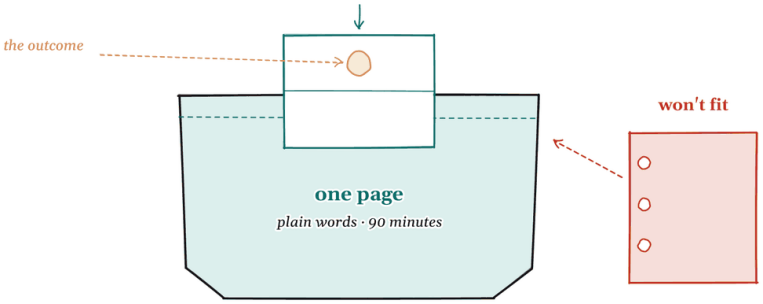


Like a family planner on the fridge door: *the whole week on one sheet, everyone reads the same days, and a magnet moves in a second.*

The canvas arrived when the hard problem in business was finding a model that worked. The web had opened new ways to reach customers and charge them. Founders needed a fast way to sketch, test and compare business models, and the canvas gave them one.

Any successor has to keep those three properties. A new canvas that needs a consultant to read it, or a week to fill in, has already failed. The Work Loop Canvas is built to the same rule. It fits on one page, it uses plain words a process owner already knows, and a team can fill a first draft in ninety minutes.

The Business Model Canvas also left behind the outcome node. Its centre is the value proposition, a statement of what a customer gets. The Work Loop Canvas keeps that habit and moves it inside. Its centre is the outcome, a statement of what one accepted result looks like.



Like a jacket pocket: *a successor canvas must still slip in as one folded page in plain words, while a thick consultant's binder will not fit.*

XXXIV

The Bots

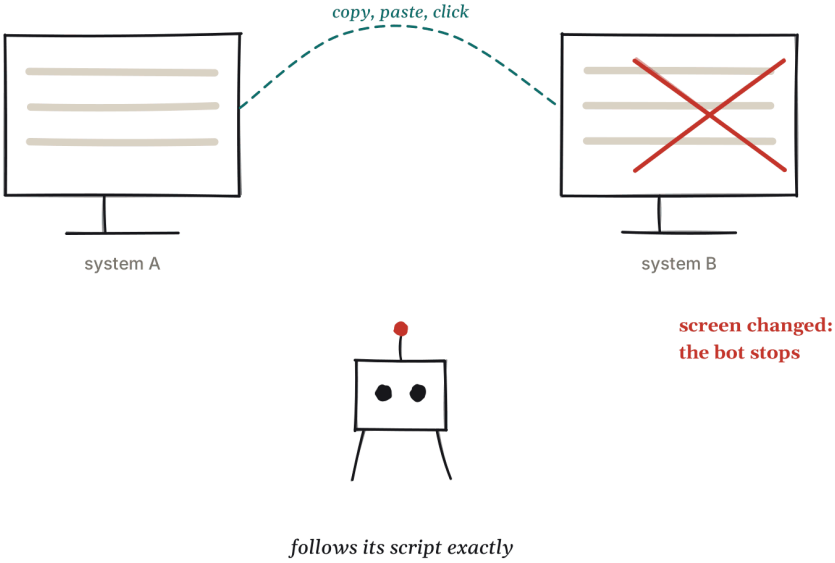
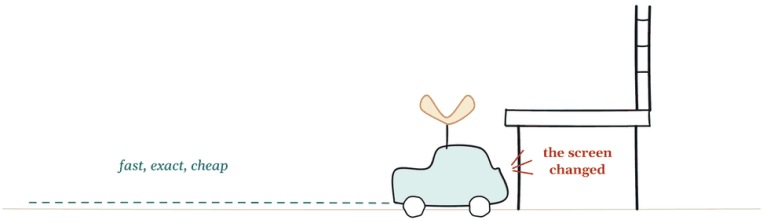


FIGURE 27 2010s: The Bots

The 2010s brought robotic process automation. Software robots clicked through the same screens people used, copying data from one system to another and filling in forms. They were fast, cheap to start and needed no change to the systems underneath. A team could automate a dull task in weeks without waiting for the IT department to rebuild anything.

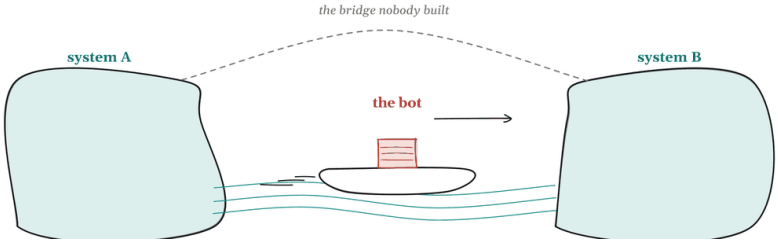
They were also brittle. A bot followed its script exactly. When a screen changed or a case fell outside the script, it stopped. Large programs ended up with teams whose job was to repair bots after every system update.



Like a wind-up toy car: *it races its track fast and exactly, but when someone moves a chair it hits the leg and stops.*

There is a deeper problem. Robotic process automation automated the workaround in the most literal sense. Re-keying data from one system into another is a compensation: it exists because the two systems were never connected. A bot that does the re-keying faster makes the compensation permanent. It recorded how people coped with disconnected systems and repeated it.

The bots left behind one useful part of the canvas, the actions node. They proved that software could act inside business systems: read a record, write a field, trigger a payment. The question they never answered was when it should. A bot has no judgment, so it either acts on everything in its script or nothing outside it.



Like a faster ferry between two banks that were never bridged: *the crossing speeds up, and the missing bridge becomes a permanent habit.*

XXXV

From Business Model to Business Engineering

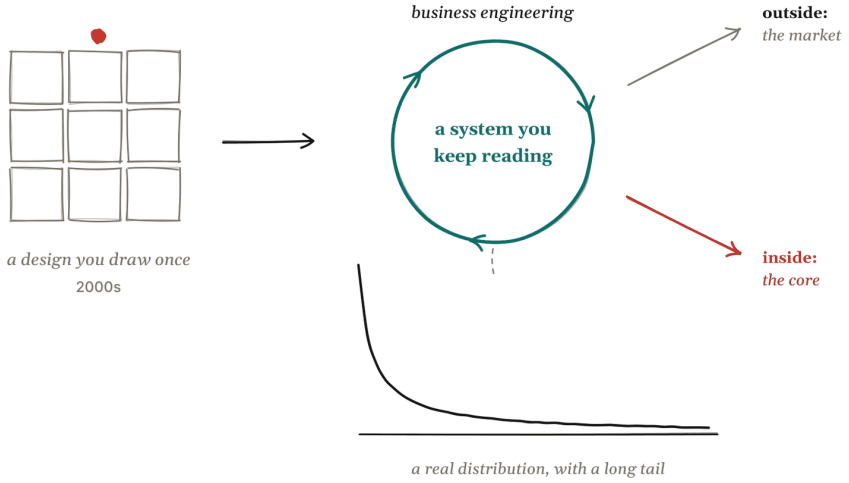
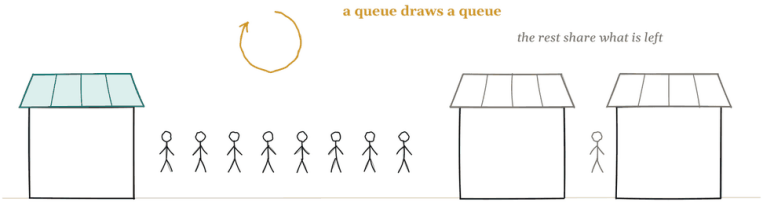


FIGURE 28 From Business Model to Business Engineering

The business model canvas treated a business as a design. A team drew it once, tested it with customers and adjusted it. That fit a world where most of a business model was a set of choices facing the market: which customers, which offer, which channel, which price.

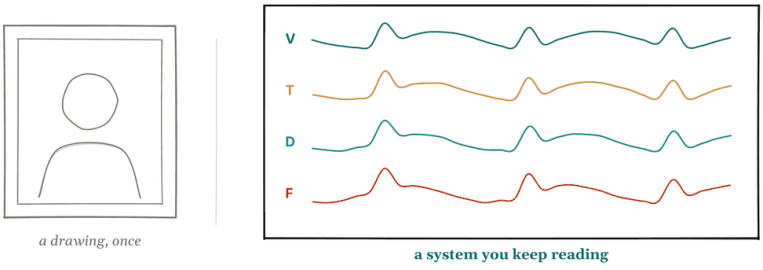
The web and its platforms broke that assumption. In the businesses that grew fastest after 2010, technology and distribution were not supporting blocks. They were the model. A search engine's results improved with every query it served. A marketplace grew more useful with every seller who joined. A software product spread through the people already using it. Growth compounded through loops, and outcomes followed power laws. A few products, customers and companies took most of the value, and the rest shared what was left.



Like stalls at a street market: *people join the longest queue, so it grows longer still, and a few stalls end up with most of the customers.*

A business like that is not a drawing on a page. It is a system running inside a real distribution of customers, cases and outcomes, and it changes every time it runs. The canvas could describe where such a business stood on a given day. It could not say how the business would behave next quarter, because that depended on feedback loops and on the shape of the distribution underneath. Drawing it was not enough. It had to be read for its structure, tested against what actually happened and adjusted as it went.

Business engineering was built to do that. It starts from the real distribution a business operates in, rather than from a list of boxes. It abstracts and compresses what it finds into mental models, tests those models against real cases and builds the intuition to notice when the distribution shifts. Its best-known instrument, VTDF, reads a business through its value, technology, distribution and financial models together, because in a platform business those four move together. **Business engineering turned the business model from a drawing into a system you keep reading.**



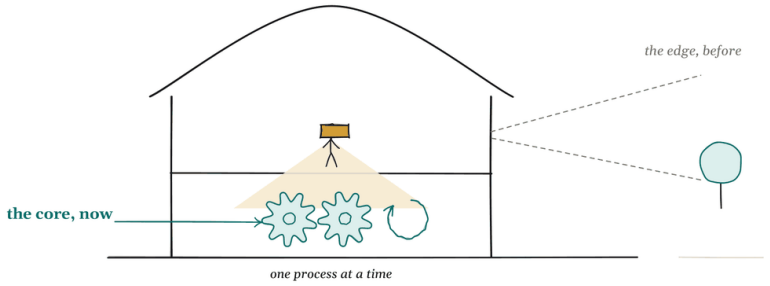
Like a bedside monitor instead of a portrait: *business engineering keeps reading four traces, value, technology, distribution and finance, as they move together.*

That shift is why business engineering is the key transition in this history, and not one more framework in the list. It changed the question from "what is the design?" to "how does this system behave in its real distribution?" Every

earlier school, from Taylor to the business model canvas, answered the first kind of question.

It also matters for a second reason. The habits that read a dynamic business from the outside are the same ones needed to read it from the inside. For the platform era, the dynamic part of a business sat at its edge, in distribution and product. The inside could stay a set of fixed procedures. Once the executor decides, that stops being true. The core of the firm begins to behave the way the platforms did: it runs on loops, it learns, it is uneven, and its tails carry the risk.

A canvas for the core therefore has to come out of business engineering, not out of the static template that came before it. The Work Loop Canvas is business engineering turned inward, applied to one process at a time. Part Two sets out the method in full.



Like a torch swung from the garden to the cellar: *the same light that read the business's edge now reads its core, one process at a time.*

XXXVI

What Each Era Left Behind

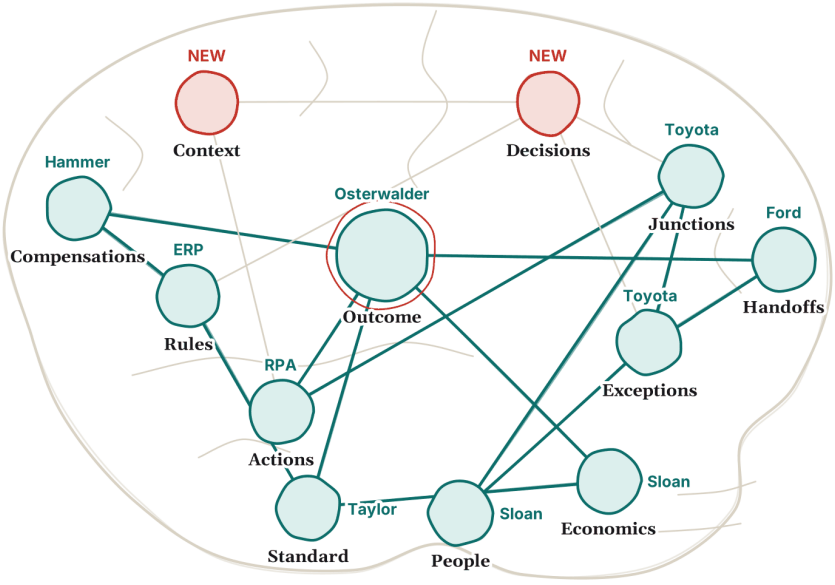
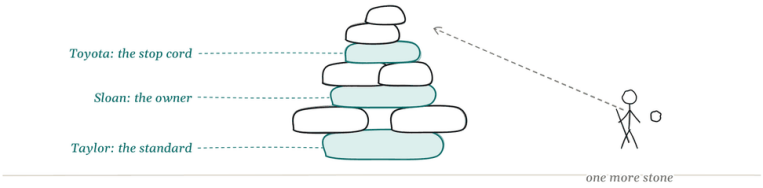


FIGURE 29 What Each Era Left Behind

Lay the century side by side and a pattern appears. Each era left behind one of the parts a modern loop needs.

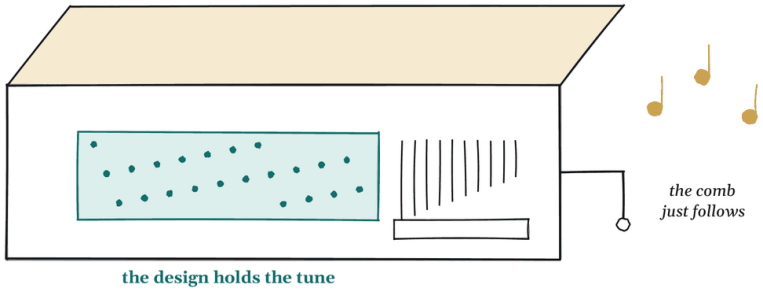
Taylor left the written standard. Ford left the designed flow and its handoffs. Sloan left the named owner and the measured economics. Toyota left the stop cord, the point where a person decides on an abnormal case, and the queue that catches what the line cannot handle. Reengineering left the instruction to strip out what no longer serves the outcome. Enterprise software left rules written into systems. Robotic process automation left software that acts inside them. The business model canvas left the habit of stating value on one page. Business engineering left the method that made the next step possible: reading a business as a running system in a real distribution, not as a fixed design.



Like a cairn built by passing walkers: *each era set down one stone, standard, flow, owner, stop cord, and the pile now marks the way.*

That accounts for ten of the twelve nodes. The two that remain are context and decisions. They are new because the executor is new.

Every earlier generation designed work for an executor that follows instructions. A worker on Ford's line, a clerk following an ERP screen and a software robot all have this in common. They do what the design tells them. The intelligence lives in the design, and the executor carries it out.



Like a music box: *the tune lives in the pinned drum, and the steel comb plays exactly what the pins tell it, nothing more.*

An AI agent is different. It reads context, interprets a case and makes decisions within the bounds it is given. That changes what the designer has to specify. It is no longer enough to describe the steps. The designer has to say what the executor may read, which decisions it may take alone and where a person must decide.

So the agentic era is the first to design for an executor that decides. Most of the parts are old and well understood. The executor is new, and it forces all the old parts into a single design. That design is the work loop.

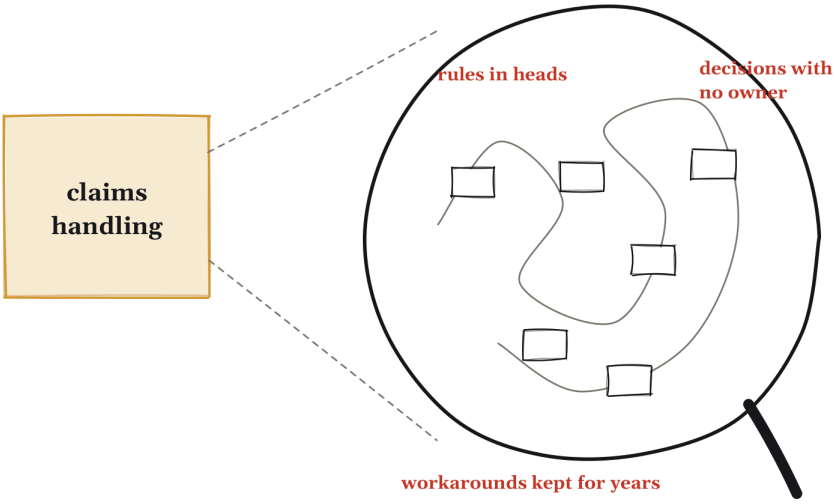
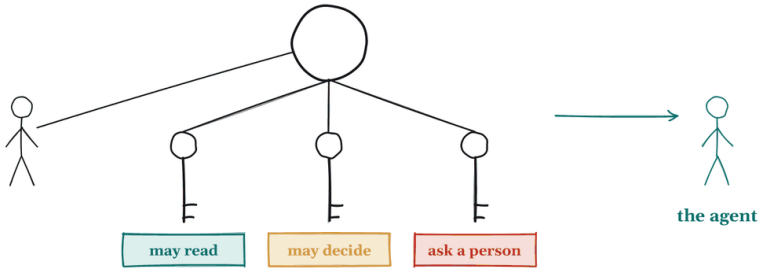


FIGURE 30 One Sticky Note Called Key Activities

A canvas is a template, and templates are easy to copy. What makes one useful is the way of thinking behind it. The Work Loop Canvas comes out of business engineering, a discipline for reading how a business actually works. This part sets out that discipline: where it starts, how it moves from observation to model and back, and how its best-known instrument, VTDF, connects to the canvas.



Like handing over a ring of tagged keys: *each tag says what the new holder may open, what they may decide alone, and where to ask.*

PART SEVEN

The Inside-Out Problem

XXXVII

Eleven Days

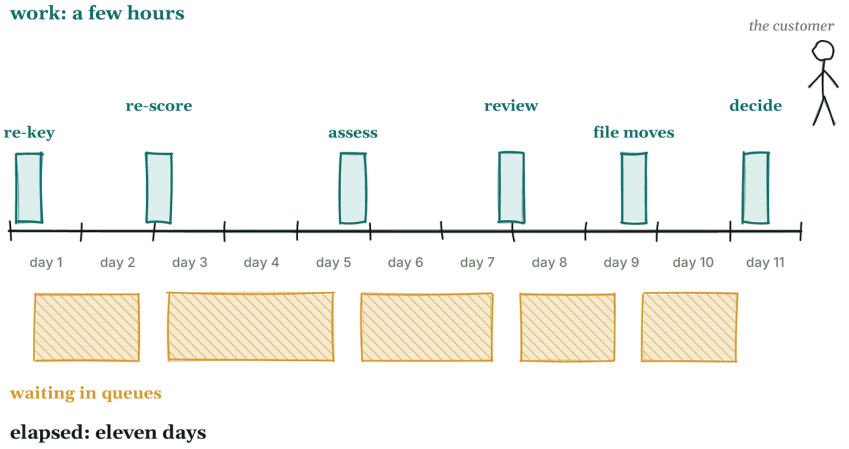
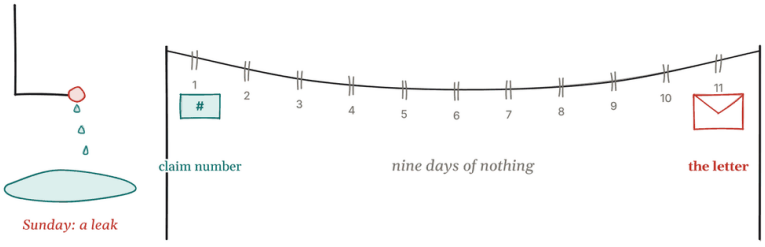


FIGURE 31 Eleven Days

A homeowner notices water spreading across the kitchen floor on a Sunday evening. A pipe behind the dishwasher has failed. By Monday morning she has photographed the damage, called her insurer and been given a claim number. She is told someone will be in touch.

Eleven days later she receives a letter with a decision and an amount.

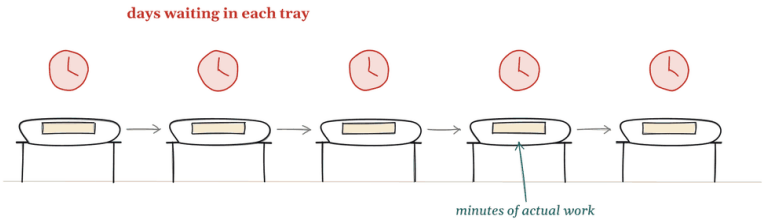


Like a washing line with eleven empty pegs: *the claim number arrives on the first day, and the letter with the answer only on the eleventh.*

Very little of those eleven days was work. Someone typed her call notes into a second system, because the call centre and the claims team use different software. A rating engine re-scored the claim overnight, because years ago it could only run at night. An adjuster assessed it. A second adjuster reviewed

the assessment, because the first step once produced errors that it no longer produces. A paper file moved between two teams on two floors. The claim waited in five queues.

None of this was anyone's mistake. Each step was added for a good reason, by a capable person, at a time when the reason was real. Over the years the process was measured, improved and documented. It is, by most standards, a well-run process.

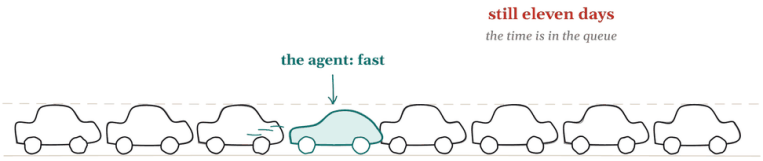


Like a letter passed along five desks: *each clerk handles it in minutes, but it lies in every in-tray for days, and nobody did anything wrong.*

Now the insurer wants to use AI. The obvious move is to take the process map and attach an agent to the steps that look automatable: the re-keying, the scoring, the first assessment. The agent will be fast. The claim will still take close to eleven days, because the time lives in the waits, the reviews and the handoffs that nobody questioned.

There is a second problem, and it is harder to see. Somewhere in those eleven days, people make decisions that nobody wrote down. Whether a borderline claim deserves a goodwill payment is one of them. Claims managers have paid them for years. The policy team says they were never authorized. Nobody noticed the disagreement, because experienced people quietly resolved it case by case. An agent cannot resolve it quietly. It will either guess or stop.

This part is about that gap. It is the gap between a business that knows what it sells and a business that knows how its work actually gets done. For most of the last twenty years the first kind of knowledge was the scarce one. Today the second kind is. It offers one instrument for closing the gap, explains where each part of it comes from, and shows how to use it from a single process to the whole firm.



Like a sports car stuck in a traffic jam: *a faster engine changes nothing when the time is spent waiting in the queue, not driving.*

XXXVIII

Outside-In, Inside-Out

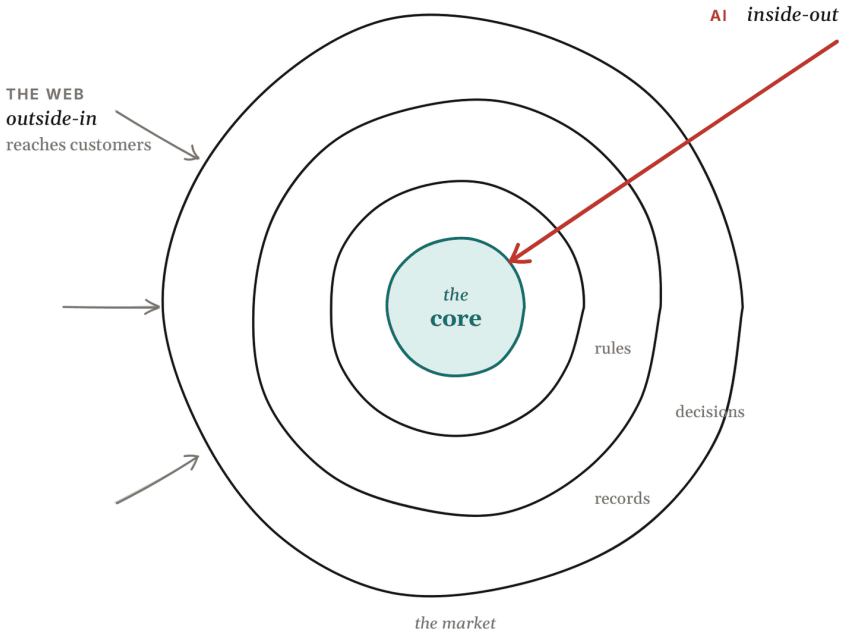
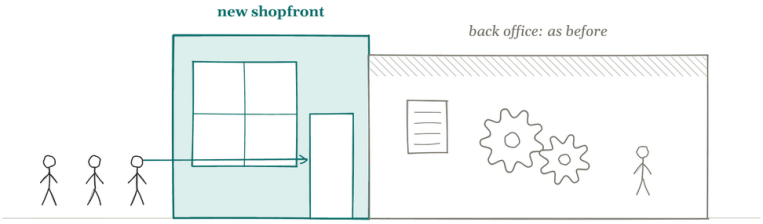


FIGURE 32 Outside-In, Inside-Out

The web was an outside-in technology. It changed how a firm met its market: how customers found it, bought from it and talked about it. Internal processes could stay largely as they were. A bank put its products online, and the back office kept running the same way. A retailer opened an online store, and the warehouse still picked orders from a list.

That is why the business model canvas, and later VTDF, focused so much on the outside. The advantage of the web era sat at the edge of the firm, in distribution, discovery and the customer interface. The Business Engineer has long described the relationship between the two technologies as Web Squared. AI does not replace the web. It compounds it, working from the opposite direction.

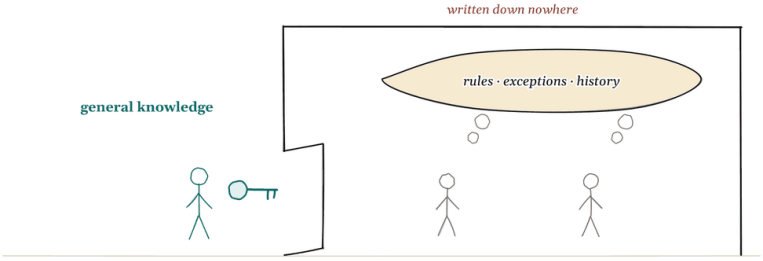


Like a shop given a new glass front: *the web changed how customers found and bought from the firm, while the workshop behind ran exactly as before.*

AI works from the inside out. Its value depends on entering the processes themselves: reading the case file, applying the rules, making the decision and writing to the system of record. A chatbot on the website is the outside-in use of AI, and it is the easy part. The hard part is the claim, the loan, the order and the invoice.

An AI model arrives with broad general knowledge and no knowledge of the firm. It does not know the firm's rules, its exceptions, its customers' history or which decisions need a manager. To be useful, it has to get inside. The inside is exactly where most firms are least documented, because for decades the people doing the work carried that knowledge for them.

So the frontier has moved. In the 2000s the scarce skill was designing the business model. Today it is mapping the core well enough for a new kind of worker to operate in it. **The hard problem moved from the business model to the process.**



Like a gifted new hire at the door on day one: *they know the trade, but the firm's rules and exceptions live only in the heads of the old hands.*

XXXIX

Why AI Stalls at the Edge of the Core

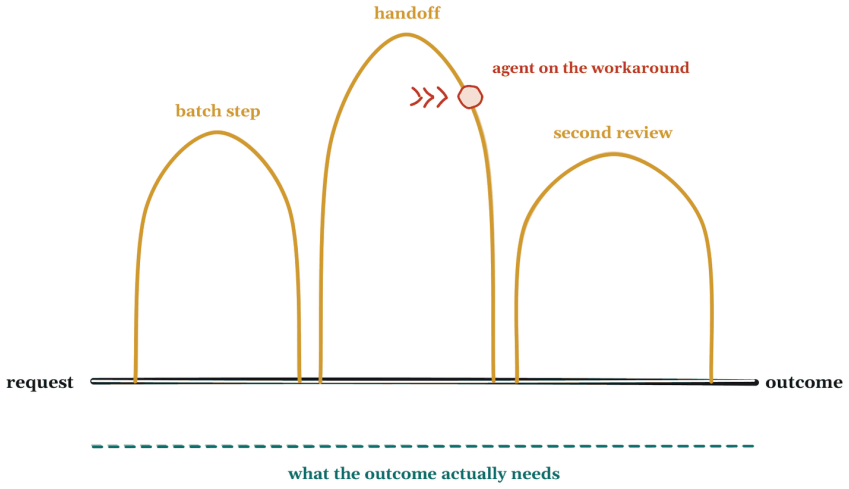
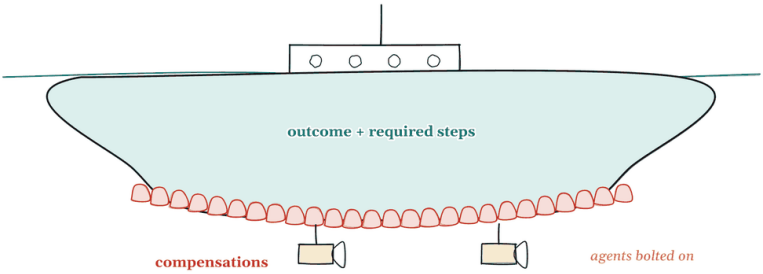


FIGURE 33 Why AI Stalls at the Edge of the Core

When a company tries to put AI into a core process, it usually starts from the process map it already has. The map shows a sequence of tasks. The team picks the tasks that look automatable and attaches an agent to each one.

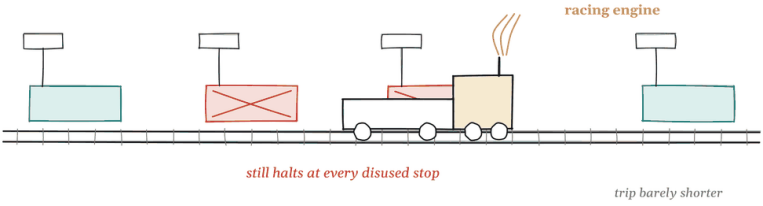
The results disappoint for a specific reason. A documented process is three things fused together. There is the outcome the process exists to produce. There is the minimum set of steps that outcome really requires. And there is a thick layer of compensations: steps that exist only to work around constraints that no longer apply.



Like a hull thick with barnacles: *the outcome and its real steps lie under years of workarounds, and the team bolts its new motors onto the crust.*

A batch step exists because a system once ran only at night. A handoff exists because two teams once sat in different buildings. A second review exists because an earlier step used to be unreliable. A form exists because a person once had to be prompted to collect the right information. Years of optimization polish these steps until they look structural. They appear on every map, they have owners and metrics, and nobody remembers why they started.

Attach an agent to them and it runs the workaround at machine speed. The pilot shows a gain on one task. The process as a whole barely moves, because most of its time and cost lived in the parts nobody questioned. This is the same mistake Hammer described in 1990, made with a more powerful tool.

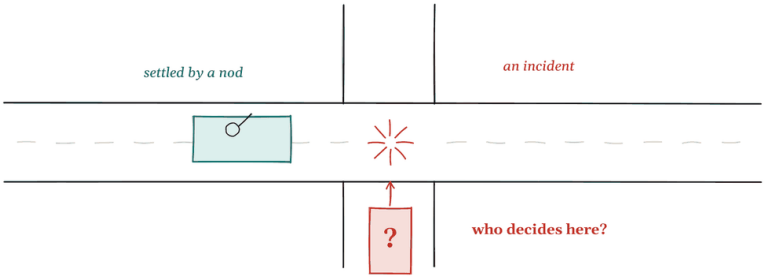


Like a racing engine in a train that still halts at every disused station: *one leg gets faster, but the journey barely shortens.*

There is a second reason pilots stall, and it matters more. Agents do not only follow steps. They make decisions. A process map shows who does each task. It almost never shows who owns each decision, which decisions a machine may take alone, and which ones the organization has never actually agreed on. Those questions stay invisible while experienced people resolve them quietly. They surface the moment something else has to act on them, usually as an incident.

The third reason is ownership. A pilot belongs to an innovation team or a technology team. The process belongs to an operations leader. When the pilot

needs a rule changed, a junction staffed or a budget moved, it has to cross that line, and it usually stops there.



Like an unwritten right of way at a country crossroads: *locals settle it with a nod, but a driverless car arriving there finds no rule at all.*

XL

The Unit Is the Work Loop

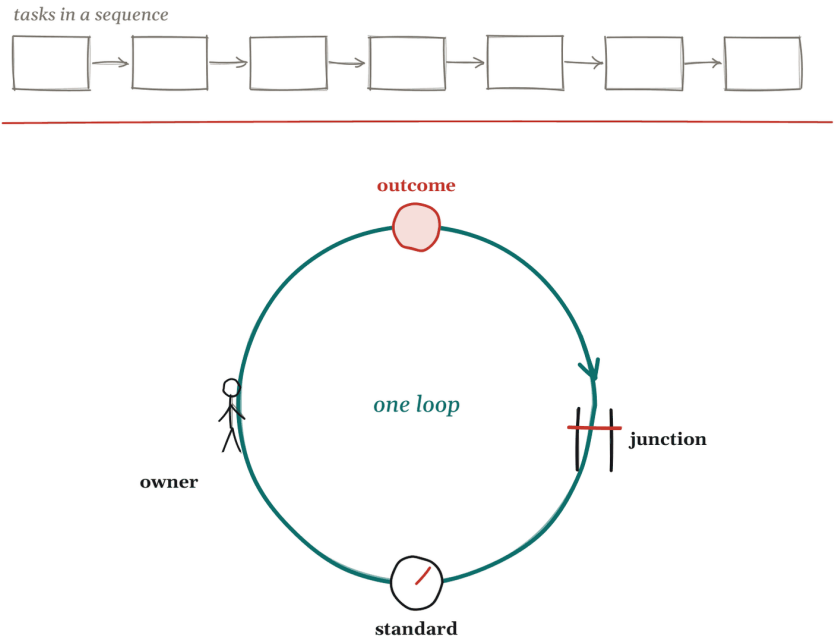
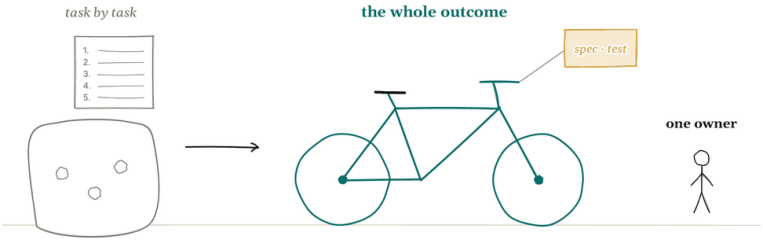


FIGURE 34 The Unit Is the Work Loop

The fix starts with changing what gets mapped. The task was the right unit for an executor that follows instructions, because instructions are written task by task. For an executor that decides, the right unit is the **work loop**.

A work loop is an outcome together with everything needed to produce it well. That includes the rules and context it depends on, the decisions it contains and the points where a person must decide. It also includes the standard every result is graded against and the person who answers for all of it.

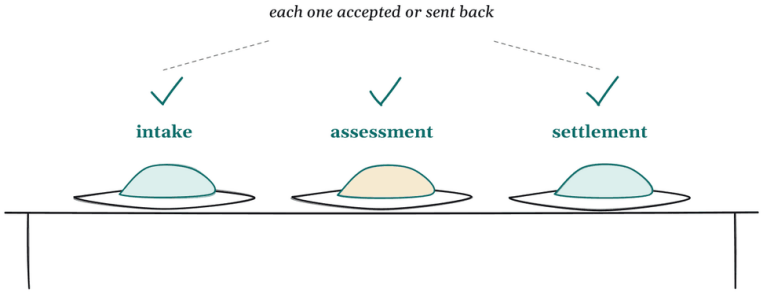


Like ordering a finished bicycle instead of a bag of parts: *the unit is the working bike, with its spec, its test ride and one maker answering for it.*

A work loop is smaller than a department and larger than a task. A claims operation might contain three: intake, assessment and settlement. Each produces something a customer or colleague can accept or reject. Each can be handed to a mixed team of people and agents as a whole, with one owner.

A useful test for the boundary of a loop is the acceptance test. If you can say what an accepted result looks like, and someone downstream can check it, you have a loop. If the only way to judge the result is to look at what happens three steps later, the loop is drawn too small. If the loop produces several different kinds of result for different customers, it is drawn too large.

Redesigning around loops tends to produce a few loops running in parallel, not a swarm of agents. That is easier to govern and easier to measure. It is also easier to explain to the people whose work changes, because each of them can see which loop they belong to and what it is for.



Like the courses of a dinner: *each plate is a whole thing a guest can accept or send back, larger than a garnish, smaller than the banquet.*

XLI

Shaped Like the Organization's Brain

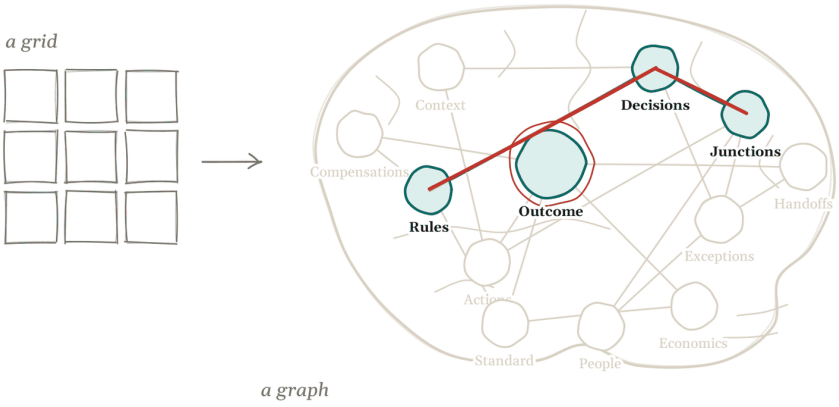
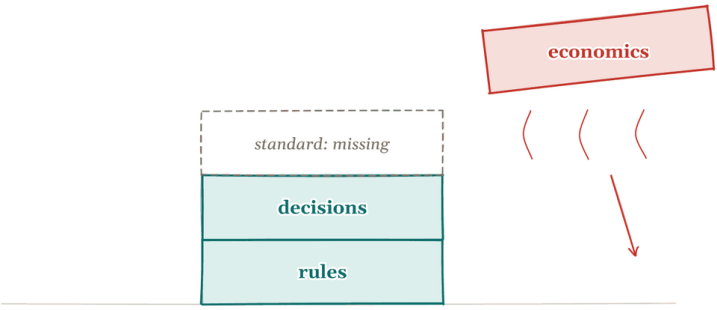


FIGURE 35 Shaped Like the Organization's Brain

The Business Model Canvas is a grid, and a grid suits its subject. A business model is a set of choices that sit side by side. A team can fill the blocks in almost any order.

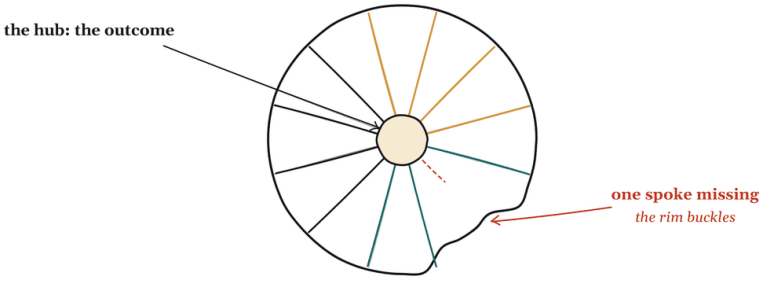
A work loop is different, because its parts depend on each other. A decision cannot be sorted until its rules are known. A junction cannot be placed until the decision behind it is sorted. The economics mean nothing without a standard that says what counts as an accepted result. Drawn as a grid, those dependencies disappear, and teams fill the easy blocks first.



Like stacking building blocks: *each rests on the one below, so a team that places the easy top block first leaves it hanging in the air.*

So the Work Loop Canvas is drawn as a graph. Twelve nodes sit in three regions around a single outcome, with lines between the nodes that depend on each other. One region holds what the loop knows and can touch. Another region decides. A third keeps the loop honest once it runs. The outcome sits in the middle, where every line leads.

The shape resembles a brain on purpose. In a brain, capability comes from regions wired together, not from any single region. The same is true of a loop. A strong rules node is useless if the decisions node is empty. A good standard is useless if nobody owns the loop.



Like a bicycle wheel: *spokes in groups around one hub, and a strong spoke is no use when its neighbour is missing, because the rim buckles there.*

The resemblance also does useful work in a room. When a node is signed, its connections light up. When a line stays dark, everyone can see which dependency is missing. Seen across the whole firm, the canvases form a network of small brains joined by handoffs. That is a fair description of how an organization actually thinks.

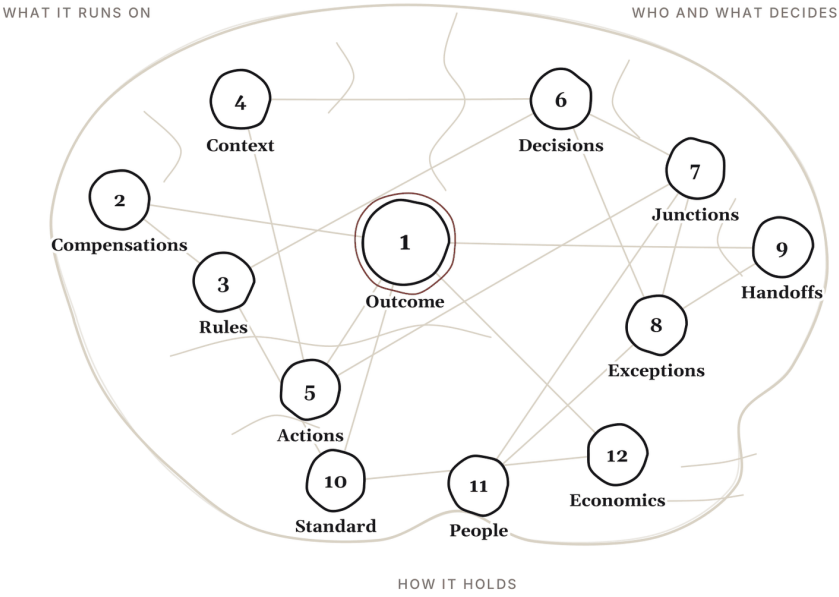
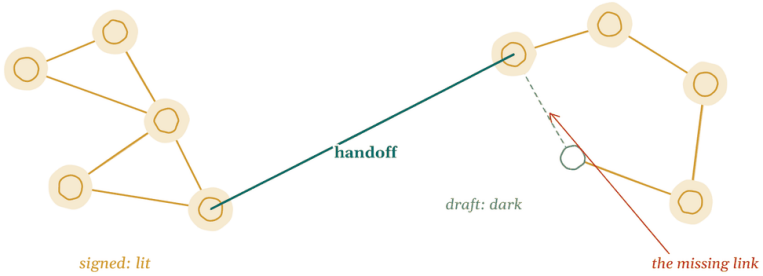


FIGURE 36 The Twelve Nodes

This part walks through the twelve nodes in the order a team should fill them. The order matters, because later answers depend on earlier ones. Each chapter gives the node's question, how to fill it, the test for when it is signed and the way it usually goes wrong. The claims assessment loop of the composite insurer runs through every chapter.

A node counts as **signed** only when its answer is written down, agreed and owned by a named person. Until then it is a draft. Draft is a sentence. Signed is a name next to it.



Like lanterns strung across a harbour: *each signed node lights its lines, a dark stretch shows the missing link, and handoffs join one cluster to the next.*

PART EIGHT

The Work Loop Canvas

XLII

Outcome

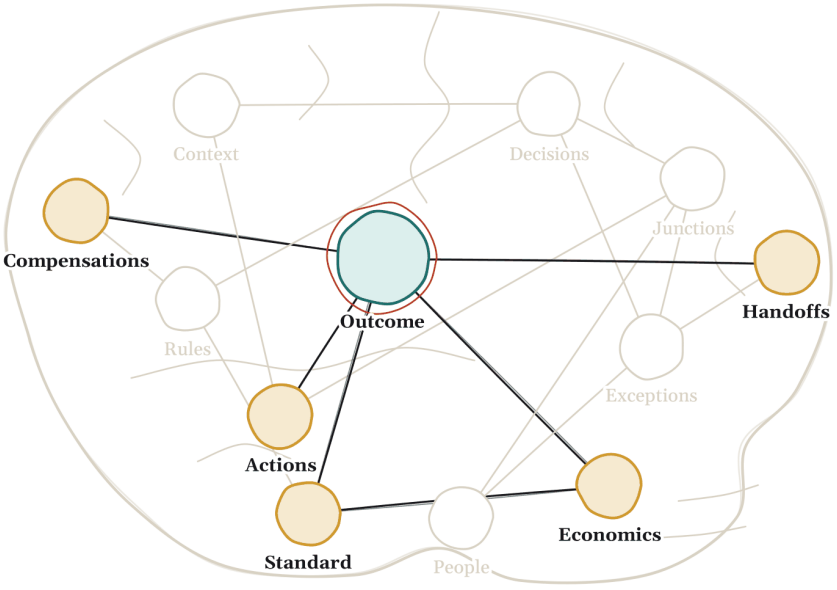
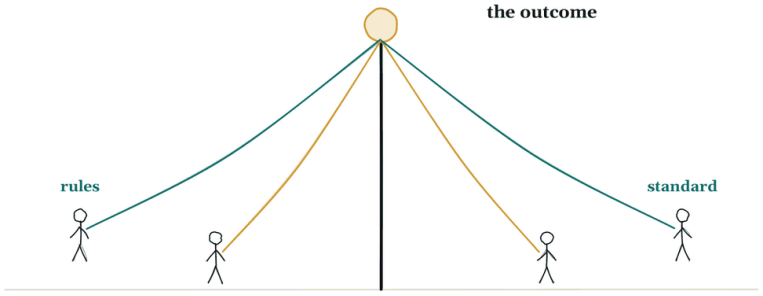


FIGURE 37 Outcome

What does one accepted result look like, and who receives it?

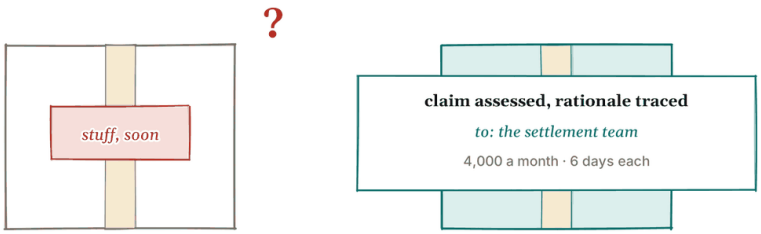
The outcome sits at the centre of the graph because every other node exists to serve it. It is the one sentence that defines the loop. Without it, rules have nothing to serve, the standard has nothing to grade and the economics have nothing to count.



Like a maypole: *every ribbon is tied to the one pole in the middle, and without the pole the ribbons have nothing to hold and fall in a heap.*

Write the outcome as a sentence a customer or colleague would recognize. "Process claims efficiently" is not an outcome. "A claim assessed, with liability and amount decided and a rationale that traces to the rules, ready for settlement" is one. The difference is that someone downstream can check the second one.

Add three facts beside the sentence: how many outcomes the loop produces in a month, how long one takes today and who receives it. The receiver matters more than it seems. A loop whose receiver is another internal team behaves differently from one whose receiver is a customer, because the cost of an error lands in a different place.

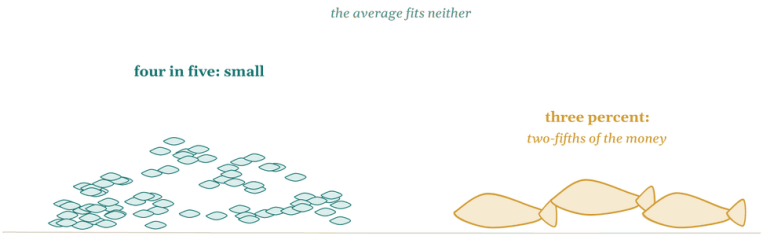


Like a parcel label: *'stuff, soon' gets lost on the way, while named contents and a named recipient let anyone downstream check it arrived right.*

Then draw the distribution of the cases the loop handles. Count them by size, type and value, and mark where the volume sits and where the money sits. This one picture shapes nearly every later node. It shows which decisions can run alone and where people are needed, and it shows which average will mislead the economics.

The outcome is signed when someone outside the team can tell an accepted result from a rejected one. The usual failure is an outcome written as an activity ("assess claims") rather than a result. Activities cannot be graded, so the standard node will stall later.

In the claims loop, the outcome sentence above was agreed in the first fifteen minutes. The volume was about four thousand claims a month. About four in five were under five thousand dollars, and the largest three percent carried around two-fifths of the money paid. Assessment took six of the claim's eleven days. The receiver was the settlement team, and through them the customer.



Like a catch sorted on deck: *a heap of small fish makes up most of the count, while a few big ones carry much of the value.*

XLIII

Compensations

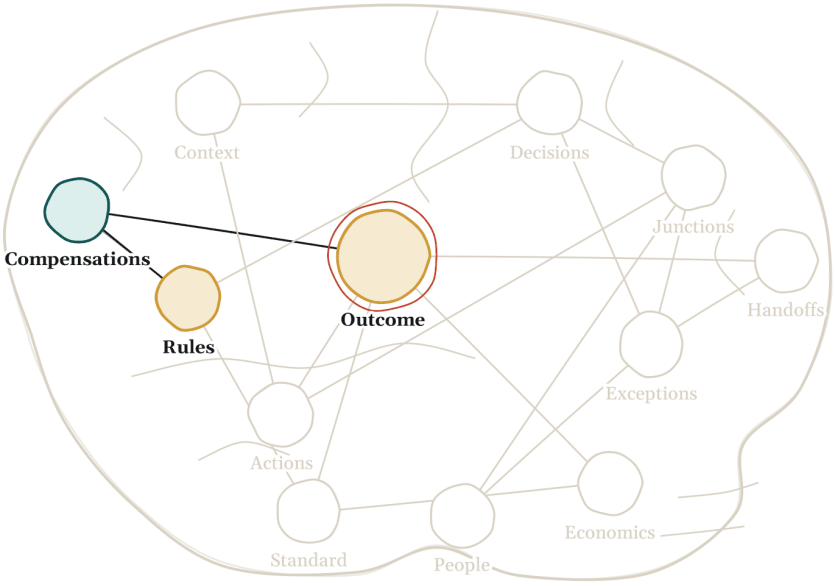
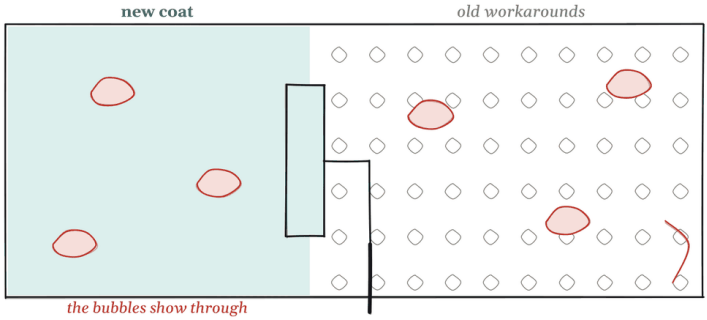


FIGURE 38 Compensations

Which steps exist only to work around a constraint that no longer applies?

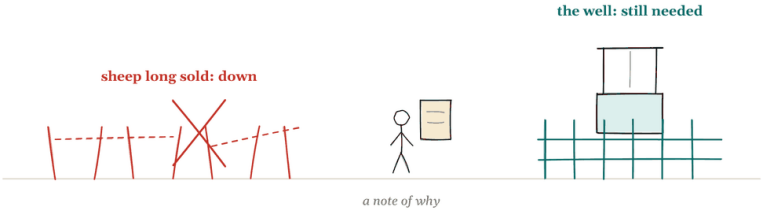
This node comes second for a reason. If a team writes rules and places agents before it strips the compensations, it will carry every workaround into the new design. Reengineering made this argument in 1990. The canvas makes it a required step.



Like painting over old wallpaper: *skip the stripping and every bubble and tear of the old workarounds comes straight through the new coat.*

Walk the current process from end to end and ask of every step: what constraint created this? Then ask whether that constraint still binds. Typical answers include a system that once ran only at night and teams that once sat apart. Others are a form that once prompted a person to collect information, a queue that once rationed scarce attention and a review that once caught errors from an unreliable step.

Strike the steps whose constraints are gone. Keep a short note of each one and the reason, because someone will ask later why it disappeared. Keep the steps the outcome genuinely needs, even when they are slow. A site inspection for structural damage is slow, but the outcome requires it.

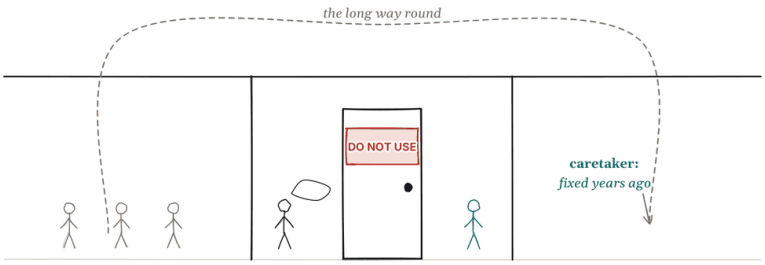


Like walking a farm and asking why each fence was built: *the pen for sheep long sold comes down, while the fence round the well stays.*

The node is signed when every remaining step is required by the outcome itself. The usual failure is deference. Teams hesitate to strike a review that a senior person introduced, or a step a regulator once asked for. The fix is to check: ask the senior person, read the regulation. Most compensations survive only because nobody asks.

In the claims loop, three steps went. The overnight re-score was struck because the rating system now runs on demand. The second review of every claim

under two thousand dollars was struck because the re-keying errors it caught no longer happen. The paper file transfer between intake and assessment was struck because both teams use the same system.



Like a 'do not use' sign on a door fixed years ago: *everyone walks the long way round until someone simply asks the caretaker.*

XLIV

Rules

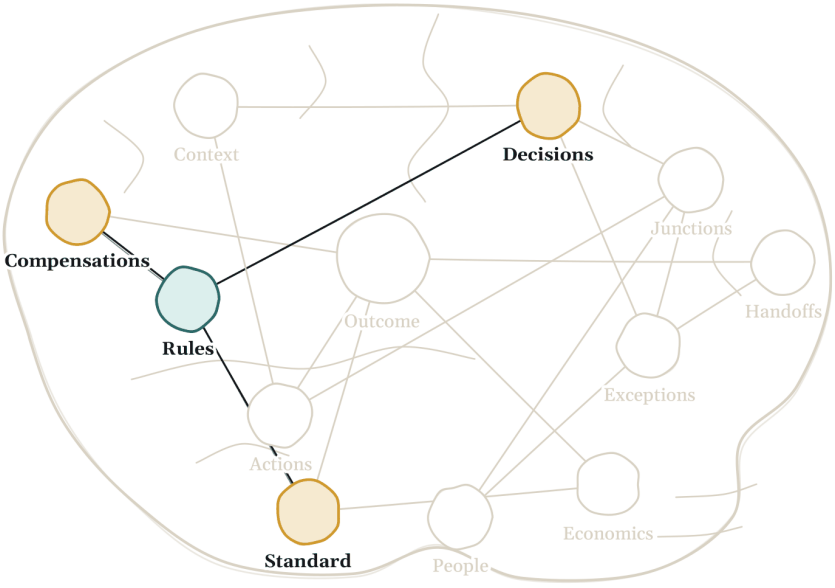
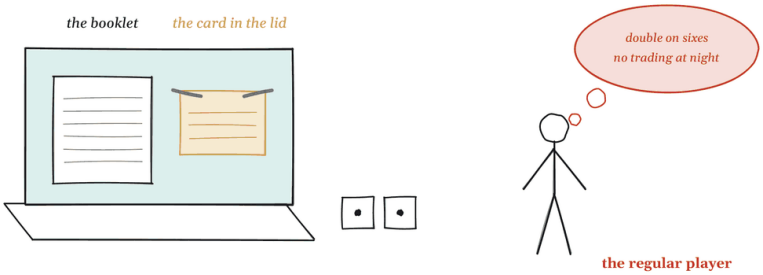


FIGURE 39 Rules

Where does each rule live today, and who owns it?

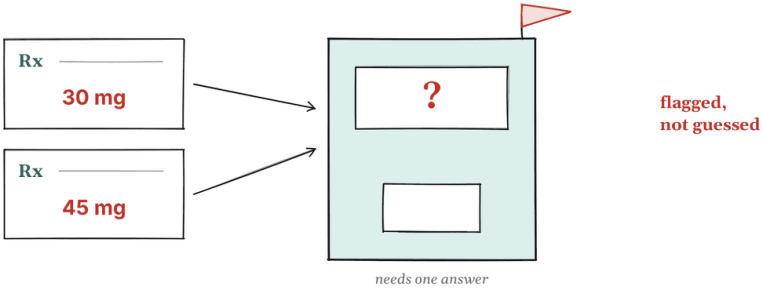
Rules are the instructions a decision has to follow: coverage conditions, approval limits, eligibility criteria, routing logic. In most firms they live in three places at once. Some are in policy documents. Some are in system configuration, written there during an enterprise software project years ago. Many live only in experienced people's heads.



Like a family board game: *some rules sit in the booklet, some on a card taped in the lid, and many only in the regular player's head.*

Gather them in one list. Interview the people who do the work as well as the people who wrote the policies, because the two groups often describe different rules. Read the configuration tables with someone who knows them. For each rule, record where it lives, who owns it and when it was last confirmed.

Contradictions will appear. Treat them as findings, not noise. A contradiction between the manual and the system means that, for some cases, the firm has been applying two different rules without knowing it. An agent cannot do that quietly. It needs one answer.

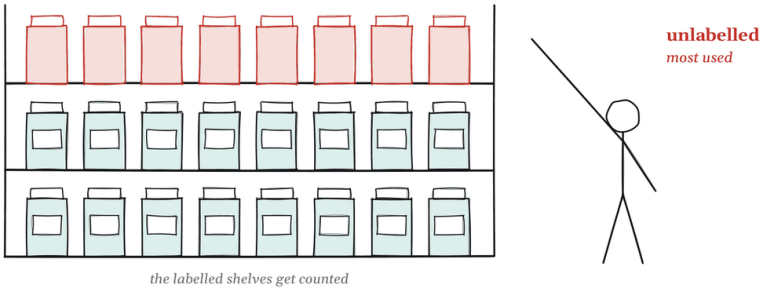


Like a pharmacy given two prescriptions with different doses: *a person might quietly pick one, but the dispensing machine needs one answer, so the clash is flagged.*

The node is signed when every rule has an owner and a date, and the contradictions are listed with an owner for each. The usual failure is stopping at the documents. The rules that matter most are often the unwritten ones, the judgment calls experienced staff make so routinely that they no longer notice them.

In the claims loop, the team found 38 rules. Twenty-one were in the claims manual, nine in the rating engine's settings and eight only in senior adjusters'

heads. Seven contradicted each other. In one, the manual covered water damage after a property had been empty for thirty days, while the engine used forty-five. The Head of Claims Policy took ownership of the list.



Like stocktaking a pantry: *counting only the labelled jars misses the unlabelled ones on the top shelf, which are the ones the cook reaches for most.*

XLV

Context

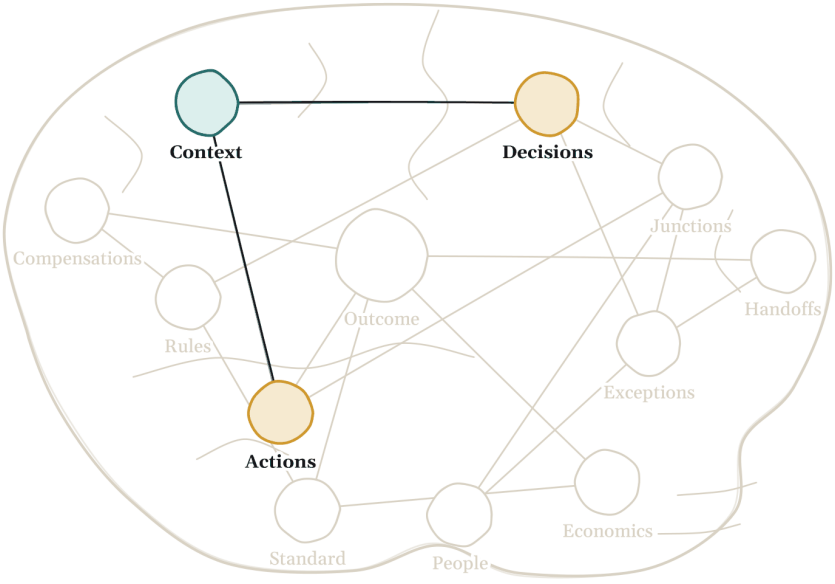
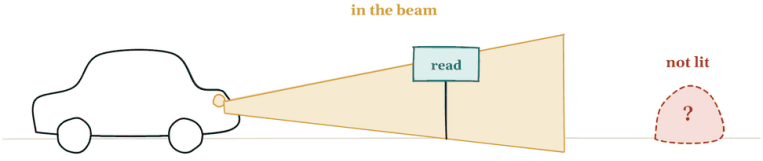


FIGURE 40 Context

What must the executor read to act well?

Context is new to the canvas. No earlier generation needed it, because the executor either knew the context already, as an experienced worker does, or did not need it, as a script does not. An agent needs it spelled out. It can only act on what it can read.

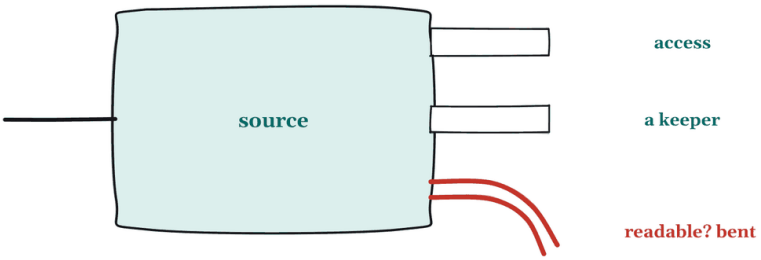


Like driving at night: *the car can only respond to what its headlights reach, so whatever the agent needs must be lit up for it to read.*

List every source a skilled person consults to do the work: records, documents, history, reference data, notes from colleagues. Then ask, for each one, three

questions. Can the executor reach it with the right permissions? Who keeps it current? Is it in a form a machine can read, or is it a scanned page, a photo or a spreadsheet on someone's laptop?

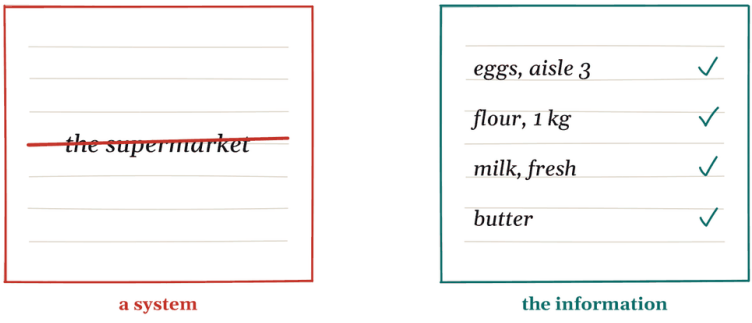
Gaps here are common and usually fixable. They are also the most frequent reason a pilot that worked on clean test data fails on real cases. The real cases depend on a source nobody listed.



Like a three-pin plug: *every source needs access, a keeper and a machine-readable form, and one bent pin stops the appliance working on real cases.*

The node is signed when each source has a keeper and the executor can reach it with the right permissions. The usual failure is listing systems instead of information. "The claims system" is a system. "The prior claims on this policy, from the claims system" is information an assessment actually uses.

In the claims loop, the sources were the policy and its endorsements, the claim notes and photos, repair estimates, prior claims on the policy and a supplier price list kept by procurement and refreshed monthly. Inspection reports arrived as scanned documents. Operations agreed to convert them on arrival, and the node was signed.



Like a shopping list: *'the supermarket' names a place, while 'eggs, flour, milk' names what the recipe actually uses, and only the second list gets you cooking.*

XLVI

Actions

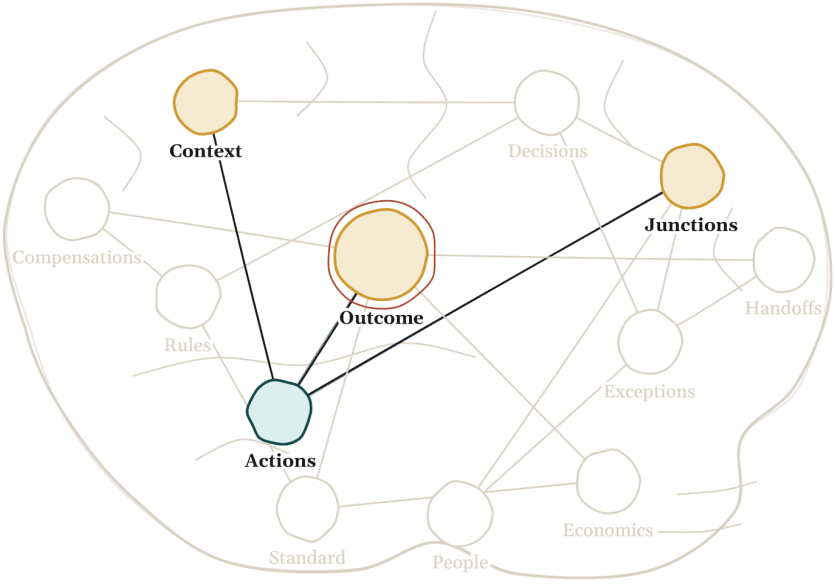
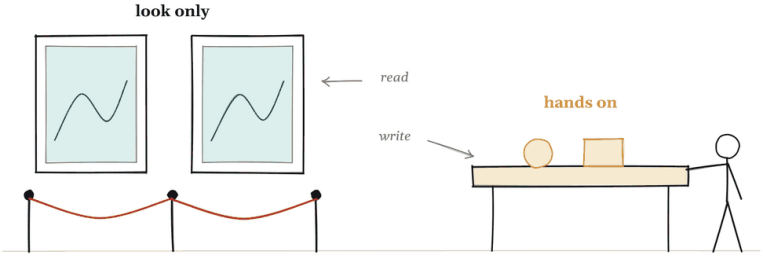


FIGURE 41 Actions

What can the executor change, and in which systems?

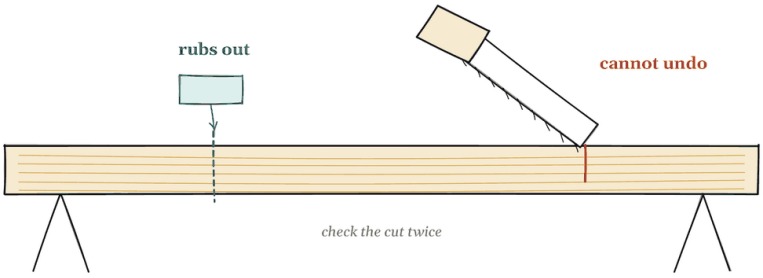
Robotic process automation proved that software could act inside business systems. This node decides what this loop's executor may actually do. It separates reading from writing, system by system.



Like a museum visit: *you may look at every painting, but you may touch only the hands-on table, and the rules differ room by room.*

List every system the loop touches. For each one, record whether the executor reads from it, writes to it or both. For each write, record the permission it needs and whether the action can be undone. Undoing matters most. A reserve figure can be changed tomorrow. A confirmed supplier booking or a released payment often cannot.

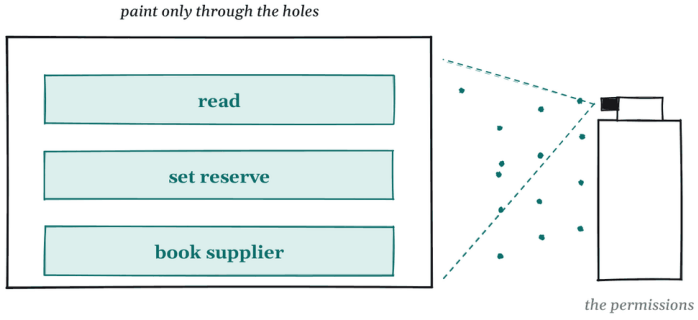
This list feeds the junctions node later. Every action that cannot be undone, or that costs a lot when wrong, is a candidate for a junction.



Like measure twice, cut once: *a pencil mark on the plank rubs out but the saw cut does not, so the cut earns the extra check.*

The node is signed when every write is listed with its system, its permission and whether it can be undone. The usual failure is granting broad permissions for convenience during a pilot and never narrowing them. A loop in production should be able to do exactly what its canvas says and nothing more.

In the claims loop, the executor reads from policy administration, the claims system and the document store. It writes to the claims system, where it sets the reserve and records the assessment, and to the supplier booking system. Reserve changes can be undone. A confirmed supplier booking cannot.



Like a stencil: *paint lands only where the shape is cut out, so the loop does exactly what its canvas says and nothing more.*

XLVII

Decisions

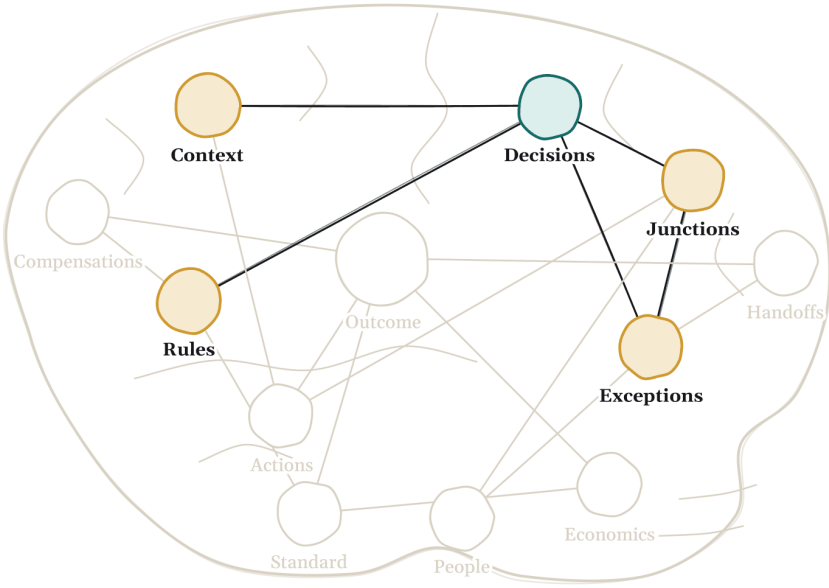
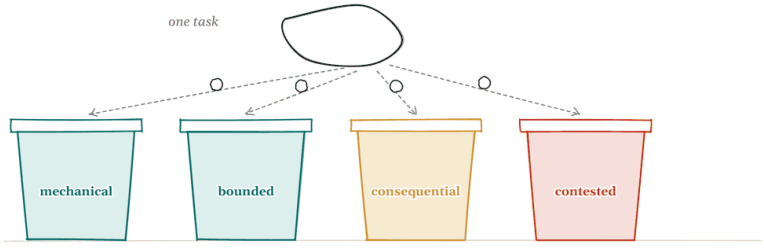


FIGURE 42 Decisions

Sort every decision: mechanical, bounded, consequential or contested.

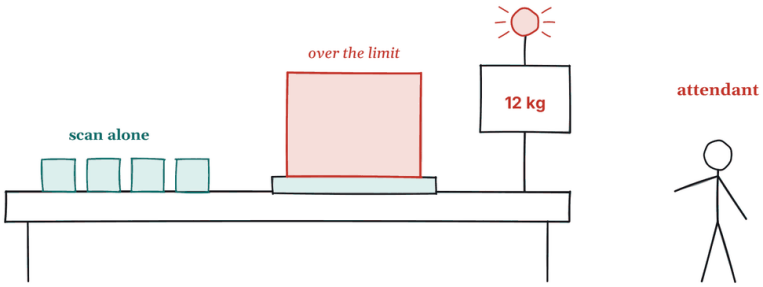
This is the node that most distinguishes the canvas from a process map. A process map shows tasks. This node lists the decisions inside them and sorts each one into four classes.



Like a recycling station with four bins: *each task is emptied out and every decision inside it goes into one of four classes.*

Mechanical decisions follow a rule with no judgment: is the policy in force, is this a duplicate, are the documents complete. They can run alone. **Bounded** decisions involve judgment within a written threshold: repair or replace below a set amount, a reserve within a range. They can run alone against the threshold, with cases outside it sent to a person. **Consequential** decisions carry enough weight that a person always makes them, whatever the evidence: declining a claim, referring someone for fraud. **Contested** decisions are the ones the organization never actually agreed on.

The classes tend to follow the distribution of cases. Mechanical and bounded decisions live in the body, where cases are many and similar. Consequential decisions cluster in the tail, where cases are rare, large and different from each other. A decision that looks bounded on the average case can turn consequential on a large one, so set thresholds by value as well as by type.

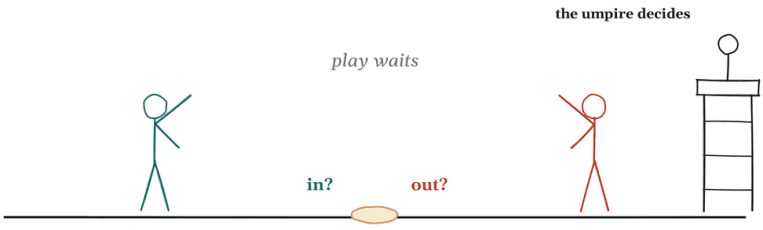


Like a self-checkout: *small items scan through on their own, but anything over the weight limit calls the attendant, whatever kind of item it is.*

Sort the decisions in the room, with the process owner, an experienced operator and the system owner present. Disagreement about which class a decision belongs to is useful information. Usually it means the decision is less settled than everyone assumed.

The node is signed when every decision is sorted and the contested ones are at zero. That last condition is strict on purpose. A contested decision cannot be engineered away. It has to be settled by someone with the authority to settle it, before the loop acts alone. It gets a chapter of its own later in this part.

In the claims loop, the sort found four mechanical decisions, four bounded, two consequential and one contested. The contested one was whether goodwill payments are allowed on borderline cover. It went to the claims committee. Until the committee decides, the node stays at Draft.



Like a disputed line call in tennis: *the players can argue, but only the umpire can settle it, and play waits until the call is made.*

XLVIII

Junctions

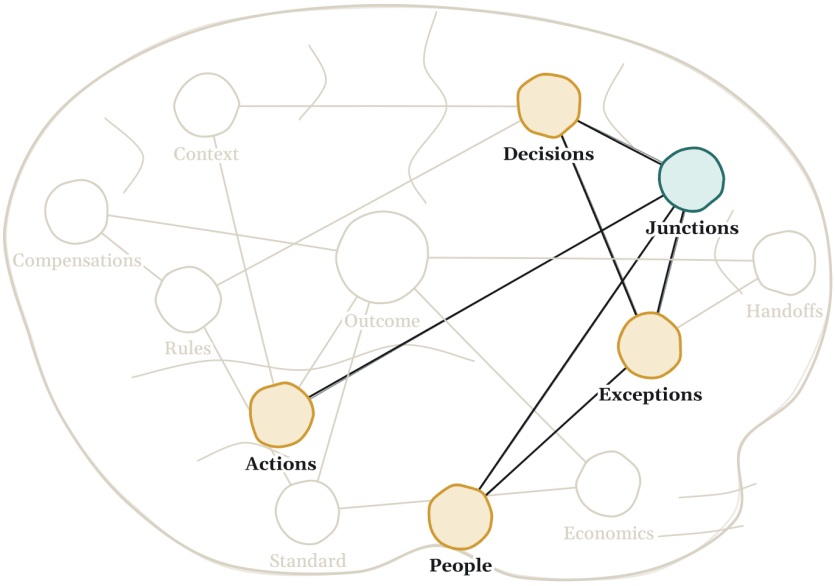
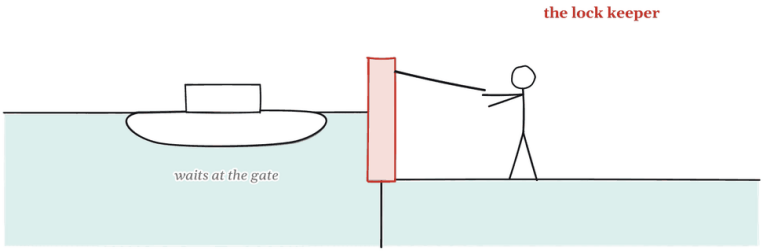


FIGURE 43 Junctions

Where does a person decide, what do they see, and how is an action reversed?

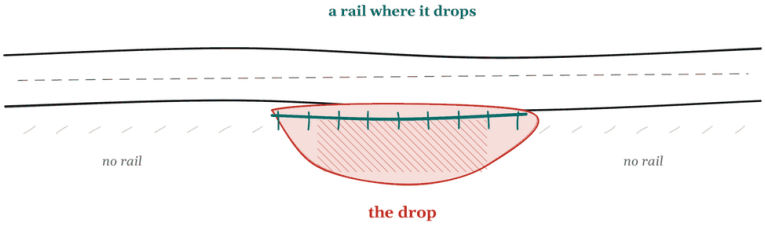
A junction is Toyota's stop cord rebuilt for an executor that decides. It is a defined point where the loop pauses and a named person decides. Every consequential decision gets one. So does every action that cannot be undone or costs a lot when wrong.



Like a canal lock: *the boat stops at a set point, the lock keeper looks and decides, and only then does the gate open.*

Place junctions by reversibility and the cost of an error, not by comfort. In a power-law domain, that usually means placing them in the tail, where a few cases carry most of the risk. Teams new to agents tend to put a person on every step, which recreates the second review the compensations node just removed. Teams that are overconfident put a person on none. The canvas asks for a reason for each junction, and the reason has to be about consequences.

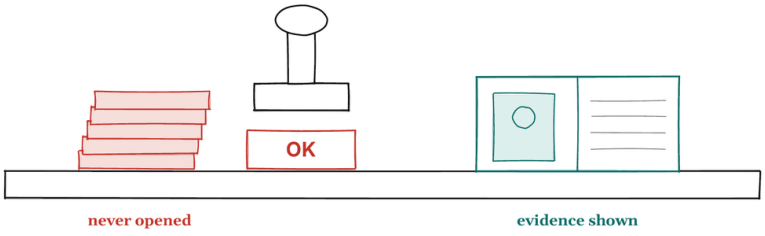
For each junction, write down three things: who decides, what evidence they see and how the action is reversed if it was wrong. Then name who can pause the whole loop. That is the andon cord at the level of the loop, and someone must hold it.



Like guard rails on a mountain road: *they go where the drop is steep, not along every straight and not nowhere, and each one has a reason.*

The node is signed when every irreversible or high-cost action passes a named junction. The usual failure is a junction with a name but no evidence. A person asked to approve an agent's decision without seeing why it was made becomes a rubber stamp within a week.

In the claims loop, three junctions were placed. Declining a claim goes to a senior adjuster, who sees the rule trace and evidence and can reopen it within thirty days. Any amount over fifteen thousand dollars goes to a team lead. A fraud referral goes to an investigations analyst. The claims operations lead can pause the loop.



Like a border officer stamping passports without opening them: *a junction that shows no evidence turns the named person into a rubber stamp.*

XLIX

Exceptions

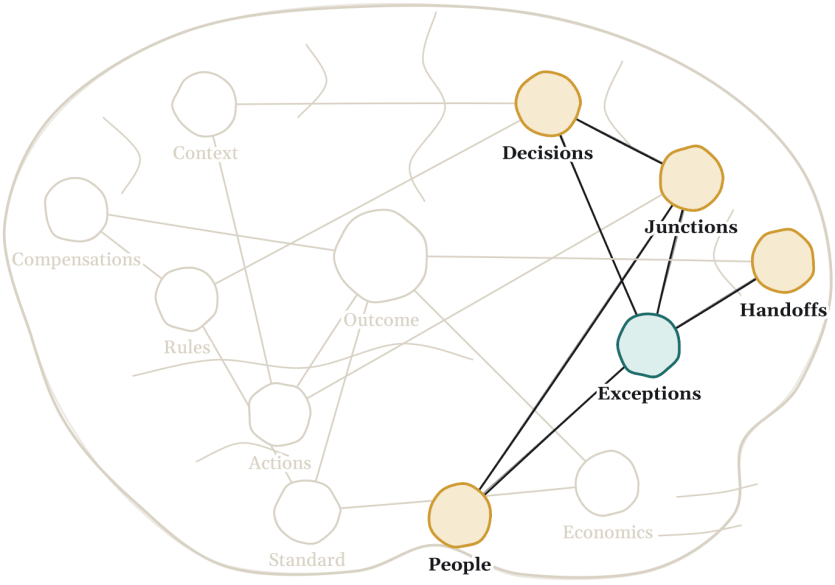
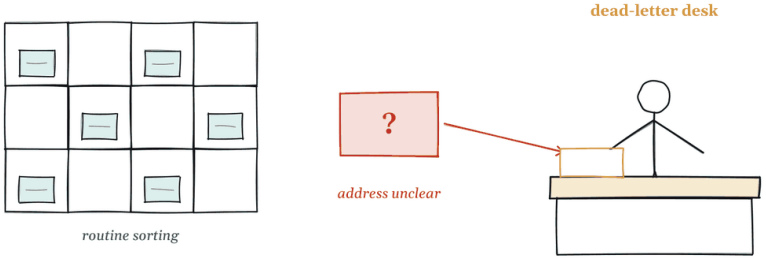


FIGURE 44 Exceptions

Where do the cases the loop cannot handle go, and who works that queue?

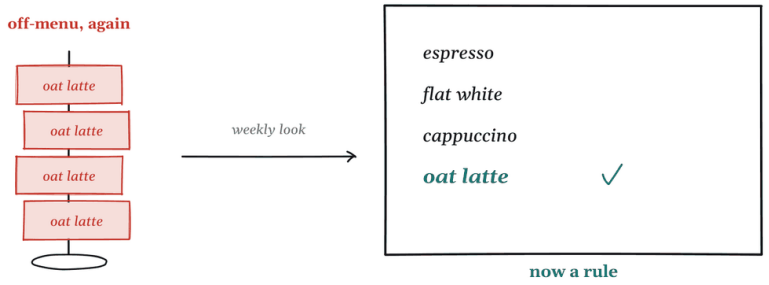
Every loop meets cases its rules do not cover: missing evidence, unusual circumstances, a policy written before a rule existed. The exceptions node gives those cases a defined place. Without one, they pile up in inboxes, or worse, the executor makes something up.



Like a post office's dead-letter desk: *a letter with an unreadable address goes to one named counter, instead of piling up or being dropped at a guessed house.*

Name the queue, its owner and a time limit. Decide what the customer sees while their case waits, because a long silence is a service failure even if the eventual decision is right. Then set a review rhythm. Once a week, someone looks at the exceptions and asks which ones repeat. A repeated exception is usually a missing rule. Writing that rule moves the case back into the routine.

The exception queue is also where shifts in the distribution show up first. New kinds of cases land here before anyone has a name for them. Watch the queue's mix as well as its size.

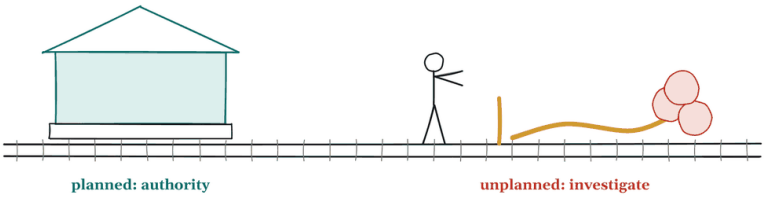


Like a café where the same off-menu order keeps coming in: *once someone notices it repeating, it gets written onto the menu and becomes routine.*

This node and the junctions node are related but different. A junction is a planned stop for a known kind of decision. An exception is an unplanned stop for a case the design did not foresee. Both need people, but the skills differ. Junctions need authority. Exceptions need investigation.

The node is signed when the queue has an owner, a time limit and a weekly review. The usual failure is an exception queue that becomes the old process under another name, with most cases ending up there. If more than a small share of cases are exceptions, the rules node is incomplete.

In the claims loop, cases outside the rules or missing evidence go to the adjuster queue. The assessment team lead owns it, with a limit of forty-eight hours. A weekly review turns repeated exceptions into rules.



Like a train's scheduled station stop beside an unscheduled stop for a fallen tree: *one needs someone with authority, the other someone who can investigate.*

L

Handoffs

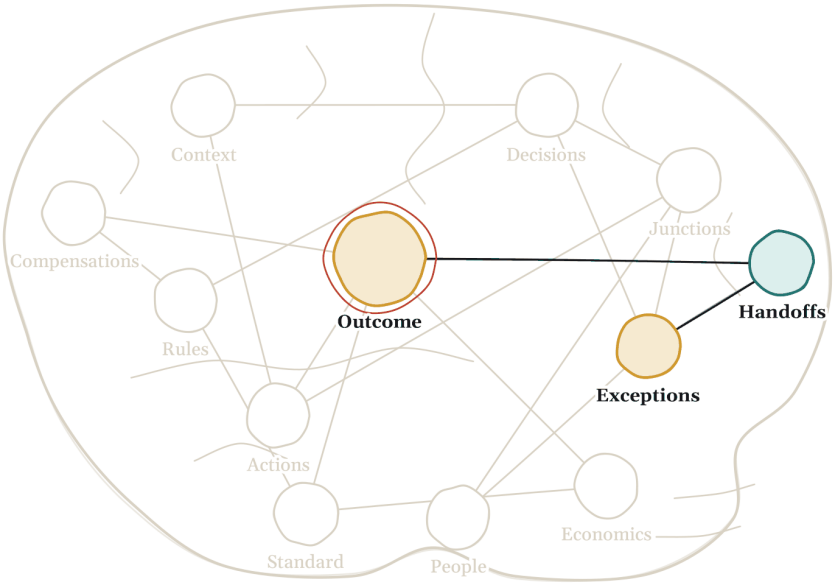
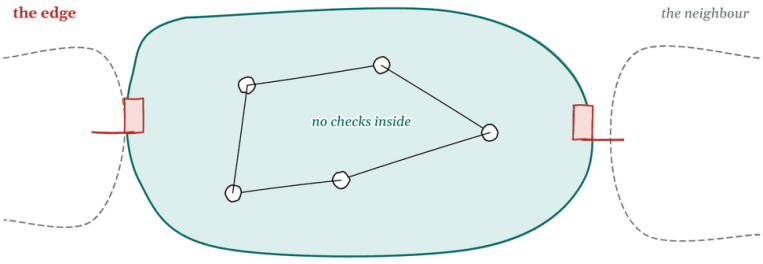


FIGURE 45 Handoffs

What does this loop receive, and what does it pass on?

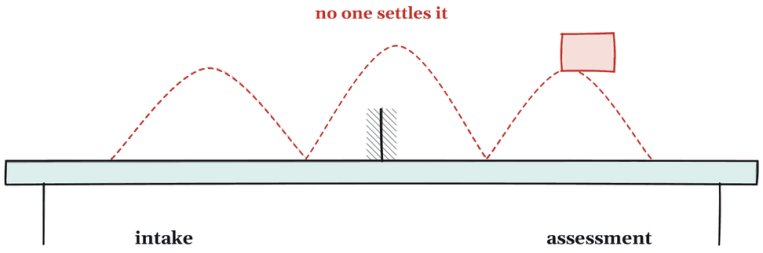
Ford made the handoff the unit of industrial work. In the work loop it moves to the edge. Inside the loop, a single owner and a single design remove most internal handoffs. What remains are the boundaries between this loop and its neighbours.



Like a free-travel zone: *inside it you drive from town to town with no checks, and the border posts sit only where it meets its neighbours.*

For each boundary, write down what the loop receives, from which loop, and in what state. Then write down what it passes on, to whom, and in what state. Finally, name who resolves a mismatch. Mismatches are where loops fail quietly. Intake sends a claim it believes is complete. Assessment finds a document missing. Without a named owner for the boundary, the claim bounces between teams.

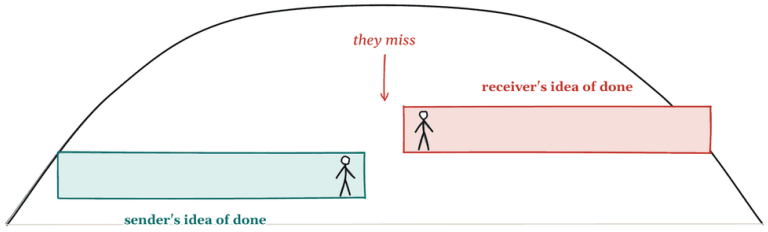
Handoffs are also where the Distribution model meets the loop. A handoff to a customer, a partner or a regulator carries more risk than a handoff to another internal loop, and deserves a stricter state definition.



Like a ball in table tennis: *with nobody named to settle a mismatch at the boundary, a claim just bounces back and forth between intake and assessment.*

The node is signed when every handoff names the state passed and who resolves a mismatch. The usual failure is defining handoffs from one side only. The sending loop's idea of complete is rarely the receiving loop's idea of complete. Both owners have to agree.

In the claims loop, assessment receives a validated claim file from intake. It passes an assessed claim with an amount and rationale to settlement, and suspected fraud to investigations. At the time of the worked run, nobody had been named to resolve mismatches with intake, so the node stayed at Draft.



Like a tunnel dug from both ends: *unless both crews agree where they will meet, the two halves miss each other in the middle.*

LI

Standard

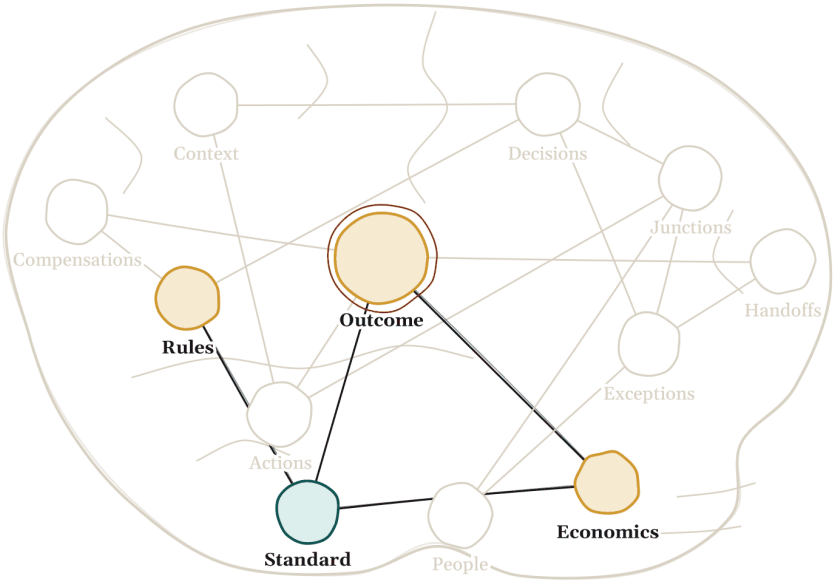
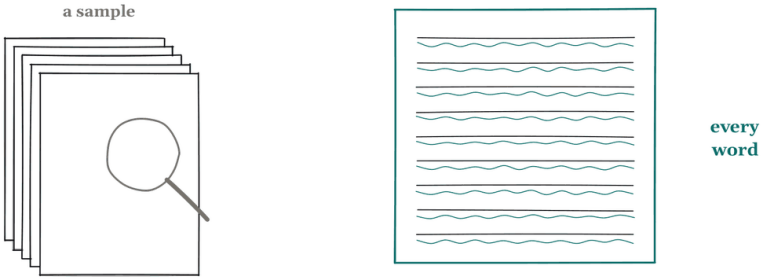


FIGURE 46 Standard

How is every outcome graded?

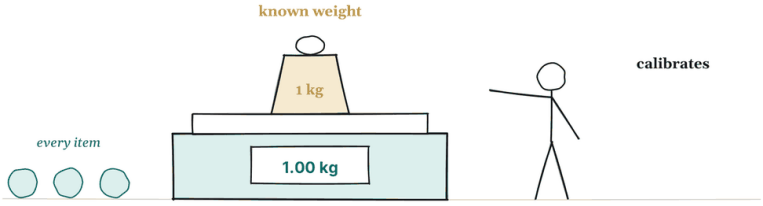
Taylor wrote the standard so people could follow it. The canvas writes it so every result can be checked. The difference is coverage. Traditional quality control checks a sample, because checking everything costs too much when people do the checking. With agents in the loop, every outcome can be graded, and it should be.



Like a spell checker beside a proofreader: *the proofreader samples a few pages, while the checker reads every word of every page.*

Write the acceptance criteria in plain language first. Then decide who or what grades each criterion: a rule, a model or a person. Rules suit criteria that can be checked mechanically, such as whether an amount is within policy limits. Models suit criteria that need reading, such as whether a rationale actually traces to the rules. People calibrate the graders, by auditing a small share of graded outcomes and correcting the graders when they drift.

Grade the tail more closely than the body. The audit share that calibrates the grader can be small for routine cases and much larger for the few cases that carry most of the value.

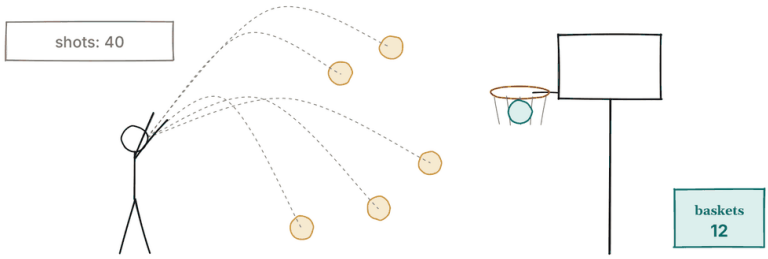


Like shop scales tested with a known weight: *the scales weigh every item, and a person checks them now and then so their readings stay true.*

The standard is also the loop′s early-warning system. A falling acceptance rate shows a problem before customers complain. It is the measure the economics node depends on, because cost per accepted outcome needs a definition of accepted.

The node is signed when every outcome is graded against a written standard. The usual failure is measuring activity instead of acceptance: claims processed per hour, tasks automated, agents deployed. None of those says whether the results were right.

In the claims loop, an assessment is accepted when the amount is within policy terms, the rationale traces to the rules and the customer letter is correct. A rules engine checks the terms. A model checks the rationale. A human audit of two percent of claims calibrates the model.



Like a basketball player counting shots taken: *the activity number climbs, but only baskets made, the accepted outcomes, show whether the work was right.*

LII

People

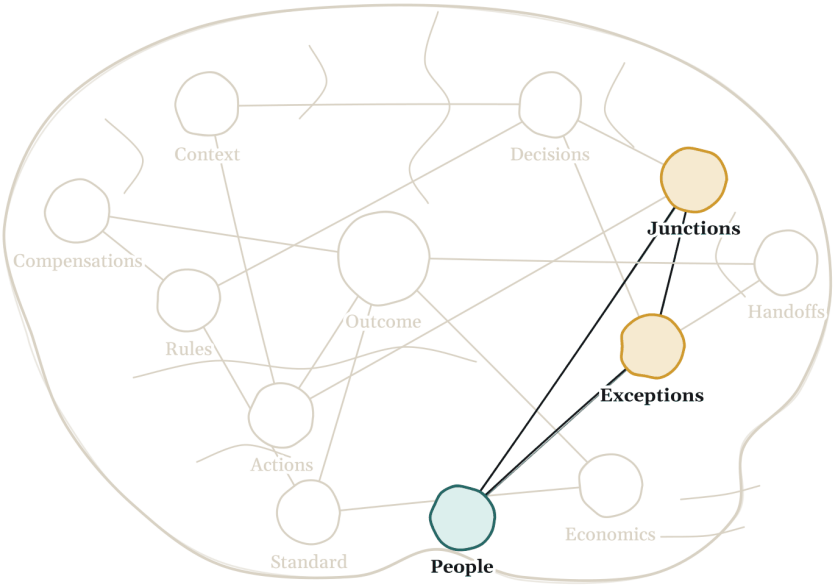
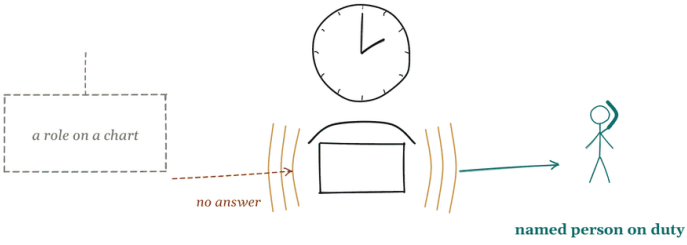


FIGURE 47 People

Who answers for the loop, and whose work changes?

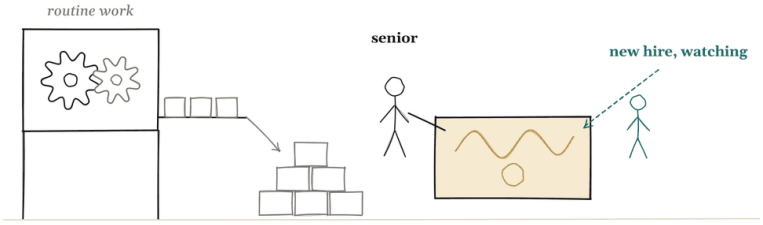
Sloan's lesson was that an outcome needs an owner. This node names one for the loop as a whole, and names the people who hold each junction. Names, not roles, because a role on a chart does not answer the phone at two in the morning.



Like a phone ringing at two in the morning: *a named person on duty picks it up, while a box on an org chart never answers.*

The node also asks a harder question: whose work changes, and how? In the claims loop, adjusters move from assessing every file to working junctions and exceptions. That is a real change in what their day looks like, and it should be said before launch, not discovered after it.

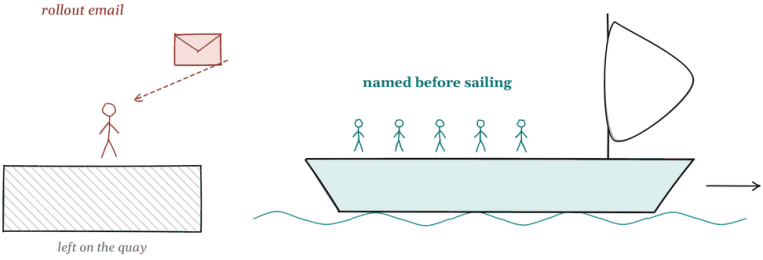
The last question is about learning. Experienced adjusters learned their judgment by assessing thousands of routine files. If agents now handle the routine files, new adjusters lose the work that used to train them. The loop needs a deliberate path for that, such as shadowing junction decisions or reviewing a sample of agent assessments with a senior colleague.



Like a stonemason's yard where a machine now cuts the plain blocks: *the apprentice must be set beside the master on the carved stones, or never learns.*

The node is signed when named people are in place before production. The usual failure is treating this node as a communications task to handle after the design is done. People who learn about a changed role from a rollout email rarely help make the loop work.

In the claims loop, the Head of Claims Operations owns the loop. Junction holders are named for all three junctions. How junior adjusters will learn was still open at the time of the worked run, so the node stayed at Draft.



Like a crew mustered on deck before the ship sails: *whoever is told by letter after departure is left on the quay and helps nobody sail.*

LIII

Economics

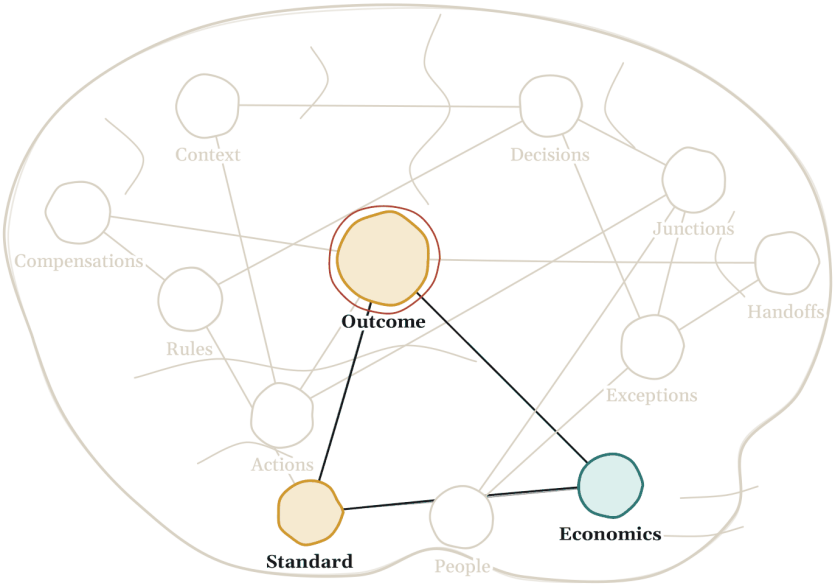
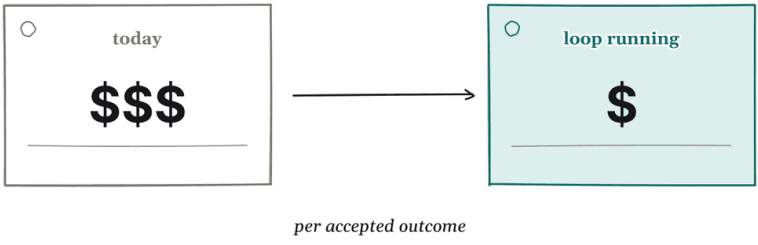


FIGURE 48 Economics

What does one accepted outcome cost, compared with the measured baseline?

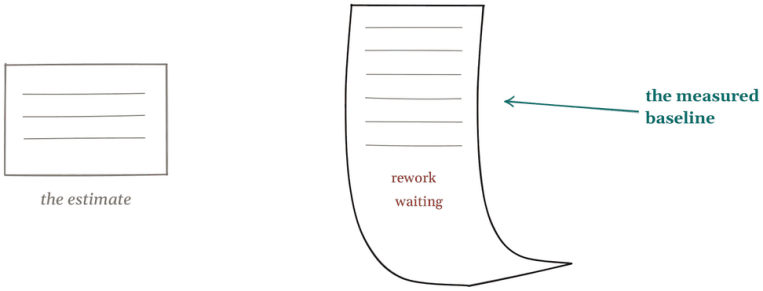
The last node turns the loop into numbers the Financial model can use. It compares the cost of one accepted outcome today with the cost once the loop runs, along with cycle time and the share of outcomes right the first time.



Like a unit-price label on a shelf: *compare the cost of one accepted outcome today with its cost once the loop runs, not the size of the bill.*

Measure the baseline before changing anything. Estimates do not count, because every estimate of today's cost is too low. It leaves out the rework, the waiting and the time experienced people spend answering questions. Then set a target for each measure.

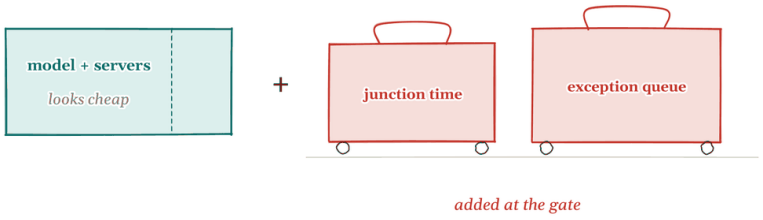
Measure by segment of the distribution, not only on average. A loop can lower the average cost per claim while raising losses on the large claims, and an average will hide that. Report the body and the tail separately.



Like a till receipt that runs far longer than the shopping list: *measure today's real cost, rework and waiting included, before trusting any estimate.*

When calculating the cost of the new loop, include everything. That means model and infrastructure costs, plus the time people spend at junctions and in the exception queue. A loop that looks cheap because its people costs were left out will look expensive the first time someone adds them back.

The node is signed when cost per accepted outcome is below the baseline, measured rather than estimated. Until the loop has run in shadow, this node can only be a draft. That is expected.

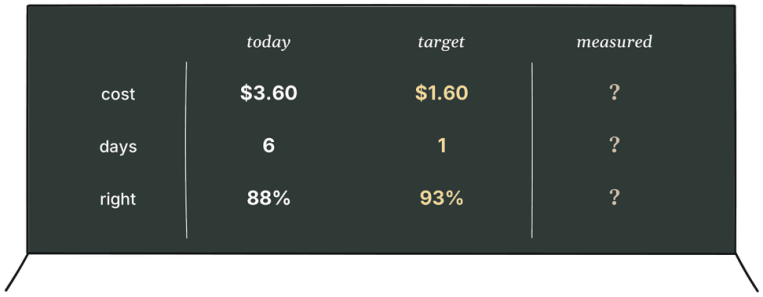


Like a budget airline fare: *the ticket looks cheap until the bags are added at the gate, so count the people time at junctions from the start.*

In the claims loop, the measured baseline was $3.60 handling cost per claim, six days in assessment and 88 percent assessed right first time. The targets were under $1.60, one day and 93 percent. The run cost was set to include

junction time and the exception queue, and had not yet been measured in shadow.

A single canvas describes one loop. A firm runs dozens. This part covers the decision that blocks most loops, the map that places all of them, the gates that pace their growth and the session that fills a canvas in an afternoon.



Like a training-ground chalkboard: *today's times are written up, the targets chalked beside them, and the trial column stays empty until the shadow run.*

LIV

The Contested Decision

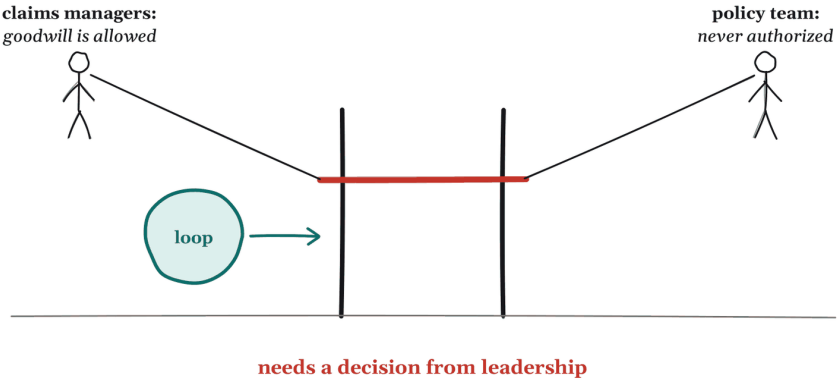
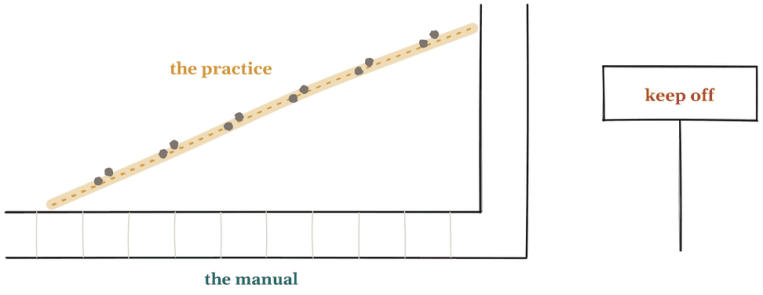


FIGURE 49 The Contested Decision

Sorting decisions in a room almost always turns up at least one that nobody can classify. The manual says one thing, the system does another, and two senior people give different answers. That is a contested decision.

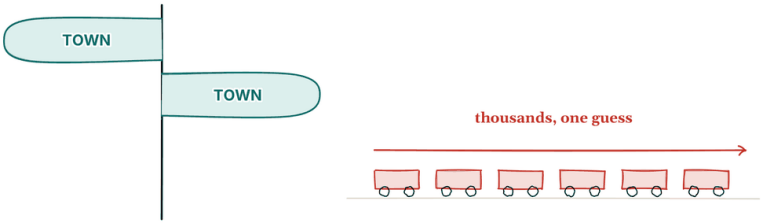
Contested decisions are not rare, and they are not signs of a badly run firm. They are what happens when a business grows faster than its rules. Two teams solve the same problem in different ways. A policy written for one market is applied in another. A manager makes a sensible exception, then another, until the exception is the practice. Experienced staff absorb the difference case by case, and nobody needs to settle it.



Like a footpath worn across a lawn beside the paved path: *a sensible shortcut, taken again and again, becomes the real route though no plan shows it.*

An agent cannot absorb it. Given a contested decision, it will either act on a guess or stop. Both look like a technology failure, though the cause is organizational. **No amount of engineering can resolve a contested decision.** It needs a choice from someone with the authority to make it, written down and owned.

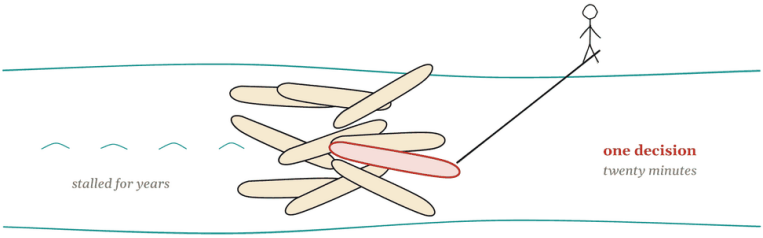
The canvas treats contested decisions as the first thing to resolve, not the last. They hold the decisions node at Draft, and that holds the loop below the Bounded gate. This is deliberate. A loop that acts alone while a contested decision is open will make that decision thousands of times, in whichever direction its guess falls.



Like a signpost pointing both ways to the same town: *a driver can ask, but a machine either stops or sends every car down whichever arm it guessed.*

Resolving one is usually quick once it is visible. The goodwill question in the claims loop took one committee meeting. The committee decided that goodwill payments are allowed up to a set amount, on claims within thirty days of the cover lapsing, with a team lead's approval. That decision created a new bounded rule and a new junction. It took twenty minutes. It had been open for years.

Many stalled AI programs have a few of these at their centre. The canvas makes them visible on the first afternoon, before anyone writes a line of code. For many firms that alone justifies the exercise.



Like a log jam on a river: *years of stalled flow, freed in twenty minutes once someone with a pole pulls out the one key log.*

LV

The Work Loop Map

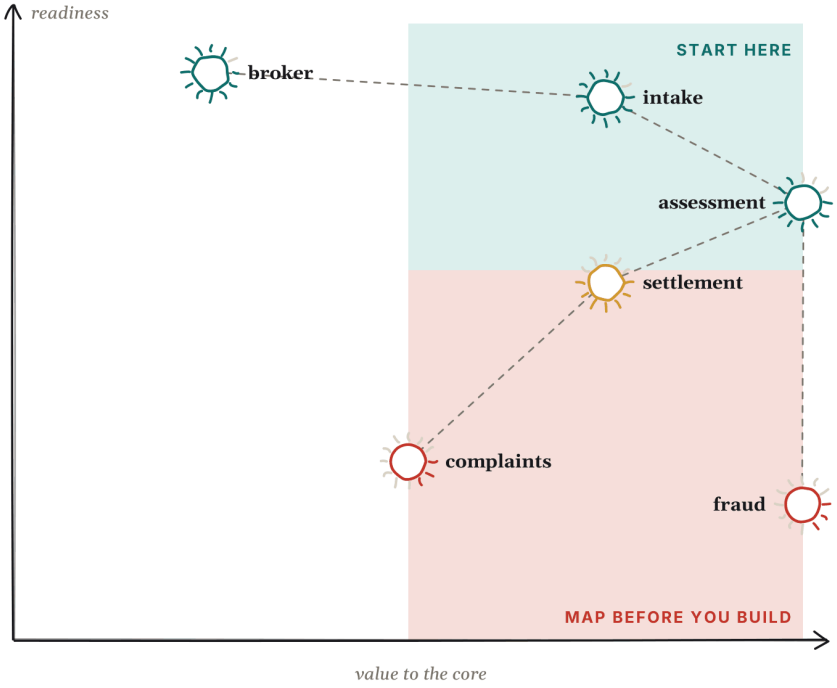
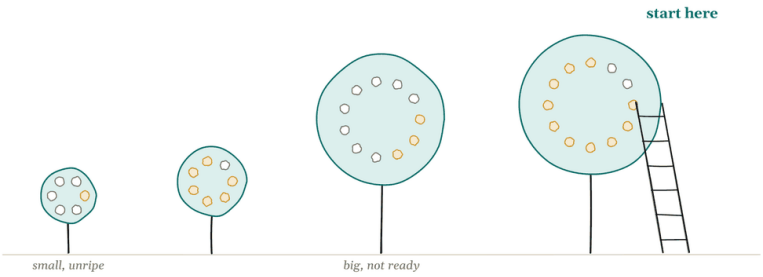


FIGURE 50 The Work Loop Map

A firm runs many loops, and they do not all deserve attention at once. The Work Loop Map places each loop on two axes.

The horizontal axis is value to the core of the business, scored from one to five. A loop scores high when it touches customers directly, carries significant cost or risk, or produces the firm's main product. It scores low when it supports other internal work. The vertical axis is readiness, read directly from how many of the loop's nodes are signed.

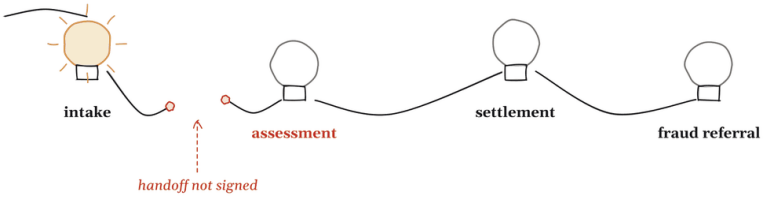


Like a fruit picker sizing up an orchard: *each tree scored by its crop and its ripeness, with the ladder going up the big ripe tree first.*

The map divides into four regions. Loops in the top right matter and are ready, so that is where to start. Loops in the bottom right matter but are not ready, so the work there is mapping, not building. Loops in the top left are ready but less important. They make good practice for a team learning the method. Loops in the bottom left can wait.

The map also draws the handoffs between loops. That turns a list of projects into a picture of the organization's wiring. It shows which loops depend on each other, and therefore which ones cannot reach the core alone. In the insurer's map, assessment depends on intake and feeds settlement and fraud referral. Assessment cannot reach the Core gate until its handoffs with intake are signed, however good the loop is on its own.

Seen this way, the map is a picture of the firm's connectome: its loops and the connections between them. It is also the most useful page to show a board. It answers three questions that most AI programs cannot: where the firm is starting, why there, and what has to be true before the next loop moves.



Like a string of lights wired one to the next: *the assessment bulb cannot glow, however good, until its wire from intake is connected.*

LVI

The Scale Gates

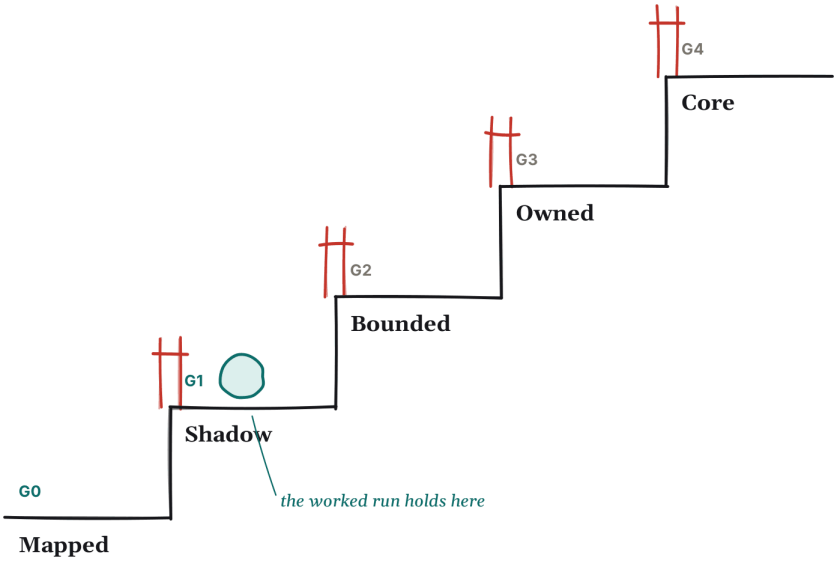
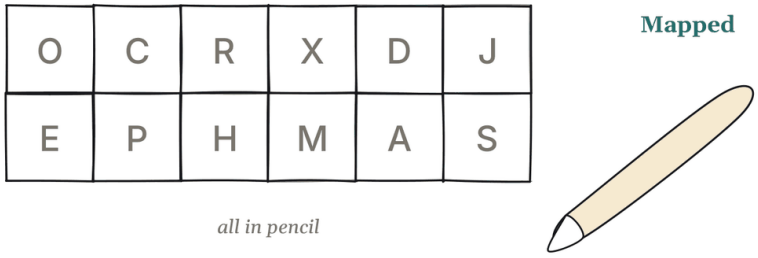


FIGURE 51 The Scale Gates

Readiness should not be a feeling. The Work Loop Canvas ties it to five gates, and each gate opens only when specific nodes are signed.

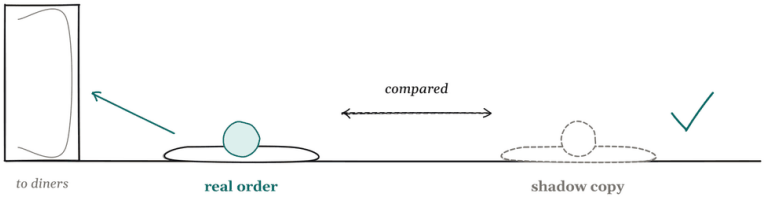
The first gate is **Mapped**. All twelve nodes have at least a draft. The loop is described but not yet agreed.



Like a crossword filled in pencil: *every one of the twelve squares has a guess, so the loop is Mapped, but none is inked and agreed yet.*

The second is **Shadow**. The loop runs beside today's process without writing to any system. Every outcome is graded against the standard, and the results are compared with what people decided. Shadow requires the outcome, compensations, rules, context and standard to be signed. It is the cheapest place to learn, because nothing the loop does reaches a customer.

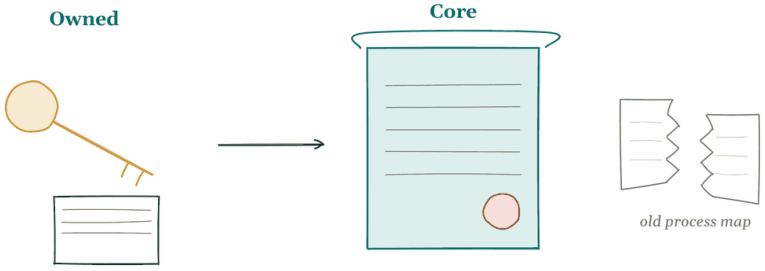
The third is **Bounded**. The loop acts alone on mechanical and bounded decisions, with junctions and the exception queue live. It adds signed actions, decisions, junctions and exceptions. This is the gate a contested decision blocks.



Like a trainee cook plating a copy of every order that never leaves the kitchen: *compared with the chef's plate, it costs nothing if it goes wrong.*

The fourth is **Owned**. The loop runs in production with a named owner, at a cost per accepted outcome below the measured baseline. It adds signed people and economics.

The fifth is **Core**. The loop writes to systems of record and hands off to neighbouring loops. The old process map is retired, and the canvas becomes the description of how the work is done. Core requires all twelve nodes, including handoffs.

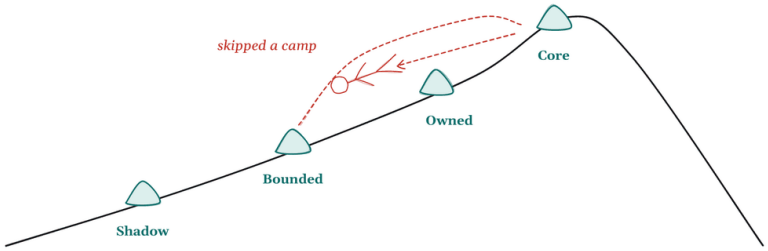


Like a tenant who becomes the owner: *first the keys and the bills in their own name, then the deeds rewritten and the old lease torn up.*

A loop moves up a gate only when the nodes that gate depends on are signed. Scale follows signatures, not confidence. A pilot that jumps a gate

has borrowed against the core. It usually repays the loan in production, as an incident that sets the whole program back.

The gates also give a program a shared language for progress. "The assessment loop is at Shadow, held by one contested decision" is a status report a board member can understand, and it says exactly what would move it.



Like a climber who moves to the next camp only after the last one:
skipping a camp borrows against the summit and is paid back as a fall.

LVII

Running the Session

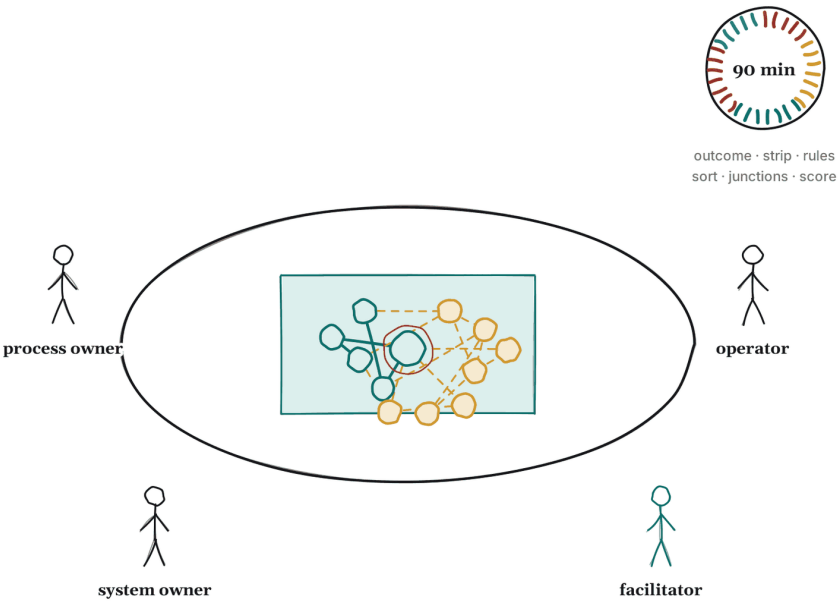
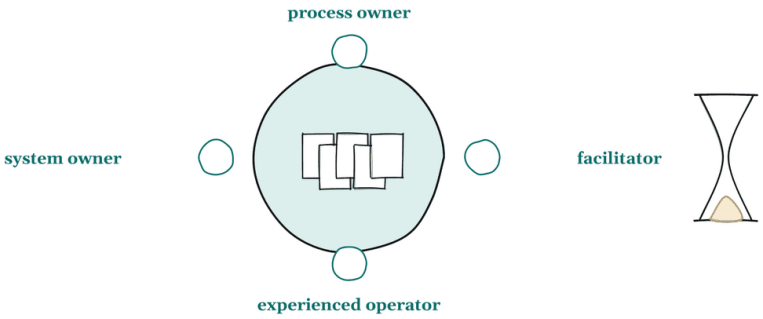


FIGURE 52 Running the Session

The Business Model Canvas spread because it came with a ritual. People knew how to run a session with it. The Work Loop Canvas needs the same.

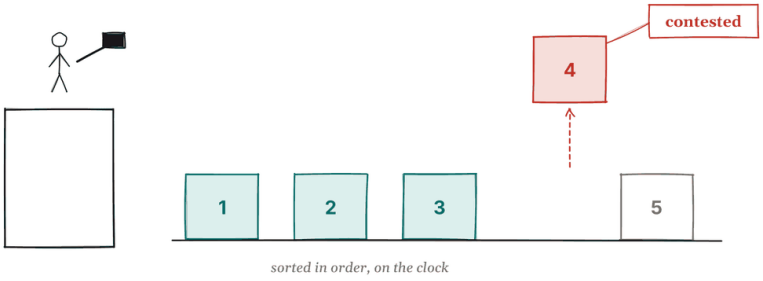
A first session takes about ninety minutes and needs four people. The process owner answers for the outcome and makes calls. An experienced operator, someone who does the work every day, knows the unwritten rules and the real exceptions. The system owner knows what the systems actually do and what the configuration says. A facilitator keeps time and holds the order of the nodes.



Like a game of bridge: *always four at the table, each seat with its own job, and a ritual everyone knows for playing the hand.*

Fill the nodes in number order. Spend the first fifteen minutes on the outcome and do not move on until it is agreed. Spend the next twenty on compensations, walking the current process step by step. Rules and context take another twenty, and the list of rules will not be finished in the room. Capture what people know and assign owners to complete it.

The decision sort is the heart of the session. Give it twenty minutes and expect disagreement. When people disagree about a decision′s class, write it down as contested and move on. Do not try to resolve it in the room unless the person with authority is present.

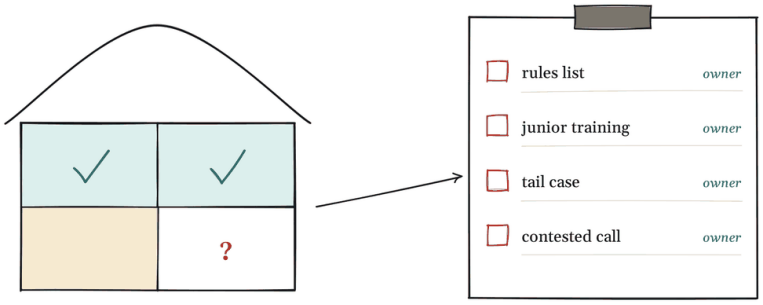


Like an auction run lot by lot to the clock: *a lot nobody agrees on is tagged and set aside for someone with authority, and the sale moves on.*

Make sure one of the cases on the table comes from the tail of the distribution: a large, unusual or expensive case. Routine cases show the steps. The tail case shows where judgment lives.

Use the last fifteen minutes for junctions, people and economics, at draft level. Close by scoring every node as Blank, Draft or Signed, and reading out the gate the loop has reached and what holds it there.

A first session rarely signs more than half the nodes. That is fine. Its real output is the list of what is missing, with an owner beside each item. Most teams reach Shadow within two or three weeks of the first session, once the rules list is complete and the contested decisions have gone to someone who can settle them.



Like a snag list after walking a new house: *half the rooms pass, and the real output is the list of what is missing, each with an owner.*

LVIII

What the Canvas Leaves Out

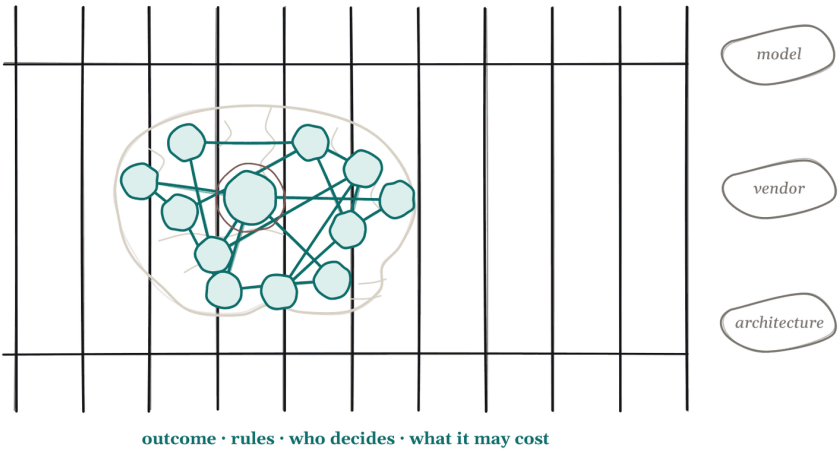
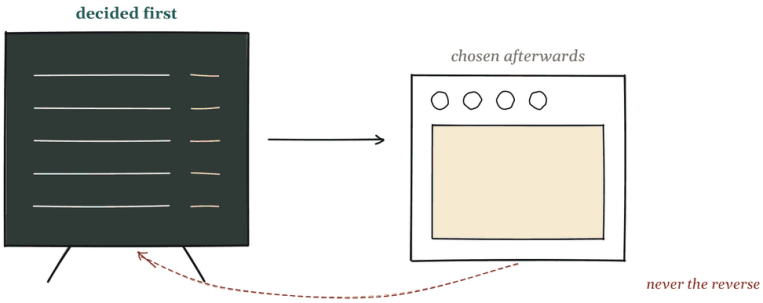


FIGURE 53 What the Canvas Leaves Out

The Work Loop Canvas deliberately says nothing about which model to use, which vendor to buy from or how to architect the system. Those choices matter, and they come afterwards.

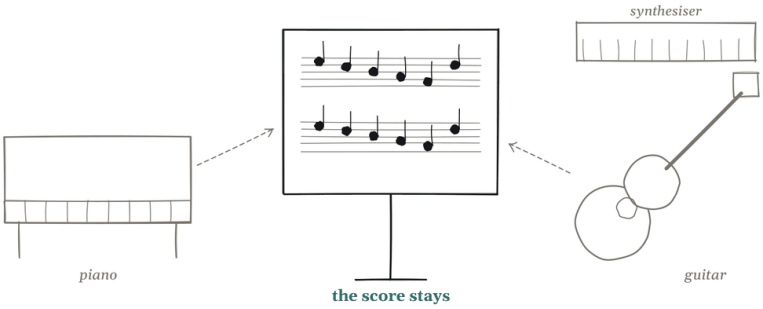
The canvas records the decisions only the business can make: what counts as a good result, which rules apply, who decides what and what it may cost. Implementation follows from those answers. A team that starts from the technology tends to make the business decisions implicitly, inside configuration files, where nobody signs them. That is the same mistake enterprise software made in the 1990s, repeated with a more capable executor.



Like writing the menu before buying the oven: *the business picks the dishes and prices first, or the oven ends up deciding the menu.*

Leaving the technology out also keeps the canvas durable. Models, vendors and architectures will change many times over the life of a loop. The outcome, the rules, the decisions and the people change much more slowly. A canvas written in business terms survives a change of model. A design written in technical terms has to be redone.

The test for a finished canvas is simple. **Someone who cannot name a single AI model should be able to read it and sign it.**



Like a page of sheet music: *the piano, the guitar and the synthesiser come and go, while the score, written in plain notes, stays the same.*

PART NINE

The Embedding Lifecycle

LIX

The Middle

Most companies can now show you an AI pilot that works. A model reads supplier invoices and proposes the right cost centre. An agent drafts replies to customer emails, and the drafts are good. A demo runs on a laptop in a meeting room, and the room is impressed.

Far fewer companies can show you the same work running inside the business a year later. By that we mean deciding real cases, writing to the systems of record, and producing results that someone has signed for.

The gap between those two pictures is the subject of this part. We will call it the middle.

The numbers on the middle are consistent, even when the surveys disagree on almost everything else. In 2025, an MIT study of enterprise AI programmes reported that only about 5% of custom enterprise tools reached production. McKinsey's survey that year found that 88% of organisations used AI in at least one function. Only about a third had begun to scale it, and no more than 10% were scaling agents in any single function. Gartner predicted that over 40% of agentic AI projects would be cancelled by the end of 2027, citing costs, unclear value and weak risk controls.

These figures are not a verdict on the technology. A pilot is a narrow test. It shows that a model can do a task on a prepared set of cases. It does not show who answers when the model is wrong, which rules it follows when the policy and the system disagree, what it costs per accepted result, or how anyone would switch it off.

Those questions are not technical. They are questions about the business. They also have to be answered in a particular order. You cannot decide who signs off a large invoice before you know which decisions the work contains. You cannot measure savings before you have measured the baseline. And you cannot measure the baseline before you know what the work really is.

Most stalled programmes did not fail at the model. They failed in the middle, because nobody owned the sequence that turns a working demo into work the business relies on.

This part describes that sequence. It is a lifecycle of ten steps, grouped into four phases. Each step answers one question, produces one or more

documents that a named person signs, and ends at a gate that says whether the work may move on. The steps run in order because each depends on the answers of the one before.

There is a deeper reason the order matters. An embedded loop combines two kinds of machinery that have been studied for a long time. One learns from examples: the model, flexible and strong on routine cases, but without guarantees. The other is written down: rules, decisions, limits and standards, precise and auditable, but never complete. The model can be bought. The written half cannot, because it describes this company's own work. The lifecycle is the sequence in which a company builds that written half, tests it against real cases, and closes the loop so that results feed back into the design.

The lifecycle is built for one particular kind of change: putting an executor that makes decisions inside a company's core processes. That is different from giving staff a writing assistant, and different from installing software that follows a script. When the executor decides, the design of the work has to say what it may decide, where a person decides instead, and how anyone would know the decisions were good. The lifecycle exists to get those answers written down and signed before the executor acts alone.

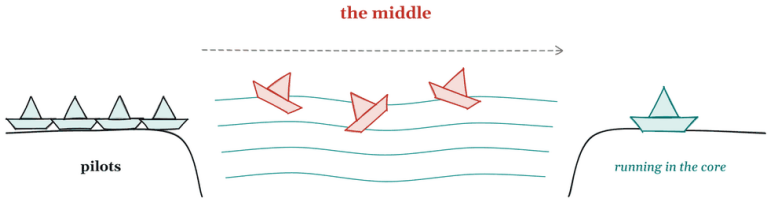
The lifecycle is for the people who carry that responsibility. That includes the executive who sponsors the change and the owner of the process being changed. It also includes the finance, risk and operations people who must agree to it, and the practitioners who run the work. No technical background is assumed. Where a term was defined earlier, it is used in the same sense.

The composite company from the body of knowledge runs through this part and the next. It is a mid-size European maker of industrial pumps and valves, with about 4,200 people and three plants. We follow one of its processes, the approval of supplier invoices, from the first reading of where value sits to the day it runs in the core. The company and its figures are illustrative. They are set within published ranges, which the notes identify, so that the arithmetic teaches something real.

The first chapters lay the ground: what embedding means, why an executor that decides changes the design of work, and what that executor is made of. They also look at what earlier lifecycles got right, explain why the lifecycle has the shape it has, and list what must be in place before the first step. The ten steps follow in order, each closing with what the foundations add. The last

chapters cover how lifecycles break, how to adapt them, how a company runs many at once, and what finishing is worth.

Read it in order the first time. After that, it is a reference you can open at whatever step your own work has reached.



Like a wide river: *many pilots launch from the near bank, few cross the middle, and only those that land do real work for the business.*

LX

What Embedding Means

Companies have used software for sixty years, and AI for more than ten. So it is fair to ask what is new about embedding it now.

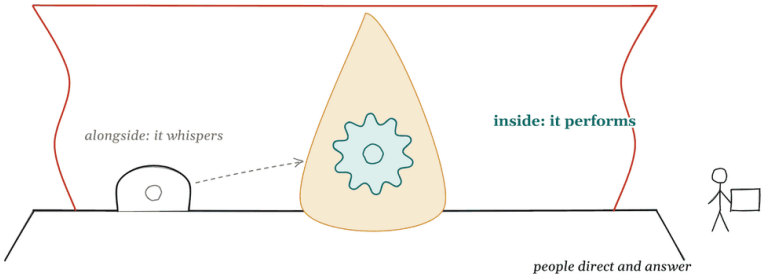
The answer starts with a distinction between two places a technology can sit. It can sit alongside the work, helping the people who do it. Or it can sit inside the work, doing part of it.

A spell-checker sits alongside the work. So does a search tool, a dashboard, or a chat assistant that drafts an email a person then edits and sends. In each case a person remains the executor. The person reads the result, decides what to do with it, and acts. If the tool is wrong, the person catches it or does not, and either way the person answered for the outcome.

Embedding means putting the technology inside the work. The executor reads the invoice, checks it against the order, decides whether it can be approved, and posts it to the ledger. Nobody retypes or approves each result. People still design the work, set its limits, decide the hard cases and answer for the outcome. They no longer touch every case.

The difference matters because the alongside position carries almost no organisational risk. If a writing assistant drafts a poor paragraph, a person deletes it. Nothing in the company's records changes. That is why assistants spread quickly: they ask nothing of the company's processes, its controls, or its accountability.

Embedding asks for all three. An executor inside the work writes to systems of record. It commits the company to suppliers, customers and regulators. Its mistakes become the company's mistakes, recorded in the ledger. So before it may act, the company has to know what it is allowed to do, under which rules, with which evidence, and who answers for it.



Like moving from the prompter's box to the stage: *alongside, the tool whispers lines to a person; inside, the executor plays the part while people direct it.*

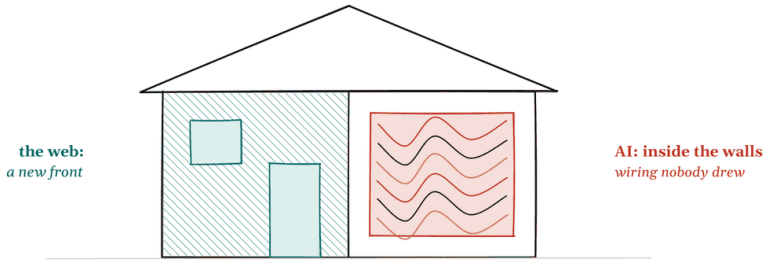
Outside in, inside out

A useful way to see why this is hard is to compare AI with the previous great wave of technology, the web.

The web worked from the outside in. It changed how companies reached customers first: the storefront, the catalogue, the marketing, the support pages. The back office changed later and more slowly, and often it did not change much at all. A company could put a modern website on top of thirty-year-old processes, and many did.

AI works from the inside out. Its value sits in the work itself, in reading documents, applying rules, making decisions and producing results. To deliver that value it has to enter the processes, and the processes are where companies are least documented. Rules live in policies, in system settings and in the heads of experienced staff, and the three rarely agree. Steps exist for reasons nobody remembers. Decisions that look routine turn out never to have been agreed.

This is why a pilot can succeed and the programme still stall. The pilot tested the executor on a clean slice of work. Embedding exposes the executor to the real process, with all its undocumented rules and buried disagreements. The lifecycle in this book is mostly a way of surfacing those things, in order, and getting someone to decide them.



Like renovating an old house: *the web repainted the front, but AI has to go inside the walls, where the old wiring was never drawn on any plan.*

What embedding is not

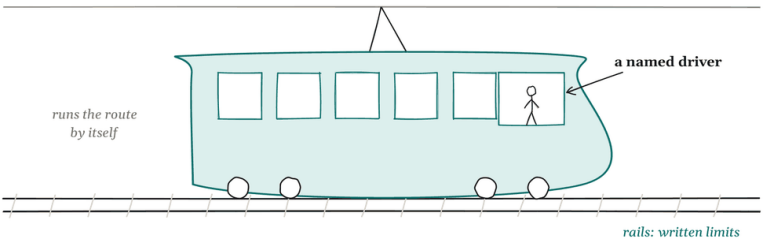
Three things are often confused with embedding, and it helps to set them aside.

Embedding is not deploying a tool. Licences can be bought and switched on in a week. A tool that staff may use is still alongside the work.

Embedding is not automating a process map. Most process maps record how work was done under old constraints. Attaching an executor to each box speeds up the steps but keeps every workaround, and it usually leaves the waiting between steps exactly where it was.

Embedding is not removing people. A well-embedded process has fewer people touching routine cases and more attention on the cases that matter. Roles change, and some disappear. But a process with no named person answering for it is not embedded. It is abandoned.

What embedding is, in one sentence: **an executor that decides runs part of a core process, inside limits the business has written down, with named people answering for the result.** The rest of this part is about getting there.



Like a tram: *it runs the route alone on rails laid in advance, with a named driver who can stop it; without one, it is a runaway.*

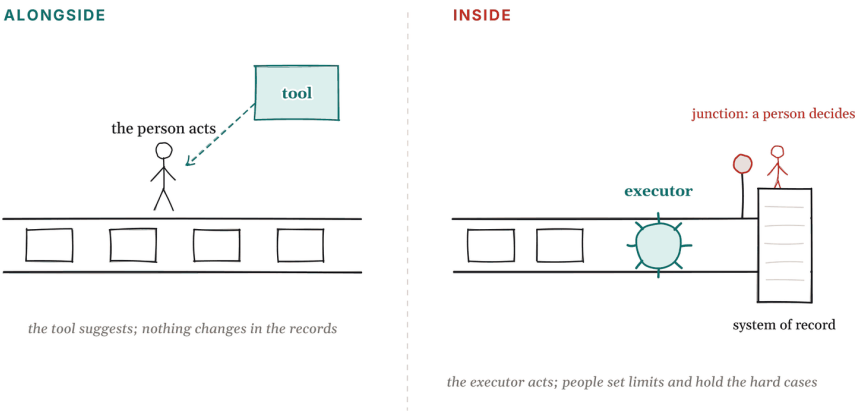


FIGURE 54 Alongside and inside. On the left, the tool helps and the person acts. On the right, the executor acts inside the work, people set its limits and hold the hard cases, and results land in the system of record.

LXI

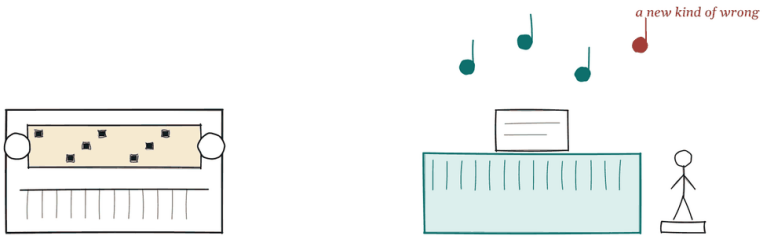
The Executor That Decides

For a century, companies designed work around executors that followed instructions. That assumption shaped every method they used, and it is the assumption that has changed.

Consider the executors a process could rely on before. A clerk followed a procedure manual. A machine on a line repeated a motion. An enterprise system applied the rules configured in its screens. A software robot clicked through a sequence of fields. Each of these needed the work to be specified in advance, step by step. Where the specification ran out, a person took over.

So process design meant writing instructions. The process map was a set of instructions for people and machines. Control meant checking that the instructions had been followed. Exceptions were whatever the instructions did not cover, and they went to a person.

A language model with access to tools is a different kind of executor. Given an outcome and some context, it can work out the steps itself. It can read a document it has never seen and decide what kind of document it is. It can apply a rule written in plain language to a case nobody anticipated. It can be wrong in ways no instruction-follower could be wrong, because it was never following an instruction in the first place.



Like a player piano beside a jazz pianist: *one plays exactly the holes in its roll; the other is given the tune, finds the notes, and can miss.*

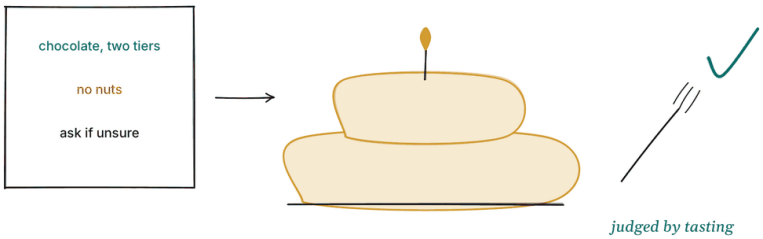
Three consequences

This changes the design of work in three ways, and each shows up later as a step in the lifecycle.

First, the unit of design moves from the task to the outcome. If the executor can find its own steps, describing steps becomes less important than describing the result, the rules and the limits. A task list tells an instruction-follower what to do. A decider needs to know what counts as done, what it may not do, and when to stop and ask.

Second, decisions become the thing to design. With instruction-followers, decisions were buried inside steps, and a person made them without anyone listing them. With a decider, every decision must be sorted. Some it can take alone. Some it can take within a written threshold. Some must always go to a person. And some, it turns out, the company has never actually agreed on.

Third, control moves from checking steps to grading outcomes. You cannot audit a decider by confirming it followed the procedure, because it did not have one. You audit it by grading what it produced, every result, against a written standard.



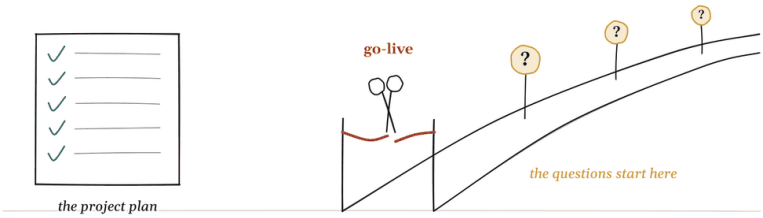
Like ordering a cake from a baker: *you describe the cake, say what it must not contain, and judge it by tasting, not by watching each step.*

Why the old methods stall

Most companies approach an AI programme with the methods they already have. These are a project plan, a process map, a requirements document and a testing phase. Each was built for instruction-followers, and each misses something a decider needs.

The process map records steps, but not which steps are workarounds. The requirements document lists functions, but not the decisions inside them or who owns each one. Testing checks that the system does what it was told, but a decider was not told; it was given an outcome. And the project plan ends at go-live, which is precisely where the questions about a decider begin. Who watches it? How is drift caught? Who may switch it off?

None of this means the old methods are useless. Several of their best ideas reappear in the lifecycle. It means they are incomplete for this executor, and the gaps sit exactly in the middle, between a pilot that works and a process the business relies on.



Like a ribbon cutting: *the project plan ends when the ribbon is cut, which is exactly when the questions of watching, correcting and stopping it begin.*

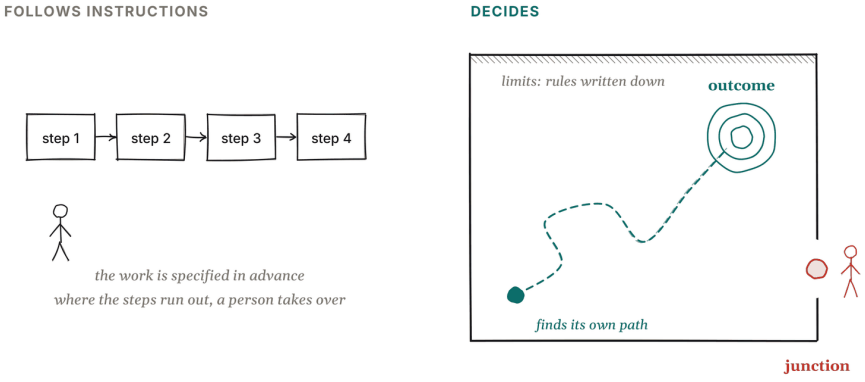
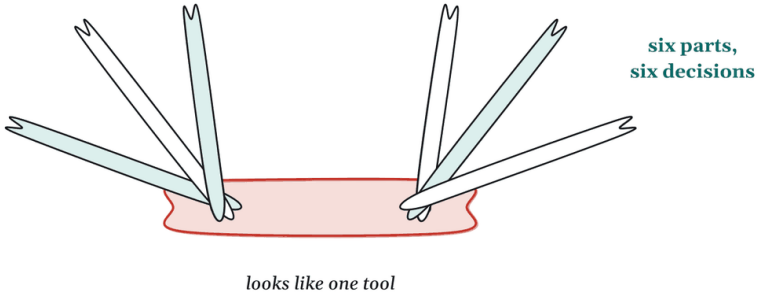


FIGURE 55 Two executors. An instruction-follower walks the written steps. An executor that decides is given an outcome, rules and limits, finds its own path, and stops at a junction where a person decides.

LXII

What the Executor Is Made Of

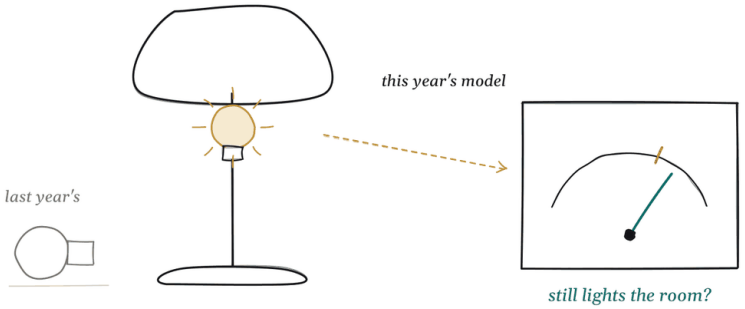
The executor that decides is often described as "the model", as if one piece of software did everything. In practice it is an assembly of six parts. A business reader does not need to build any of them. But it helps to know what they are, because each one maps to a decision the lifecycle asks the business to make.



Like a pocket knife: *it looks like one tool, but open it and there are six parts, each asking the owner to decide what it may do.*

The model

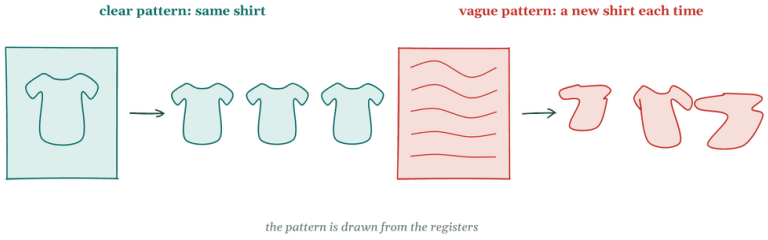
At the centre is a language model: a system trained on very large amounts of text that can read, reason about and write language. It is what lets the executor understand an invoice it has never seen, or apply a rule written in plain words. Companies rarely build their own. They buy access from a provider or run an openly available model on their own infrastructure. Models improve quickly and are replaced often. That is one reason the lifecycle grades outcomes rather than trusting a particular model: when the model changes, the grading shows whether the loop still meets its standard.



Like a bulb in a lamp: *you buy it, swap it often, and trust a light meter, not the box, to tell you the room is still lit.*

The instructions

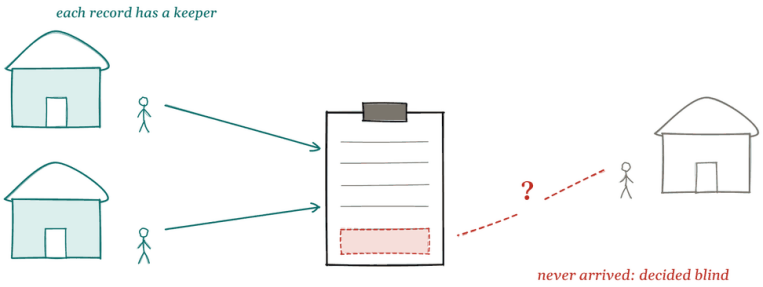
The model is told what its job is: the outcome, the rules, the limits, and what to do when a case falls outside them. In a well-designed loop, these instructions come straight from the documents the lifecycle produces. The rule register, the decision register and the junction register become the executor's brief. A loop whose rules were never gathered can only give its executor vague instructions, and vague instructions produce inconsistent decisions.



Like a sewing pattern: *a clear one, traced from agreed measurements, gives the same shirt every time; a vague one gives a different shirt from every cut.*

The context

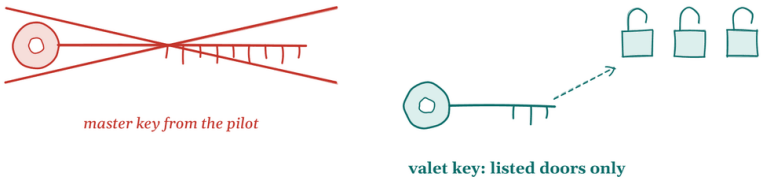
To act well, the executor must read the information the case needs: the order, the receipt, the contract, the customer's history. That information lives in the company's systems. The executor reaches it through connections that someone has to build, authorise and keep working. The context node of the canvas lists each source and its keeper. A source the executor cannot reach is a decision it will make blind.



Like a doctor's patient file: *each record is fetched from the clinic that keeps it, and a record that never arrives means a diagnosis made blind.*

The actions

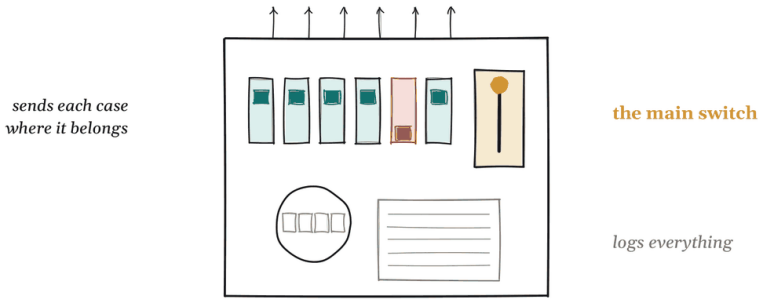
The executor changes things through tools: posting an invoice, updating a record, sending a message. Each tool is a permission granted in a particular system. This is where the business has the most control and the most exposure. The actions node lists each one with a single question: can it be undone? Permissions should match that list exactly. Permissions granted broadly for a pilot are the commonest source of executors doing things nobody intended.



Like a valet key instead of the master key: *it opens only the doors on the list, because broad keys handed out for a trial cause surprises.*

The platform

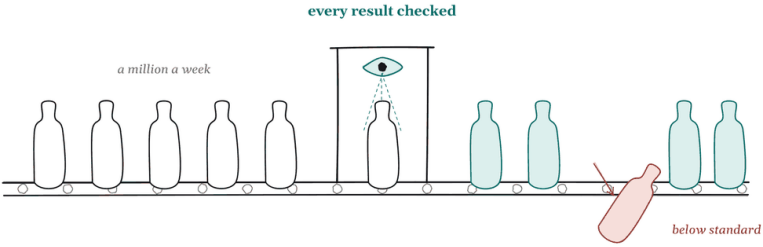
Around these parts sits the platform that runs the loop. It routes each case, calls the model, applies the limits, sends consequential cases to the junctions and exceptions to the queue, and records everything that happened. It also holds the switch that stops the loop. Some companies build this layer and many buy it. Either way, the business should insist on three things from it: a complete log of every case, a way for the named person to switch the loop off, and access for the grader to every result.



Like a building's fuse box: *it routes power to every room, trips at a limit, keeps the meter running, and has one main switch someone can pull.*

The grader

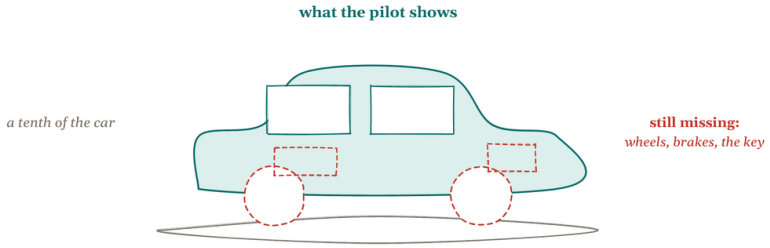
The last part watches the others. It checks every outcome against the standard, using rules, a second model trained to judge, and a small human audit. The grader is how anyone knows the loop is working. It is also the part most often left out of pilots, because a pilot can be judged by eye on a few dozen cases and a running loop cannot.



Like the inspection gate on a bottling line: *every bottle passes the sensor and is checked against the standard, because nobody can eyeball a million bottles.*

Where the pilot fits

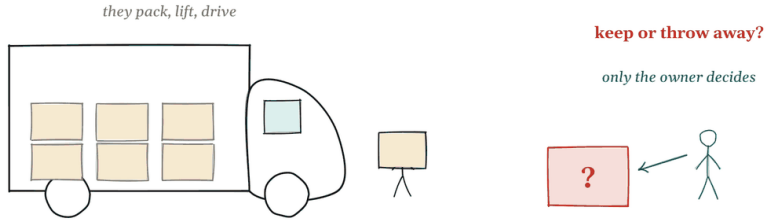
A pilot usually builds the first two parts well. It shows a capable model following good instructions on a prepared set of cases. Sometimes it connects a little context. It rarely touches real actions, almost never includes the platform's switch or a full log, and seldom has a grader. Measured by effort, a pilot is often about a tenth of what a loop running in the core requires. That is not a criticism of pilots. It is a description of the middle. Most of the remaining work is the business deciding what the other parts should allow, and the lifecycle is the order in which those decisions are made.



Like a car body on a showroom stand: *it gleams, yet the wheels, brakes and the key that switches it off are most of the work.*

Build or buy

Every part can be built, bought or rented, and the choice changes over time. The lifecycle does not depend on it. Whoever supplies the executor, the business still has to sign the rules, sort the decisions, place the junctions, write the standard and measure the economics. A vendor can run the platform. It cannot decide which disputed invoices the company is willing to pay.



Like hiring a removal firm: *they can pack, lift and drive, but only the owner can decide which boxes are kept and which are thrown away.*

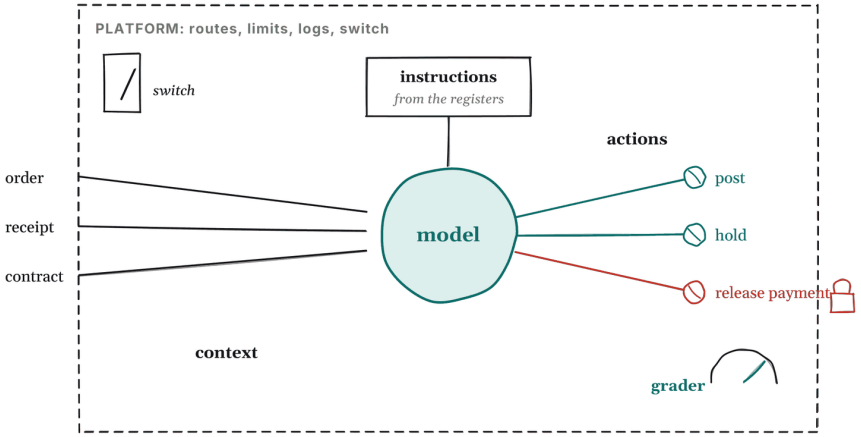
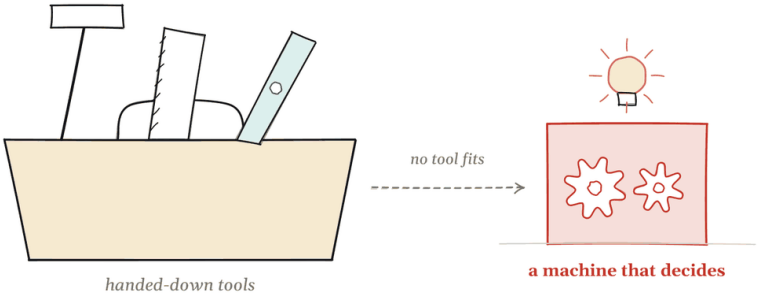


FIGURE 56 What the executor is made of. A model at the centre, briefed by instructions drawn from the signed registers, reading context through connections, acting through permissioned tools, run inside a platform that logs every case and can be switched off, with a grader on every outcome.

LXIII

Lifecycles Before This One

The idea of a lifecycle is old. Organisations have long broken large changes into stages, with a review between each. Several of those lifecycles shaped how companies work today. Each solved a real problem, and each carries something this lifecycle keeps. None was built for an executor that decides.

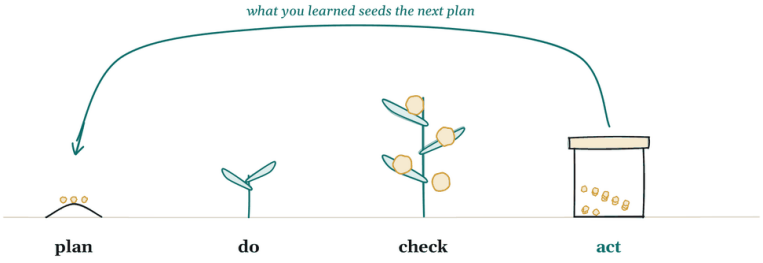


Like a toolbox handed down for generations: *every old tool still earns its place, but none was made for the machine now on the bench.*

The plan-do-check-act cycle

In the 1930s, the statistician Walter Shewhart described production as a cycle of specifying, producing and inspecting. W. Edwards Deming carried the idea to Japanese industry in the 1950s, and it became known as plan, do, check, act. Its lasting contribution is the loop itself: work is never finished, and what you learn from checking feeds the next plan.

This lifecycle keeps that shape. Its last step returns to its first.

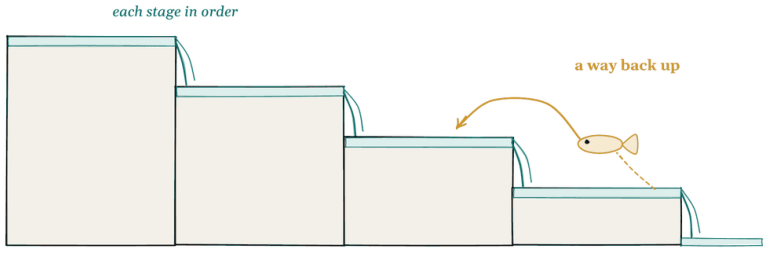


Like a gardener who saves seeds: *what this year's harvest teaches, and its best seeds, go straight back into next spring's planting.*

The waterfall

In 1970, Winston Royce published a paper on managing large software projects. It is remembered for a diagram of stages cascading from requirements to operations, each completed before the next began. The diagram became the "waterfall" and dominated large projects for decades. Royce himself warned that the pure sequence was risky and recommended building a first version to learn from. That warning was mostly ignored.

The waterfall's lasting idea is that some things must be settled before others. Its failure was treating every stage as complete before the next began, with no way back. This lifecycle keeps the order but expects to loop back when a later step uncovers something an earlier one missed.

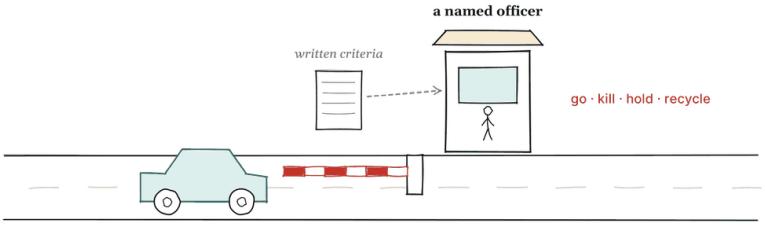


Like a waterfall with a fish ladder beside it: *the water still falls step by step, but a fish can climb back when a lower pool finds something missed.*

Stage-gate reviews

In 1990, Robert Cooper described the stage-gate process for new products. Work moves through stages, and between each stage sits a gate where named reviewers decide go, kill, hold or recycle against written criteria. The gates protected companies from spending heavily on products that should have been stopped early.

This lifecycle takes its gates from that idea. Each step ends with a gate, and each gate has a named signer and written criteria.

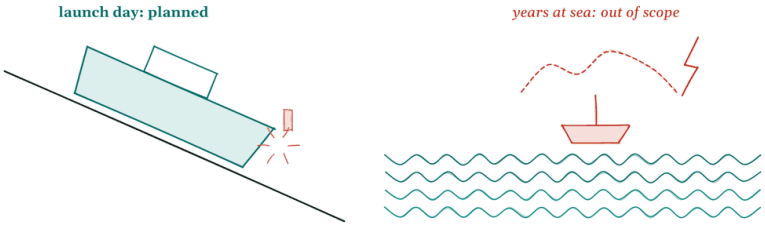


Like a border post: *at each crossing a named officer checks the papers against written rules, then lifts the barrier, holds the car, or sends it back.*

The data-mining cycle

In 1999, a consortium of companies published CRISP-DM, the cross-industry standard process for data mining. It ran from business understanding through data understanding, preparation, modelling and evaluation to deployment, with arrows looping back between phases. It was the first widely used lifecycle for building models, and it put business understanding first.

Its blind spot was the end. Deployment was a single phase, often a report. What happened when a model ran inside a process for years, with its inputs drifting and its owners changing, was mostly out of scope.



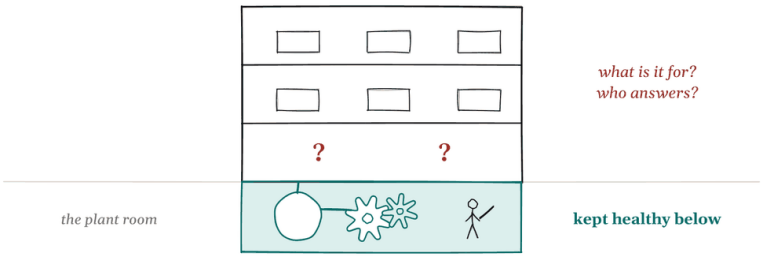
Like a ship launch: *the design and the build are planned with care, the bottle breaks, and the years at sea afterwards are nobody's plan.*

Service management and MLOps

Two later traditions addressed running systems over time. Service management, codified in the British government's ITIL guidance from 1989, organised how IT services are designed, handed over, operated and improved. Machine-learning operations, which grew after a 2015 paper on the hidden technical debt in machine-learning systems, applied similar discipline to models: monitoring, retraining, versioning.

Both are essential, and both sit mostly below the business. They tell you how to keep a system healthy. They do not tell you which decisions it should be

allowed to make, who answers for them, or whether the process it serves was worth automating in the first place.

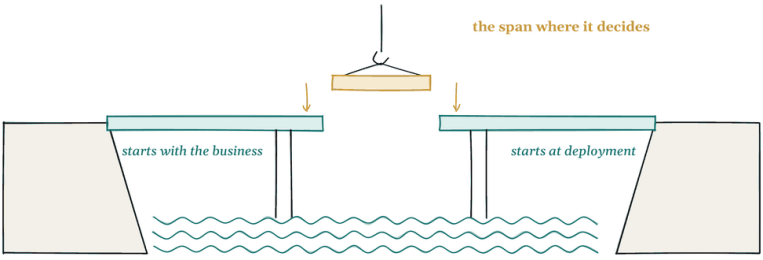


Like a building's plant room: *engineers keep the boiler and pumps healthy below, but what the floors above are for is decided somewhere else.*

What was missing

Put these lifecycles side by side and a gap appears. Some begin with the business but end at deployment. Others begin at deployment but assume the business questions were settled. None was designed for an executor that decides inside a core process, where the business questions and the operating questions cannot be separated.

This lifecycle tries to close that gap. It keeps the loop from plan-do-check-act and the order from the waterfall. It keeps the gates from stage-gate reviews and puts business understanding first, as CRISP-DM did. And it keeps the long view of service management. What it adds is a sequence of business decisions, each signed, that has to be completed before an executor that decides may act alone.



Like a bridge begun from both banks: *one crew starts from the business, one from deployment, and the middle span where the executor decides was never laid.*

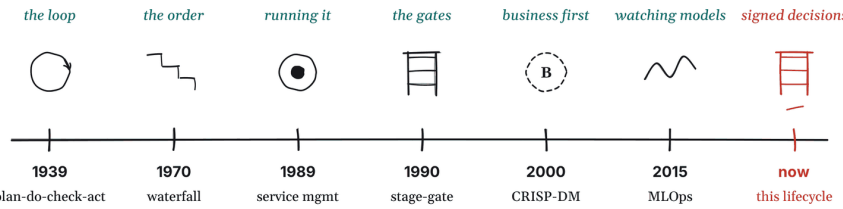


FIGURE 57 Lifecycles before this one. Each earlier lifecycle left one idea this lifecycle keeps; none was built for an executor that decides.

LXIV

The Lifecycle at a Glance

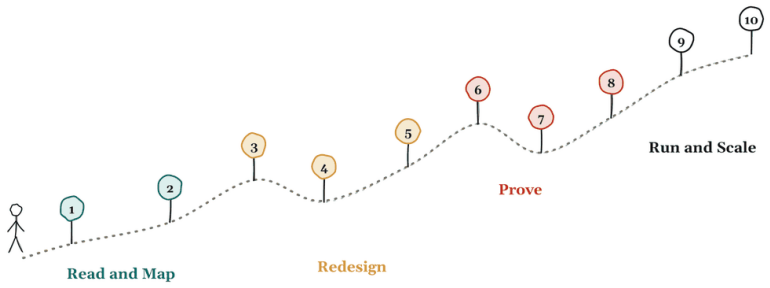
The lifecycle has ten steps in four phases. Before taking them one by one, it helps to see the whole.

The first phase, Read and Map, establishes what is true today. Step 1, Read, looks across the company's loops and asks where the value sits and where to start. Step 2, Map, takes one loop and records how its work really runs, with a measured baseline of time, cost and quality.

The second phase, Redesign, decides what the work should become. Step 3, Strip, removes the steps that exist only because of constraints that no longer apply. Step 4, Sort, gathers the rules and sorts every decision into its class, and it puts the contested ones in front of someone who can settle them. Step 5, Design, draws the loop: the junctions, the exception queue, the handoffs and the standard it will be graded against.

The third phase, Prove, earns trust in stages. Step 6, Shadow, runs the loop beside the current process without touching anything, and compares results. Step 7, Bound, lets it act alone on the routine body of cases, inside written limits. Step 8, Own, puts named people in place and measures the economics against the baseline.

The fourth phase, Run and Scale, makes the loop part of the business and moves on. Step 9, Core, runs it in production with drift checks, a monthly review and a tested way to switch it off. Step 10, Scale, reviews the whole map again, decides which loop comes next, and plans how people's roles change. Its output is the input to the next Read.



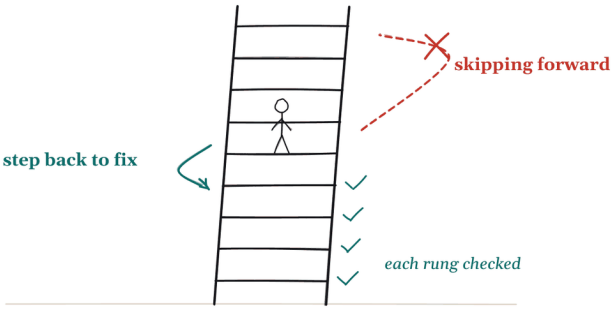
Like a trail map at the trailhead: *ten waymarks in four stretches, seen whole before the walker takes the first step.*

Three rules that hold across all ten steps

Each step answers one question, and the answer is written down. A step that ends with a discussion but no document has not ended.

Each step ends at a gate with a named signer. The signer is the person whose authority the answer needs. That is usually the process or loop owner, sometimes a control owner, finance or the executive sponsor. The gate either opens or it does not.

The steps run in order, but the lifecycle can loop back. If shadow running reveals a rule nobody recorded, the work returns to Sort for that rule, and the rule register is signed again. Looping back is normal. Skipping forward is the failure.



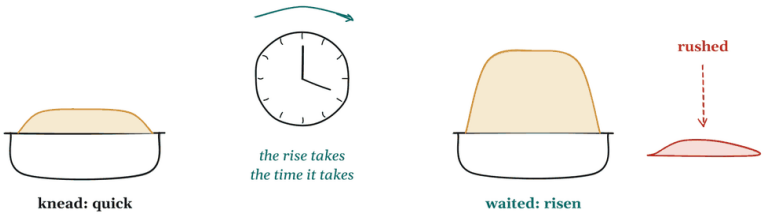
Like climbing a ladder: *each rung is checked as you go, stepping back down to fix one is normal, and leaping rungs is how people fall.*

How long it takes

For a single loop of moderate size, the first six steps usually take between four and six months. Read takes about two weeks, Map three, Strip one, Sort three, Design two, and Shadow six. Bound and Own add two to three months, mostly measurement. A loop typically reaches Core between seven and ten months after Read begins.

That can sound slow next to a pilot built in three weeks. But the pilot answered one question, whether a model can do the task. The lifecycle answers the forty or so questions a business needs answered before it relies on the work. Most of the elapsed time is waiting for evidence: six weeks of shadow running, eight weeks of measured economics. That time cannot be compressed without guessing, and guessing is what stalls programmes in the middle.

The second loop is faster than the first. The map already exists, the people know the steps, and the templates have been used once. By the third or fourth loop, a company is usually running several lifecycles at once, each at a different step. A later chapter describes how.

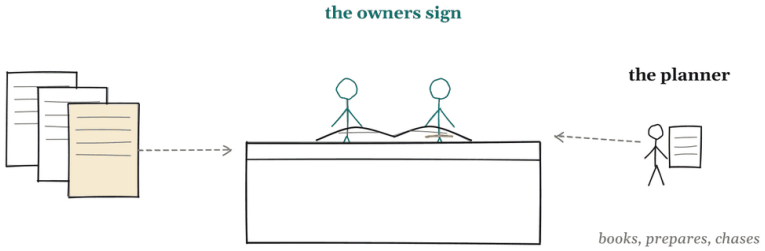


Like bread dough proving: *kneading is quick, but the rise takes the hours it takes, and a loaf rushed into the oven collapses.*

Who runs it

Every lifecycle needs a person who carries it from step to step: who books the sessions, prepares the documents, chases the signatures and knows what each gate needs. In this book that person is called the business engineer. The title matters less than the job. It can be held by someone in finance, operations, transformation or IT. It must be held by someone who understands the business well enough to ask the right questions, and the technology well enough to know what the executor can and cannot do.

The business engineer does not sign the gates. Owners and control holders sign. The business engineer makes sure there is something worth signing.



Like a wedding planner: *books the room, prepares the papers and chases everyone, but never signs the register; the people who own the promise do.*

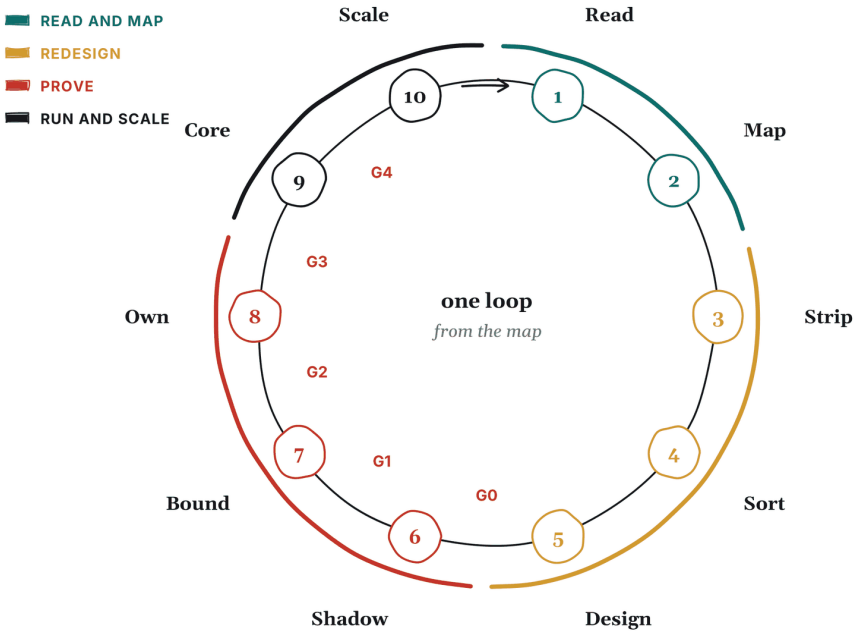
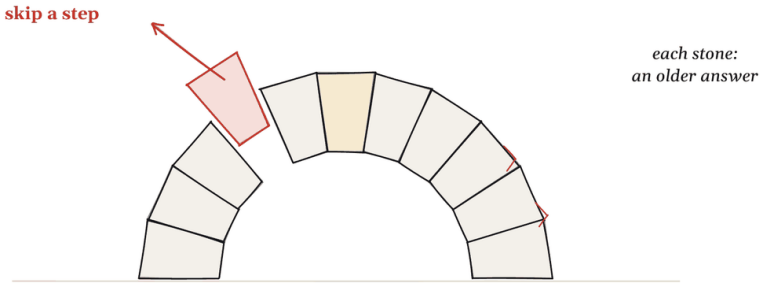


FIGURE 58 The lifecycle at a glance. Ten steps in four phases, with the Scale Gates along the way. Step 10 feeds the next Read, so the lifecycle closes into a loop.

LXV

Why the Lifecycle Has This Shape

The ten steps are not an arbitrary checklist. Each one answers a question that older fields of knowledge have already worked on. Knowing those answers makes the steps easier to run, and easier to defend when someone wants to skip one.

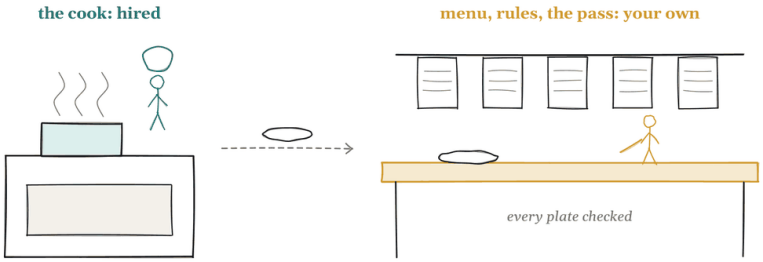


Like a stone arch: *every stone rests on an answer older fields already proved, and pulling one out to save time brings the arch down.*

Two kinds of machinery

For most of the history of computing, there were two ideas of how to make a machine reason. One wrote the reasoning down as explicit rules, in the tradition of George Boole's laws of thought. Rules can be read, checked and audited, but they break at the edges, where the world is richer than anyone wrote down. The other let machines learn from examples. Learned models handle variety well, but they come with no guarantees, and they are strong and weak in ways nobody can see in advance.

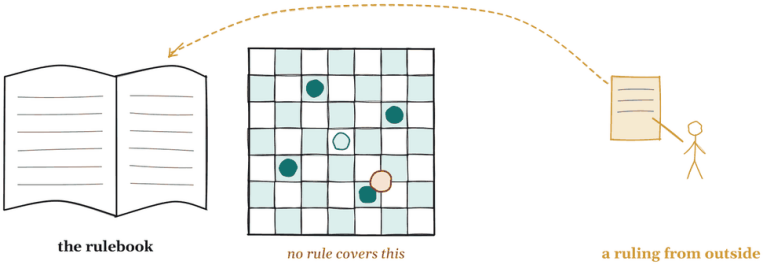
Researchers now combine the two, and call the result neurosymbolic AI. An embedded loop is such a combination. The model is the learned half, and a company can buy it. The written half is the rule register, the decision register, the limits, the junctions and the standard. A company cannot buy that half, because it describes its own work. Steps 2 to 5 build it. Steps 6 to 9 test it and keep it true.



Like a restaurant kitchen: *you can hire a skilled cook, but the menu, the house rules and the pass where each plate is checked must be your own.*

Rules never cover everything

In 1931, the logician Kurt Gödel showed that any rule system powerful enough to be useful leaves some truths it cannot reach from inside. Businesses meet a practical version of that limit in three forms: the tail of rare cases, the tacit knowledge in experienced people's heads, and the decisions the company never agreed on. The lifecycle does not pretend these away. Sort puts contested decisions in front of someone with authority, because only a person outside the rule system can extend it. Design sends the tail to people at junctions. Shadow surfaces the tacit rules that people apply without having written them down.

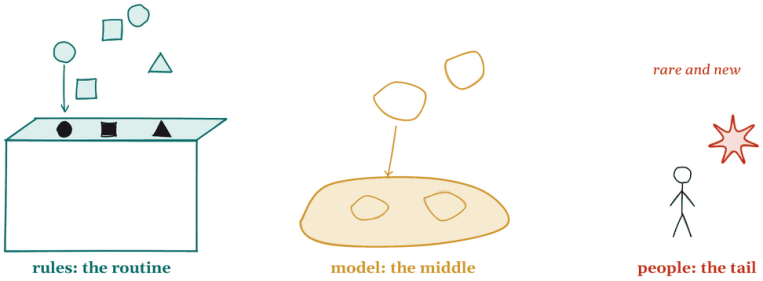


Like a rulebook at a game table: *some positions arise that no rule covers, and play waits until a referee outside the book writes a new ruling.*

Only variety absorbs variety

The cybernetician W. Ross Ashby stated a law that every designer of control meets: a regulator must have at least as much variety as the disturbances it faces. A loop that meets a hundred kinds of case cannot be run by a rule set that knows ten. The lifecycle spreads the variety across three things. Written rules take the routine body, where cases repeat. The model adds variety in the middle, where cases differ in ways rules would struggle to list. People hold the

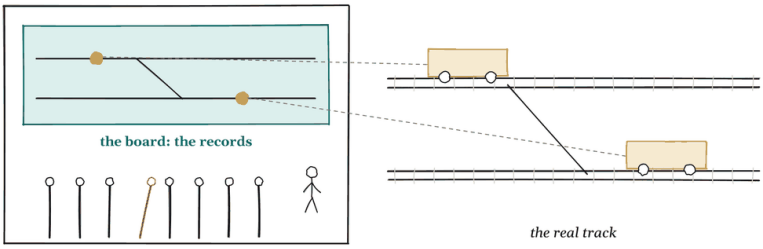
tail, where cases are rare, consequential and new. Design is largely the work of making that allocation explicit.



Like a child's shape sorter: *ten holes cannot take a hundred shapes, so common ones drop through, odd ones press into soft clay, and rare ones go to a hand.*

The record set is the regulator

Ashby's colleague Roger Conant proved with him a second result: every good regulator of a system must be a model of that system. A company that wants to control an embedded loop must hold a model of it. The map, the baseline, the registers, the standard, the shadow report and the charter are that model. They are not paperwork produced for an auditor. They are what lets the company see what the loop is doing and correct it. That is why every step ends in a signed document.

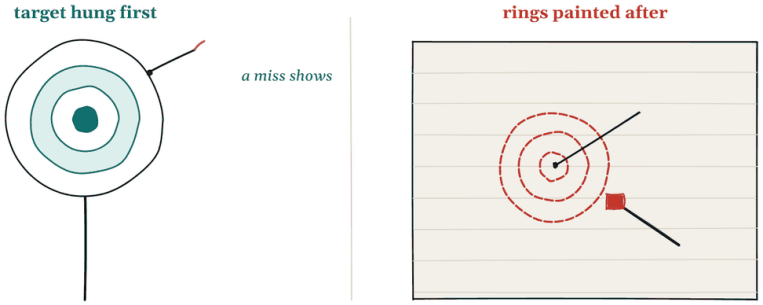


Like a railway signal box: *the signaller steers trains they cannot see by a board that mirrors the track, so the board is the control, not paperwork.*

Claims that can fail

The philosopher Karl Popper argued that a claim worth making is one that a result could prove wrong. Shadow and Bound are built that way. The agreement the loop must reach, body and tail apart, is written down before the cases arrive. The limits are written before the loop acts. If the evidence falls

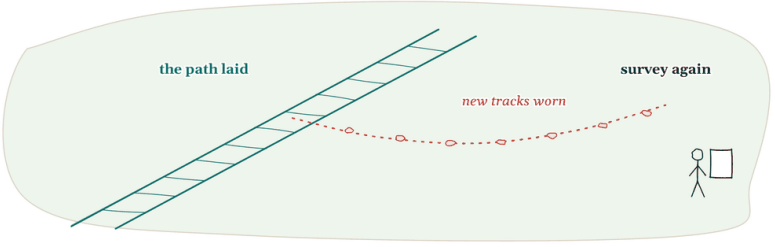
short, the gate does not open. A lifecycle that measures first and sets the bar afterwards can always declare success, and so it learns nothing.



Like an archer who paints the rings after the arrow lands: *always a bullseye, nothing learned. Hang the target first, and a miss can show.*

The loop changes what it reads

Norbert Wiener described control as a closed loop: act, sense the result, compare it with the goal, adjust. Core is the loop running. But a running loop also changes the distribution it was designed for. Suppliers adapt, volumes shift, the tail grows or shrinks. That is why the last step re-reads the distribution and returns to the map. **The lifecycle is itself a closed loop: its output is the next reading.**



Like paving a footpath across a park: *once it is laid, people walk differently and wear new tracks, so the gardener surveys the lawn again.*

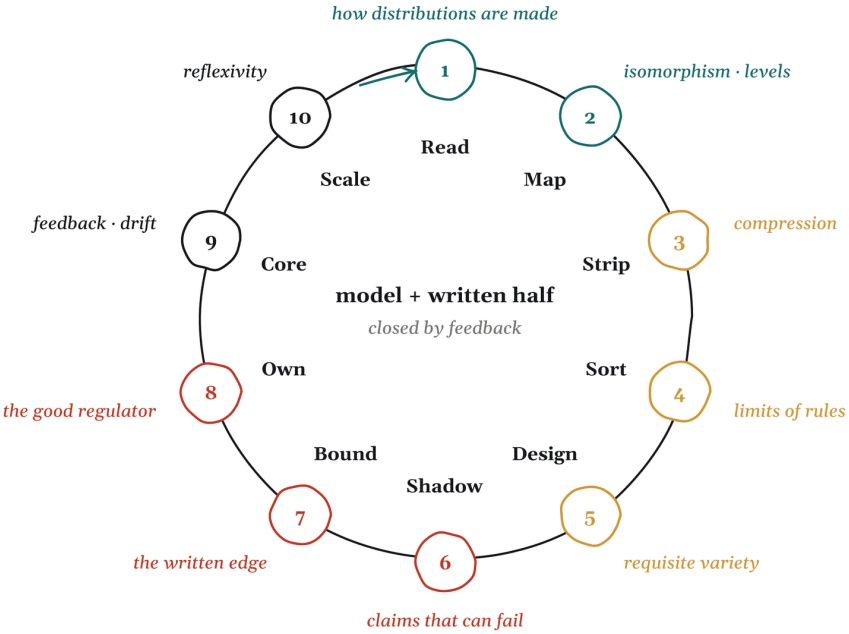
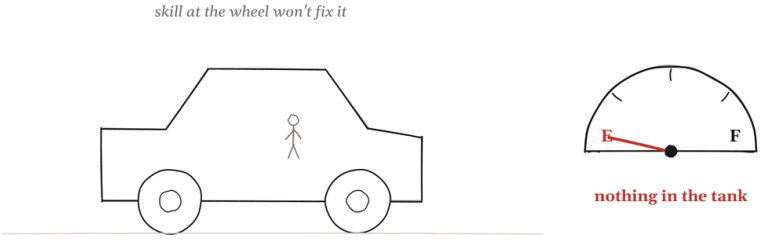


FIGURE 59 Why the lifecycle has this shape. Each step answers a question older fields already worked on: reading distributions, mirroring the work, compression, the limits of rules, requisite variety, falsifiable tests, the good regulator, feedback and reflexivity.

LXVI

Before Step One

The lifecycle assumes a few things are in place before the first reading. Most stalled programmes lacked at least one of them, and no amount of work at later steps makes up for it.



Like a car with an empty tank: *however skilled the driver, nothing done at the wheel later makes up for what was never put in.*

Five conditions

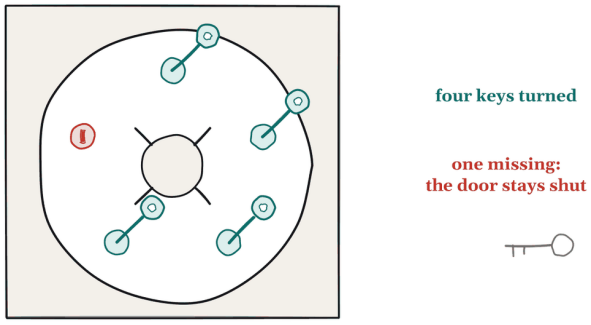
The first is a sponsor with authority over the loops in question. Embedding crosses departments, and only someone senior enough to settle a disagreement between them can keep the lifecycle moving. In most companies that means an executive committee member. A sponsor who cannot settle a contested decision will watch the programme stop at Sort.

The second is a business engineer: one named person who will carry the first loop from Read to Core. It can be a part-time role for the first loop, but it cannot be nobody. Lifecycles run by committee rarely reach Bound.

The third is time from the people who know the work. Mapping, stripping and sorting need operators, team leads and process owners in the room, for several half-days over a few weeks. If their managers will not release them, the map will be drawn by people who do not do the work.

The fourth is access. Someone must be able to pull a few months of cases for the distribution and the baseline, and later connect the executor to the systems it must read. A loop whose data cannot be reached cannot be read, let alone embedded.

The fifth is finance's agreement to measure. Finance should agree at the start that it will sign the baseline method and, later, the economics. Programmes that bring finance in at the end usually find their savings disputed.

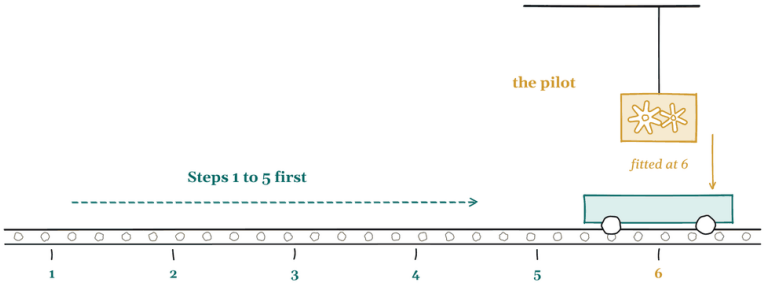


Like a vault that opens only when five keys turn: *the sponsor, the business engineer, the operators' time, data access and finance each hold one.*

If a pilot already exists

Many companies reach this point with a pilot already built. That is useful, not wasted, but it does not let the loop skip ahead. The right move is to run Steps 1 to 5 quickly around the pilot. Check on the map that this loop is the right one to start with. Map its real process and measure the baseline. Strip, sort and design. Then bring the pilot in at Step 6, where it becomes the executor in shadow. Most of what the pilot built survives. What changes is that it now runs against signed rules, inside designed junctions, and is graded against a written standard.

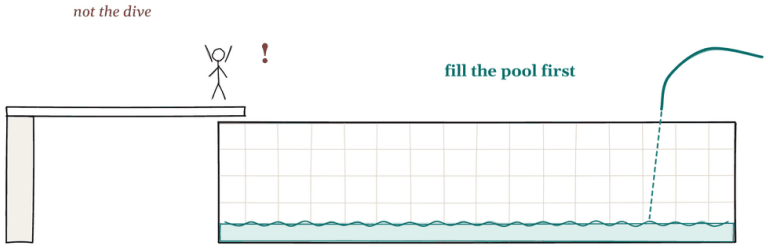
If Step 1 shows that the pilot's loop is a poor place to start, it is better to learn that in two weeks than in two quarters. The pilot can wait on the map until its loop's turn comes.



Like an engine built before the car: *keep it, run the chassis down the line first, then fit it at its station, the sixth step.*

Who should not start yet

A company that cannot meet the five conditions should start with the conditions, not the loop. That usually means one decision by the executive committee: name a sponsor, name a business engineer, free the operators' time, open the data, and bring finance in. It is a short list, and it is the difference between a programme that crosses the middle and one that stalls in it.



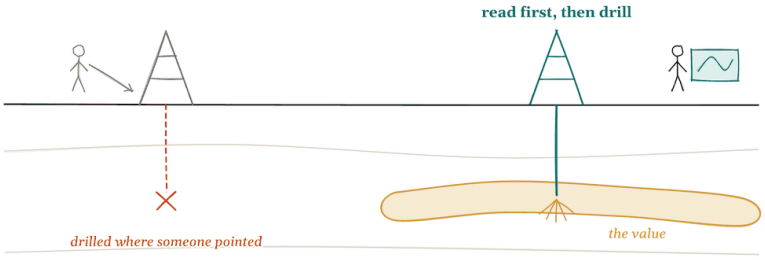
Like a diving board over an empty pool: *the first job is filling the pool, not perfecting the dive.*

LXVII

Step 1: Read

The step at a glance	
Question	Where does value sit, and which loop do we start with?
Starts when	An executive sponsor has asked for AI to reach the core, and has given the time of the people who know the work
Signed	The Work Loop Map, with the first loop chosen
Signer	The executive sponsor
Exit gate	One loop selected, its owner named, its distribution read
Typical duration	Two weeks

The lifecycle begins with a reading, not a project. Before anyone touches a process, the company needs a picture of its loops, where value sits among them, and which one to start with. Most programmes skip this step. They start with whatever a vendor demonstrated, or whatever a keen manager volunteered. Then they discover months later that they have embedded AI into work that did not matter much.



Like a prospector reading the survey before drilling: *most wells go down wherever someone pointed and come up dry, while the reading finds where the value lies.*

The work

The reading has three parts.

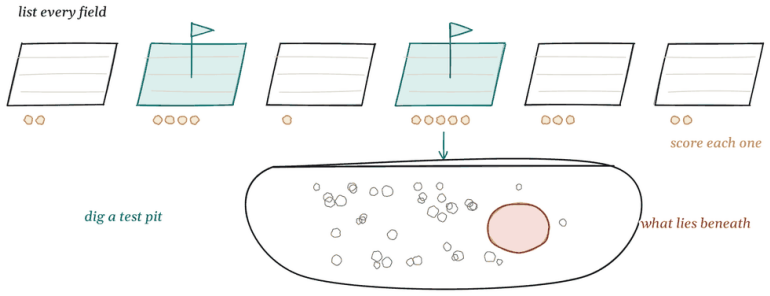
The first is listing the loops. A working session with the heads of the main functions lists the outcomes the company produces, one per line, in plain words. Approved supplier invoices. Resolved customer tickets. Qualified leads.

Closed month-end books. A mid-size company usually finds between twenty and fifty. Each is named by its outcome, not by the department that produces it, because many outcomes cross departments.

The second is placing each loop on the map. Two scores are enough for a first reading. Value to the core asks how much this loop matters to the business: its volume, its cost, its effect on customers, cash or risk. Readiness asks how close it is to being embedded: how well its rules are known, how measurable its outcome is, how much of its context is already in systems. Both are scored from one to five by the people who know the work, and the scores are argued over. The argument is useful. It is often the first time a leadership team has compared its processes side by side.

The third is reading the distribution of the candidate loops. For the two or three loops that score highest, someone pulls a few months of cases and looks at their shape. How many cases are there? How do their value and difficulty spread? Is there a routine body and a consequential tail, or are the cases all alike, or all different?

The distribution decides more than it seems to. A loop with a large routine body and a clear tail is a good first candidate, because the executor can take the body while people keep the tail. A loop where every case is different and consequential, such as a merger review, is a poor first candidate however valuable it is. A loop with high volume but trivial value may be easy, but it will not teach the company much or pay for the effort.



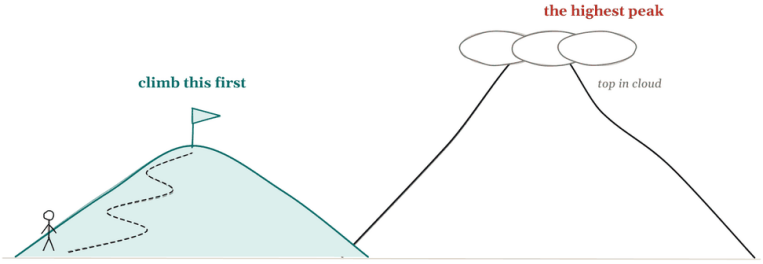
Like a farmer choosing where to plant first: *list every field, score each one, then dig test pits in the best two to see what lies beneath.*

Choosing the first loop

The first loop has a job beyond its own savings. It teaches the company the lifecycle. So it should have four properties. Its cases repeat, so evidence accumulates quickly. Feedback is fast, so the company learns within weeks

rather than quarters. Its outcome can be measured. And its owner is willing. A first loop owned by a reluctant manager will stall at Sort, where the owner has to decide things.

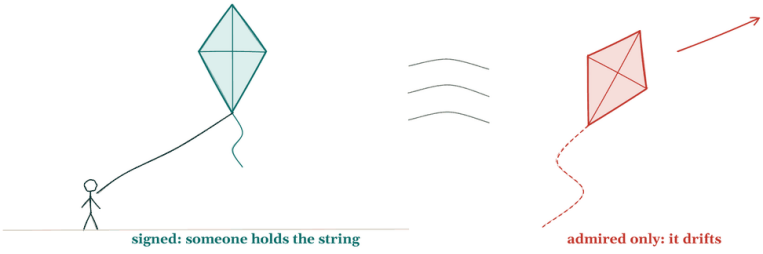
The best first loops are rarely the most valuable ones on the map. They sit in the top right quadrant, high on value and high on readiness, with a body and a tail. The most valuable loops often sit lower on readiness. Their turn comes once the company has learned the steps.



Like the first walk of a hiking holiday: *not the highest peak, but a near hill with a clear path and a visible top.*

What gets signed

The executive sponsor signs the Work Loop Map with the first loop marked. The signature commits three things: that this loop goes first, that its owner is named and has agreed, and that the people who know the work will be available for the next steps. A map that is admired but unsigned commits nothing, and the programme drifts.

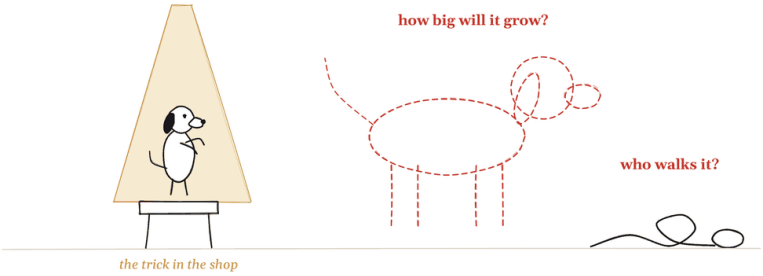


Like a kite: *a map that is only admired drifts off on the first wind, while a signed one has someone holding the string.*

How it fails

The common failure is choosing by enthusiasm instead of by the map. The loop that a vendor demonstrated, or the one a senior leader cares about, goes

first regardless of readiness. A second failure is skipping the distribution. Without it, nobody knows whether the loop has a body that an executor could run, and the design that follows treats every case the same. A third is naming an owner who has not agreed. Owners are named in a meeting they did not attend, and the loop has no one who will make its decisions.

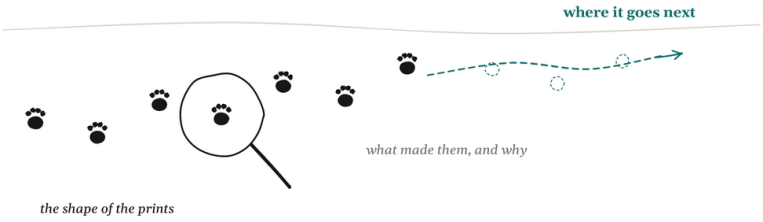


Like choosing a puppy because it did a trick in the shop: *nobody asked how big it grows, or whether anyone has agreed to walk it.*

What the foundations add

A distribution is the footprint of a mechanism. Many small independent causes produce a normal shape, where the average case is typical. Compounding, where big gets bigger, produces a heavy tail, where a few cases carry most of the value. Rules and thresholds leave their own marks, such as a pile of invoices just under an approval limit. So the reading should ask why the shape is what it is, not only what it is. A shape whose cause is understood can be predicted when the business changes. A shape that is merely observed cannot.

The reading also segments on three axes, not one. Value is the usual axis. Difficulty and risk can cut differently: a small invoice from a new supplier in a new country may belong in the tail even though its amount is trivial. Write the reading down as a prediction, such as "most invoices will stay under €5,000 next year". Later steps can test it, and Scale will.



Like a tracker reading prints in the snow: *the shape tells you what made them, and knowing that lets you predict where the trail goes next.*

At the manufacturer

The manufacturer's chief financial officer sponsored the programme. In two half-day sessions, the heads of finance, operations, sales, service, HR and IT listed thirty-four loops across six chains of work, from quoting to collecting cash.

Placing them took longer than listing them. Order fulfilment scored highest on value, but its rules were scattered across three plants and its readiness was low. Customer quoting was valuable and contested, and nobody wanted to start there. Supplier invoice approval scored four on value and three on readiness. It was the most frequent case in the company, it touched every plant, and finance could measure it to the cent.

The finance team pulled six months of supplier invoices. There were about 7,400 a month. Their shape was the clearest finding of the reading. Eighty-one percent of invoices were under €5,000, and they carried 14% of the spend. Sixteen percent fell between €5,000 and €50,000 and carried 40%. The last 3%, each over €50,000, carried 46% of everything the company paid its suppliers.

That was a routine body and a heavy tail, which made it the right first loop. The head of accounts payable agreed to own it. The CFO signed the map with the invoice loop marked first and goods receipt marked to map next.



Like a day's catch: *the heap of small fish is most of the count but little of the weight, while three big ones weigh almost half.*

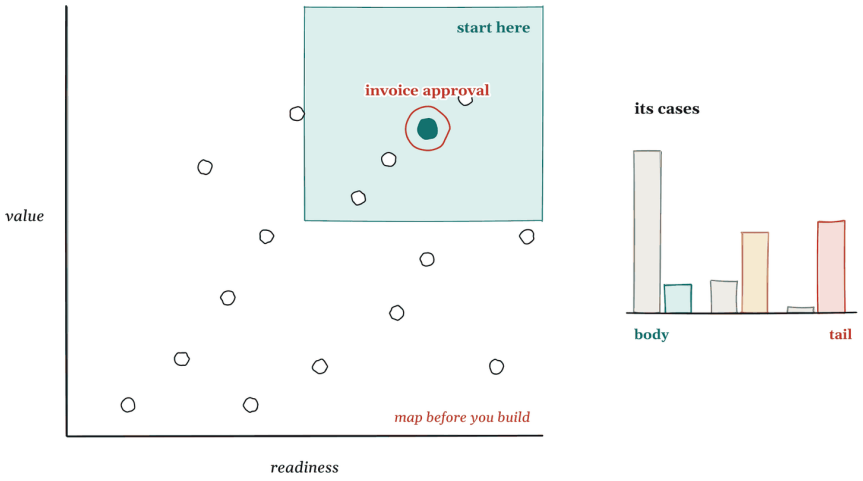


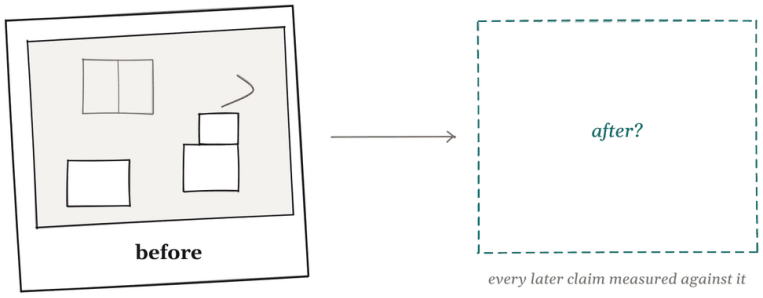
FIGURE 60 Step 1, Read. The loops placed by value to the core and readiness, with the first loop chosen in the upper right, and the shape of its cases read before anything else.

LXVIII

Step 2: Map

The step at a glance	
Question	How does the work really run today, and what does it cost?
Starts when	The loop is chosen and its owner has agreed
Signed	The current-state process map and the baseline sheet
Signer	The process owner
Exit gate	A map the people who do the work recognise, and a measured baseline
Typical duration	Three weeks

Mapping is where the lifecycle meets the real work for the first time. The aim is a picture of how the loop runs today, as the people who do it would describe it, together with numbers for what it costs. Both will be used for months. The map tells the next steps what to strip and sort. The baseline is what every later claim of improvement is measured against.



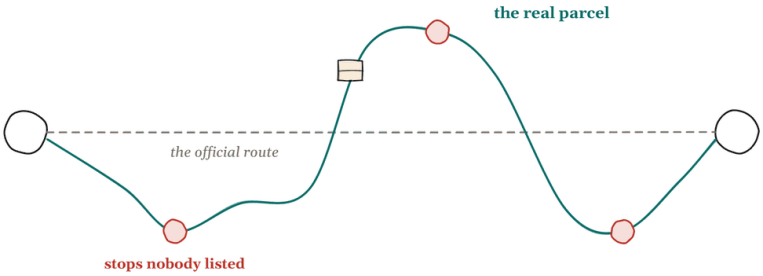
Like the before photo of a renovation: *taken of the room as it really is, so that every later claim of improvement is held up against it.*

The work

The map is drawn as swimlanes. Each lane is a role, a team, or an outside party such as a supplier or a customer. The steps run from left to right in the order they happen, each in the lane of whoever does it. Every step records the system or channel it uses, how many minutes of hands-on work it takes, and how long the case waits before the next step starts.

Two habits make the map honest. Draw it with the people who do the work, not the people who manage it. Managers describe the process as designed, and operators describe it as it runs. And walk real cases through it. Pick five recent cases, including one that went wrong, and follow each from start to finish. The steps the official map leaves out appear in the first hour.

While mapping, mark each step that looks like a workaround: re-keying between systems, printing for a signature, a batch job that runs weekly, an email chase between teams. Mark each step that redoes something that should have been right the first time. Do not argue about them yet. That is Step 3's job.



Like following five real parcels with a tracker: *the official route is a straight line, and the extra stops nobody listed show up within the first hour.*

The baseline

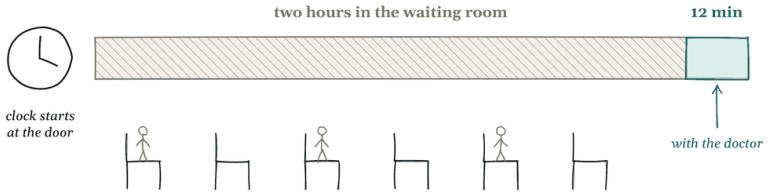
The baseline is the part most programmes skip and later regret. It records, for a sample of real cases, five numbers. These are the volume, the cost per case, the elapsed time from start to finish, the share of cases that needed an exception, and the share that were right the first time.

Cost per case should be fully loaded. It counts the people's time at their real cost, the systems, the rework, and the cost of errors that reach the outside. Elapsed time should be measured from the moment the case arrives, not from when someone first touches it. The waits are usually most of the time.

Two derived numbers help. Hands-on time divided by elapsed time is the flow efficiency of the loop. In most office processes it is shockingly low, often under 5%. The count of handoffs, each time the case moves between lanes, tells you where delay and errors gather.

The baseline must be measured, not estimated. Estimates are almost always too optimistic about the current process, which makes later savings look

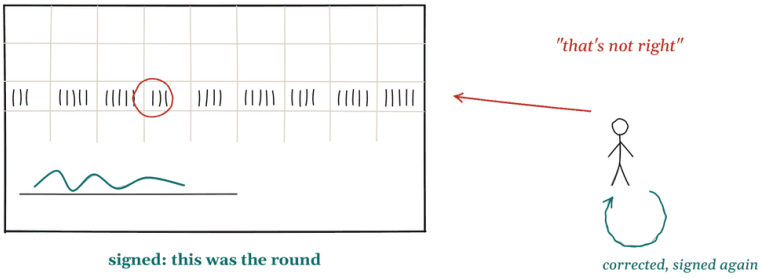
smaller than they are. Or they are too pessimistic, which makes the business case look better than it will turn out to be. Either way the economics at Step 8 will rest on sand.



Like a clinic visit timed from the moment you walk in: *twelve minutes with the doctor, two hours waiting, so the waits are most of the time.*

What gets signed

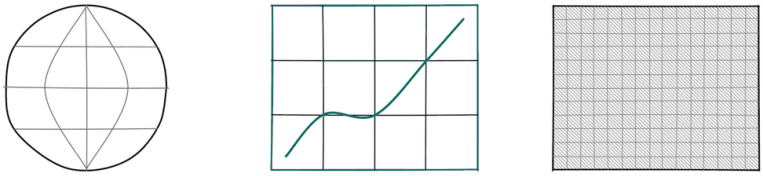
The process owner signs the map and the baseline sheet. The signature says two things: the map is how the work runs, and the baseline is the number we will be held to. If an operator later says "that's not how we do it", the map goes back to be fixed, and the owner signs again.



Like a golfer signing the scorecard: *it says these were the strokes, and if a marker spots a wrong one, it is corrected and signed again.*

How it fails

The most common failure is mapping the official process. The procedure manual is copied into lanes, nobody walks a real case, and the workarounds stay invisible. A second is mapping at the wrong grain. A map with three boxes hides everything, and a map with ninety boxes cannot be read. Somewhere between eight and twenty steps is usually right for one loop. A third is skipping the baseline because "we know roughly what it costs". Nobody does.



Like choosing a map for a city walk: *a globe hides every street, a plan of every brick cannot be read, and the street map in between works.*

What the foundations add

A map means something only when its structure mirrors the work. Douglas Hofstadter called this isomorphism: symbols carry meaning when they change as the thing they describe changes. A map copied from the procedure manual mirrors the manual, not the work. That is why the map is drawn from walk-throughs of real cases and checked against the system log.

Mapping is also a choice of level. The same invoice can be described as keystrokes, as steps, as decisions or as one line in the accounts. Keystrokes are too fine to redesign, and the accounts are too coarse. The useful level for embedding is steps and the decisions inside them, because that is the level at which work can be stripped, sorted and handed to an executor. Every map leaves things out. The test is whether what it leaves out would have changed the design.



Like a mirror and a portrait: *the mirror moves as you move, the portrait keeps an old pose, so the map is drawn from the work itself.*

At the manufacturer

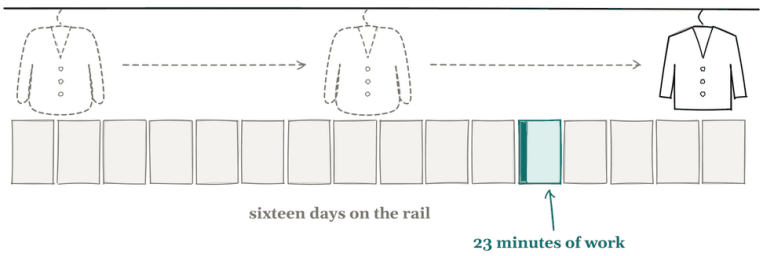
The invoice loop had six lanes: the supplier, a scanning desk, the accounts payable clerks, the budget owners across the plants, the financial controller,

and treasury. Two half-day sessions with four clerks and a team lead drew fourteen steps.

The walk-through of real cases changed the map twice. The official procedure said invoices arrived through the supplier portal. In fact, about half still arrived as PDF attachments, and a clerk printed and scanned them so the scanning desk could read them into the system. The procedure said budget owners approved in the ERP. In practice, most printed a batch of invoices every Friday, signed them, and sent them back by internal mail.

Finance measured 1,200 invoices over six weeks. The fully loaded cost was €11.80 per invoice. Elapsed time from receipt to payment-ready averaged sixteen days. Twenty-four percent of invoices raised an exception, and 71% matched their order and receipt the first time. Hands-on work averaged twenty-three minutes per invoice, which put flow efficiency well under 1%. Nine clerks did the work.

The numbers sat within published ranges. Industry surveys put the average cost of processing an invoice at around nine to ten dollars, with the best performers far lower. The baseline was not unusual. It was simply measured for the first time. The head of accounts payable signed the map and the baseline sheet.



Like a coat left at the dry cleaner's for sixteen days that gets twenty-three minutes of cleaning: *the invoice spent almost all its time waiting.*

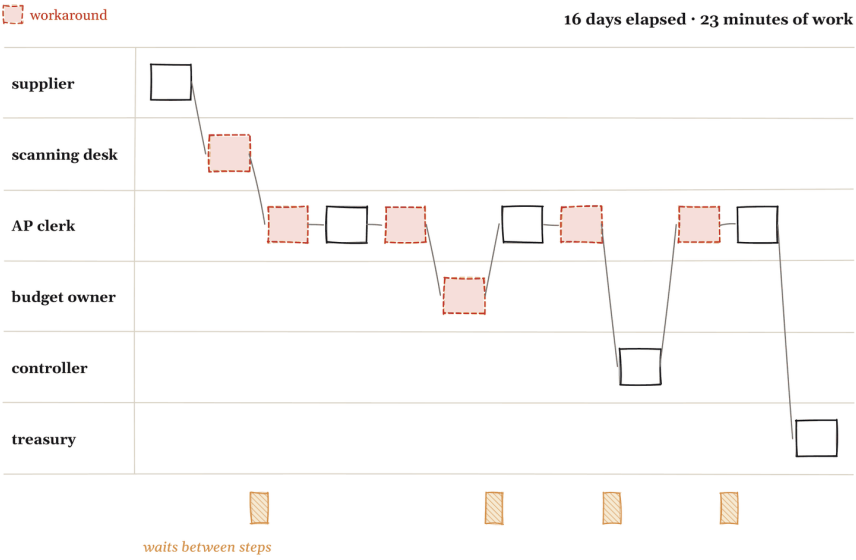


FIGURE 61 Step 2, Map. The loop drawn as it really runs: six lanes, fourteen steps, workarounds marked in red, and long waits between steps. Twenty-three minutes of work take sixteen days.

LXIX

Step 3: Strip

The step at a glance	
Question	Which steps exist only because of constraints that no longer apply?
Starts when	The current-state map and the baseline are signed
Signed	The strip register
Signer	The process owner
Exit gate	Every step on the map is marked struck or kept, with the constraint named
Typical duration	One week

Every documented process is a mixture of three things. The first is the outcome itself. The second is the steps the outcome genuinely needs. The third is a layer of workarounds for constraints that once applied and no longer do. A batch step exists because a system once ran overnight. A handoff exists because two teams once sat in different buildings. A second review exists because errors were once frequent. A printed form exists because approvals were once on paper.

These workarounds are called compensations, because each compensates for a constraint. Over years they become indistinguishable from the real work. People are trained in them, controls are built on them, and optimisation polishes them until they look essential. Attach an executor that decides to a process full of compensations, and it will run the workarounds faster. It will also keep every wait between them.

Strip comes before any rule is written down, for that reason. There is no point gathering the rules of a step that should not exist.



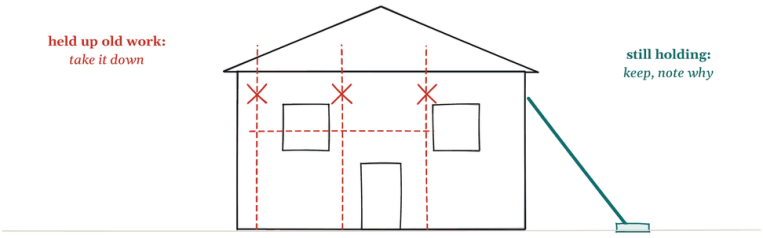
Like a path that still bends round trees felled years ago: *walkers keep every detour, and a faster walker simply takes the same bends sooner.*

The work

Take the signed map and walk through it step by step. For each step, ask one question: what constraint is this step compensating for, and does that constraint still apply?

The answer falls into three groups. Some steps serve the outcome directly: matching an invoice to its order is part of approving it. Keep these. Some compensate for a constraint that is gone: re-keying a PDF exists because the systems were not connected, and they now can be. Strike these. Some compensate for a constraint that still applies: a four-eyes check on large payments compensates for the risk of fraud, which has not gone away. Keep these, but record the constraint, because Step 5 may meet it differently.

Record each step in the strip register with its status and the constraint behind it. The constraint is the important column. A step struck without a named constraint is an opinion. A step struck because "the weekly batch dates from the 2009 ERP, which now posts in real time" is a finding.

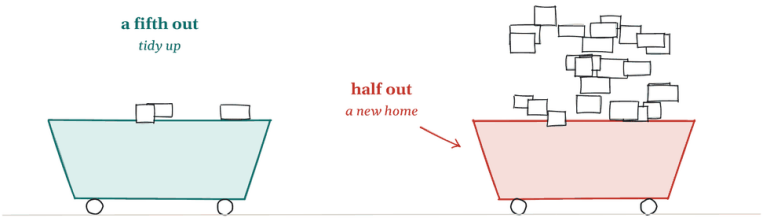


Like clearing scaffolding from a finished house: *take down the poles that held up work long done, keep the prop still holding a wall, and note why.*

The strip ratio

The share of steps struck is the strip ratio. It tells you what kind of work you are about to do.

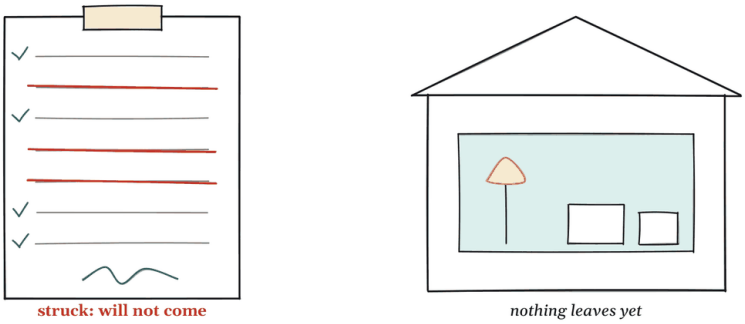
A low ratio, under a fifth, means the process is mostly real topology. Embedding it is largely a matter of letting an executor run the steps that remain. A high ratio, over a third, means the process is mostly accretion. Embedding it is a redesign, and the loop that comes out will look little like the map that went in. Neither is better. But a team that expects a light automation project and finds a 45% strip ratio needs to know before it plans the next three months.



Like clearing a house before moving: *if a fifth goes in the skip you tidy up, if half does you are planning a new home, so measure first.*

What gets signed

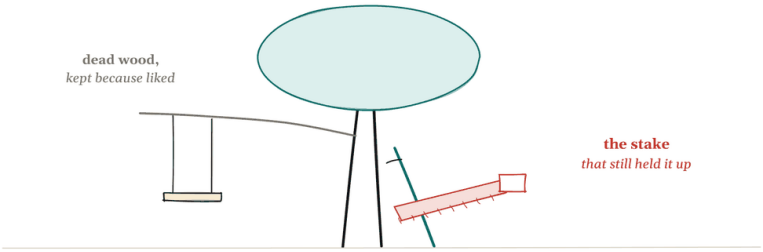
The process owner signs the strip register. Struck steps are not removed yet. Nothing changes in the live process at this step. The signature says that these steps will not be carried into the design, and that the owner accepts the reason given for each.



Like a packing list before a move: *the owner strikes what will not come to the new house, but nothing leaves the old one yet.*

How it fails

The usual failure is stripping too little because every step has a defender. Someone built the weekly reconciliation, and someone else relies on its output. The way through is to ask what the output is used for and whether the new loop could produce it directly. A second failure is stripping controls. A review that looks redundant may be the only thing standing between the company and a fraud. If the constraint still applies, the step stays, or its purpose moves to a junction or the standard. The third failure is quietly skipping the step, which leaves every compensation in place for the executor to inherit.

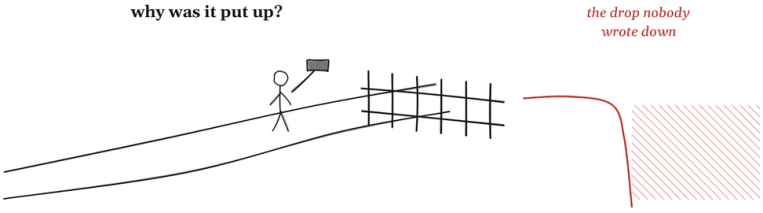


Like pruning a tree: *dead branches stay because someone hung a swing on them, while the stake still holding the tree up is sawn through.*

What the foundations add

Stripping is compression. The shortest description of a process that still predicts its results is the best understanding of it, and every compensation is a line in the description that predicts nothing anymore. It answers to a constraint that has gone. Removing it makes the rules that remain fewer and truer, which matters twice over once a model reads them. A model follows a long, inconsistent procedure as faithfully as a short, clear one, so it will faithfully automate the noise.

There is a discipline to it. The writer G. K. Chesterton advised against removing a fence until you know why it was put up. Each compensation is struck only after someone has named the constraint it answered and confirmed that the constraint is gone. A step struck without that check sometimes turns out to have been holding up a rule nobody wrote down, and Shadow will find it the hard way.



Like Chesterton's fence across a lane: *find out why it was put up, and confirm the reason has gone, before you take it down.*

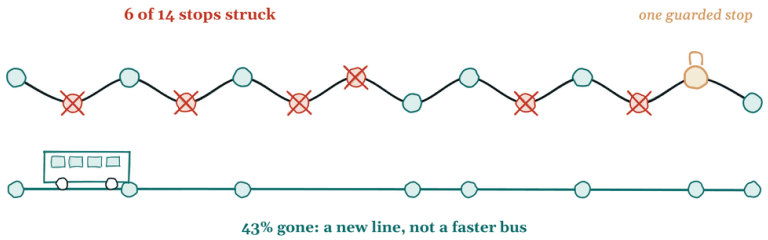
At the manufacturer

The clerks and their team lead went through the fourteen steps in a single afternoon. Six were struck.

Printing and scanning PDF invoices compensated for a scanning desk that predated email intake. The invoice platform could now read attachments directly. Budget owners' printed Friday approvals compensated for a time when approvals were on paper, and the ERP had supported electronic approval for years. Weekly batch posting compensated for an ERP job that had once run weekly and now ran continuously. A second clerk re-checked every invoice under €1,000. That step had been added after an error spike years before, and the errors had long since stopped. Clerks chased goods receipts by email because the receiving teams had no way to see which invoices were waiting. And a manual duplicate check against a spreadsheet compensated for a duplicate rule the ERP now applied itself.

Eight steps were kept. One of them, treasury's call-back to verify changed supplier bank details, compensated for fraud risk that was very much alive. It stayed, marked as a likely junction.

The strip ratio was 43%. That told the team it was redesigning the loop, not just speeding it up. The head of accounts payable signed the register.



Like an old bus route of fourteen stops: *six served places that had moved away, one guarded stop stayed, and what remained was a new line.*

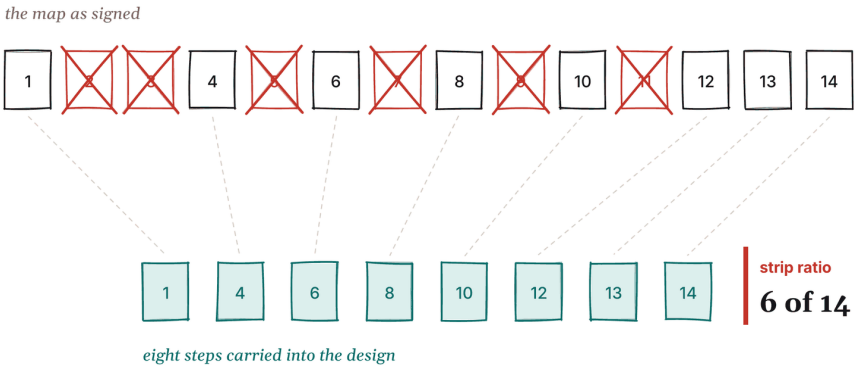


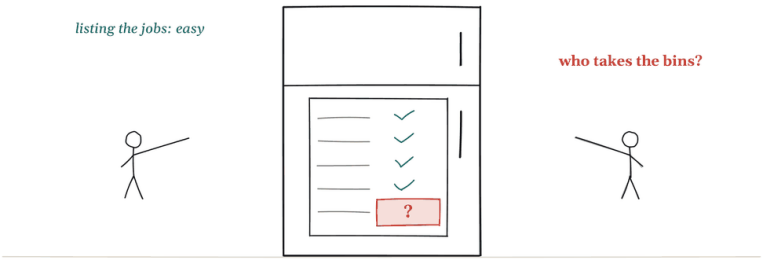
FIGURE 62 Step 3, Strip. Fourteen steps go in; six are struck, each with the constraint it compensated for. Eight remain. A strip ratio of 43% means the loop will be redesigned, not just sped up.

LXX

Step 4: Sort

The step at a glance	
Question	Which rules apply, and who decides what?
Starts when	The strip register is signed
Signed	The rule register, the decision register and the contested-decision log
Signer	The process owner; the executive sponsor or the relevant leaders for contested decisions
Exit gate	Every rule owned and dated, every decision classed, no contested decision left open
Typical duration	Three weeks, of which the contested decisions often take most

Sort is where the lifecycle meets the company's unspoken agreements, and where the most important findings of the whole programme usually appear. It has two halves. The first gathers the rules the loop follows. The second sorts every decision the loop contains.



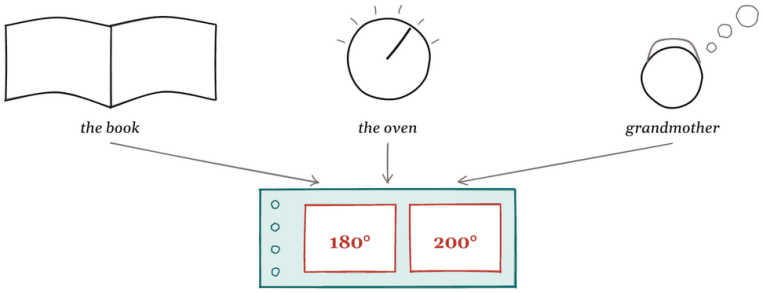
Like a house-share writing its chores down for the first time: *listing the jobs is easy, and the real argument is who takes out the bins.*

Gathering the rules

The rules of a loop live in three places. Some are documented in policies and procedures. Some are configured in systems, as tolerances, approval limits and validation checks. And some live in the heads of experienced staff: the supplier who always invoices in two parts, the cost centre that is never charged for freight, the threshold below which nobody really checks.

All three must be gathered into one rule register. Each rule gets a source, an owner, and a date. Rules in people's heads are written down in the words of the person who holds them, then confirmed by the owner.

Then compare. Where the policy and the system disagree, or two experienced clerks apply different rules to the same case, record a contradiction. **Contradictions are findings, not noise.** Each one is a place where the executor would have had to guess, and where the current process has been guessing for years. Each needs an owner to decide which rule stands.



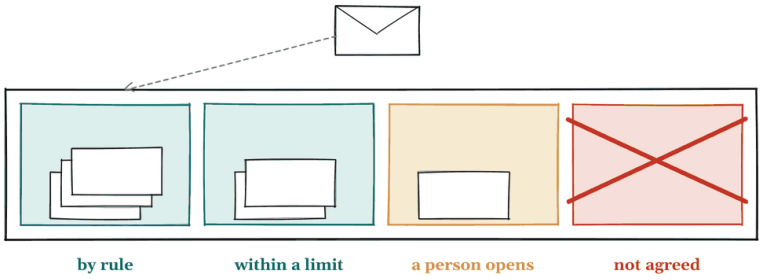
Like gathering a family's recipes from the cookbook, the oven settings and grandmother's memory into one binder, and finding two temperatures for the same cake.:

Sorting the decisions

Next, list every decision the remaining steps contain. A decision is any point where the work could go more than one way. Is this a duplicate? Does the price match within tolerance? Which cost centre? Approve or hold? Phrase each as a question.

Then sort each into one of the four classes. Mechanical decisions follow a rule completely. Bounded decisions need judgement inside a written limit. Consequential decisions carry a high cost of error or cannot be undone, so a person always decides. Contested decisions are ones the organisation has never agreed on.

The sort is partly technical and mostly organisational. A decision is mechanical only if its rule is written and uncontradicted. It is bounded only if someone has signed the limit. It is consequential because of its stakes, not because someone is nervous. And it is contested if, when the question is asked in a room, two senior people answer differently and both are sure they are right.



Like a mailroom's four pigeonholes: *letters handled by rule, ones settled within a limit, ones a person must open, and a fourth boarded up until agreed.*

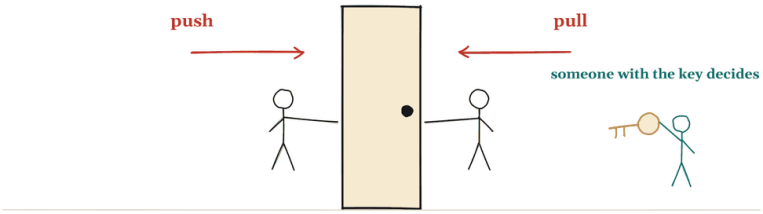
The contested decisions

Contested decisions deserve their own section because they decide the pace of everything after this step.

They are found by asking. Put each decision to the process owner and the people around them, and listen for disagreement. They are usually decisions that the old process handled by escalation, by habit or by not looking. Pay a disputed invoice to keep a key supplier happy? Give a goodwill refund outside the policy? Accept a late claim from a long-standing customer? Each has been answered case by case for years, differently each time.

No engineering can resolve a contested decision. An executor cannot be given a rule the company does not have. Someone with the authority to decide must decide, write the answer down, and sign it. In practice this is often a short meeting that has been avoided for years. It is also one of the most valuable meetings the programme will hold, because the same decision was going to surface as an incident the first time an executor met it.

The exit gate of Sort is strict on this point. **A loop with an open contested decision cannot pass.** It can wait, and the rest of the work can continue. But nothing goes to Design until the contested-decision log shows every entry decided, by whom and when.



Like a door with one person pushing and another pulling: *no new hinge will fix it; someone with the key decides which way it opens.*

What gets signed

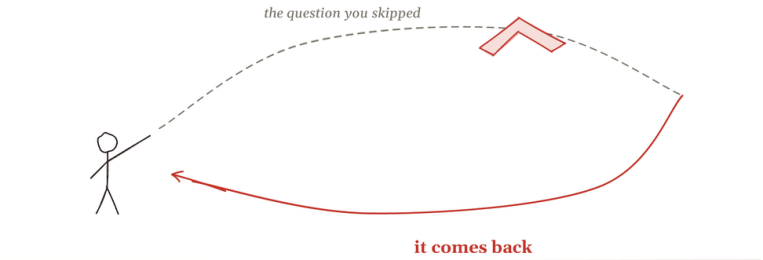
The process owner signs the rule register and the decision register. The contested-decision log is signed by whoever settled each entry, usually the executive sponsor or the leaders who share the decision.



Like selling a house: *the owner signs the deeds, while each boundary dispute is signed by the neighbours who settled it.*

How it fails

The first failure is treating Sort as documentation. A team collects the policies into a spreadsheet, does not interview the operators or read the system settings, and finds no contradictions because it never compared anything. The second is classing decisions by comfort. Everything is "consequential" because nobody wants to let go, so the executor is left with nothing to do. Or everything is "mechanical" because the pilot looked good, and the loop takes decisions nobody authorised. The third, and the most damaging, is noticing a contested decision and moving on. It will come back.



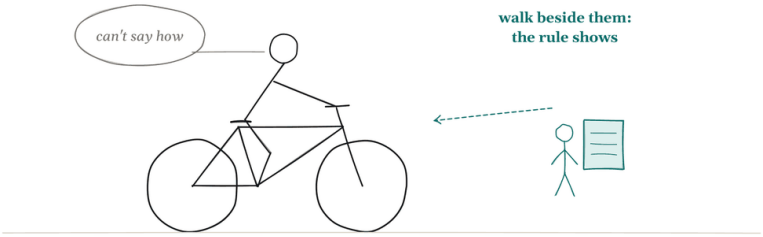
Like a boomerang: *throw an awkward question away unanswered and it comes back, usually the first time an executor meets it.*

What the foundations add

Sort is where the limits of rules become practical. Some rules can be gathered from documents and systems. Others live as tacit knowledge, and as the philosopher Michael Polanyi put it, people know more than they can tell. Asking experts to list their rules rarely works. Walking real cases with them does, because the rule shows up as a decision they make and can then explain.

The contested decisions are the deepest case. They cannot be settled inside the rule system, because the organisation never agreed on the answer. Gödel's result has a business analogue here: the system has to be extended from outside. Only a person with authority can do that, which is why Sort escalates rather than guesses.

The decision register matters for one more reason. Decisions written apart from the process steps can be given to an executor directly, tested one by one, and changed without redrawing the map. That separation is what makes the written half of the loop maintainable.



Like asking a cyclist how they keep their balance: *they cannot tell you, but walk beside them and the rule shows in what they do.*

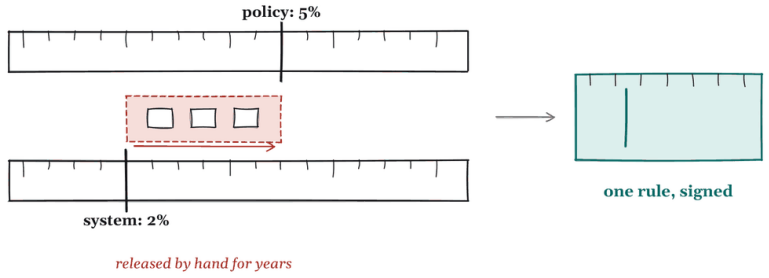
At the manufacturer

The rules came from four sources. These were the finance policy manual, the ERP configuration, the invoice platform's settings and three days of interviews with clerks. Fifty-eight rules made it into the register. Nine contradicted one another.

The largest contradiction was the price tolerance. The policy allowed a 5% difference between invoice and order before an invoice needed approval, while the ERP blocked anything over 2%. Clerks had been releasing blocked invoices between 2% and 5% by hand for years, which explained a good share of the exceptions. The controller decided the tolerance would be 2% for orders under €50,000 and 1% above, and the policy was changed to match. Smaller contradictions covered who could approve a non-order invoice under €500, and how to round invoices in foreign currencies.

Seventeen decisions remained after Strip. Eight were mechanical, such as duplicate detection, matching to an order and receipt within tolerance, and tax-code validation. Five were bounded, including coding a small non-order invoice to the right cost centre and accepting a partial delivery within limits. Three were consequential: releasing invoices over €50,000, accepting changed supplier bank details, and approving a non-order invoice over €5,000.

One was contested. When a supplier of a single-source part disputed a deduction and threatened to hold shipments, should the company pay the disputed amount to protect supply? Procurement said yes. Finance said never. Each had acted on its view for years. The CFO and the chief operating officer settled it in a thirty-minute meeting three weeks after it was logged. Payment was allowed up to €25,000, with the procurement director's signature and a record in the dispute log. Above that, it went to the CFO. The answer went into the contested-decision log with both their names.



Like two rulers marked at different lengths: *the policy said 5%, the system 2%, clerks bridged the gap by hand until the controller set one rule.*

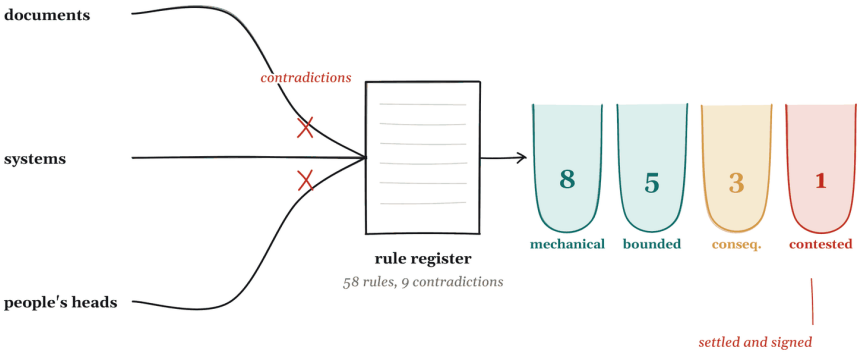


FIGURE 63 Step 4, Sort. Rules from documents, systems and people's heads flow into one register, where contradictions surface. Every decision is sorted into its class; the contested one is settled and signed before Design.

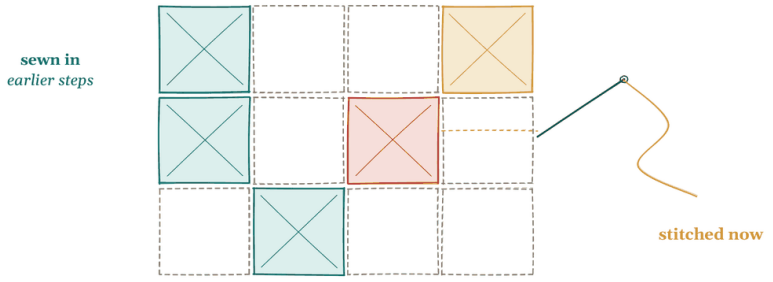
LXXI

Step 5: Design

The step at a glance	
Question	What does the loop look like, including where people decide and how it is graded?
Starts when	The rule and decision registers are signed and the contested-decision log is closed
Signed	The Work Loop Canvas, the junction register and the standard
Signer	The loop owner
Exit gate	Gate G0 reached, with the nodes G1 needs signed
Typical duration	Two weeks

Design turns the findings of the first four steps into a loop. Up to now the lifecycle has described the old work and decided what to keep. Now it draws the new work: what the executor reads and changes, where people decide, where unhandled cases go, how the loop connects to its neighbours, and how its results will be graded.

The design is captured on the Work Loop Canvas. Several of its twelve nodes are already filled from earlier steps. The outcome came from Read. The compensations came from Strip. The rules and decisions came from Sort. Design fills the rest and signs the ones the next gate needs.



Like a quilt of twelve patches: *some were already sewn in earlier steps, and design stitches in the rest so the new work is whole.*

The work

Context comes first. List everything the executor must read to act well: the order, the receipt, the contract price, the supplier record. For each, name the system that holds it and the person who keeps it accurate. Mark any source the executor cannot reach, or that exists only as a scan. A gap here becomes a wrong decision later.

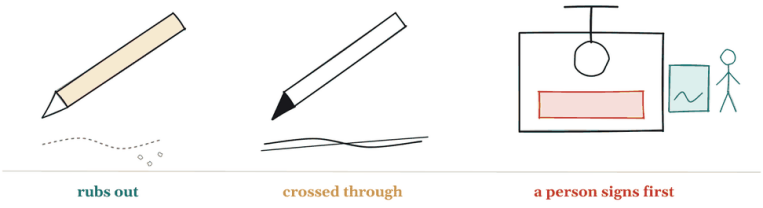
Actions come next. List everything the executor may change, in which system, with which permission. For each, answer one question: can it be undone, and at what cost? Reading is cheap. Posting an invoice can be reversed. Releasing a payment cannot.

Then place the junctions. Every consequential decision and every action that cannot be undone needs one. For each, write down who decides, what evidence the executor puts in front of them, and how a wrong decision is reversed. Then name the person who can switch the whole loop off, and how they would do it.

Then design the exception queue. Some cases will fall outside every rule. They go to a single queue with a named owner and a time limit. A weekly review looks at what landed there and turns repeated exceptions into new rules.

Then the handoffs. Write down what the loop receives, from which loop, and what it passes on, to which loop. For each, name who resolves a mismatch when the two sides disagree about what was sent.

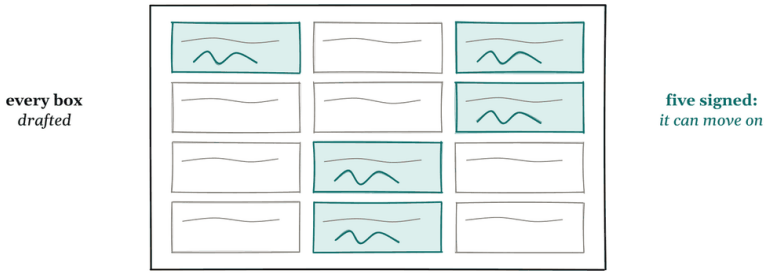
Finally, the standard. Write down how every outcome will be graded. A standard has criteria, the things an accepted outcome must satisfy, and a grading method. The usual pattern combines rules that check every case, a model that grades every case against the criteria, and a small human audit that keeps the model grader honest. The audit should look harder at the tail.



Like pencil, pen and printing press: *pencil rubs out, ink can be crossed through, but before the press runs a person signs the proof.*

Gate G0

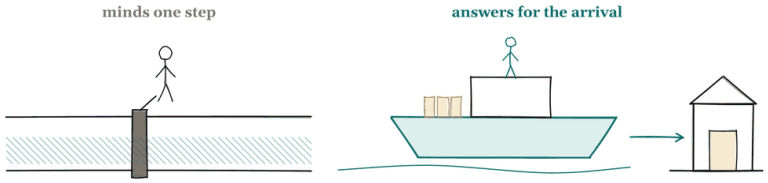
When every node on the canvas has a draft, the loop has reached G0, Mapped. For the loop to pass the next gate, five nodes must also be signed: the outcome, the compensations, the rules, the context and the standard. Design is the step where those signatures happen.



Like an application form: *every box filled in makes it complete, but it moves on only when the five boxes that need a signature are signed.*

What gets signed

The loop owner signs the canvas, the junction register and the standard. In many companies the process owner becomes the loop owner at this point. It is a small change of title that matters. A process owner looks after steps. A loop owner answers for an outcome, including the parts an executor produces.

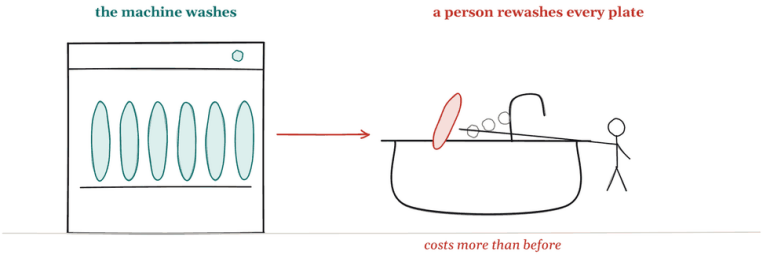


Like a lock-keeper becoming a ship's captain: *one minds a single step of the canal, the other answers for the cargo arriving at port.*

How it fails

The first failure is designing for the demo. The loop is drawn around what the pilot did well, and the junctions and exceptions are left for later. Later never comes. The second is placing junctions by comfort instead of by reversibility. Every step gets a human approval "to be safe", the executor does nothing a person does not repeat, and the loop costs more than the old one. The third is writing a standard that grades activity instead of outcomes. "Invoices

processed per hour" is not a standard. "Correct supplier, amount, tax and cost centre, with no duplicate, ready within terms" is.



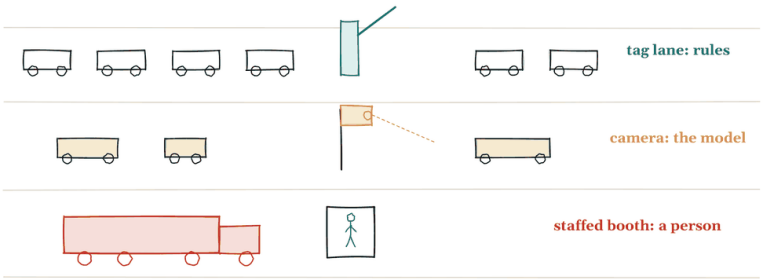
Like rewashing every plate the dishwasher cleaned, just to be safe: *the machine runs, nothing is saved, and the kitchen costs more than before.*

What the foundations add

Design allocates variety. Ashby's law says the loop as a whole must have as much variety as the cases it meets. Written rules take the body, where cases repeat and can be listed. The model takes the variety in between, reading cases that differ in ways no one could enumerate. People take the tail at the junctions. A design that asks rules to cover the tail becomes brittle. One that asks the model to cover it becomes unaccountable. One that sends everything to people has embedded nothing.

The strongest designs pair the two kinds of machinery inside each decision. The model proposes and the rules check: a match is suggested, then verified against the order and the receipt. A coding is drafted, then tested against the account rules. This is the pattern that made systems such as AlphaGeometry reliable, applied to ordinary work.

Design also builds the sensors with the actuators. The standard and the grader are designed at the same time as the actions, because a loop that cannot see its own results cannot be corrected. That is the closed loop drawn on paper, before anything runs.



Like a toll plaza: *tag lanes pass the regular cars, a camera reads the odd plates in between, and a staffed booth takes the rare oversized load.*

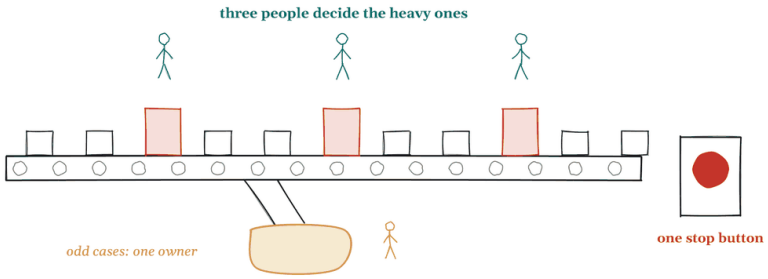
At the manufacturer

The executor would read the order, the goods receipt, the supplier record and contract prices. Contract prices turned out to live in PDF agreements, not in the ERP, for about a third of suppliers. Procurement agreed to load the prices for the top two hundred suppliers into the ERP before shadow running began. Below that, price checks would rely on order prices alone.

Three junctions were placed. Invoices over €50,000 went to the financial controller, who would see the matched order, the receipt, the contract price and the supplier's payment history. New or changed bank details went to a treasury analyst, who would call the supplier on a number held separately. Non-order invoices over €5,000 went to the budget owner of the cost centre. The head of accounts payable could switch the loop off from the platform's console. Invoices would then return to the clerks' queue within the hour.

The exception queue went to the accounts payable team lead, with a limit of two working days and a review every Monday. The standard required the right supplier, amount, tax and cost centre, no duplicate, and readiness within payment terms. Rules would check every invoice. A model grader would grade every invoice against the criteria. A finance quality analyst would audit 3% of invoices at random, and 20% of those over €10,000.

The head of accounts payable signed the canvas as loop owner. With the outcome, compensations, rules, context and standard signed, the loop stood at G0 and was ready for shadow running.



Like a packing line with three staffed checkpoints for the heaviest items, a reject bin with a named owner and one red stop button.:

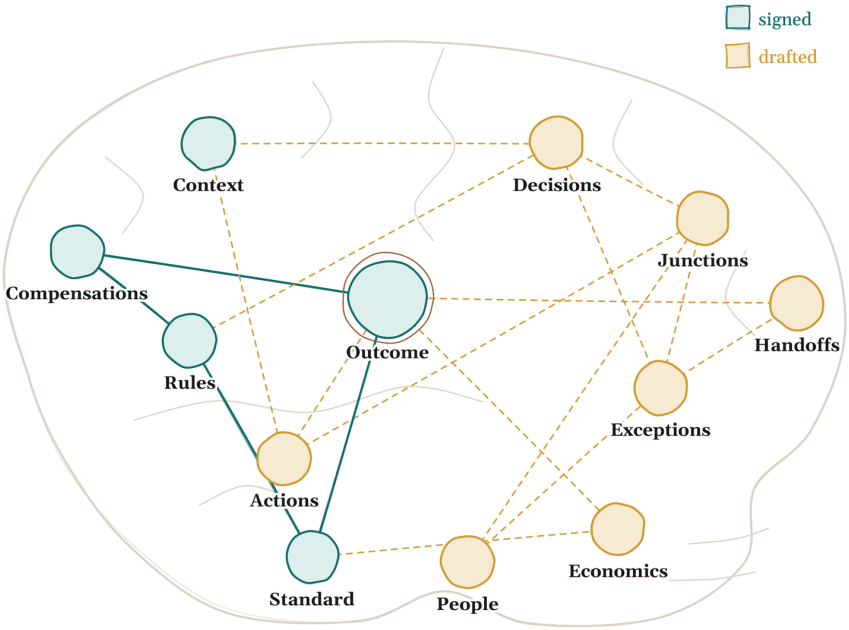


FIGURE 64 Step 5, Design. The canvas with every node drafted (G0, Mapped). The five nodes the next gate needs are signed: outcome, compensations, rules, context and standard.

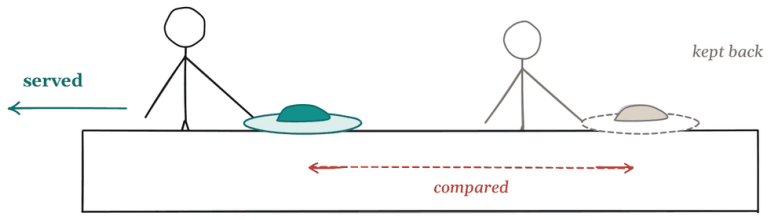
LXXII

Step 6: Shadow

The step at a glance	
Question	Does the loop reach the same results as people, without acting on anything?
Starts when	Gate G0 is reached and the G1 nodes are signed
Signed	The shadow report
Signer	The loop owner and the grader
Exit gate	Gate G1 passed: agreement measured on the body and the tail, every disagreement explained
Typical duration	Four to eight weeks

Shadow is the first time the executor meets the real stream of cases. It reads every case the current process receives and reaches its own result. But it changes nothing. People keep doing the work as before. At the end of each case, the two results are compared.

The point is not to prove the executor is perfect. It is to learn, on real cases and at no risk, where it agrees with people, where it disagrees, and why.



Like a trainee chef cooking every dish beside the head chef: *only the head chef's plate goes out, and the two plates are compared each time.*

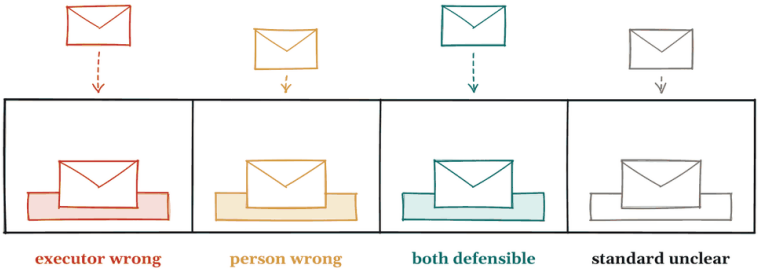
The work

Run the loop in parallel for long enough to see the whole distribution. A week of routine cases teaches little. The tail cases, which are rarer and more consequential, need several weeks to arrive in useful numbers. For most loops, four to eight weeks is enough.

Measure agreement separately for the body and the tail. A single overall number hides the most important fact. An executor can agree with people on 98% of routine cases and 80% of large ones, and the overall figure of 95% would describe neither.

Then study every disagreement. Each falls into one of four groups. The executor was wrong, and the cause is a missing rule or a gap in context. The person was wrong, which happens more often than teams expect, and it is worth recording. Both were defensible, which usually points to a rule that is vaguer than anyone thought. Or the standard itself was unclear, so the grader could not decide.

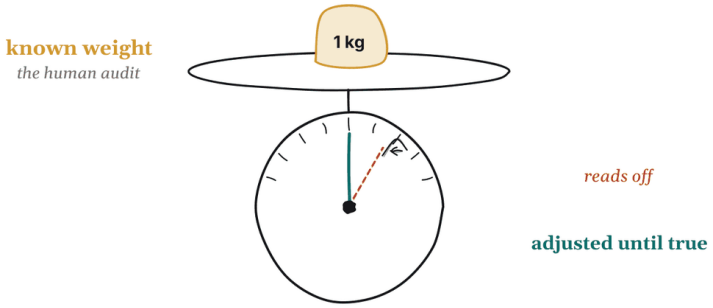
Each group sends the work somewhere. Missing rules go back to Sort, and the rule register is signed again. Context gaps go back to Design. Unclear criteria go to the standard. The shadow report records every change.



Like a post room with four pigeonholes: *every disagreement is filed by cause, and each hole is emptied by the desk that can fix it.*

The grader

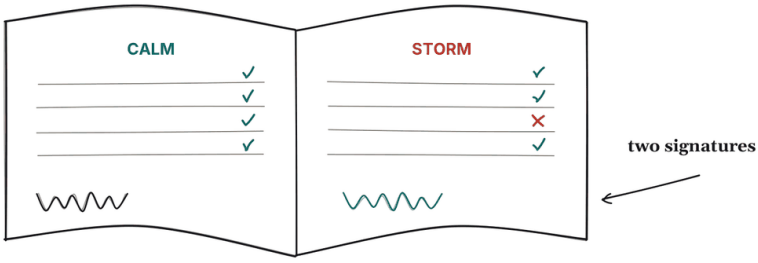
Shadow is also where the standard is tested. The model grader and the human audit both grade the executor's results and the people's results. When they disagree with each other, the grader is calibrated. By the end of shadow, the loop owner should trust the grader enough to rely on it, because from the next step onward the grader is the main way anyone knows the loop is working.



Like a market scale checked against a known weight each morning: *once its needle matches, traders trust it to weigh the whole day unwatched.*

What gets signed

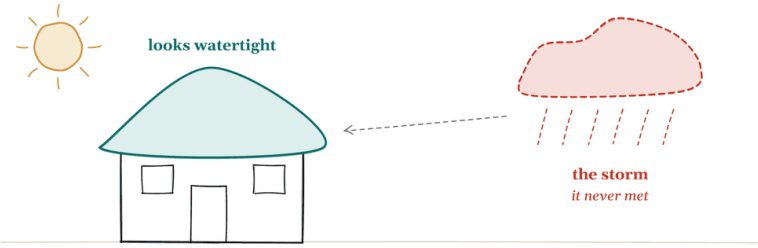
The loop owner and the grader sign the shadow report. The report states the agreement on the body and the tail, the disagreements by group, the changes made to rules, context and the standard, and the remaining known weaknesses. G1 is passed when the owner judges the evidence sufficient to let the loop act on part of the work.



Like a trainee pilot's logbook: *calm and stormy hours are tallied apart, the faults are listed, and two signatures release the next stage.*

How it fails

The first failure is too short a shadow. Two weeks of clean routine cases produce a reassuring number that says nothing about the tail. The second is reporting one overall agreement figure. The third is treating disagreements as the executor's fault by default. Teams that do this never discover the rules their own people apply inconsistently, which are often the more valuable finding. The fourth is changing the executor during shadow without recording why. When the loop later fails, nobody can say which version was proven.



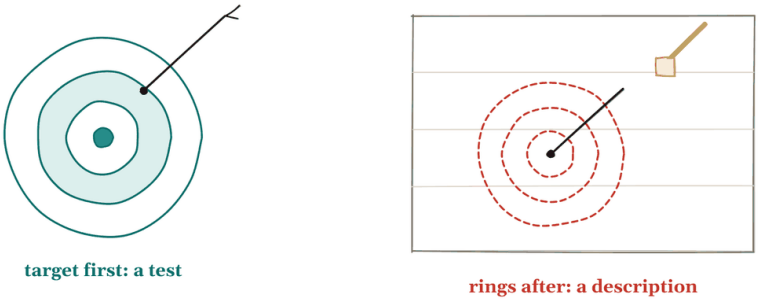
Like judging a new roof after two sunny weeks: *it looks watertight only because the storm that would test it never came.*

What the foundations add

Shadow is a test that can fail, and it should be set up that way. The agreement targets for the body and the tail are written into the G1 nodes before the first case arrives. Results that miss them stop the gate. Setting the bar after seeing the numbers turns the test into a description.

Shadow also maps what researchers call the jagged frontier. A learned model is strong and weak in places nobody can predict from a demonstration. Only the real distribution of cases shows where its edges are, band by band. That is why shadow runs on the live stream of cases, not on a curated set, and why the tail needs weeks to arrive in useful numbers.

Finally, shadow is where tacit rules surface. Every disagreement where the person was right and the executor wrong is a rule someone applied without writing it down. Every disagreement where the executor was right is a rule people had stopped following. Both feed the written half of the loop.



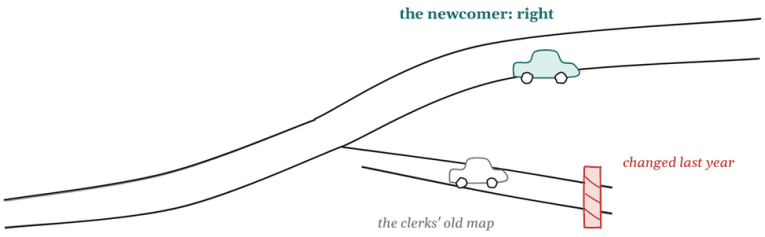
Like an archery target fixed before the shot: *painting the rings around wherever the arrow landed turns a test into a description.*

At the manufacturer

Shadow ran for six weeks on 9,800 invoices. Agreement with the clerks' results was 97.8% on invoices under €5,000 and 81% on invoices over €50,000. The overall figure of 94% appeared once in the report, with a note on why it should not be quoted alone.

Two kinds of error accounted for most disagreements. Credit notes from some suppliers looked like invoices, and the executor read several as invoices. And invoices covering more than one order confused the matching. Two rules were added to the register and signed again. Procurement corrected the supplier records that had caused one cluster of mismatches.

A third finding surprised the team. In about one in five disagreements on mid-sized invoices, the grader sided with the executor. Clerks had been applying an old cost-centre rule that finance had changed the year before. The rule was clarified and the clerks were briefed. The head of accounts payable and the finance quality analyst signed the shadow report, and the loop passed G1.



Like veterans driving by last year's road map: *in one disagreement in five the newcomer was right, because the clerks followed a rule finance had changed.*

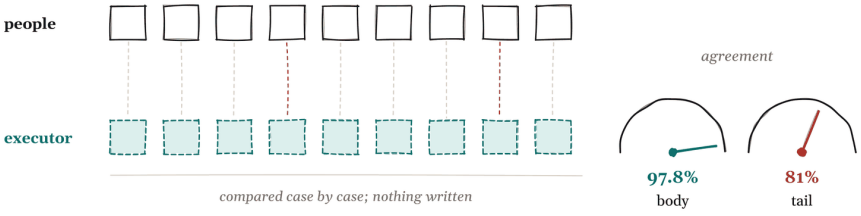


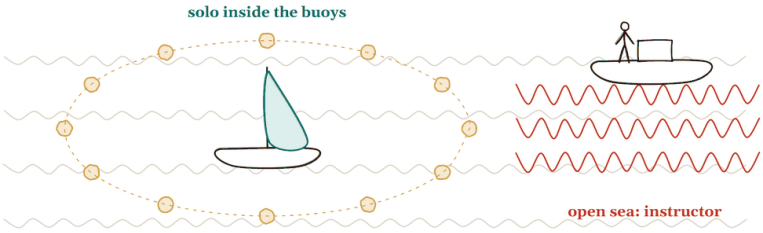
FIGURE 65 Step 6, Shadow. The executor and the people work the same cases side by side; nothing the executor does reaches the records. Agreement is read separately for the body and the tail.

LXXIII

Step 7: Bound

The step at a glance	
Question	Can the loop act alone on part of the work, inside written limits?
Starts when	Gate G1 is passed
Signed	The gate review record
Signer	The control owner, with internal audit consulted
Exit gate	Gate G2 passed: the loop acts alone inside its limits, and the switch-off has been tested
Typical duration	Four weeks of bounded running before the review

Bound is the first time the executor acts. It does so only inside limits written down in advance, usually the routine body of the distribution. Everything outside the limits still goes to people. The step turns the evidence of shadow into a decision about trust, and it puts that decision in front of the person responsible for control.



Like a learner sailing solo inside a ring of buoys: *the routine water is theirs alone, while the open sea still needs the instructor aboard.*

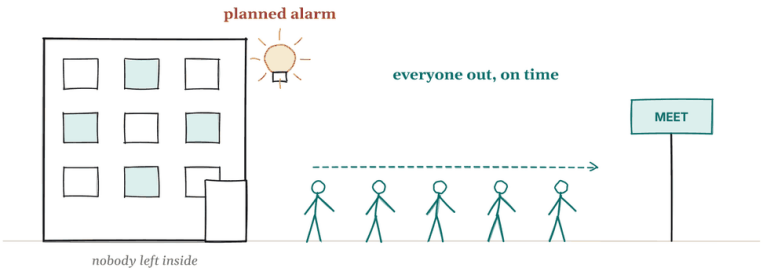
The work

Write the limits. They are drawn from the shadow evidence, not from hope. The cases where agreement was highest, the decisions that were mechanical or bounded, and the actions that can be undone define the first scope. Limits are usually expressed as plain conditions. Below a value. Within a tolerance. From a known counterparty. Of a known type.

Then complete the nodes the second gate needs. The actions, the decisions, the junctions and the exception queue must all be signed. The executor now writes to systems, so its permissions must match the actions node exactly. Permissions granted "for convenience" during the pilot are removed.

Then test the switch-off. Before the loop acts on its first live case, the person named as able to stop it does so, in a planned test. Cases should flow back to people within the stated time, and nothing should be lost. A switch-off that has never been tested is a hope.

Then run inside the limits for several weeks, grading every outcome, and bring the evidence to a gate review.

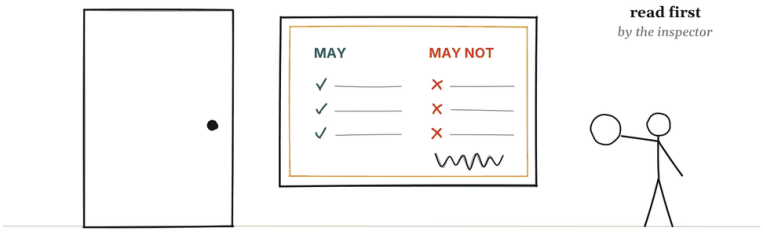


Like a fire drill before the building opens: *the alarm is pulled on purpose, everyone files out on time, and nobody is left inside.*

The gate review

The gate review is a short meeting with a written record. The control owner, such as the financial controller or the head of risk for the process, reviews the limits and the graded results. They also review the exceptions, the junction decisions and the switch-off test. Internal audit is consulted, so that the controls it will later test are the ones the loop actually has.

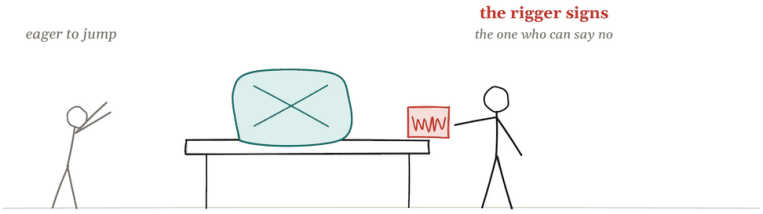
The record states what the loop may do alone, what it may not, and what evidence supported the decision. It is the document an auditor or regulator will ask for first.



Like a permit framed by a building's door: *it says plainly what may happen inside and what may not, and it is the first thing an inspector reads.*

What gets signed

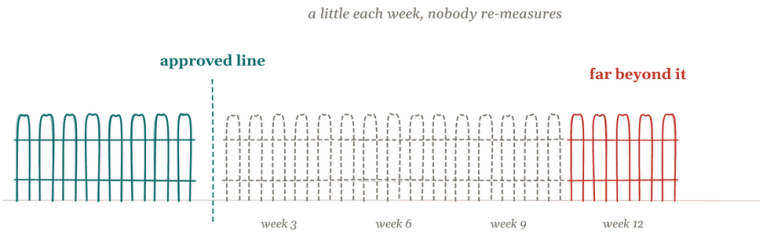
The control owner signs the gate review record. This is deliberately not the loop owner. The loop owner is responsible for the outcome and naturally wants the loop to run. The control owner is responsible for the company's controls. Bound is the step where those two views meet, and the signature belongs to the one whose job is to say no.



Like a parachute checked by the rigger, not the jumper: *the jumper is eager to go, so the signature belongs to the one whose job is to refuse.*

How it fails

The first failure is setting limits by ambition rather than evidence. The pilot handled large cases well in a demo, so large cases go into scope. The second is letting the scope creep. Each week someone adds another supplier, another value band, another type, and nobody holds a new review. Each extension of scope is a new gate, however small. The third is never testing the switch-off. The fourth is giving the executor broader system permissions than its actions node allows.

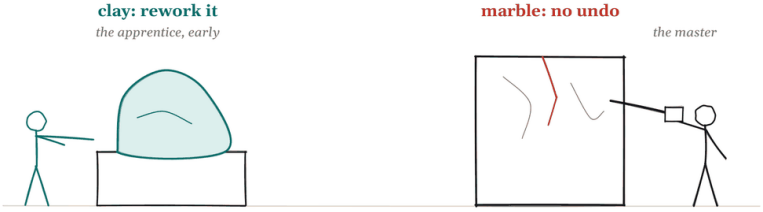


Like a garden fence moved out one section a week: *each move looks tiny, nobody measures again, and soon it stands far beyond the approved line.*

What the foundations add

The bounded scope is the written edge of the rule system. Inside it, the loop acts alone because the evidence says it can. Outside it, cases go to people. Drawing that line explicitly is the practical answer to the limits of rules: not pretending the rules cover everything, but saying exactly where they stop.

The line follows reversibility as much as value. An action that can be undone cheaply can be trusted to the loop earlier, because a mistake is a correction, not a loss. An action that cannot be undone, such as a payment, waits behind more evidence and a person at the junction. The control owner signs because their role is to stand outside the loop and say no. In Gödel's terms, they are the authority from outside the system that decides when it may be extended.



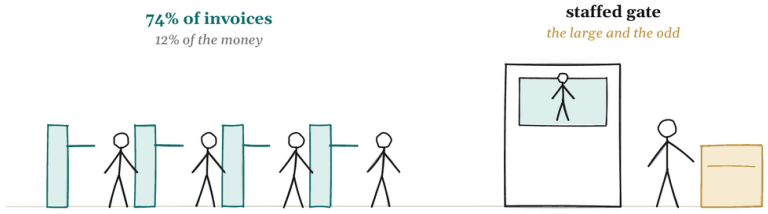
Like a sculptor's studio: *the apprentice may shape clay early because a mistake can be reworked, while the cut in marble waits for the master.*

At the manufacturer

The limits followed the shadow evidence closely. The loop could act alone on invoices backed by an order, under €5,000, within the 2% tolerance, from suppliers the company had paid for at least six months. That was 74% of all invoices by count, and about 12% of spend. Everything else stayed with the clerks and the junction holders.

The platform's permissions were cut back to the actions on the canvas. A service account from the pilot that could edit supplier records was removed. In the first week, the head of accounts payable switched the loop off during a planned test. Invoices returned to the clerks' queue in twenty minutes, and none was lost.

After four weeks the financial controller held the gate review. The graded results showed 98.6% of bounded invoices met the standard, and the exception queue had cleared within its limit every week. Internal audit asked for one change: the dispute log required by the contested decision was to be linked to the payment record. The controller signed the record, and the loop passed G2.



Like ticket gates at a station: *most travellers pass the automatic gates alone, while the few with heavy or odd luggage go to the staffed gate.*

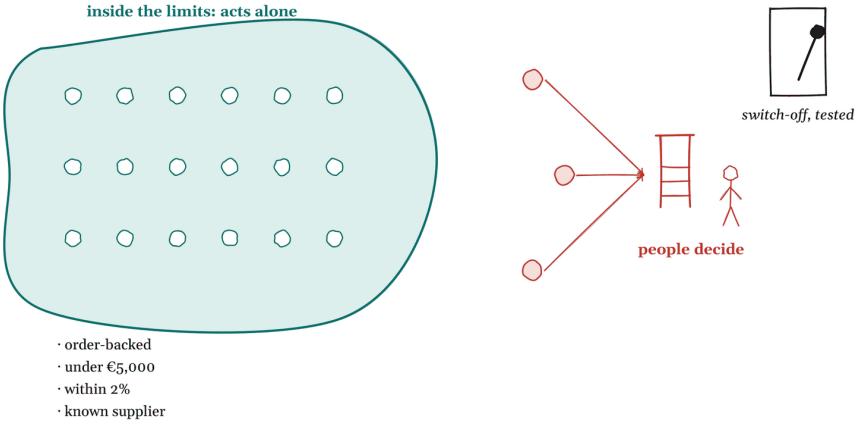


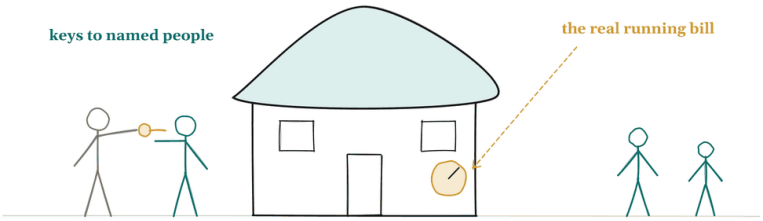
FIGURE 66 Step 7, Bound. The executor acts alone only inside a fenced scope drawn from the shadow evidence. Everything outside goes through gates to people, and the switch-off is tested before the first live case.

LXXIV

Step 8: Own

The step at a glance	
Question	Are named people in place, and what does one accepted outcome really cost?
Starts when	Gate G2 is passed
Signed	The loop charter and the economics sheet
Signer	The loop owner for the charter; finance for the economics
Exit gate	Gate G3 passed: people named and in post, cost per accepted outcome measured against the baseline
Typical duration	Six to ten weeks, mostly measurement

Own is the step that makes the loop part of the organisation rather than a project inside it. It does two things. It puts named people into every role the loop needs, and it measures, over enough weeks to trust the number, what the loop costs per accepted outcome compared with the baseline from Step 2.



Like a house handed from builder to household: *the keys pass into named hands, and the real running bill is measured month by month.*

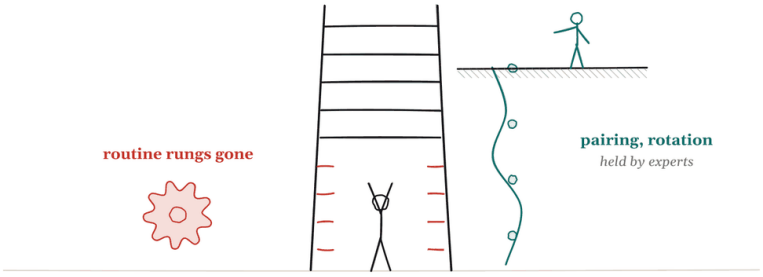
The people

A loop running at scale needs five kinds of role filled by named people, not by job titles.

The loop owner answers for the outcome. The junction holders decide the consequential cases. The exception owner works the queue and runs the weekly review. The grader keeps the standard calibrated. And someone owns adoption: the people whose work changes, how they move into new roles, and how new staff will learn the work when routine cases no longer teach it.

That last point is easy to miss. In the old process, junior staff learned by handling simple cases until they were ready for hard ones. When the executor takes the simple cases, that path disappears. The charter has to say how people will learn the judgement the junctions need. Common answers are pairing with junction holders, rotating through the exception queue, and reviewing graded cases.

The people part of Own also includes the plain question of whose job changes and how. This is better answered openly and early than discovered by rumour. Most loops move people from routine work to exceptions, junctions, supplier relationships or the next loop to be embedded. Some roles shrink. The charter records the changes agreed, and the role change plan at Step 10 carries them across the map.



Like a ladder whose lowest rungs the machine has taken: *new staff need another way up to the hard cases, such as a rope held by experts.*

The economics

The economics compare the loop's cost per accepted outcome with the baseline. Each part of that phrase matters.

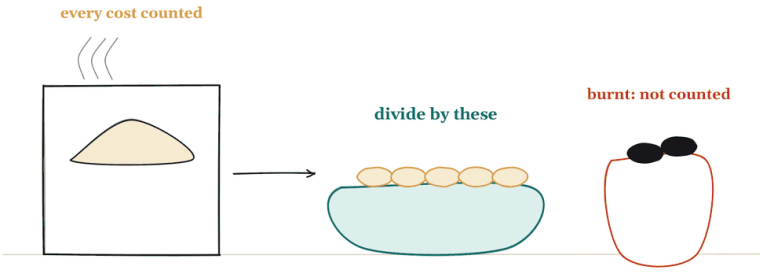
Cost is fully loaded, as the baseline was. It includes the model and platform costs, the people at junctions and in the exception queue, the grading and audit, and the rework. Leaving out the people still working in the loop is the most common way to flatter the number.

Per accepted outcome means dividing by the outcomes that met the standard, not by the cases processed. A loop that processes many cases quickly but sends a fifth back for rework is not cheap.

Compared with the baseline means against the measured figure from Step 2, over a comparable period. And the body and the tail are reported separately. A

loop can cut the cost of routine cases sharply and leave the tail unchanged, and the average would hide both facts.

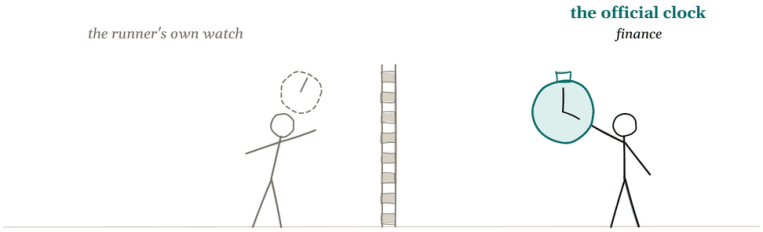
Measure for long enough to cover a full business cycle where possible, such as a month-end or a quarter-end. Six to ten weeks is typical.



Like pricing a bakery's bread by loaves that pass, not loaves baked: *count flour, oven and staff, then divide by the good ones only.*

What gets signed

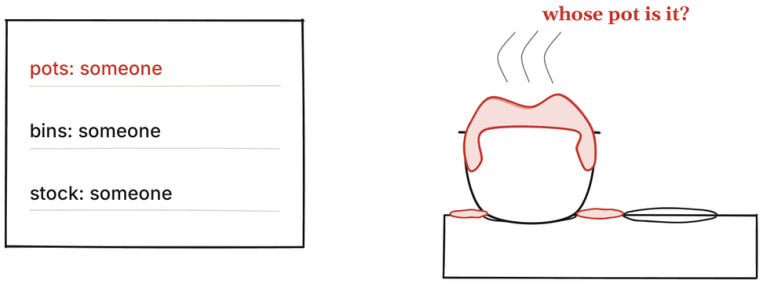
The loop owner signs the loop charter, which names every role and records the agreed changes to people's work. Finance signs the economics sheet, usually the planning and analysis lead. It is finance's number, not the programme's.



Like a race timed by the official's stopwatch, not the runner's wristwatch: *finance holds the clock, so the cost figure is theirs, not the programme's.*

How it fails

The first failure is naming roles, not people. The charter says "junction holder: finance" and nobody in finance knows they hold it. The second is flattering the economics. The model cost is counted, but the exception queue, the grading, the audit and the platform are not. The third is skipping the adoption owner, which leaves the change to people's work unmanaged and the next generation of experts untrained. The fourth is declaring savings that depend on headcount changes nobody has agreed.

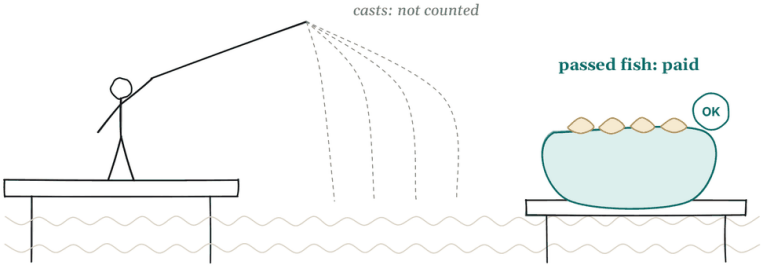


Like a duty rota that says 'someone' on every line: *when the pot boils over, nobody knows it is theirs to deal with.*

What the foundations add

Named people are part of the regulator, not decoration around it. When the loop meets something its rules cannot handle, a person extends the rules. The exception owner turns a recurring exception into a new rule. The junction holder decides a case no rule anticipated. The loop owner settles what the register leaves open. A loop with no named owners has no way to grow its written half, so it slowly falls behind the work.

Cost per accepted outcome is the feedback signal for economics. It counts only results that met the standard, so it cannot be improved by doing more work badly. The economist Charles Goodhart warned that a measure made into a target stops measuring. Activity counts, such as cases processed, fail that test. Accepted outcomes, counted against a measured baseline, hold up.



Like paying an angler by fish that pass the market inspector, not by casts: *casting more cannot raise the pay, only good catch can.*

At the manufacturer

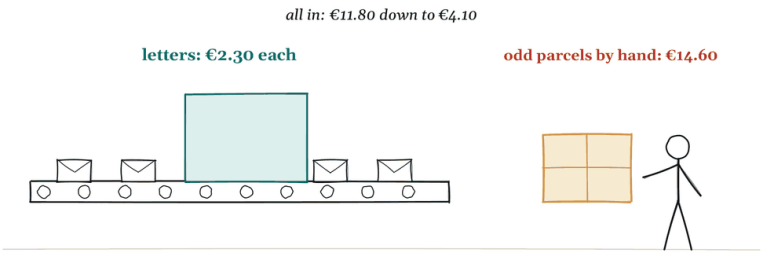
The charter named the head of accounts payable as loop owner. The financial controller, a treasury analyst and fourteen budget owners were named as

junction holders. The accounts payable team lead owned the exception queue, and the finance quality analyst the grading. The finance operations manager took on adoption.

The changes to people's work were agreed with the clerks over three weeks. Of the nine clerks, four moved to work the exception queue and supplier queries. Three moved to the supplier onboarding team, which the map had marked as a loop to embed later. One moved to treasury. One retirement during the period was not replaced. New clerks would rotate through the exception queue and sit with the controller for junction decisions during their first three months.

Finance measured eight weeks across a quarter-end. The fully loaded cost per accepted invoice was €4.10, against the baseline of €11.80. That figure included the platform, the model usage, grading, audit and every hour spent at junctions and in the exception queue. Reported separately, invoices in the body cost €2.30 each and those in the tail €14.60, because the tail still needed people. Elapsed time from receipt to payment-ready fell from sixteen days to 3.4. The share right first time rose from 71% to 93%.

The head of accounts payable signed the charter. The planning and analysis lead signed the economics sheet. With the people and economics nodes signed, the loop passed G3.



Like a sorting office after the machine arrived: *each letter costs far less all in, while odd parcels still go by hand and cost the most.*

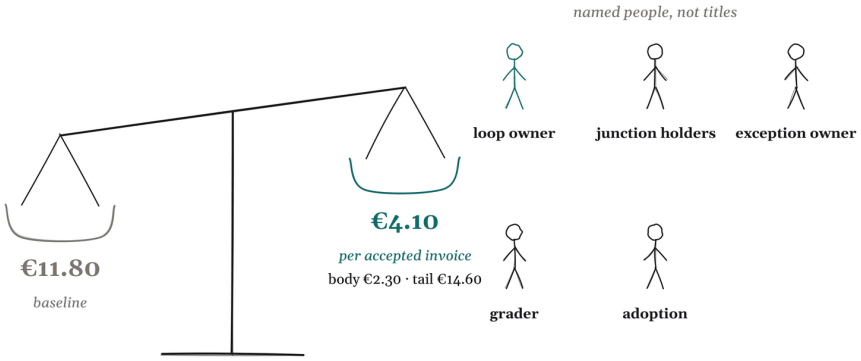


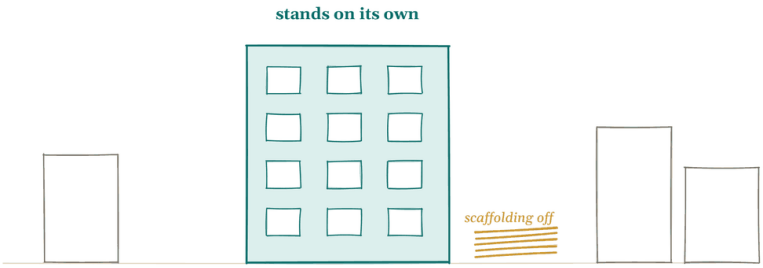
FIGURE 67 Step 8, Own. Named people fill every role the loop needs, and the cost of one accepted outcome is weighed against the measured baseline, body and tail apart.

LXXV

Step 9: Core

The step at a glance	
Question	Does the loop run as part of the business, and would we know if it stopped working well?
Starts when	Gate G3 is passed
Signed	The operating review, each month, and the switch-off plan
Signer	The loop owner
Exit gate	Gate G4 passed: all twelve nodes signed, the first operating reviews held, the switch-off tested on schedule
Typical duration	Ongoing; G4 is usually confirmed after the first two or three monthly reviews

Core is the step where the loop stops being a programme and becomes how the work is done. Its full scope is switched on. The old process is retired. From here on, the loop is run, measured and reviewed like any other part of the business. The difference is that an executor that decides is doing much of the work, so the reviews have to watch for things an instruction-follower would never do.

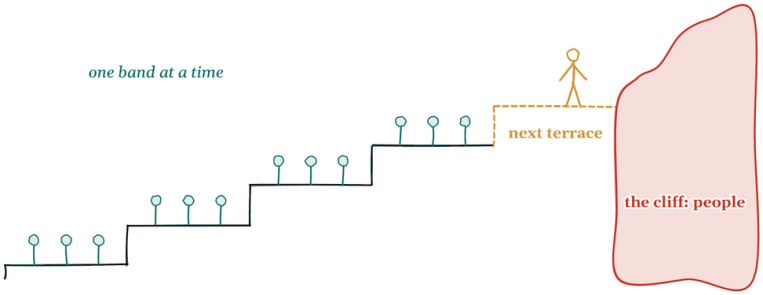


Like a finished building with its scaffolding packed away: *it now stands on its own and is looked after like every other building on the street.*

Widening the scope

Core usually begins by widening the limits set at Bound, one band at a time. Each widening is small, evidenced and recorded. It might mean larger values, newer suppliers, other case types. A widening is a gate in miniature, with the control owner's agreement. The tail stays with the junctions. Core does not

mean the executor decides everything. It means the loop, people included, runs the whole outcome.



Like a vineyard adding one surveyed terrace a season: *each new step is checked before planting, and the steep cliff above is left to people.*

Drift

Every loop that runs long enough will drift. Its inputs change: a supplier redesigns its invoices, a new product line appears, a regulation changes a tax rule. The model behind the executor may be updated by its provider. People at the junctions get used to agreeing with the evidence and stop reading it closely. None of these shows up as an error message. They show up as a slow change in the share of outcomes that meet the standard.

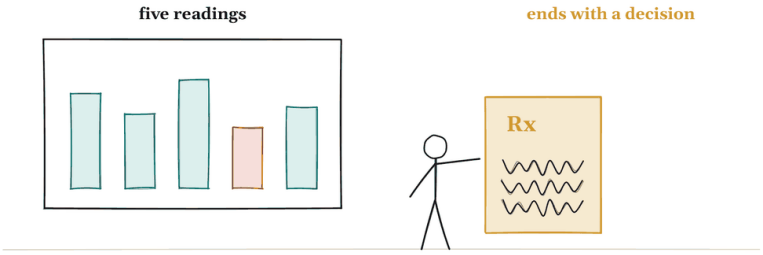
Drift is caught by grading every outcome, as the standard requires, and by a weekly sample in which a person compares a handful of the executor's results with their own judgement. The grading shows when quality moves. The sample shows when the grader itself has started to miss things.



Like a clock that loses a minute a week: *it never stops or beeps, so only setting it against the radio time signal shows the slow error.*

The operating review

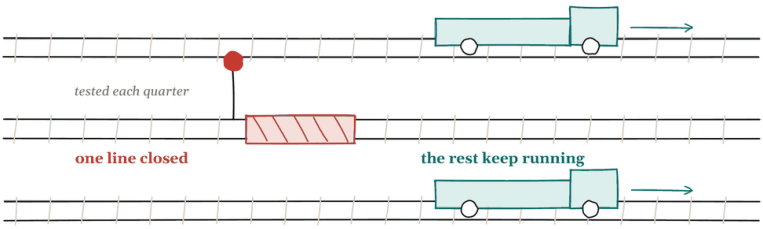
Once a month, the loop owner holds an operating review. It looks at five things. The first is the share of outcomes meeting the standard, body and tail separately. The second is the cost per accepted outcome. The third is the exception queue, including the oldest case and the repeated exceptions that should become rules. The fourth is the junction decisions, including how often junction holders overrode the executor's evidence. The fifth is any drift found. Each review ends with decisions, which are recorded. A review that ends without decisions is a status meeting.



Like a monthly check-up: *five readings are taken, and the visit is worth having only if it ends with something written on the prescription pad.*

The switch-off plan

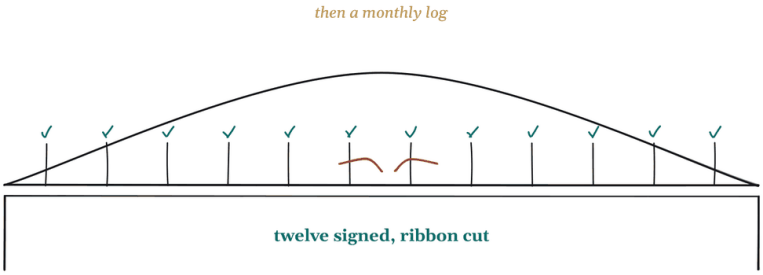
The switch-off was tested once at Bound. In Core it becomes a plan. It records who can stop the loop, how, how quickly cases return to people, and what happens to cases in flight. It is tested on a schedule, usually every quarter. It is also used, without hesitation, when drift or an incident makes the executor's results untrustworthy for a part of the work. Switching off a part of the scope for a few days is normal operations, not failure.



Like a railway closing one line for repairs while trains run on the others: *it is practised every quarter and done without hesitation when needed.*

What gets signed

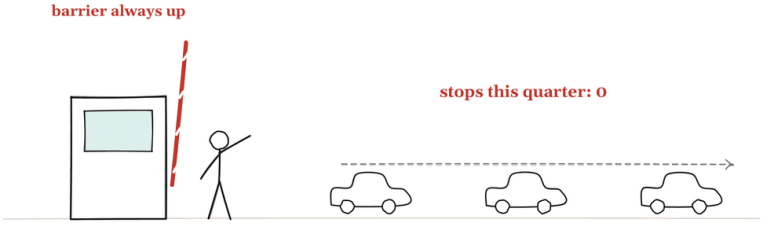
The loop owner signs each monthly operating review and the switch-off plan. With all twelve canvas nodes signed and the first reviews held, the loop passes G4. That is the formal end of the embedding. The loop is in the core.



Like a new bridge at its opening: *all twelve inspections are signed, the ribbon is cut, and from then on a monthly log keeps it in service.*

How it fails

The first failure is declaring victory at go-live. The programme team disbands, the reviews stop after two months, and nobody notices the drift until an auditor does. The second is letting junctions become rubber stamps. Junction holders approve the executor's recommendation every time, so the junction exists on paper only. Overrides near zero for months are a warning, not a success. The third is never using the switch-off, even when it is clearly needed, because switching off feels like admitting failure. The fourth is widening the scope without recording it, until nobody can say what the loop is allowed to do.



Like a checkpoint guard who waves every car through: *the barrier exists on paper, and months with no stops are a warning sign, not a success.*

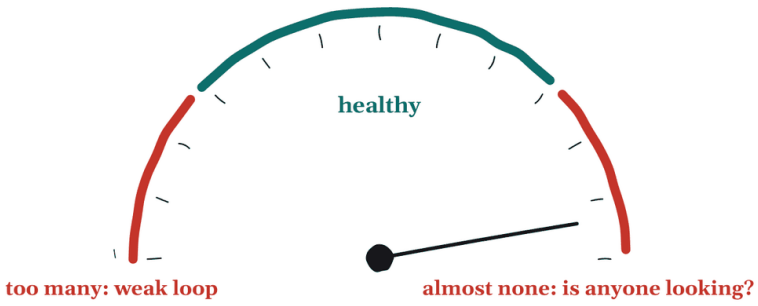
What the foundations add

Core is the closed loop running. Norbert Wiener described it in 1948: act, sense the result, compare it with the goal, adjust. Every outcome is graded

against the standard, the grades are read at the monthly review, and the review changes rules, limits or context. A loop that acts without that cycle is running open, however well it was designed.

Drift is what open loops miss. The world the loop reads keeps moving: suppliers change formats, volumes shift, a new regulation adds a check. A learned model trained on the old distribution goes on answering confidently in the new one. Drift checks compare today's cases with the ones the loop was proven on.

The engineer Lisanne Bainbridge named the ironies of automation: the more reliable the automation, the less practised the people who must step in when it fails. Override rates are therefore a health signal in both directions. Too many overrides mean the loop is weak. Almost none, for months, can mean people have stopped looking.



Like an engine gauge red at both ends: *too many overrides mean a weak loop, and almost none for months can mean nobody is looking.*

At the manufacturer

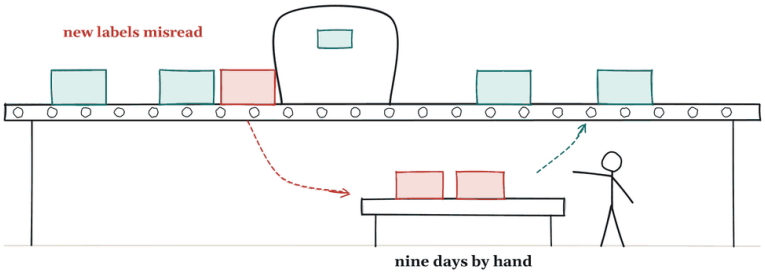
The full scope went live in month eight. Non-order invoices under €5,000 were added, with the cost-centre coding as a bounded decision. New suppliers came into scope after three months of history instead of six. Invoices over €50,000 stayed with the controller, and bank-detail changes with treasury.

In its third month in Core, one of the company's largest suppliers changed the layout of its invoices without warning. The executor began misreading the order reference on about 1.6% of that supplier's invoices. The grader flagged a drop in quality for that supplier within two days. The head of accounts payable routed all of that supplier's invoices to the clerks for nine days while the reading was corrected and re-tested. Then they were returned to the loop. The operating review that month recorded the event, the response and one new

rule: any supplier whose quality drops below the standard for two days in a row is routed to people automatically.

The monthly reviews also tracked junction overrides. The controller overrode the executor's evidence on about one large invoice in twenty, usually because of information the executor could not see, such as a phone call with the supplier. The review treated that rate as healthy and asked for those reasons to be logged, so the context node could be improved.

The head of accounts payable signed the switch-off plan and each operating review. After the third review, all twelve nodes were signed, and the loop stood at G4.



Like a grocer whose scanner misreads one supplier's new labels: *those crates go to the hand-check bench for nine days, then back onto the belt.*

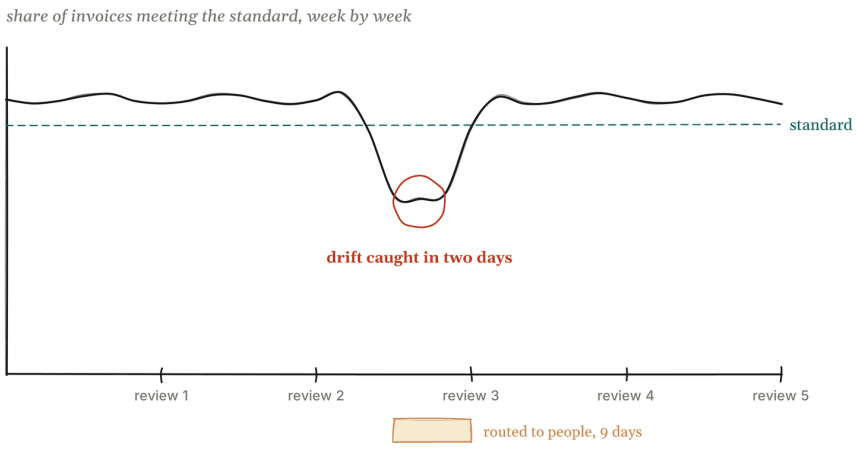


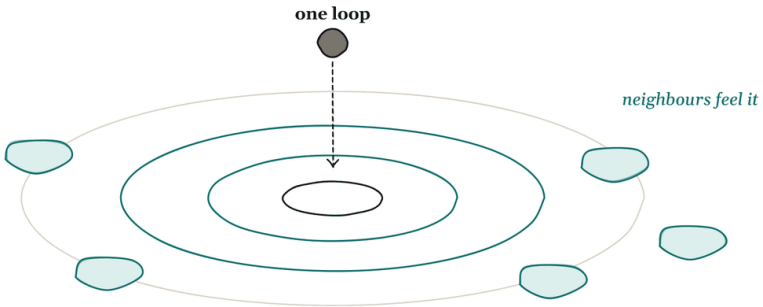
FIGURE 68 Step 9, Core. Quality is graded on every outcome. When a supplier changed its invoice layout, quality for that supplier dipped, the grader caught it, and its invoices went to people for nine days.

LXXVI

Step 10: Scale

The step at a glance	
Question	What does this loop change across the map, and which loop comes next?
Starts when	At least one loop has passed G3, and usually G4
Signed	The map review and the role change plan
Signer	The executive sponsor
Exit gate	The map updated, the next loops chosen, the changes to roles agreed; the next Read begins
Typical duration	A quarterly review, about two weeks of preparation

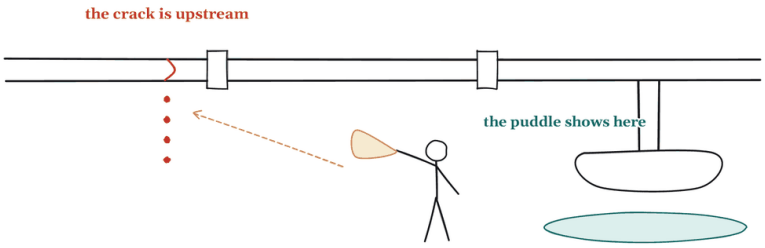
The last step looks up from the single loop to the whole map. Embedding one loop changes things around it. Work that was waiting for that loop now arrives sooner. Problems that loop used to absorb now show up at its edges. People have moved. And the company has learned how long each step really takes. Scale turns those changes into decisions about what comes next.



Like a stone dropped into a lily pond: *the ripples spread from the one spot and reach every neighbouring pad, changing what sits around it.*

Reading the edges

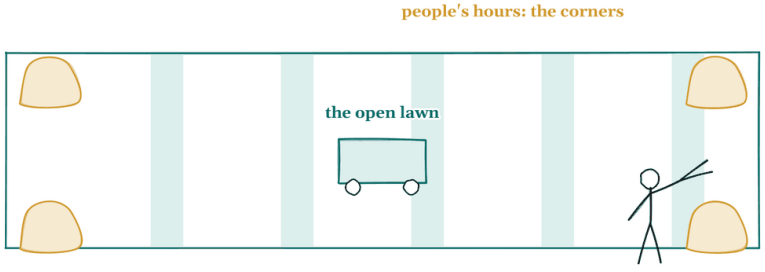
The best guide to the next loop is the handoffs of the last one. A loop running in the core makes its neighbours visible. The exceptions it cannot clear usually point to a loop upstream that sends it incomplete work. The waiting it can no longer explain usually sits in a loop downstream. Reading the exception queue by cause is often the fastest way to see which loop should go next.



Like a plumber tracing a puddle: *the water pools under the sink, but following the drips back points to the cracked pipe upstream.*

Re-reading the distribution

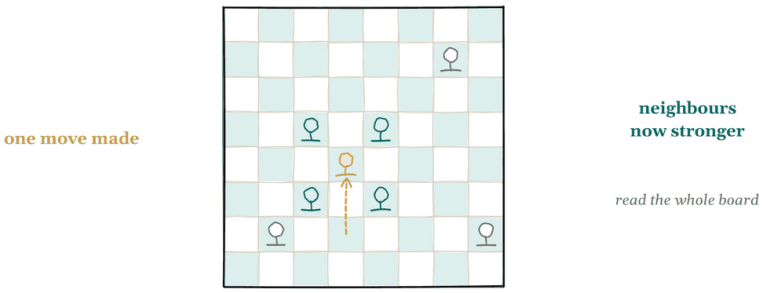
Scale also returns to the distribution. Once an executor runs the body of a loop, the people's time concentrates in the tail. That changes what the company should improve next. It may be the junction evidence, so tail decisions are faster. It may be a new loop for a class of tail cases. Or it may be a decision that the tail is exactly where people should stay.



Like a garden after a robot mower takes the open lawn: *the gardener's hours move to the edges and corners, so better shears now matter most.*

The map review

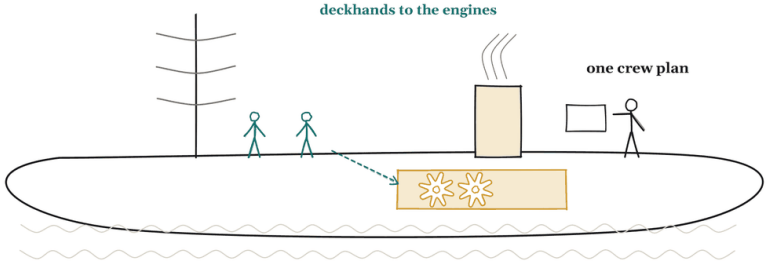
The executive sponsor holds a map review, usually each quarter. It updates every loop's readiness, since the work on one loop often raises the readiness of its neighbours. It records the state of each lifecycle in progress, step by step. And it chooses the next loops to Read. The review is signed by the sponsor. It is the same document that began the lifecycle at Step 1, one cycle later.



Like a chess player studying the whole board after each move: *one move changes its neighbours' strength, so the next move is chosen from the new position.*

The role change plan

Every loop that reaches Own changes some people's work. Across a map, those changes add up, and they need a plan larger than any single loop. The role change plan records which roles are growing, such as junction holders, exception owners, graders and business engineers. It records which are shrinking, where people are moving, and how expertise will be rebuilt now that routine cases no longer teach it. It is signed by the sponsor, because it commits the company, not just one team.

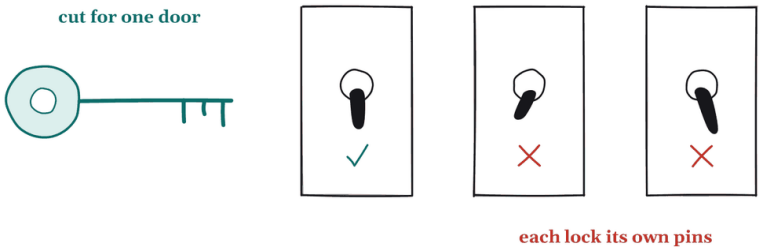


Like a ship refitted from sail to engine: *deckhands become engineers, and the captain plans the whole crew's moves, not one mast at a time.*

How it fails

The first failure is never coming back to the map. The first loop succeeds, the team moves straight to the next project someone suggests, and the company ends up with a scattering of embedded loops that do not connect. The second is scaling by copying. The design of one loop is applied to another without running the lifecycle, on the theory that they are similar. They rarely are. Each loop's rules, decisions and distribution are its own. The third is leaving the role

changes to each loop, so that across the company nobody sees the whole shape of the change to people's work.

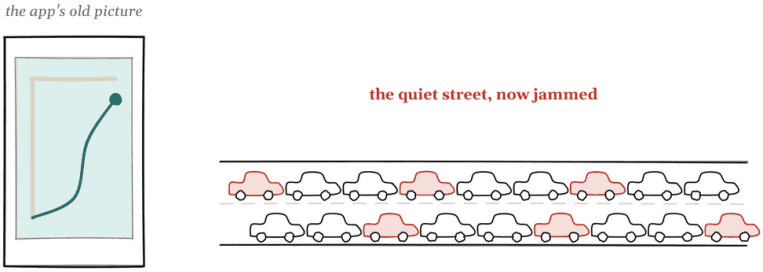


Like cutting one key and trying it in every door: *the first lock opens, but each of the others has pins of its own.*

What the foundations add

A running loop changes the distribution it reads. The financier George Soros called this reflexivity: actions based on a picture of the world alter the world the picture describes. Once the body of invoices runs alone, suppliers learn which invoices clear fastest, the tail shifts, and some old exceptions disappear while new ones appear. The reading made at Step 1 is now out of date, partly because the loop succeeded.

So Scale re-reads before it moves on. Hofstadter called a system that models itself and acts through the model a strange loop. A company that embeds AI becomes one: it builds a model of its work, acts through it, and the action changes the work. The loop stays honest only if the model is redrawn from the real work each time, not from the previous model. **That is what turns one embedded loop into a capacity to embed.**



Like a traffic app that sends everyone down the quiet street: *the street fills, so the picture must be redrawn from the real traffic each time.*

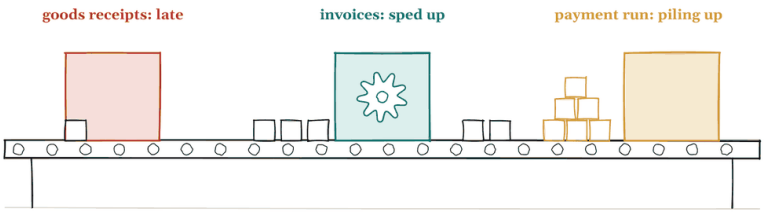
At the manufacturer

The first map review came in month twelve. Reading the invoice loop's exception queue by cause showed that missing goods receipts caused 38% of the exceptions that remained. The receiving teams at the three plants still confirmed deliveries late, or not at all. The goods receipt loop, already marked on the map, moved to the top. It entered Read the following month.

The payment run was chosen second. With invoices ready in three days instead of sixteen, the weekly payment run had become the constraint between an approved invoice and a paid supplier.

Re-reading the invoice distribution showed where people's time now went. The 3% of invoices over €50,000 took more than half of the remaining hours, mostly at the controller's junction. The review asked for better evidence on large invoices, starting with contract terms and delivery history in one view.

The role change plan recorded the moves from the invoice loop and planned the next ones. The receiving teams' work would change with the goods receipt loop. Two of the former clerks, now in supplier onboarding, would help map it. The CFO signed the map review and the plan, and the lifecycle began again at Read.



Like an assembly line after one station is sped up: *work now piles up at the next station, and the late parts from upstream become obvious.*

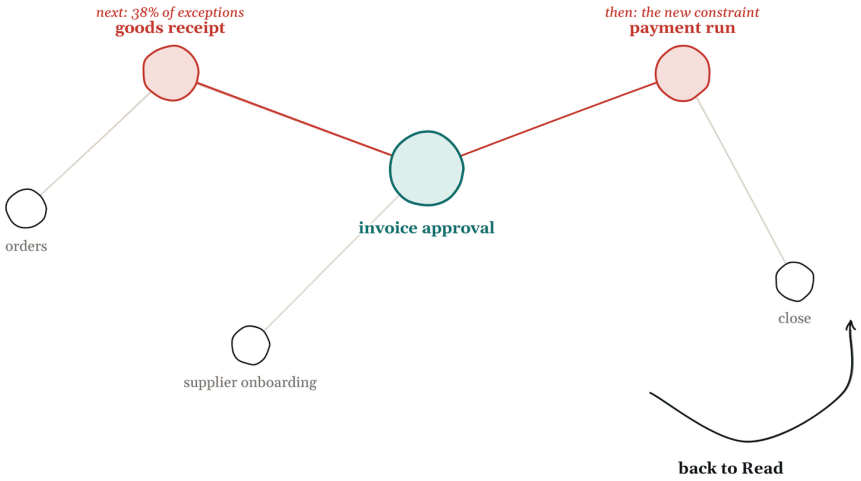
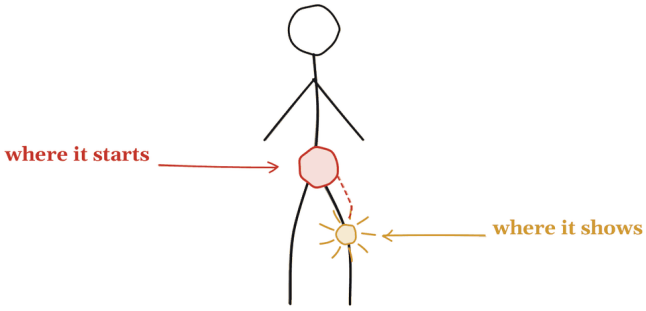


FIGURE 69 Step 10, Scale. The embedded loop lights up its neighbours: missing goods receipts point upstream, the payment run waits downstream, and the map review sends the lifecycle back to Read.

LXXVII

How Lifecycles Break

Each step chapter listed its own failures. This chapter steps back and looks at the patterns, because most stalled programmes fail in one of a small number of ways. Each failure has a step where it could have been prevented, and that step is rarely the one where the failure shows.

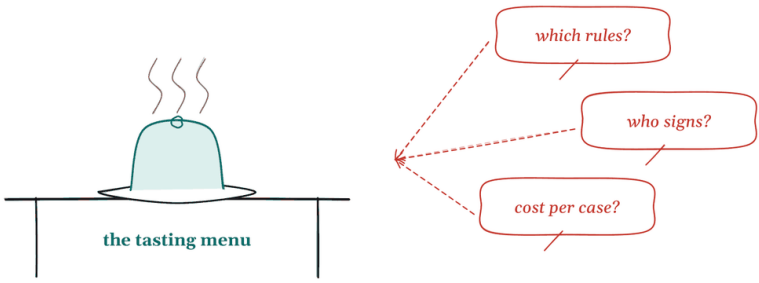


Like referred pain: *the knee hurts, but the cause sits in the hip, so each failure must be traced back to the step that could have prevented it.*

Starting in the middle

The most common pattern is a programme that begins at Step 6. A pilot built in a few weeks is put beside the real process, the results look good, and the team prepares to switch it on. Then the questions arrive. Which rules is it following? Who approves the large cases? What happens to the cases it cannot handle? What does it cost per accepted result, against what baseline? None was asked earlier, so all are asked at once, at the point where every answer delays launch.

The prevention is in Steps 1 to 5. A pilot is valuable evidence that the executor can do the task. It belongs inside Shadow, after the work of Read, Map, Strip, Sort and Design has been done.

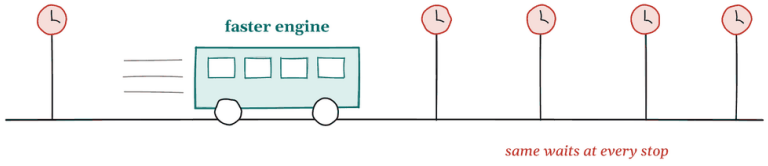


Like a restaurant that serves its tasting menu before getting its licences: *the food wins praise, then every question arrives on opening night and each one delays it.*

Automating the workaround

The second pattern is a loop that works and saves little. The executor runs every step of the old map faster. The waits between steps stay where they were, and the elapsed time barely moves. Often the executor is diligently re-keying data between two systems that could have been connected, or completing a review that guarded against an error nobody makes any more.

The prevention is Step 3. Strip before you sort, and sort before you design.



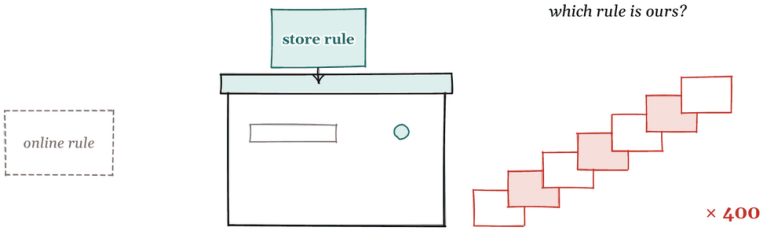
Like a faster engine in a bus that still calls at every old stop: *the drive between stops shrinks, but the waits remain and the journey barely shortens.*

The incident that was a decision

The third pattern is an incident that turns out not to be a technical failure. The executor paid a disputed invoice, refunded a customer outside policy, or accepted a claim that one manager would have rejected. Investigation shows that it followed one of two rules that the company had been applying inconsistently for years. The executor did not create the disagreement. It made the disagreement visible, at scale, in the ledger.

The prevention is Step 4, and specifically the contested-decision log. Every contested decision found and settled before Design is an incident that does not happen.

A composite case shows the pattern. A retailer embedded its returns-and-refunds loop. Within a month, the executor had approved several hundred refunds on items bought outside the return window, from customers with long purchase histories. The investigation found two rules in use. Store managers had always made goodwill exceptions for loyal customers, while the online team never had. The executor had learned from the store data. Nobody had ever decided which rule was the company's, and the refunds were the first time anyone had to.



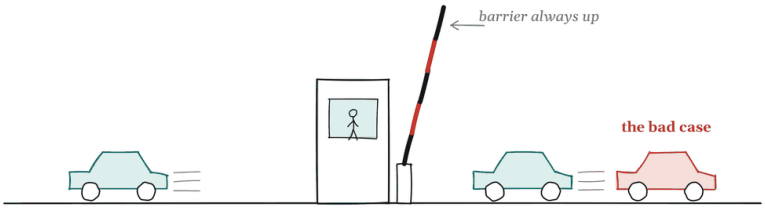
Like a photocopier fed one shop's note: *an exception made quietly for years comes out by the hundred, and someone must finally decide the company's rule.*

Junctions that do not decide

The fourth pattern appears months into Core. Junction holders approve the executor's recommendation almost every time, often without opening the evidence. The junction exists on the canvas, but nobody is really deciding. When a bad case arrives, it passes the junction as easily as the good ones.

The prevention is partly in Step 5, which designs what evidence a junction holder sees. It is partly in Step 9, whose operating review tracks override rates and treats a rate near zero as a warning.

A composite bank met this pattern in its customer onboarding loop. Compliance analysts held the junction for high-risk applicants. After six months, they were approving 99.6% of the executor's recommendations, often in under a minute. A sample review found that the evidence screen showed the executor's conclusion first and the source documents three clicks away. The screen was redesigned to show the documents first and the recommendation last. Override rates rose to a level the head of compliance considered credible.



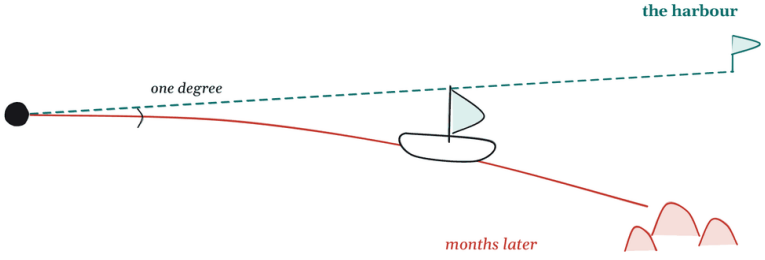
Like a toll barrier stuck in the raised position: *the booth is staffed, yet every car drives straight through, the bad one as easily as the rest.*

Drift nobody watched

The fifth pattern is slow decline. Quality falls by a point or two a month as inputs change, until a customer, a regulator or an auditor notices. The programme team has moved on. The operating reviews became status meetings and then stopped.

The prevention is the standard from Step 5, which grades every outcome, and the operating review from Step 9, which reads the grades body and tail separately every month.

In a composite insurer, a claims intake loop ran well for a year. The model provider then released an update, and the loop began classifying a small share of water-damage claims as maintenance issues. The overall quality figure barely moved, because the share was small. The tail did move: larger claims were being routed wrongly. The insurer had stopped reading body and tail separately after the first quarter. A broker's complaint found the problem, four months late.

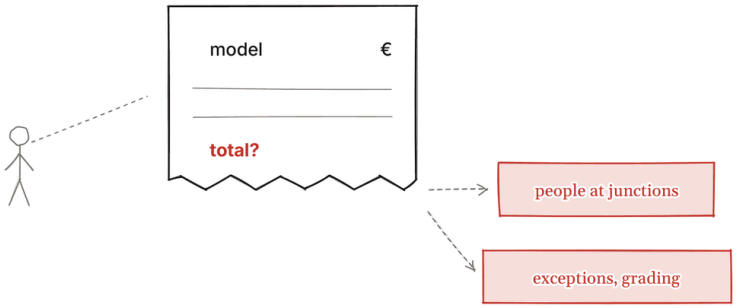


Like a ship one degree off course: *nothing looks wrong on any single day, and only regular checks of position against the chart catch it before the rocks.*

Savings that were never measured

The sixth pattern is a loop that runs well and still cannot defend its budget. When finance asks what it saved, the answer compares an estimate of the old cost with a partial count of the new one. The model cost is counted, but the people at junctions, the exception queue and the grading are not. Finance does not believe the number, and it is right not to.

The prevention is the baseline at Step 2 and the economics at Step 8, measured the same way, signed by finance.

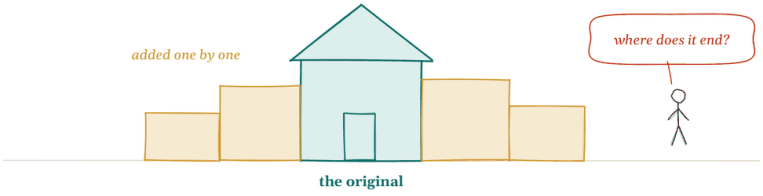


Like a receipt with its bottom torn off: *the model's charge is printed, but people at the junctions, exceptions and grading are missing, so finance rejects the total.*

Scope that crept

The seventh pattern is a loop whose limits nobody can state. It began with small, order-backed invoices. Over a year, case types, value bands and counterparties were added one at a time, each by an informal agreement. When an auditor asks what the executor is allowed to do, the answer takes a week to assemble.

The prevention is the gate review at Step 7 and the rule at Step 9 that every widening of scope is recorded and agreed by the control owner.

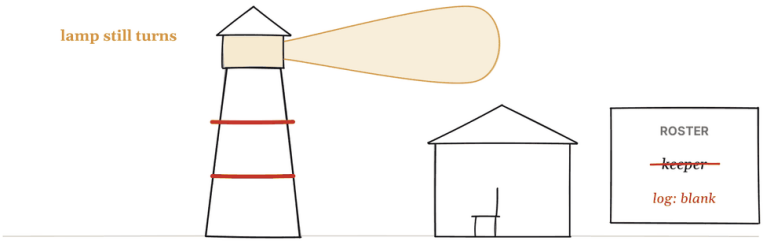


Like a cottage extended one room at a time by handshake: *years later nobody can show the inspector where the approved building ends.*

A loop with no owner

The last pattern is the quietest. The loop runs, but the person who owned it has moved jobs, and nobody took the role formally. Exceptions age. Reviews stop. The switch-off plan names someone who has left.

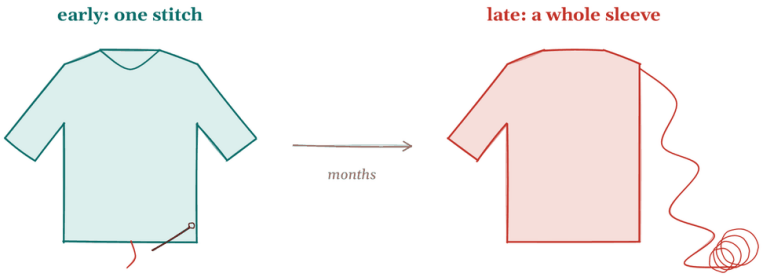
The prevention is Step 8. The charter names people, not roles. And the operating review at Step 9 should check, every month, that every named person is still in post.



Like a lighthouse whose keeper moved away: *the lamp still turns each night, but nobody keeps the log, and the roster still names someone long gone.*

Where the failures cluster

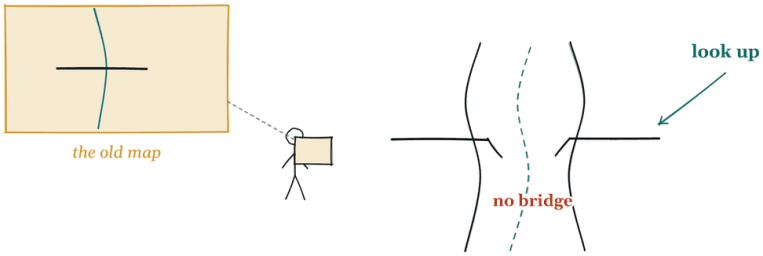
Put these patterns on the lifecycle and a shape appears. Most failures show up late, in Shadow, Bound or Core, where they are expensive. Most are prevented early, in Map, Strip, Sort and Design, where they are cheap. **That asymmetry is the strongest argument for running the steps in order.** The early steps feel slow because their value is in the failures that never arrive.



Like a loose thread on a jumper: *one stitch fixes it early, and left alone it unravels a whole sleeve later, so the cheap early steps come first.*

What the failures have in common

Read through the foundations, the eight patterns reduce to two. In some, the loop runs open: nobody senses its results, so drift, creeping scope and unmeasured savings go unnoticed. In the others, the company's model of its own work stops mirroring the work. The workaround is automated as if it were a rule, the incident turns out to be a decision nobody made, the owner on the charter has left. The cure in both cases is the same. Keep the loop closed, and redraw the model from the real work rather than from the last version of the model.



Like a walker reading an old map instead of the ground: *look up to see where you are, and redraw the map from the path itself.*

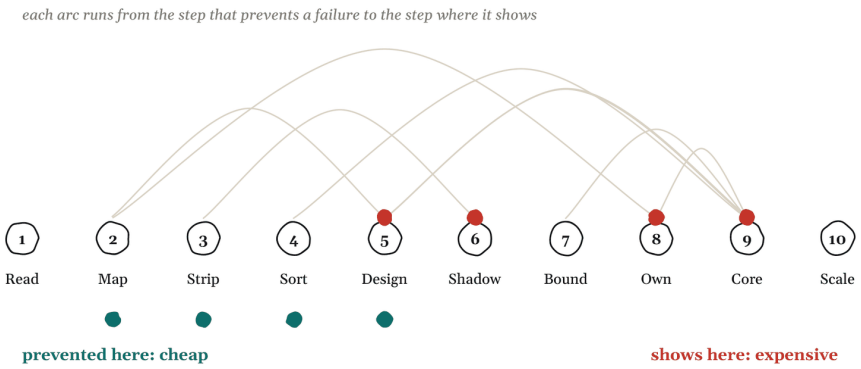
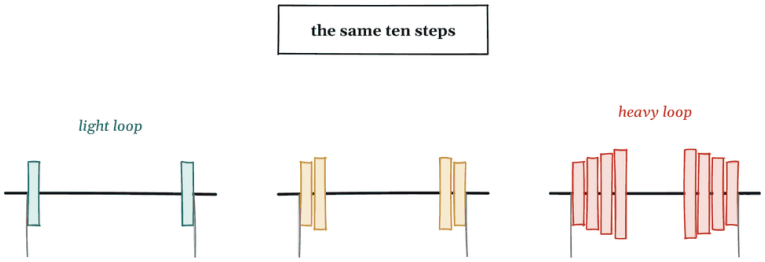


FIGURE 70 Where failures show and where they are prevented. Most failures appear late, in Shadow, Bound or Core, where they are expensive. Most are prevented early, where they are cheap.

LXXVIII

Adapting the Lifecycle

The ten steps hold for every loop. How long each takes, and what it emphasises, depends on the loop. This chapter covers the six adaptations that come up most often. None of them removes a step or a signature. They change the weight each step carries.

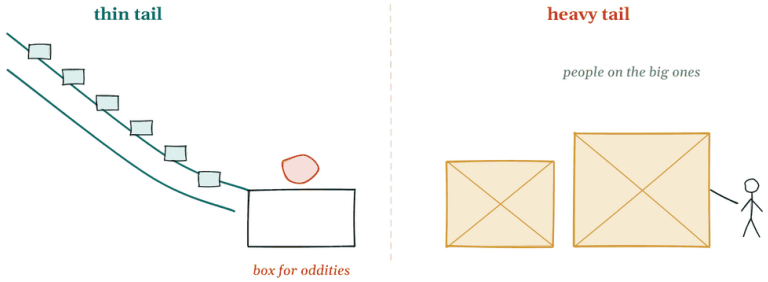


Like one set of exercises for every athlete: *none is skipped, but the weight on each bar is set for the person lifting it.*

By the shape of the distribution

In a loop whose cases are mostly alike, such as standard timesheet approvals or routine address changes, the tail is thin. Shadow can be shorter, because agreement on the body tells you most of what you need. Bound can cover most of the volume from the start, and junctions are few. The risk in these loops is complacency: a rare odd case arrives months later and finds no route to a person. The exception queue matters more than the junctions.

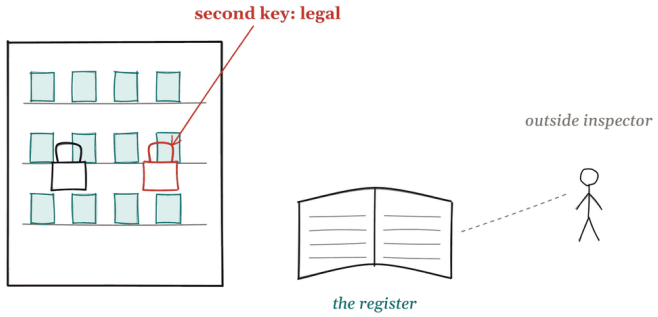
In a loop with a heavy tail, such as large claims, complex contracts or high-value credit, most of the value and risk sit in a few cases. Shadow must run long enough to see enough of them, often two or three months. Bound covers a smaller share of the value, even if it covers most of the volume. The junction evidence deserves the most design effort, because that is where people will spend their time. In these loops the executor's biggest contribution is often preparing the tail cases for people, not deciding them.



Like two post rooms: *one sorts near-identical letters down a chute with a box for oddities, the other gets a few heavy crates that need people.*

Regulated loops

Some loops fall within rules that govern automated decisions. Examples include hiring and managing people, credit and insurance pricing, access to essential services, and anything regulated as high risk under laws such as the European Union's AI Act. For these, two additions are needed. Legal or compliance joins Sort, to confirm which decisions the law requires a person to take or review. And it joins the gate review at Bound, to confirm that oversight, logging and monitoring meet the legal standard, not just the company's own. The junction register and the switch-off plan become legal evidence. Write them so that someone outside the company could follow them.

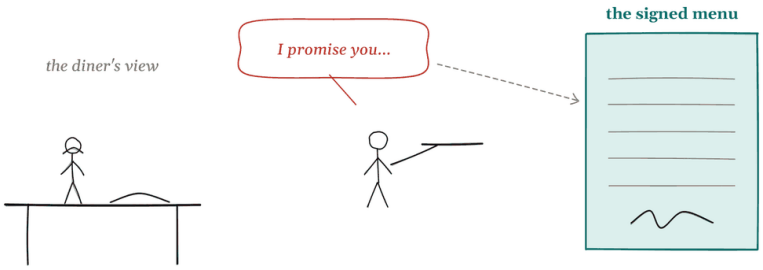


Like a pharmacy's controlled-medicines cabinet: *a second key holder from outside the counter, and a register written so an inspector can follow every entry.*

Customer-facing loops

When the receiver of a loop's outcome is a customer, the standard has to include the customer's view, not only internal correctness. A support reply can be accurate and still fail the customer. Complaints, repeat contacts and

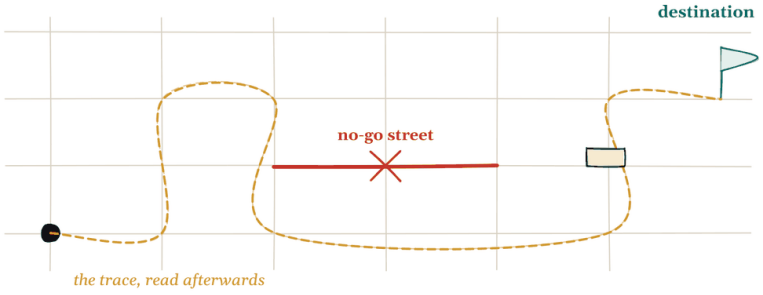
reopened cases become part of the grading. Junctions often sit where the executor would make a promise on the company's behalf: a refund, a delivery date, an exception to policy. A company is generally bound by what its agents tell customers, so the executor's promises must stay inside signed limits.



Like a waiter: *a correct dish can still leave the diner unhappy, and whatever the waiter promises, the restaurant must honour, so promises stay within the signed menu.*

Loops whose steps are generated

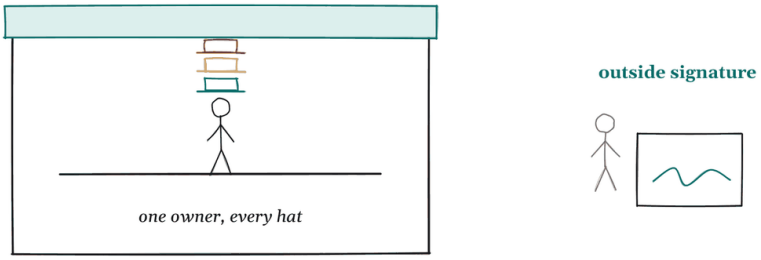
In some loops the executor does not follow a set sequence at all. It plans its own steps for each case: research for a proposal, investigation of an incident, drafting an analysis. Mapping still matters, because it records what the work achieves today and what it costs. But the design moves up. The steps become a trace of what the executor did, not a specification. The canvas carries the weight: outcome, rules, limits on actions, junctions and the standard. Grading every outcome becomes even more important, because there is no fixed path to audit.



Like a taxi with no fixed route: *you set the destination, the fare limit and the streets it may not take, then read its route trace afterwards.*

Small companies

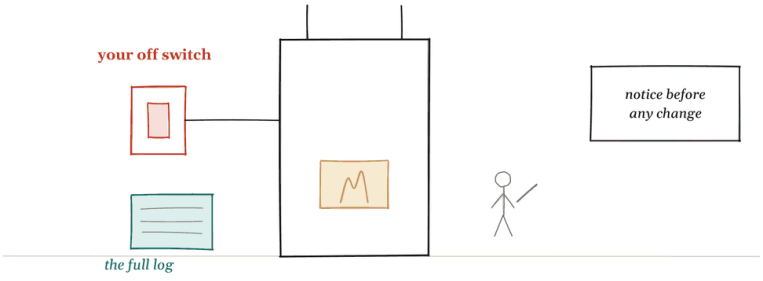
A company of a few hundred people may have one person who is sponsor, process owner and control owner at once. The lifecycle still applies, and the steps are faster because fewer people must agree. One separation should survive even in a small company. The person who answers for the loop's results should not be the only person who signs its limits at Bound. If nobody else is available, an outside accountant or adviser can take the control owner's role for that one review.



Like a corner shop whose owner wears every hat: *the work is quicker, but an outside accountant still signs the limits the owner works within.*

Loops run by a vendor

Some executors are bought as a service: the vendor's platform, the vendor's model, sometimes the vendor's staff at the junctions. The lifecycle does not change, but the contract has to make it possible. The company needs the full log of every case, the ability to switch the loop off, access for its own grader or auditor, and notice before the vendor changes the model. It also needs the baseline and economics measured on the company's own numbers, not the vendor's case studies. A vendor that will not agree to these terms is selling a pilot, not an embedded loop.



Like a contractor running your boiler: *insist on the full service log, your own off switch, access for your inspector and notice before any part is changed.*

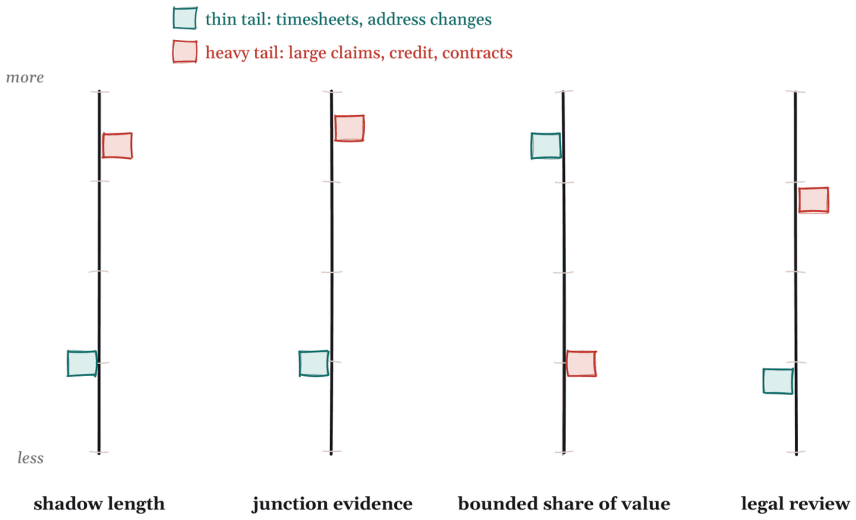
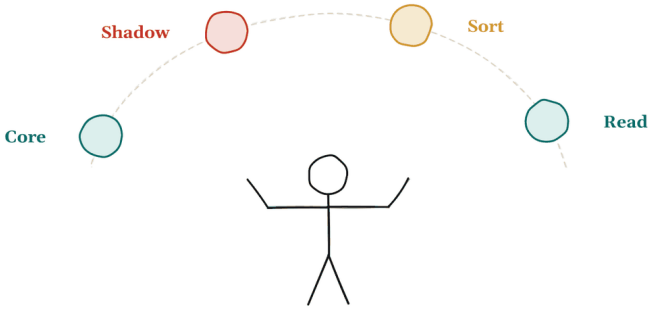


FIGURE 71 Adapting the lifecycle. The steps and signatures stay the same for every loop; what moves is the weight each one carries. A thin-tailed loop and a heavy-tailed loop set the same four controls very differently.

LXXIX

Running Many Loops at Once

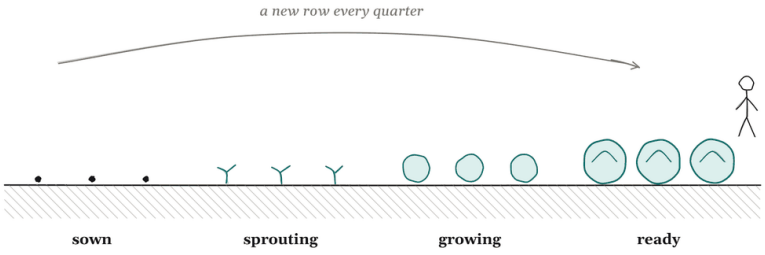
The lifecycle has been described for one loop. A company that takes embedding seriously soon runs several at once, each at a different step. Running many lifecycles is a discipline of its own, and it has a few rules.



Like a juggler: *several balls in the air at once, each at a different point of its arc, which is a skill of its own.*

Stagger the steps

Loops should not move through the lifecycle in lockstep. The steps draw on different people. Read and Scale draw on the executive sponsor. Map and Strip draw on operators. Sort draws on process owners and leaders. Shadow and Bound draw on graders and control owners. Own draws on finance. Staggering the loops spreads the load. A sensible pattern is to start a new Read each quarter, so that at any time one loop is being mapped, one redesigned, one proven and one running.

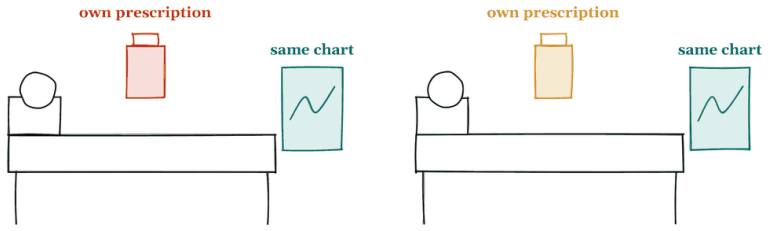


Like sowing a new row of lettuce each month: *one row sprouts, one grows, one is picked, so the gardener's work and the harvest stay steady.*

Share the instruments, not the designs

Every loop uses the same instruments. These are the map, the canvas, the registers, the gate review record, the charter, the economics sheet and the operating review. Shared instruments let leaders compare loops, let auditors find what they need, and let people move between loops without relearning the paperwork.

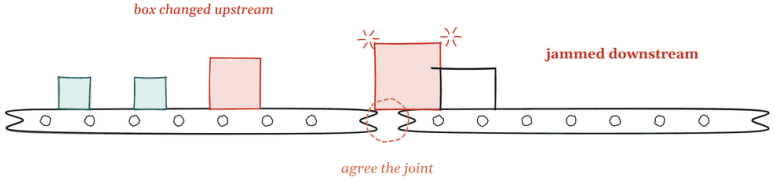
What should not be shared is the design. Each loop's rules, decisions, junctions and distribution are its own. Two loops that look alike on the map often turn out to differ on the one decision that matters.



Like a hospital ward: *every patient gets the same chart and the same thermometer, but each gets a prescription written for their own case.*

Watch the handoffs between loops

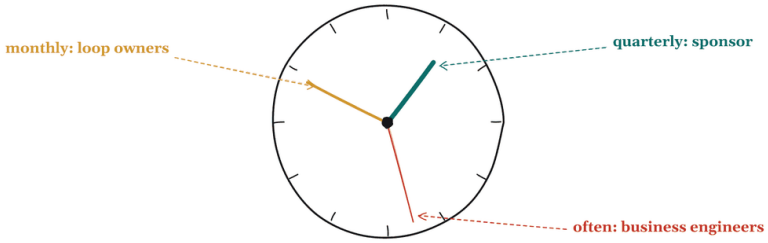
As more loops reach the core, the edges between them matter more than their insides. A loop that sends work to another must agree what it sends, in what state, and who resolves a mismatch. Those agreements live in the handoffs node of each canvas. They should be reviewed whenever either loop changes, because a small change upstream can fill a downstream exception queue overnight.



Like two conveyor belts joined end to end: *a small change to the boxes upstream jams the next machine overnight, so the joint is agreed and rechecked.*

One operating rhythm

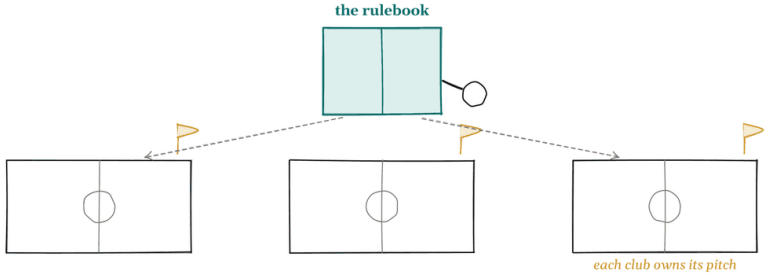
A company running many loops benefits from a single rhythm. Each loop owner holds the monthly operating review for their loop. The sponsor holds the quarterly map review, reading the state of every lifecycle and every loop in the core. The business engineers who run the lifecycles meet more often, compare notes, and keep the instruments consistent. The rhythm is simple, but without it each loop drifts into its own habits, and the company loses the ability to see its embedding as a whole.



Like a clock's three hands: *each turns at its own pace, the quick one, the monthly one, the quarterly one, but all turn on one face.*

The business engineering function

At some scale, usually around the fifth or sixth loop, companies give the lifecycle a home. It can be a small team in transformation, finance or operations. Its job is to run lifecycles, train loop owners, keep the instruments, and hold the map. It does not own any loop; owners do. It owns the method, and it is accountable for the method being followed.



Like a referees' association: *it owns no club and plays no match, but it trains the referees, keeps the rulebook and answers for how it is applied.*

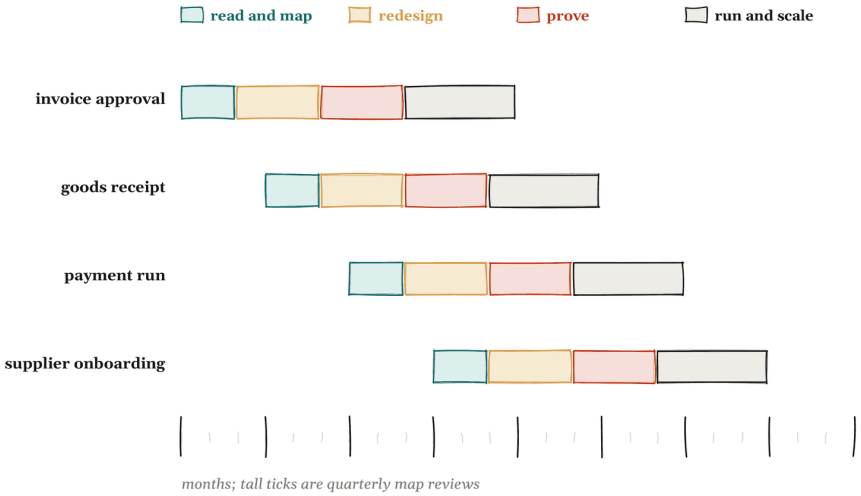
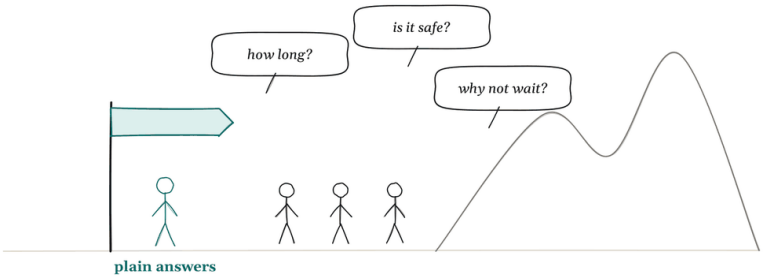


FIGURE 72 Running many loops at once. A new Read starts each quarter, so at any time one loop is being mapped, one redesigned, one proven and one running, all on one operating rhythm.

LXXX

Questions Leaders Ask

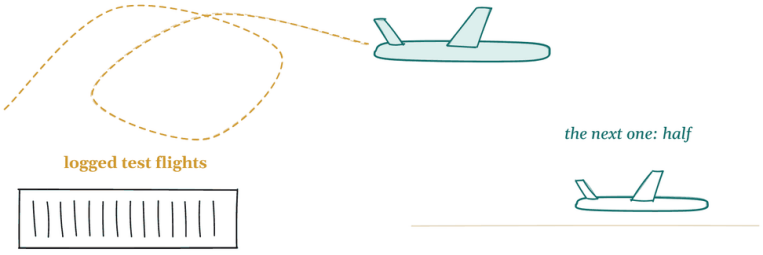
Executives meeting the lifecycle for the first time tend to raise the same handful of questions. Here are the most common, with direct answers.



Like a mountain guide's briefing at the trailhead: *every group asks the same few questions, and a good guide answers each one plainly before setting off.*

"Seven to ten months is too slow."

It is slower than a pilot and faster than most programmes. A pilot reaches a demo in weeks, but the surveys suggest most never reach production at all. The lifecycle's time is mostly spent collecting evidence: weeks of shadow running, weeks of measured economics. Remove it and the evidence is replaced by hope. The second loop takes roughly half as long, and a company running several loops at once sees progress every month, not every year.

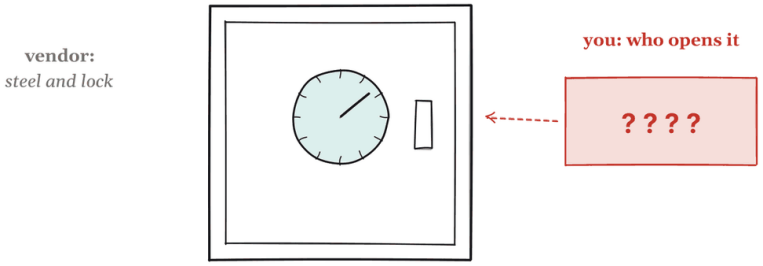


Like test flights for a new aircraft: *weeks of logged flying before passengers board, and the next plane of the same type needs about half as long.*

"Our vendor says its product does all of this."

A vendor can supply the model, the platform and often the grading. It cannot decide which of your rules is right when two contradict, who in your company

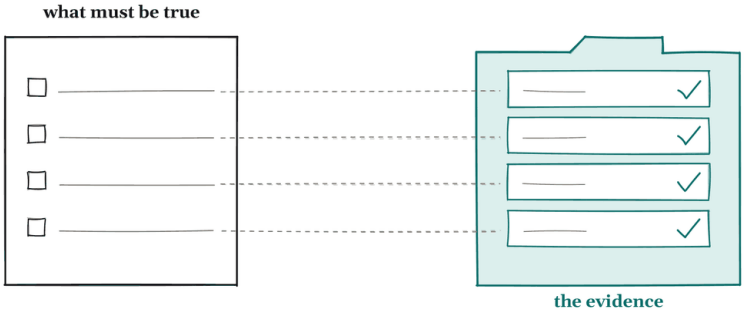
releases a large payment, or whether to pay a disputed invoice to protect supply. Those decisions are the substance of Steps 3 to 8. Ask the vendor which of your signed documents its product produces. The honest answer is none, though a good product makes several of them easier to keep.



Like buying a first-rate safe: *the maker supplies the steel and the lock, but only you can decide who holds the combination and what goes inside.*

"We already have an AI governance framework."

Good. Governance frameworks say what must be true: oversight, risk assessment, monitoring, accountability. The lifecycle is how a particular loop comes to make those things true, step by step, with records. Appendix L maps the two. Companies with a framework usually find the lifecycle supplies the evidence their framework asks for.

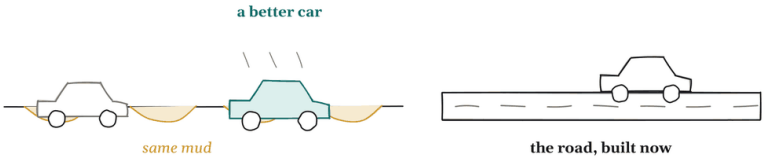


Like an exam syllabus and a coursework folder: *the syllabus says what must be shown, and the folder is the evidence, piece by piece, that it was.*

"Models are improving so fast. Why not wait?"

Better models make the executor more capable. They do not make the company's rules consistent, settle its contested decisions, measure its baselines or name its junction holders. That work is the same whichever model is used, and it takes the same time. A company that waits for better models will meet

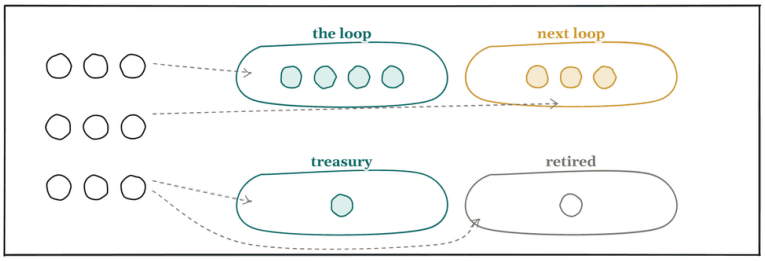
the same middle with a better model. A company that does the work now can swap in better models as they arrive, and its grading will show whether they help.



Like waiting for a faster car before building the road: *the new car sinks in the same mud, while a road built now carries every later car.*

"Will this cost jobs?"

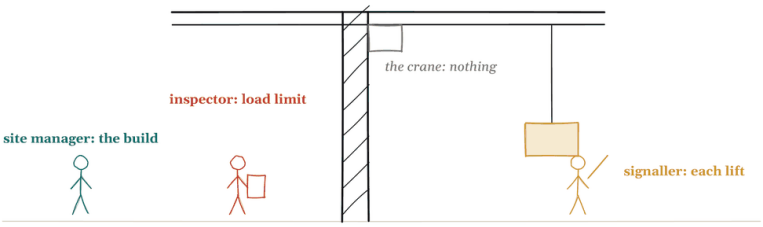
It changes jobs, and in some loops it reduces the number of people doing routine work. At the manufacturer, nine clerks became four on the loop, three moved to another loop, one moved to treasury and one retirement was not replaced. The lifecycle does not decide headcount. It insists that changes to people's work are agreed openly at Own and planned across the map at Scale, and that the company plans how people will learn the judgement the junctions need. Programmes that hide these questions lose the trust of the people whose knowledge they need at Map and Sort.



Like a coach redrawing the tactics board in front of the squad: *of nine players, four keep their places, three move positions, one transfers, one retires.*

"Who is accountable if the executor gets it wrong?"

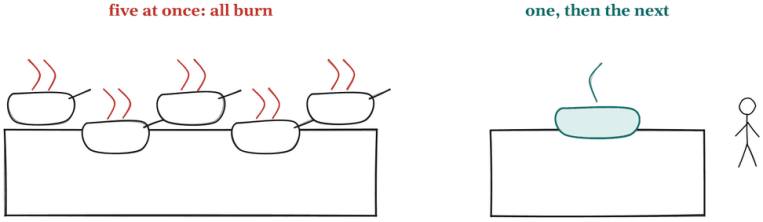
The loop owner answers for the outcome, as they would for any team's work. The junction holder answers for the decisions taken at the junction. The control owner answers for the limits signed at Bound. The executor is never accountable, because it cannot be. The lifecycle exists largely to make this answer clear before the first live case, rather than after the first incident.



Like a building-site crane: *the site manager answers for the build, the signaller for each lift, the inspector for the load limit, the crane for nothing.*

"Can we run several loops in parallel from the start?"

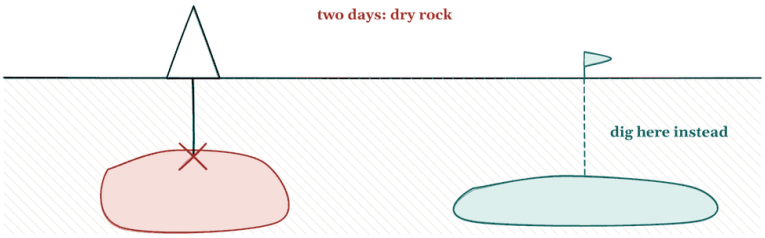
Not usually. The first loop teaches the company the steps, the instruments and how long things really take. Running five first loops at once multiplies the learning cost and usually stalls all five at Sort, where the same leaders are needed for every contested decision. Start one. Start the second once the first passes Shadow. Then start one a quarter.



Like a new cook: *five pans at once all burn while waiting for the same hands, so master one dish, then add a pan at a time.*

"What if a step shows we picked the wrong loop?"

Stop and go back to the map. Finding at Map that a loop's baseline is already excellent, or at Sort that its decisions are mostly contested, is a success of the lifecycle. Two weeks spent learning that is far cheaper than six months spent embedding the wrong loop.

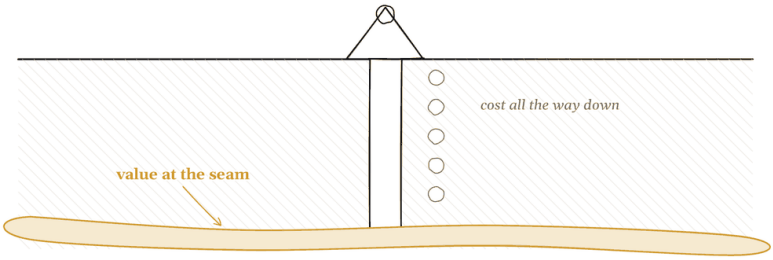


Like a test bore before digging a well: *hitting dry rock after two days is a success, far cheaper than months spent digging in the wrong place.*

LXXXI

What Finishing Is Worth

It is worth being precise about where the value of embedding appears, because it explains both why the middle is so expensive and why the lifecycle pays for itself.

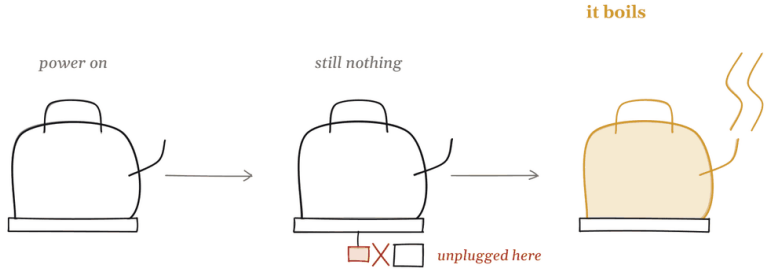


Like a mine shaft: *money is spent all the way down, but the ore lies only at the seam, so know exactly how deep it is.*

Value arrives late

A pilot produces evidence, not value. Shadow produces more evidence and some valuable findings, such as rules nobody knew were inconsistent. But it saves nothing, because people are still doing all the work. The first real savings arrive at Bound, on the body of the distribution. They grow at Core, as the scope widens and the old process is retired.

So the value curve of an embedding programme is flat for months and then rises steeply. A programme that stops anywhere before Bound has spent most of its cost and captured almost none of its value. That is the economic shape of the middle, and it is why so many surveys find so little return. Most programmes are spending money in the flat part of the curve.



Like a kettle: *it draws power for minutes with nothing to show, then boils, and switching off just before the boil buys only lukewarm water.*

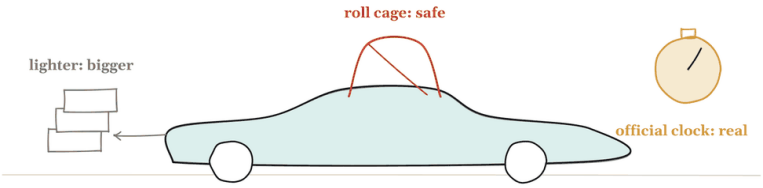
What the steps protect

The early steps look like cost, but they protect the later value in three ways.

They make the savings bigger. Strip removes waits and steps that an executor would otherwise have run faster but kept. At the manufacturer, most of the drop in elapsed time came from the six struck steps, not from the executor's speed.

They make the savings real. A measured baseline and a fully loaded economics sheet mean finance believes the number, so the loop keeps its budget and the next loop gets one.

They make the savings safe. A contested decision found at Sort, a junction placed at Design and a switch-off tested at Bound each prevent a single incident. One such incident can cost more than the loop saves in a year, and can stop the whole programme.

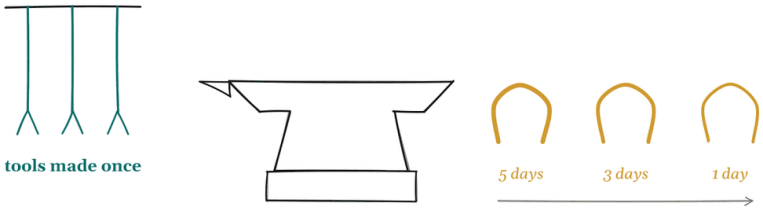


Like preparing a race car: *stripping weight makes it faster, the official clock makes the lap count, and the roll cage stops one crash ending the season.*

The value of the second loop

The first loop through the lifecycle is the most expensive. The map is drawn for the first time, the instruments are new, the people are learning the steps. The second loop is cheaper and faster. By the fourth or fifth, the cost of each lifecycle falls sharply, and the company has something more valuable than any single loop: the capacity to embed.

That capacity is what separates the few companies that report large returns from AI from the many that do not. They did not find better models. They built the ability to move loop after loop through the middle.

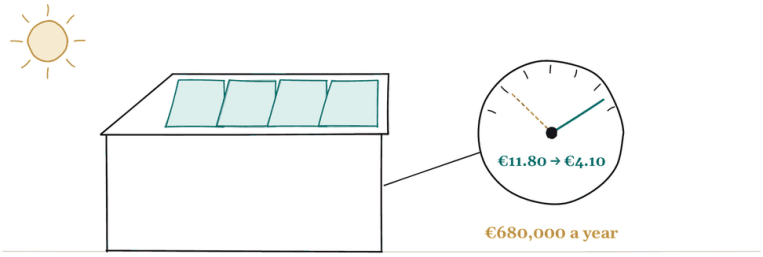


Like a new blacksmith: *the first horseshoe takes days because the tools are made too; by the fifth, shoes come fast and the forge is the asset.*

At the manufacturer

The invoice loop cost about four months of effort before it saved anything. The cost included the time of the clerks and the controller, the business engineer, the platform and the shadow period. At Bound, savings began on three quarters of invoices. At Core, the measured cost per accepted invoice of €4.10 against €11.80 saved the company about €680,000 a year on invoice processing. Paying suppliers on time also let it capture early-payment discounts it had been missing, and it had fewer late-payment disputes.

The CFO's own summary at the first map review was simpler. The invoice loop had paid for itself in its first year of Core. The goods receipt loop would take half as long to reach Bound. And the company now knew how to do this.



Like solar panels on a factory roof: *months of cost before any saving, then each unit costs a third as much, and they pay back within a year.*

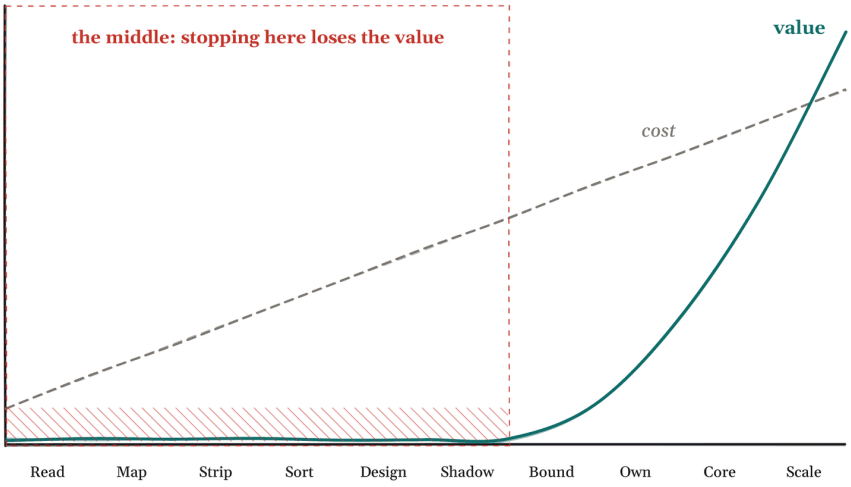


FIGURE 73 What finishing is worth. Cost accumulates from the first step, but value arrives only when the executor acts, at Bound, and grows at Core. A programme that stops before Bound spends most of its cost for almost none of its value.

PART TEN

The Work Loop Library

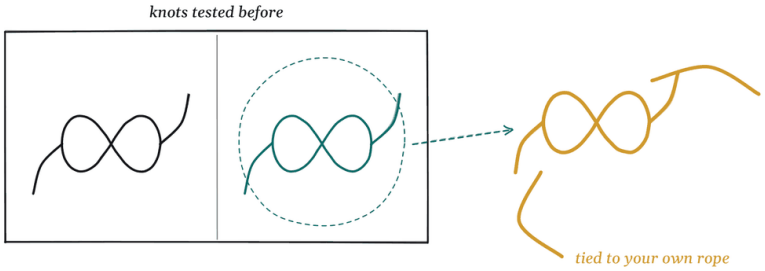
LXXXII

A Library of Work

In 1977, the architect Christopher Alexander and his colleagues published *A Pattern Language*. It described 253 patterns for building towns, streets and houses. Each pattern named a problem that recurs, described the forces behind it, and gave the core of a solution. A window place, a light on two sides of every room, a small public square. None of the patterns was a blueprint. Each was a solution that could be adapted a thousand times without ever being the same twice. And each pattern linked to larger ones it helped complete and smaller ones that completed it.

Software engineers borrowed the idea in the 1990s, and it changed how they designed programs. The reason it travels is simple. Most problems are not new. Someone has met them before, in another building or another company, and what they learned can be written down so that the next person starts further ahead.

This part applies that idea to the work inside a company. It is a library of work loops: the recurring units of work that almost every company runs, from qualifying a lead to reconciling an account. A work loop is one outcome owned end to end, such as an approved invoice or a resolved ticket, together with the steps, rules, decisions and people that produce it. Companies differ in what they sell. They differ far less in how they bill, buy, hire, close their books, answer customers and move goods.



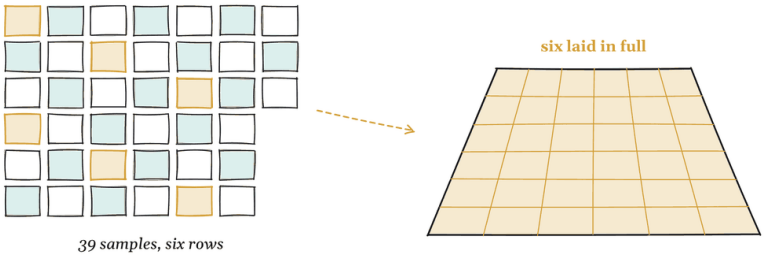
Like a sailor's book of knots: *most problems have been met before, so you pick the tested knot and tie it to your own rope.*

What the library holds

The library covers six chains of work: lead to cash, procure to pay, hire to retire, record to report, issue to resolution and order to delivery. Between

them they hold thirty-nine loops. Each loop has an entry. An entry says what the loop produces and where its work comes from and goes. It describes the shape of the loop's cases and the mechanism that makes that shape. It names the laws and standards that force design choices, and sorts the decisions inside the loop into four classes. It says how the work should split between written rules, a model and people. It gives the measures that matter, with published benchmarks and their sources, and the companies that have embedded AI into the loop, as read in 2026. And it says plainly what is not yet known.

Six of the thirty-nine are then followed through the whole embedding lifecycle, one from each chain. For those six, every document the lifecycle asks someone to sign is shown filled in. A blank form teaches almost nothing. Crafts have long solved this by keeping finished examples: pattern books of whole houses, legal precedents, reference designs. The six filled loops are the library's finished examples.



Like a tile showroom: *thirty-nine sample tiles on the wall in six rows, and six floors laid in full so you can see them finished.*

Why entries are written this way

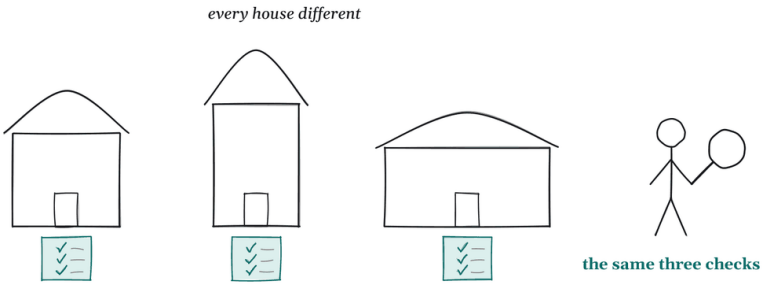
Every entry follows the same lens, because embedding AI into a loop always turns on the same questions. Three come from older fields of knowledge.

The first is the shape of the cases. A loop whose cases are alike can be handled by rules. A loop where a few cases carry most of the value has a routine body and a consequential tail, and the two must be designed differently. The shape is the footprint of a mechanism: many small independent causes give an even spread, while compounding, such as a few large customers or suppliers, gives a heavy tail.

The second is how variety is shared. The cybernetician W. Ross Ashby showed that a regulator must have as much variety as the situations it faces. In a loop, written rules take the cases that repeat. A learned model takes the variety in

between. People take the tail, at junctions placed where an action is costly or cannot be undone.

The third is what the company must write down. A model can be bought. The rules, decisions, limits and standards that describe this company's own version of the loop cannot. Each entry points to where that written half is hardest to build, which is usually a decision the company has never agreed.

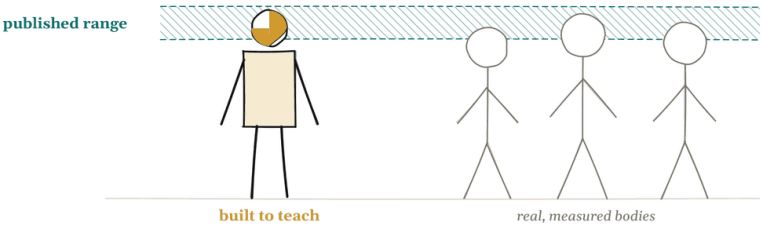


Like a house surveyor who checks every house for the same three things: *how its cases spread, who handles which variety, and what must be written down.*

The company and the evidence

The six filled loops belong to one composite company: a mid-size European maker of industrial pumps and valves, with about 4,200 people, three plants and revenue of about €1.1 billion. Its figures are illustrative. They are authored to teach and are not drawn from any company.

The reference entries are not illustrative. Every figure, law and deployment in them comes from a published source, named in the text. Where no reliable figure is published, the entry says so. The manufacturer's numbers are set within the published ranges, so a reader can see where an illustrative loop sits against the field.



Like a crash-test dummy: *built to teach and belonging to nobody, yet sized inside the measured range of real bodies.*

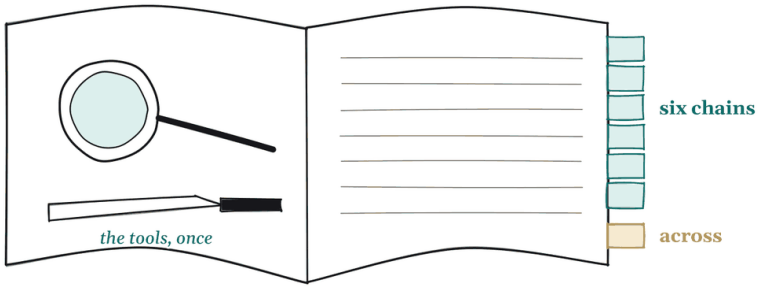
How to use the library

The next chapter describes the template kit: eighteen templates, each with its purpose, the step where it is signed, who signs it, and what to write in every field.

The chains then follow one at a time. Each opens with the chain and its reference data, then gives an entry for every loop in the chain, then follows its featured loop through the lifecycle with every template filled in.

A closing chapter reads across the library. It sets out what thirty-nine loops have in common, where the hard decisions cluster, and how to choose and adapt an entry for your own company.

The templates also come as files: narrative documents as Word files, and the registers and sheets as one Excel workbook, described in Appendix K.



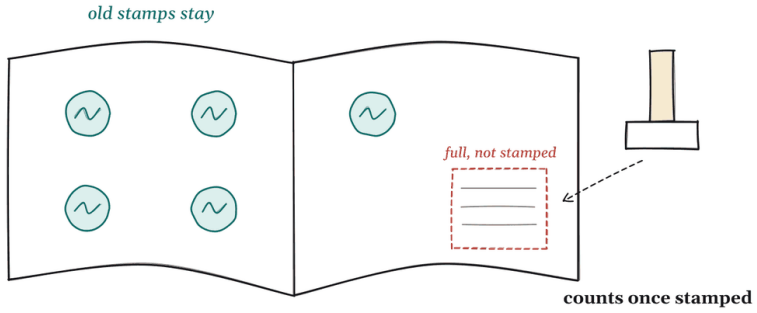
Like a cookbook: *the front pages show the knives and pans once, the chapters go course by course, and the last tab compares whole menus.*

LXXXIII

The Template Kit

The kit has eighteen templates, one for each document the lifecycle asks someone to sign. Each is described here with its purpose, the step where it is signed, who signs it, and its fields with guidance on what to write. The chapters that follow show every template filled in for a real loop. The same templates come as files: narrative documents as Word files, and the registers and sheets as one Excel workbook.

Three rules apply to all of them. A template is finished when it is signed, not when it is full. Every figure written into one is either measured or marked as an estimate. And a template that is signed again after a change keeps its earlier versions.



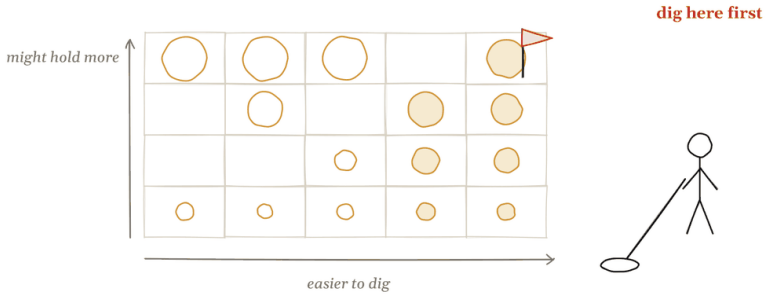
Like a car's service book: *a page counts once the garage stamps it, not when it is filled in, and earlier stamps are never torn out.*

T01. Work Loop Map

Lists the company's loops, places them by value to the core and readiness, and marks the first loop. Signed at Step 1, Read, by the executive sponsor.

Field	What to write
Company or unit	The scope of the map
Loops	One line per outcome, named by the outcome, not the department
Chain	The chain each loop belongs to, such as procure to pay
Value to the core (1-5)	Volume, cost, effect on customers, cash and risk; record the reasons

Field	What to write
Readiness (1-5)	Known rules, measurable outcome, available context, willing owner, settled decisions
Handoffs	Which loop sends work to which
Risk category	Whether the loop falls within rules on automated decisions
First loop	The loop chosen, why, and its owner, who has agreed
Distribution read	Share of cases and of value by band for the first loop
Next to map	Loops to read next

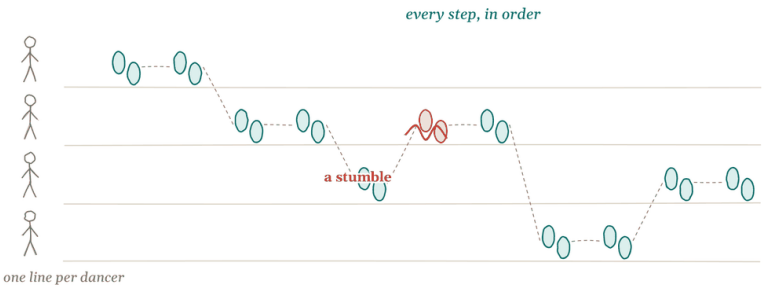


Like a detectorist's grid over a beach: *each square marked by what it might hold and how easy it is to dig, with a flag on the first.*

T02. Process Map

Records how the loop's work runs today, drawn with the people who do it. Signed at Step 2, Map, by the process owner.

Field	What to write
Lanes	Each role, team or outside party
Steps	In order, each in its lane
System or channel	For each step
Hands-on minutes	Sampled, for each step
Wait after (hours)	Time the case waits before the next step
Flags	Workaround or rework
Cases walked through	At least five real cases, one that went wrong

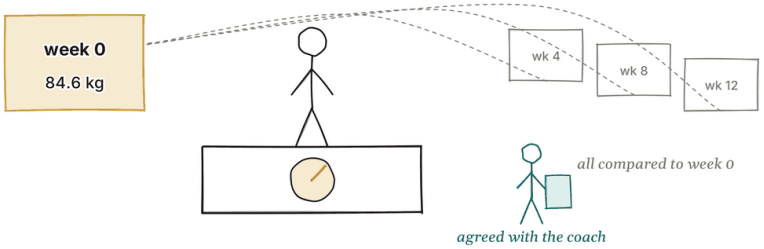


Like a dance notated with its own dancers: *each dancer gets a line, every step is written in order, and the stumbles are marked.*

T03. Baseline Sheet

Measures the loop before it changes; every later claim is compared with it. Signed at Step 2, Map, by the process owner.

Field	What to write
Sample and period	Number of cases and dates
Volume	Cases per month
Fully loaded cost per case	People at real cost, systems, rework, errors
Elapsed time	Arrival to accepted outcome, median and 90th percentile
Exception rate	Cases leaving the normal path
First-time-right	Cases accepted without rework
Flow efficiency	Hands-on time divided by elapsed time
Handoffs	Lane changes per case
Method agreed with finance	Yes or no, and by whom

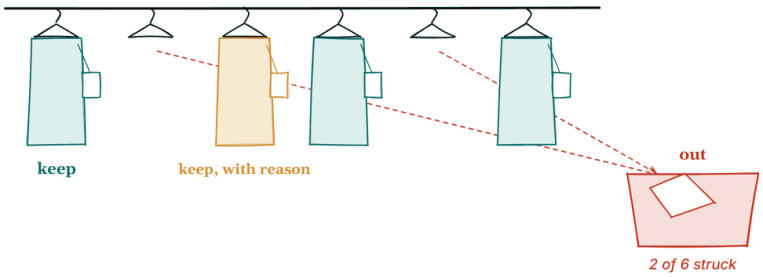


Like the first weigh-in before a diet, agreed with the coach: *every later claim of progress is measured against that one number.*

T04. Strip Register

Records, for every mapped step, whether it is struck or kept, and the constraint behind it. Signed at Step 3, Strip, by the process owner.

Field	What to write
Step	As on the signed map
Status	Struck, kept, or kept with constraint
Constraint	What the step compensates for
Still applies?	Yes or no, with the reason
Where it goes	For kept constraints: junction, standard or step
Strip ratio	Struck steps divided by mapped steps

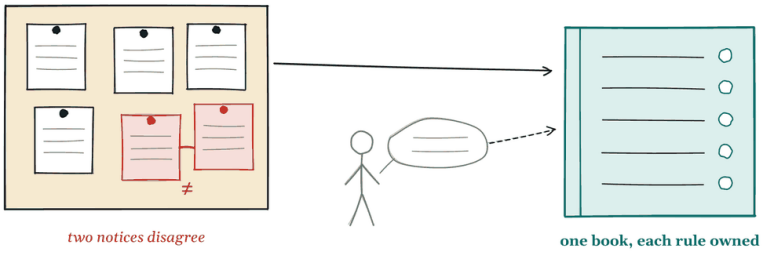


Like clearing a wardrobe with a tag on every hanger: *out, keep, or keep for a stated reason, and a count of what went.*

T05. Rule Register

Collects every rule of the loop in one place, with its source and owner. Signed at Step 4, Sort, by the process owner.

Field	What to write
Rule	In plain words
Source	Document, system configuration, or person
Owner	Who decides the rule
Date confirmed	When the owner confirmed it
Contradiction	The rule it conflicts with, if any
Decision	Which rule stands, decided by whom

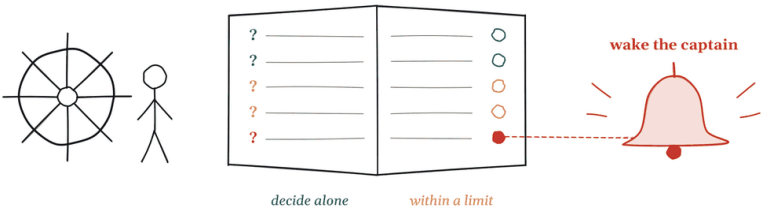


Like a sports club writing one rulebook from pinned notices and the groundsman's memory, with an owner for each rule, and catching two notices that disagree.:

T06. Decision Register

Lists every decision in the loop, as a question, with its class. Signed at Step 4, Sort, by the process owner.

Field	What to write
Decision	Phrased as a question
Class	Mechanical, bounded, consequential or contested
Reason	Why this class
Limit	For bounded decisions, the signed limit
Junction	For consequential decisions, which junction

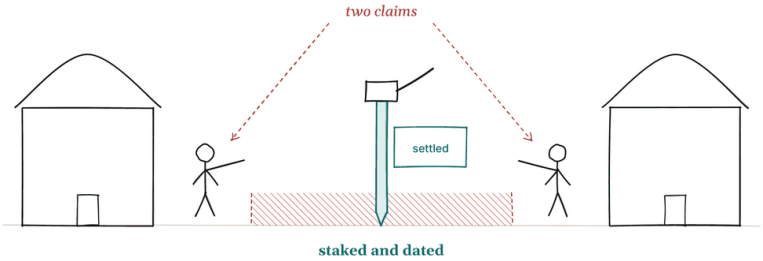


Like a captain's night orders: *each likely question is written down, with what the watch decides alone, what within a set limit, and when to wake the captain.*

T07. Contested-Decision Log

Records each decision the organisation never agreed on, and how it was settled. Signed at Step 4, Sort, by the the leaders who settle each entry.

Field	What to write
Decision	The question
Positions	Who holds which view
Logged	Date found
Settled by	Name and role
Answer	What was decided, with any limit
Date settled	When
Resulting class	Usually consequential with a named junction



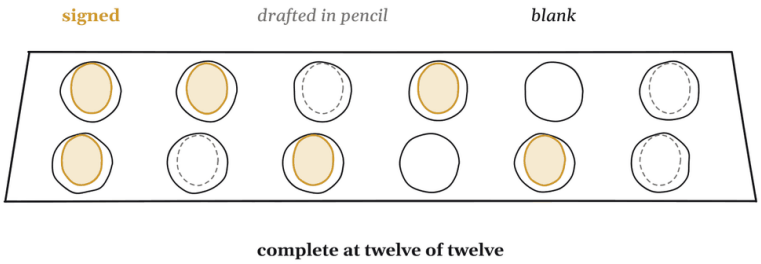
Like a boundary quarrel between neighbours: *each claim is written down, then someone with authority drives in a stake and the date is recorded.*

T08. Work Loop Canvas

Describes the loop on one page in twelve connected nodes, each blank, drafted or signed. Signed at Step 5, Design, by the loop owner.

Field	What to write
Outcome	What one accepted result looks like, and who receives it
Compensations	From the strip register
Rules	From the rule register
Context	Sources the executor reads, each with a keeper
Actions	What it may change, where, and whether it can be undone
Decisions	From the decision register
Junctions	From the junction register
Exceptions	Queue owner, time limit, weekly review

Field	What to write
Handoffs	What comes in and goes out, and who resolves a mismatch
Standard	From the standard
People	From the loop charter
Economics	Baseline and target
Gate	G0 to G4, from the signatures

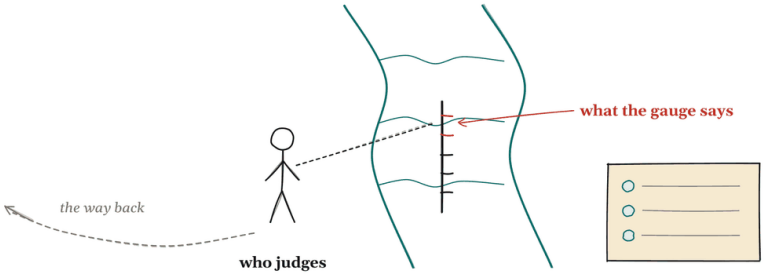


Like a twelve-egg carton: *each cup is empty, holds a pencilled egg, or a real one, and the box is complete when all twelve are real.*

T09. Junction Register

Defines every point where a person decides instead of the executor. Signed at Step 5, Design, by the loop owner.

Field	What to write
Junction	The decision or action
Why	Cost of error or irreversibility
Who decides	Named person
Evidence shown	What the executor puts in front of them, in what order
How reversed	If the decision was wrong
Switch-off holder	Who can stop the whole loop

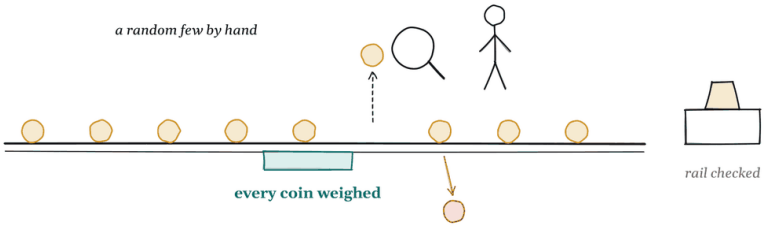


Like a hiker's route card for river crossings: *at each ford it names who judges the water, what gauge they read, and the way back.*

T10. Standard

Defines an accepted outcome and how every outcome is graded. Signed at Step 5, Design, by the loop owner.

Field	What to write
Criteria	What an accepted outcome must satisfy, stated as outcomes
Rule checks	Checked on every case
Model grader	Criteria it grades on every case
Human audit	Share audited at random, and the higher share for the tail
Customer signals	For customer-facing loops: complaints, repeat contacts, reopened cases
Calibration	How often the grader is checked, and by whom

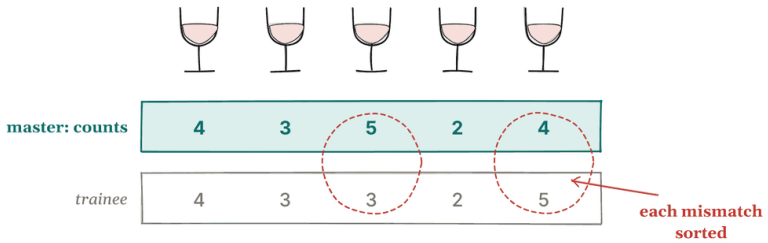


Like a coin mint: *every coin crosses the weighing rail, an inspector pulls a random few, and the rail itself is checked against a test weight.*

T11. Shadow Report

Reports how the executor's results compared with people's on real cases, without acting. Signed at Step 6, Shadow, by the loop owner and grader.

Field	What to write
Period and cases	Dates and number
Agreement, body	Share matching, routine cases
Agreement, tail	Share matching, consequential cases
Disagreements by group	Executor wrong, person wrong, both defensible, standard unclear
Changes made	To rules, context or the standard, with dates
Known weaknesses	What the executor still does badly
Recommendation	Pass G1, extend, or return to an earlier step

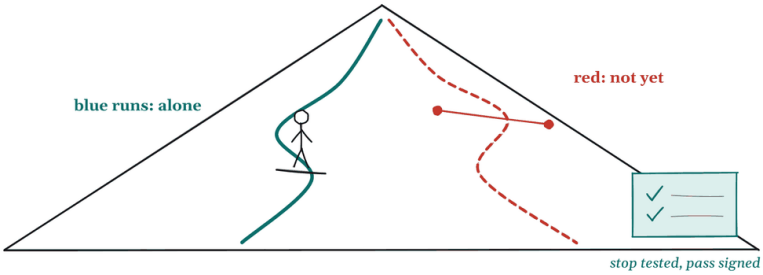


Like a trainee scoring the same wines beside the master taster: *only the master's card counts, and every mismatch is sorted and explained.*

T12. Gate Review Record

Records the decision to let the executor act alone inside written limits. Signed at Step 7, Bound, by the control owner.

Field	What to write
Scope alone	Conditions: value, tolerance, counterparty, type; share of cases and of value
Evidence	Graded results during bounded running
Permissions	Matched to the actions node; anything removed
Switch-off test	Date, time to return cases, cases lost
Audit consulted	Who, and any notes
Decision	Pass G2, with conditions

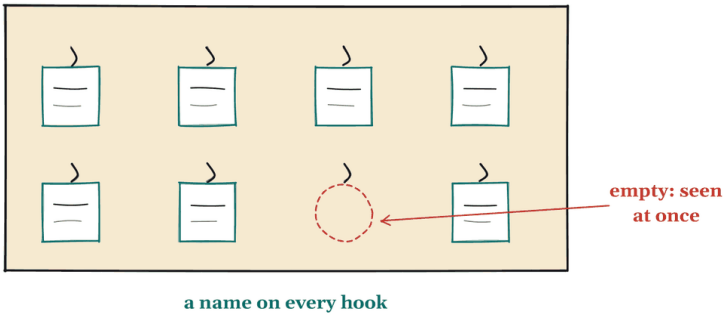


Like a ski school passing a pupil for the blue runs alone: *the stop is tested first, and the pass covers those runs and no others.*

T13. Loop Charter

Names the people in every role the loop needs and records changes to people's work. Signed at Step 8, Own, by the loop owner.

Field	What to write
Loop owner	Name
Junction holders	Names, one per junction
Exception owner	Name
Grader	Name
Adoption owner	Name
Control owner	Name
Changes to work	Who moves where, agreed with whom
Learning path	How new staff learn the judgement the junctions need

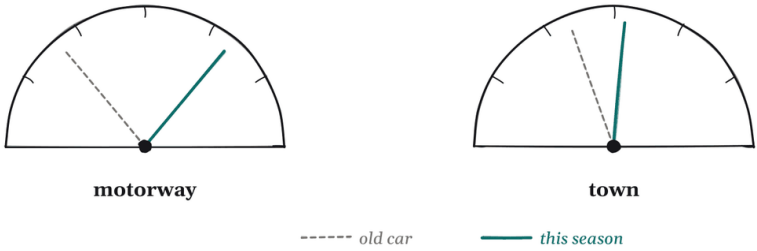


Like a lifeboat station's crew board: *a named tag hangs on every hook, from coxswain to mechanic, and an empty hook shows at once.*

T14. Economics Sheet

Compares cost per accepted outcome with the baseline, body and tail apart. Signed at Step 8, Own, by finance.

Field	What to write
Period	Weeks measured, including a business cycle
Cost lines	Platform, model, grading, audit, junction time, exception queue, rework, adoption
Accepted outcomes	Outcomes meeting the standard
Cost per accepted outcome	Overall, body and tail
Baseline	From the baseline sheet
Time and quality	Elapsed time and first-time-right, before and after
Annualised effect	On measured volume



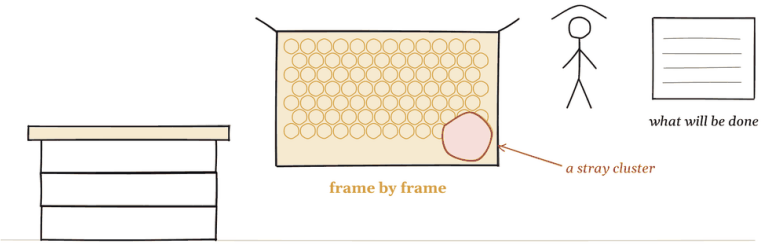
Like a car's fuel log over a full season: *the cost of each good mile on motorway and in town, kept apart, beside the old car.*

T15. Operating Review

Monthly review of a loop in the core, ending in recorded decisions. Signed at Step 9, Core, by the loop owner.

Field	What to write
Standard met	Body and tail
Cost per accepted outcome	This month
Exception queue	Oldest case, repeated exceptions to turn into rules
Junctions	Override rate and reasons

Field	What to write
Drift	Any segment where quality moved, and the response
Scope changes	Any widening, with the control owner's agreement
People	Every named role still filled
Decisions	What was decided, by whom

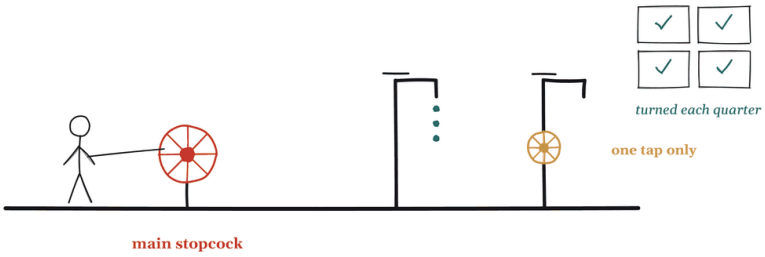


Like a beekeeper's monthly hive inspection: *frame by frame, check the brood, the stores and any stray cluster, then write down what will be done.*

T16. Switch-Off Plan

Says who can stop the loop, how, and what happens to cases in flight. Signed at Step 9, Core, by the loop owner.

Field	What to write
Holder and deputy	Names
How	The control used
Time to return cases	Target
Cases in flight	What happens to them
Partial switch-off	How to route one segment or counterparty to people
Test schedule	Usually quarterly
Last test	Date and result

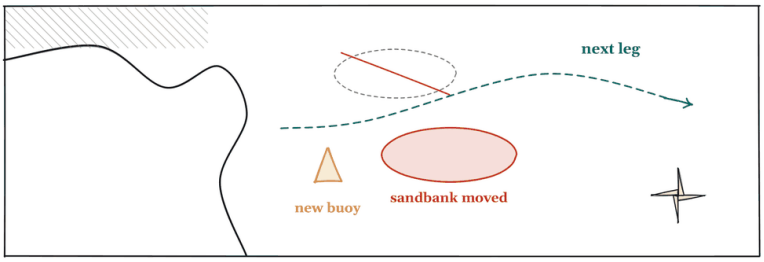


Like a house's water stopcock: *a named person knows where it is and turns it each quarter so it never seizes, and a side valve shuts one tap.*

T17. Map Review

Quarterly update of the map: state of each lifecycle and the next loops to read. Signed at Step 10, Scale, by the executive sponsor.

Field	What to write
Loops in progress	Each with its current step and gate
Readiness changes	Loops whose score moved, and why
Edges read	Exceptions by cause pointing to upstream loops; waits pointing downstream
Distribution re-read	Where people's time now sits
Next to read	Loops chosen, with owners

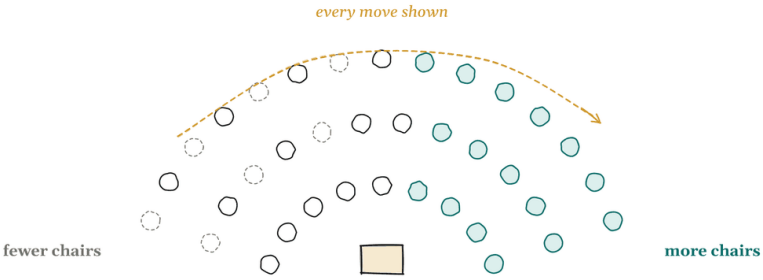


Like a sailor's chart corrected each quarter from the notices: *new buoys inked in, moved sandbanks redrawn, and the next leg pencilled on.*

T18. Role Change Plan

The map-wide plan for how roles and expertise change as loops are embedded. Signed at Step 10, Scale, by the executive sponsor.

Field	What to write
Roles growing	Junction holders, exception owners, graders, business engineers
Roles shrinking	And by how much
Moves	Where people are going, by loop
Rebuilding expertise	How people will learn the tail
Agreed with	Employee representatives where they exist

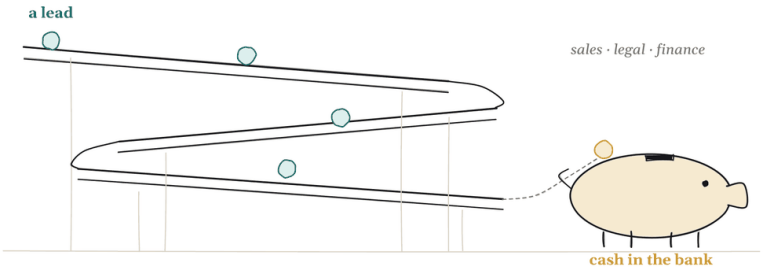


Like re-seating an orchestra for a new repertoire: *fewer chairs in one section, more in another, and every player is shown where they move.*

LXXXIV

The Lead-to-Cash Chain

Lead to cash is the chain that turns a prospective customer into money in the bank. It runs from qualifying a lead, through quoting, contracting, checking credit, booking the order and invoicing, to collecting payment and applying it to the right invoices. It touches sales, legal, finance and customer service. Most of its loops are customer-facing, so their standards must include the customer's experience as well as internal accuracy.



Like a marble run: *a prospect drops in at the top, rolls past sales, legal and finance in turn, and lands in the bank as cash.*

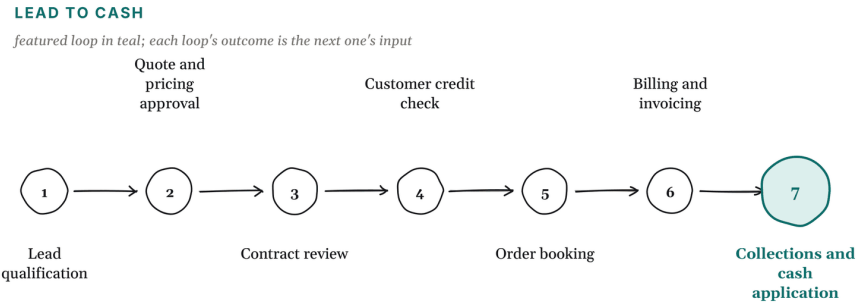


FIGURE 74 The lead to cash chain. Its loops in handoff order, with the featured loop, collections and cash application, in teal.

The loops and the evidence

The table gives, for each loop in the chain, the shape of its cases and a published benchmark. Shapes marked as estimates are reasoned from the nature of the work where no published distribution was found.

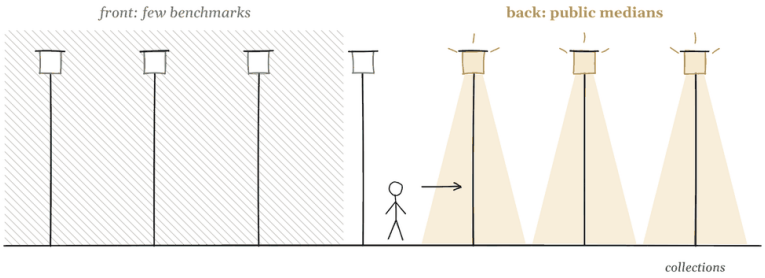
Loop	Shape of cases	Published benchmark
Lead qualification	Heavy tail (estimate)	No reliable public benchmark for cost or time per qualified lead
Quote and pricing approval	Heavy tail (estimate)	No public benchmark for cost or time per quote
Contract review	Heavy tail (estimate)	$3,800 per basic contract for top performers against $6,900 for others (IACCM, 2017, the latest public figure)
Customer credit check	Heavy tail	Median of 5 days to approve credit (APQC)
Order booking	Mixed	Median of about 80% of orders processed without a person touching them (APQC)
Billing and invoicing	Mixed (estimate)	Median of $2.50 to produce a customer invoice (APQC)
Collections and cash application	Heavy tail	Median of 38 days sales outstanding and 45% of receipts applied automatically (APQC); 0.4% of receivables more than 91 days past due (Credit Research Foundation, 2026)

Named public deployments in the chain are concentrated in lead qualification and cash application. Results are as reported by the companies or their vendors.

Loop	Company	Reported result
Lead qualification	Salesforce, own inbound sales	40% less time to qualify a lead year on year
Contract review	LogicMonitor	Initial review time cut by about 90% (vendor-reported)
Cash application	Keurig Dr Pepper	98% of cash applied with no manual touch (vendor-reported)
Cash application	Sysco	More than 85% straight-through cash application (vendor-reported)
Collections	Ferrero	67% fewer average days delinquent (vendor-reported)

The featured loop is collections and cash application. It shows a distribution whose tail is defined by customer rather than by case, a contested decision

between sales and finance, and an economic effect that comes mostly from cash released rather than cost saved.



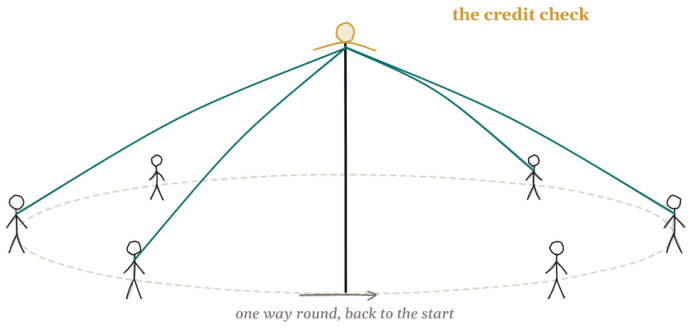
Like a street lit only at its far end: *billing and collections have public medians and named results, while lead qualification and quoting stand mostly in the dark.*

LXXXV

The Loops of Lead to Cash

Lead to cash runs from a prospect's first contact to cleared cash in the bank. Its seven loops hand work forward in a fixed order. Lead qualification passes an opportunity to quoting, and the accepted quote goes to contract review. The credit check sets limits and terms, and order booking turns the deal into a released order. Billing turns delivery into a receivable, and collections turns the receivable into cash. The credit check sits in the middle and exchanges work with four other loops. Collections sends payment history back to it, so the chain closes on itself.

Most of these loops have a heavy tail: a few customers, deals or accounts carry most of the value and the risk. Order booking and billing are the exceptions. Their bodies are large and uniform, and their tails are made of errors that started upstream. The evidence is uneven. The front of the chain has almost no published benchmarks and few named deployments, while the back has public medians and a long list of automation results. Contested decisions gather in the middle, where sales meets legal and finance over price, terms and credit.



Like a maypole dance: *the loops circle in fixed order and close the ring, while the credit check stands at the centre holding ribbons to four.*

Lead qualification

At a glance	
Outcome	A lead accepted as a sales-qualified opportunity, or rejected with a reason
Shape of cases	Heavy-tailed (estimate)

At a glance	
Decisions	2 mechanical · 1 bounded · 1 consequential · 0 contested
Hardest to undo	Routing a strategic account away from a senior seller
Anchor figure	Average email reply to a B2B demo request of 11h54m across 114 companies (Workato, published 2026, data likely older)

Lead qualification decides whether a prospect is worth a seller's time. Leads arrive from marketing as form fills, chats and event responses, and customer master data supplies account records for matching. An accepted lead goes to quote and pricing approval as an opportunity in the CRM, the customer relationship management system. A rejected lead returns to marketing nurture with its reason.

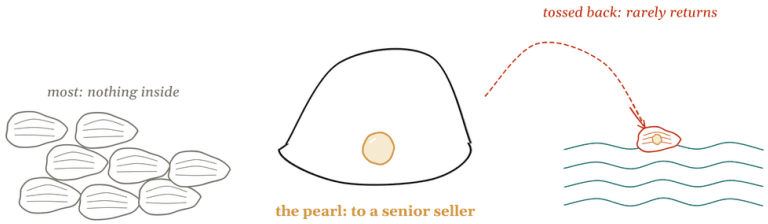
No distribution of lead value has been published, so the shape is an estimate. Most inbound leads are low-intent and never buy. Value grows with the size of the buyer, so a few enterprise-fit leads carry most of the pipeline. Salesforce reports that leads its agent qualified became opportunities 90% more valuable than average web leads, which fits a skew without measuring it. The body is volume to sort fast; the tail is a handful of accounts where delay is expensive.

The rules shape the design once rather than each case. From 2 August 2026, Article 50 of the EU AI Act requires that a prospect talking to an AI agent is told so. GDPR and the ePrivacy Directive require a lawful basis before outreach. Matching and routing are mechanical. Accepting a lead against written criteria such as budget, authority, need and timing is bounded. Sending a strategic account straight to a senior seller is consequential. No contested decision is recorded.

Written rules can take de-duplication, routing and the disclosure text. A model adds what rules cannot: reading a conversation, judging need and timing, and writing the handoff summary. People hold the junction where a large account is spotted, because a strategic prospect sent to nurture rarely comes back.

Speed is what published evidence supports. Workato tested 114 B2B companies and found average replies of 11h54m by email and 14h29m by phone, and about 20% never answered by email. Firms with a routing tool replied in 3h32m. Salesforce's survey of 5,500 sellers put non-selling work at 70% of their time in 2024. No reliable figure is published for cost per qualified lead or for conversion to opportunity.

As read in 2026, Salesforce runs Agentforce on its own website and reports 40% less time to qualify and a $20 million pipeline lift, vendor-reported. A second Salesforce agent reached 100,000 lower-intent leads and created 1,500 opportunities. Agibank and EasyPark run qualifying agents without published results. Faster answers attract more inbound leads, so a running loop may grow its body faster than its tail.



Like opening oysters: *most hold nothing, a rare one holds a pearl for a senior seller, and tossing that one back by mistake is the costly error.*

Quote and pricing approval

At a glance	
Outcome	A priced quote within discount policy, or with a recorded approved exception, sent to the customer
Shape of cases	Heavy-tailed (estimate)
Decisions	1 mechanical · 1 bounded · 1 consequential · 1 contested
Hardest to undo	Sending the customer a quote with a discount beyond policy
Anchor figure	No reliable published figure

This loop turns a qualified opportunity into a price the customer can accept. Pricing strategy supplies price lists, discount limits and the approval matrix, the table of who may approve which discount. The accepted quote passes to contract review with any non-standard terms. Deal value and requested payment terms go to the credit check.

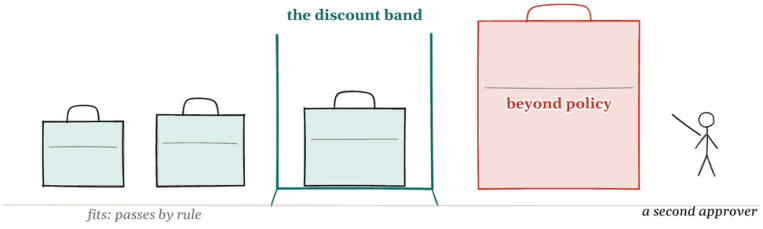
No spread of quote values is published, so the shape is an estimate. Deal size grows with customer size, and large strategic deals attract deep discounts and bundled terms. Most quotes sit at list price or inside policy and could be approved by rule. A few large ones carry most of the margin risk and most of the approval effort.

For US-listed companies, section 404 of the Sarbanes-Oxley Act (SOX) requires controls over financial reporting. Discount approval is one of them, so someone other than the seller must approve discounts beyond set thresholds. The Robinson-Patman Act requires a documented defence when competing resellers pay different prices. Competition law requires review of pricing algorithms for collusion risk. Configuring a bundle at list price is mechanical, and approving a discount inside the band is bounded. A deal beyond policy is consequential. Pricing for a new market, matching a competitor or pricing a reseller is contested. Finance, sales and legal weigh margin, share and legal exposure differently, and many companies have never settled it.

Price books can carry the body. A model can assemble the quote, apply the band and write the case for an exception on one page. People hold the deep-discount approval, since a price that reaches the customer is hard to withdraw. The contested question belongs with the leaders who own it, settled before design rather than argued on every deal.

An accepted quote is correctly priced, inside policy or carrying a recorded exception, and traceable to its approver. Gartner reported in 2025 that AI agents cut quote turnaround by more than 90%, but it gave no median. APQC tracks manual order fixes caused by pricing without a public value. No reliable figure is published for quote cycle time, quotes per seller or cost per quote, so each company must build its own baseline.

As read in 2026, the one reported deployment is Qohash, whose quoting agent built on Claude cut turnaround from one week to 15 minutes, by its own account. No named public company has published results for an AI quoting agent. Once the loop runs, sellers who learn the band the executor approves alone will cluster requests just below it.



Like the bag-size frame at a boarding gate: *quotes that fit the discount band pass by rule, and the oversized deal waits for a second person to approve.*

Contract review

At a glance	
Outcome	A customer contract within the playbook, or with approved deviations, signed by both parties and filed
Shape of cases	Heavy-tailed (estimate, with sector evidence)
Decisions	1 mechanical · 1 bounded · 1 consequential · 1 contested
Hardest to undo	Signing a deviation on liability, indemnity or intellectual property
Anchor figure	$3,800 per basic contract for top performers against $6,900 for others (IACCM, 2017)

Contract review produces a signed customer contract within the playbook, legal's set of standard clauses and acceptable fallbacks, or with approved deviations. It receives the accepted quote and the customer's redlines or own paper. The executed contract goes to order booking, the billing schedule to billing, and payment terms and security to the credit check.

Value lost to poor contracting varies about sevenfold by sector, according to World Commerce & Contracting and Deloitte in 2023. It runs near 3% in technology and near 20% in retail. No distribution of effort per contract is published, so the case shape is an estimate. Most sales contracts are on standard paper. Large deals on customer paper are different, because each negotiated clause interacts with the others and risk compounds. That small group takes most of legal's time.

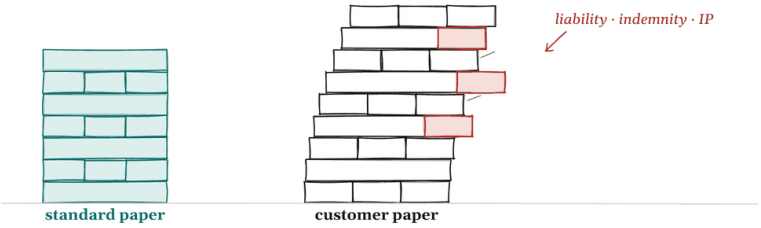
ABA Formal Opinion 512 keeps lawyers fully responsible for AI-assisted work, so legal sign-off cannot pass to a model. Signature authority policies let only named officers bind the company. The revenue standards ASC 606 and IFRS 15 require finance to judge whether terms such as side letters change when revenue is recognised. Extracting terms is mechanical, and comparing redlines with playbook fallbacks is bounded. Accepting a change to liability, indemnity or intellectual property is consequential. Walking away over terms is contested, because sales and legal weigh revenue and risk differently.

Rules can flag exact clause matches. A model reads free-text redlines, proposes the allowed fallback and drafts the first markup for a lawyer to confirm. The junction sits at liability, indemnity and intellectual property, where exposure

outlives the signature. The walk-away question needs a standing threshold from the heads of sales and legal.

Every signed contract should match the playbook or carry a recorded approval for each deviation. Aberdeen Group put completion at 11.5 days for the best firms against 21.2 for others in 2017. IACCM put a basic contract at $3,800 for top performers and $6,900 for others that year; nothing newer is public. World Commerce & Contracting and Icertis reported in 2023 that poor contracting loses nearly 9% of contract value.

As read in 2026, LogicMonitor, a private company, uses AI to review and redline agreements. Gartner reports first review time down about 90% and about $100,000 a month in return. HERE Technologies digitised 70,000 legacy contracts with Icertis, by the vendor's account. No public company has published sell-side results. Once standard paper clears in hours, sellers gain a reason to steer customers onto it, which thins the tail.



Like a block tower: *standard paper stands as built, but on customer paper each clause pulled and changed shifts the others, until liability brings it down.*

Customer credit check

At a glance	
Outcome	An approved credit limit and terms, or a requirement to prepay or give security, with the order released or blocked
Shape of cases	Heavy-tailed (measured)
Decisions	1 mechanical · 1 bounded · 1 consequential · 1 contested
Hardest to undo	Releasing a blocked order or granting a large limit to a weak customer
Anchor figure	Median of 5 days to approve credit across 4,740 companies (APQC, read in 2026)

The credit check gives a customer a limit and terms, or requires prepayment or security. Work arrives from quoting, contract review, order booking and collections. The limit and the release or block go to order booking. A risk class goes to collections, where it sets how each account is chased.

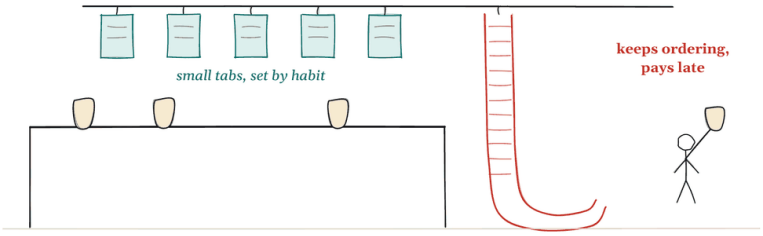
This tail is measured. The Credit Research Foundation found a median 87.9% of US receivables current in the second quarter of 2026, with only 0.40% more than 91 days past due. Loss compounds in a few accounts, because a struggling customer keeps ordering while it pays late. Policy adds bunching, since most companies approve small limits automatically and review large ones. The body is small limits set by table; the tail is a few exposures where one wrong limit becomes bad debt.

The rules tighten whenever the customer is a person, including a sole trader. In the SCHUFA case of December 2023, the EU Court of Justice held that a credit score strongly driving a decision is an automated decision under GDPR Article 22. That brings rights to human intervention and an explanation. The EU AI Act treats credit scoring of natural persons as high-risk. US Regulation B requires notice of business credit denials. SOX requires an approver separate from sales above thresholds. Building the score is mechanical, and setting a limit within policy bands is bounded. Approving a large limit or releasing a blocked order is consequential. Credit for a strategic but risky customer is contested between sales and finance, and often ends with the chief financial officer.

Score tables can take the small limits. A model reads financial statements and payment patterns and drafts the reasons a decline must give. People hold large limits and releases, because goods shipped on credit cannot be recalled.

An accepted outcome is a limit within policy, with recorded reasons and a prompt release or hold. APQC's open database, read in 2026, shows a median of 5 days to approve credit. APQC's 2026 consumer products figures put cost at $0.27 per $1,000 of revenue. The share of receivables over 91 days late serves as the quality check.

As read in 2026, the named deployments come from HighRadius and use rules and machine learning more than generative agents. The Mosaic Company reports approvals 55% faster, Chevron Phillips Chemical onboarding 61% faster, and Tech Data review efficiency up more than 120%. All are vendor-reported. Refused customers never build a payment history, so a running check learns only from those it approved.



Like a bar tab: *most regulars run small tabs set by habit, but one customer who keeps ordering while paying late runs up the loss.*

Order booking

At a glance	
Outcome	A complete, priced, credit-released sales order created in the ERP and confirmed to the customer
Shape of cases	Mixed: uniform body, varied exception tail
Decisions	1 mechanical · 2 bounded · 1 consequential · 0 contested
Hardest to undo	Releasing an order with a restricted-party or export-control hit
Anchor figure	Median of 80% of orders with no human intervention, across 425 companies (APQC, read in 2026)

Order booking creates a priced, released sales order in the ERP, the company's core transaction system, and confirms it to the customer. Purchase orders arrive by EDI, the structured electronic exchange of business documents, or by portal, email or PDF. Terms come from contract review and the release from the credit check. The order goes to fulfilment in the order-to-delivery chain and its lines to billing.

The shape is mixed, and it is measured. APQC's median is 80% of orders handled without a person, and top performers reach 94%. That body is uniform: the same match against master data and contract price, repeated thousands of times. The remainder is varied. APQC counts its causes separately as pricing, contract, master-data and invoicing problems. Each starts in a different upstream loop, so the tail is several small tails rather than one.

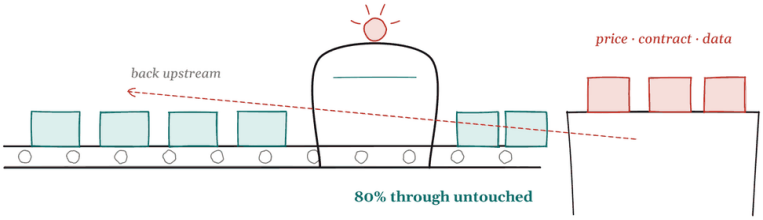
Trade law sets the hardest edge. US export regulations require that an order matching a denied party is held until a trade compliance officer clears it. The

EU dual-use regulation and sanctions rules require the same. Matching order data is mechanical. Promising a delivery date is bounded: available-to-promise rules compute it from stock and capacity, and a planner may override. Resolving a price or quantity mismatch within tolerance is also bounded. Releasing an order with a screening hit is consequential. No contested decision is recorded.

Rules carry matching, tolerances and delivery promises. A model adds the reading of free-form orders from email and PDF, where untouched rates usually fall. People hold mismatches outside tolerance, which need a call to the customer, and screening hits. An export-controlled order cannot be called back once shipped.

An accepted order is complete, priced to contract and confirmed on the first pass. APQC's open database, read in 2026, puts the median effort to create an order at 1.5 hours. A 2018 APQC study found a 2-day median order cycle for the most automated quartile against 8 days for the least. Cost per order is tracked but not public.

The named evidence is older automation. Accord Healthcare reported just under 65% of orders needing no manual work in 2018, and MEDRAD cut entry errors from 1.6% to under 0.4%. Micro Focus reports orders processed four hours faster, by its vendor's account. As read in 2026, no generative-agent deployment has published results. As the body clears untouched, the exception queue becomes the clearest record of what upstream loops get wrong.



Like a self-checkout: *most baskets scan through untouched, and the few that call an attendant each trace back to a different upstream fault.*

Billing and invoicing

At a glance	
Outcome	A correct, compliant invoice delivered to the customer on time, with the receivable posted

At a glance	
Shape of cases	Mixed (estimate): uniform body, error-driven tail
Decisions	2 mechanical · 1 bounded · 1 consequential · 0 contested
Hardest to undo	Billing a disputed milestone under ambiguous contract terms
Anchor figure	Median cost of $2.50 per invoice across 2,856 companies (APQC, read in 2026)

Billing sends a correct invoice to the customer's payables team and posts the receivable. Order lines and tax data come from order booking. A delivery confirmation from fulfilment triggers the bill, and contract review supplies milestones. The receivable goes to collections, entries go to period close, and disputes go to complaint handling in the issue-to-resolution chain.

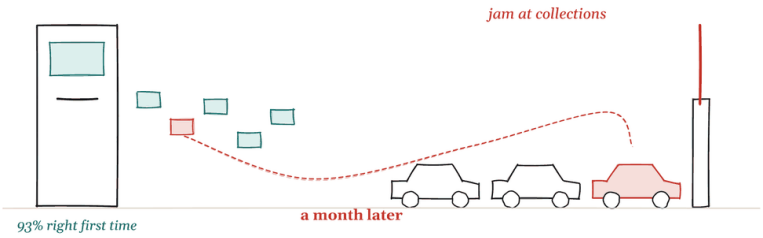
Volume is high and even. APQC's 2026 consumer products benchmarks show about 26,700 invoices per person, and across companies 93% of invoice lines are right first time. The other 7% form a tail that drives inquiries, deductions and late payment, so a billing error becomes a collections problem a month later. A value skew across customers is expected but has not been measured.

Tax law writes most of the rules, and most are mechanical. The EU VAT Directive fixes what an invoice must contain. Its 2025 amendment, VAT in the Digital Age, makes structured e-invoicing mandatory for intra-EU business trade from 2030. Italy already clears invoices through a government channel, and Germany has required businesses to accept e-invoices since 2025. SOX requires independent approval of credit notes above thresholds. Generating the invoice and setting its tax code are mechanical. Issuing a credit note is bounded. Billing a disputed milestone under unclear terms is consequential.

Tax engines and e-invoicing formats should carry the body by rule. A model earns its place in the tail, reading the contract behind an inquiry and proposing the credit note policy allows. People hold disputed milestones and usage bills, which affect revenue recognition. Once a disputed invoice reaches the customer, undoing it takes a credit note and a harder conversation.

An accepted invoice is correct, compliant, on time and posted. APQC's open database, read in 2026, puts the median cost at $2.50 per invoice. Its 2026 consumer products figures add 2 days to generate complete billing data and 70% of lines invoiced automatically. Count inquiries and deductions per thousand invoices too, because billing quality surfaces downstream.

As read in 2026, no named company has published results for an AI agent that generates invoices. Cross-industry productivity figures are public only for consumer products. As e-invoicing mandates spread, the body will grow more uniform and the tail will narrow to contract interpretation.



Like a car-park ticket machine: *thousands of tickets print correctly, and the odd misprint turns up weeks later as a jam at the exit barrier.*

Collections and cash application

At a glance	
Outcome	Payments matched to the right invoices and posted, overdue balances collected or resolved
Shape of cases	Heavy-tailed (measured)
Decisions	2 mechanical · 1 bounded · 2 consequential · 0 contested
Hardest to undo	Writing off a balance or sending it to an agency or to court
Anchor figure	Median of 38 days sales outstanding and 45% of receipts applied automatically (APQC, read in 2026)

The last loop turns receivables into cleared cash. Payments arrive from banks with remittance advice, the customer's note of which invoices a payment covers. Each payment is matched and posted, and overdue balances are chased. Cleared cash and write-offs go to period close and payment behaviour returns to the credit check. Deductions, amounts customers withhold for shortages or promotions, go to complaint handling.

The shape is measured. APQC's median shows 92% of invoices paid in full the first time, and the Credit Research Foundation found 0.40% of receivables beyond 91 days in mid-2026. Atradius found in 2012 that 30% of B2B receivable value in Western Europe was paid late, yet only about 3% of invoices defaulted. A few large customers hold much of the balance, and their

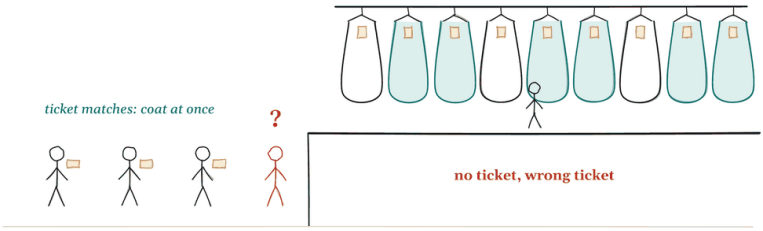
habits compound. In cash application, short payments, missing remittance and deductions absorb the manual effort.

The EU Late Payment Directive caps business terms at 60 days unless expressly agreed. SOX requires independent approval of write-offs. The expected-loss standards, CECL and IFRS 9, require management sign-off on the bad-debt allowance each period. Matching payments and dunning, the scheduled sequence of reminders, are mechanical. Accepting a short payment within tolerance is bounded. Payment plans, account holds and write-offs are consequential. No contested decision appears in the published evidence.

Matching rules and worklist ranking carry the body. A model reads PDF and email remittances, where unapplied cash builds up, and drafts reminders suited to the account. People hold plans, holds and write-offs. Sending a balance to an agency or to court is the least reversible step in the chain, because it usually ends the relationship.

Days sales outstanding (DSO) is the average number of days revenue waits to be collected. APQC's open database, read in 2026, shows a median DSO of 38 days and collections cost of $0.31 per $1,000 of revenue. It shows 45% of receipts applied automatically. The Hackett Group found a 70% median auto-match among software users in 2025. Measure DSO alongside cost, because cash released usually outweighs processing saved.

As read in 2026, every named result comes from HighRadius and is vendor-reported. Keurig Dr Pepper reports 98% no-touch cash application, and Sanofi 94% across 50 countries. Ferrero reports 67% fewer days delinquent, and Danone recovers $25.5 million a year from invalid deductions. Most is machine learning, not generative agents. Cleaner payment histories change the credit check, and so who reaches collections at all. The next chapter follows this loop through its whole lifecycle.

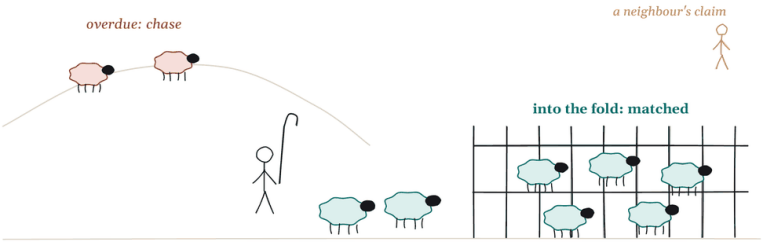


Like a cloakroom counter: *most payments arrive with a ticket and match at once, while the few with no ticket or the wrong one take the time.*

LXXXVI

Collections and Cash Application

At the composite manufacturer, the loop receives payments from about 2,800 business customers. It matches each payment to the invoices it settles, chases what is overdue, and resolves the deductions customers take. Its outcome is cleared cash in the ledger and accurate receivables for treasury and the close.



Like a shepherd at dusk: *counting each sheep into the fold, going after the strays, and settling a neighbour's claim, until every animal is accounted for.*

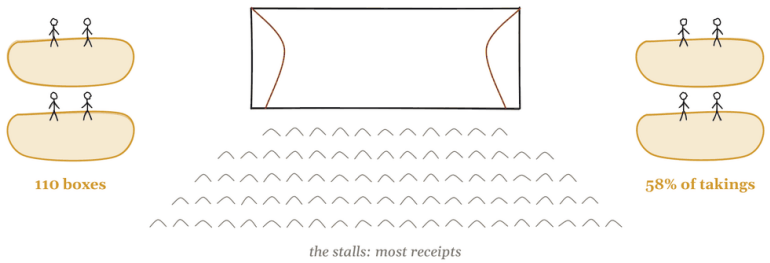
Step 1, Read: the map entry

The chief financial officer's map placed the loop at value four and readiness four. Bank data was already structured, the outcome was measurable to the cent, and the head of credit and collections wanted to own it.

Field	Entry
Loop	Collections and cash application
Chain	Lead to cash
Value to the core	4: about 9,600 receipts a month; cash and working capital directly affected
Readiness	4: bank feed in place; rules mostly written; owner willing; one known dispute between sales and finance
Handoffs	In from billing and the bank; out to the close, credit checks and dispute handling
Risk category	Not high risk; late payment rules apply to dunning
Owner	Head of credit and collections (agreed)
Distribution read	By case: 72% of receipts pay one invoice in full with a clear reference; 19% short-pay or deduct; 9% arrive with unclear remittance. By customer: the largest 110 customers, 4% of the total, hold 58% of receivables

Field	Entry
Signed	Chief financial officer

The distribution had two dimensions. The routine body was clear by case. But the tail was defined by customer: a small receipt from one of the 110 largest customers mattered more than a large one from a small customer, because of the relationship at stake.



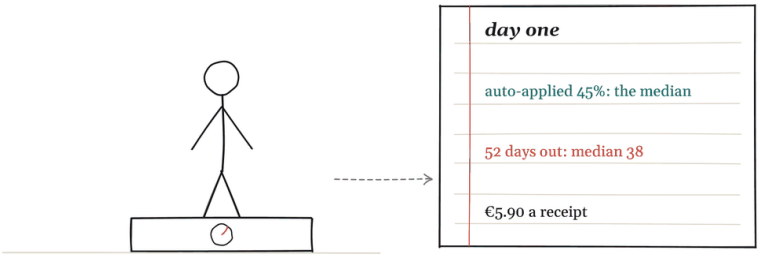
Like the boxes at a theatre: *a few seats hold most of the takings, so a small request from a box outranks a big one from the stalls.*

Step 2, Map: the baseline sheet

Mapping with the cash application clerks and collectors drew twelve steps across six lanes: customer, bank, cash application, collections, customer service and sales.

Field	Entry
Sample and period	14,200 receipts and 4,100 collection actions over six weeks
Volume	9,600 receipts a month; 1,400 collection calls a month
Fully loaded cost per receipt applied	€5.90, including collections effort allocated per receipt
Receipts applied automatically	45%, by existing system rules
Unapplied cash	6 days average before a receipt was applied
Days sales outstanding	52
Receivables over 90 days past due	1.3%
Deductions	34 days average to resolve
People	5 cash application clerks, 6 collectors
Method agreed with finance	Yes, planning and analysis lead

The automatic application rate sat exactly at the published median. Days sales outstanding sat well above it, partly because of long terms with distributors.

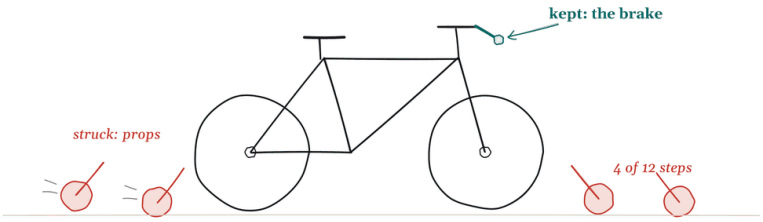


Like weighing in on day one: *one reading sits right at the average, another well above it, and every later result is judged from this page.*

Step 3, Strip: the strip register

Step	Status	Constraint	Still applies?
Download bank statements to a spreadsheet and re-key unmatched lines	Struck	No bank feed into the ledger when the process was designed	No: the feed has run since 2021
Email customers for remittance advice	Struck	Remittances arrived in a shared mailbox nobody read systematically	No: the mailbox can be read by the executor
Weekly printed unapplied-cash report for the controller	Struck	The controller once had no system access	No
Dunning letters written by hand from a template	Struck	System dunning switched off during a 2019 migration and never restored	No
Credit manager approves write-offs	Kept	Write-offs reduce assets and need authority	Yes: becomes a junction
Match receipts to open items	Kept	Required by the outcome	Yes

Four of twelve steps were struck, a strip ratio of 33%. That put the loop between automation and redesign.



Like stabilisers taken off a bicycle: *four steps were props for limits long gone, so they came off, while the brake, the write-off approval, stays.*

Step 4, Sort: rules, decisions and the contested log

The rule register held 41 rules. Six contradicted each other. The two that mattered most are shown.

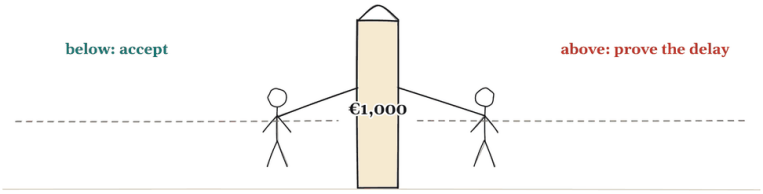
Rule	Source	Contradiction	Decision
Write off short payments under €50 without approval	Finance policy	Collectors wrote off up to €250	Controller: €50 stands; €50 to €250 becomes bounded with a weekly report
Credit hold at 60 days overdue	Credit policy	For the largest customers, holds were applied at 90 days in practice	Credit manager: 60 days for all, with the account manager informed at 45

The decision register held fourteen decisions.

Decision	Class	Limit or junction
Does the payment reference match an open invoice?	Mechanical	
Does amount and customer match a single open item?	Mechanical	
Which reminder is due under the dunning schedule?	Mechanical	
Apply a short payment within tolerance	Bounded	Up to €50
Match a multi-invoice remittance with no references	Bounded	Above a confidence threshold set in shadow
Rank the collection worklist	Bounded	By amount, age and customer tier
Write off a balance over €250	Consequential	Credit manager
Place a customer on credit hold or agree a payment plan	Consequential	Credit manager, account manager informed

Decision	Class	Limit or junction
Accept a deduction for late delivery without proof	Contested	See log

Field	Entry
Decision	Do we accept a distributor's deduction for late delivery without proof that the delivery was late?
Positions	Sales: accept, to protect the relationship. Finance: never accept without proof
Logged	Week 1 of Sort
Settled by	Chief financial officer and chief commercial officer
Answer	Accept without proof up to €1,000 per invoice for customers with fewer than two such claims a quarter. Above that, customer service confirms the delay from shipment records
Date settled	Week 3 of Sort
Resulting class	Bounded under €1,000; consequential above, with a junction held by the customer service lead



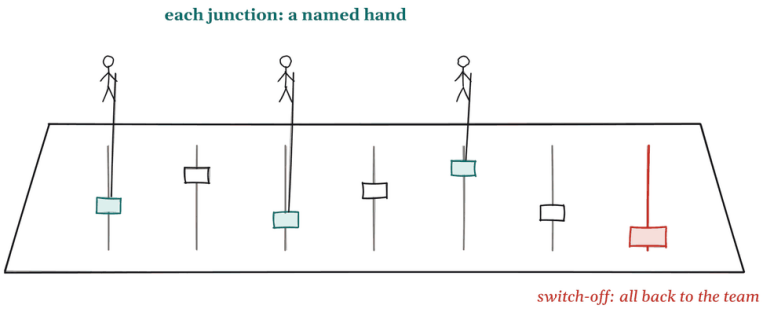
Like neighbours settling a boundary: *sales and finance pulled opposite ways until their two leaders drove one post at €1,000, and now both sides know the line.*

Step 5, Design: junctions and the standard

Junction	Who decides	Evidence shown	How reversed
Write-off over €250	Credit manager	Invoice, payment history, dispute notes, prior write-offs for the customer	Reinstate the balance and resume collection
Credit hold or payment plan	Credit manager	Ageing, order pipeline, account manager's note	Release the hold the same day
			Rebill the customer

Junction	Who decides	Evidence shown	How reversed
Deduction over €1,000 for late delivery	Customer service lead	Shipment records, promised and actual dates	
Second reminder to one of the 110 largest customers	Account manager	The reminder text and open items	Not sent until approved
Switch-off	Head of credit and collections	Grading dashboard	All receipts and reminders return to the team within 30 minutes

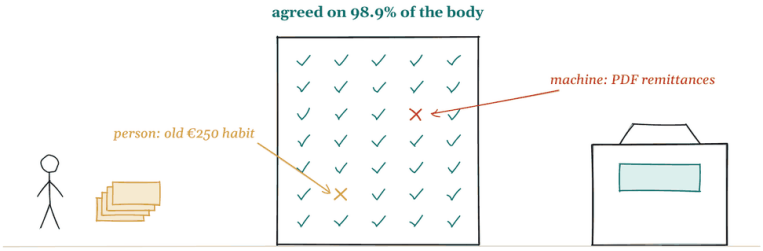
Field	Entry
Criteria	Each receipt applied to the right customer, invoices and amounts; no receipt unapplied for more than two working days; reminders sent on schedule and never to a customer in an open dispute
Rule checks	Every receipt balances to the bank line; every reminder checked against open disputes
Model grader	Every application and every reminder graded against the criteria
Human audit	2% at random; 10% for the 110 largest customers
Customer signals	Complaints about reminders; disputes reopened
Calibration	Monthly, by the finance quality analyst



Like a sound desk: *each risky channel has a fader under a named hand, and one master fader returns the whole mix to the team.*

Step 6, Shadow: the report

Field	Entry
Period and cases	Six weeks; 14,400 receipts; 2,900 reminders
Agreement, body	98.9%
Agreement, tail	86%: unclear remittances and the 110 largest customers
Disagreements	Executor wrong 52%: remittance tables in PDF from three large distributors; credit notes netted against payments. Person wrong 31%: collectors still applying the old €250 write-off limit. Both defensible 12%. Standard unclear 5%
Changes made	Two rules added; remittance reading corrected for the three distributors; write-off limit briefed to collectors
Known weaknesses	Remittances that refer to purchase orders rather than invoices
Recommendation	Pass G1

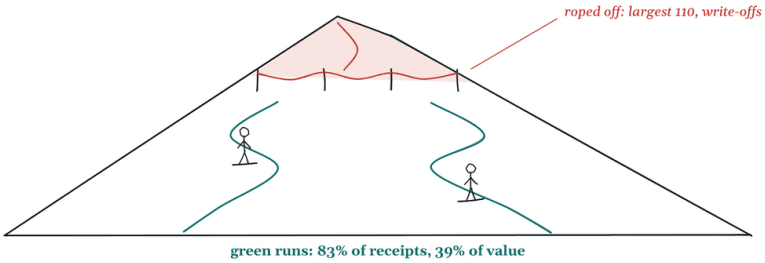


Like two cashiers counting the same till: *they agreed on almost every note, and each difference was traced, sometimes to the machine, sometimes to an old habit.*

Step 7, Bound: the gate review record

Field	Entry
Scope alone	Receipts matched above the confidence threshold, for customers outside the largest 110; short payments under €50; first reminders for customers outside the largest 110. 83% of receipts and 39% of receipt value
Evidence	Four weeks bounded: 99.2% of bounded applications met the standard; no reminder sent to a customer in dispute
Permissions	Apply cash, post deductions to the dispute queue, send first reminders. No permission to write off or change credit limits

Field	Entry
Switch-off test	Week 1: work returned to the team in 15 minutes; nothing lost
Audit consulted	Internal audit: keep the text of every reminder sent with the customer record
Decision	Pass G2. Signed by the financial controller



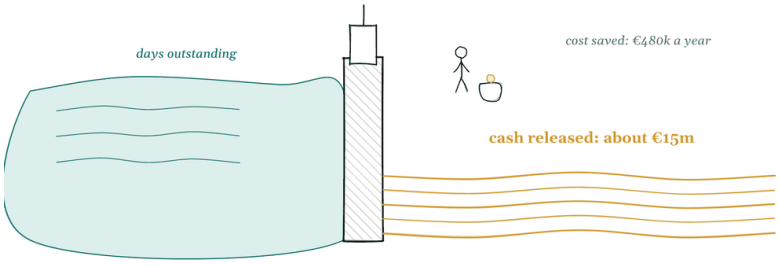
Like a skier cleared for the green runs only: *most of the mountain's traffic, little of its value at risk, and a rope across the steep slopes above.*

Step 8, Own: charter and economics

Field	Entry
Loop owner	Head of credit and collections
Junction holders	Credit manager; customer service lead; 22 account managers for the largest customers
Exception owner	Cash application lead, unapplied cash cleared within one working day
Grader	Finance quality analyst
Adoption owner	Finance operations manager
Control owner	Financial controller
Changes to work	Cash application: five clerks to two on exceptions. Collections: six collectors to four, focused on the largest customers and disputes. Two moved to a new dispute resolution team, two to the customer credit check loop; one left by choice and was not replaced
Learning path	New collectors work the exception queue for two months and sit in on credit manager junctions

Field	Entry
Period	Eight weeks across a quarter-end
Cost lines	Platform and model; grading and audit; junction time; exception queue; dispute team share; adoption
Cost per accepted receipt	€1.70 overall; body €0.90; tail €6.80
Baseline	€5.90
Applied automatically	45% to 88%
Unapplied cash	6 days to 1.2 days
Deductions resolved	34 days to 19 days
Days sales outstanding	52 to 47
Annualised effect	About €480,000 a year in processing cost. Five days less outstanding on €1.1 billion of revenue releases about €15 million of working capital
Signed	Planning and analysis lead

The processing saving was real but small next to the cash released. That is common in this chain, and it is why finance should measure days outstanding as well as cost.

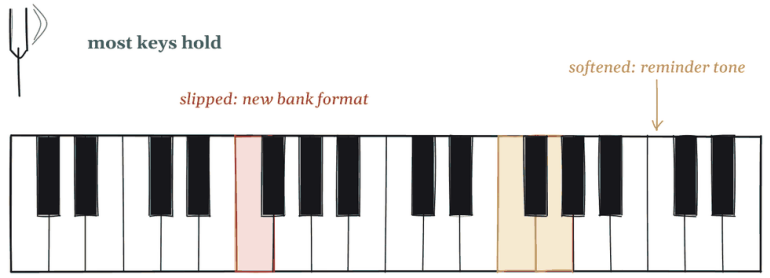


Like opening the sluice on a mill pond: *the miller saves a little labour, but the real gain is the water released downstream, the cash.*

Step 9, Core: an operating review

Field	Entry
Month	Second month in Core

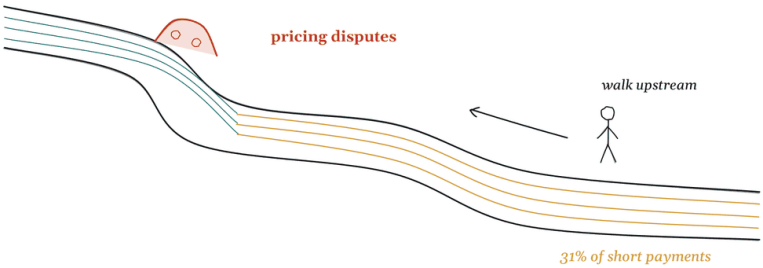
Field	Entry
Standard met	Body 99.4%; tail 93%
Cost per accepted receipt	€1.60
Exception queue	Oldest case 1 day. Repeated exception: receipts referring to purchase order numbers; rule added
Junctions	Credit manager overrode one write-off proposal in eight, asking for further collection
Drift	A bank changed its statement format; its receipts went to people for one day while reading was corrected
Customer signals	Two complaints from long-standing distributors about reminder tone; template revised and a rule added for customers over ten years old
Decisions	Widen bounded scope to second reminders outside the largest 110, agreed by the controller



Like tuning a piano each month: *most keys still hold, one string slipped when the room changed, and two notes were softened after listeners complained.*

Step 10, Scale: what the edges showed

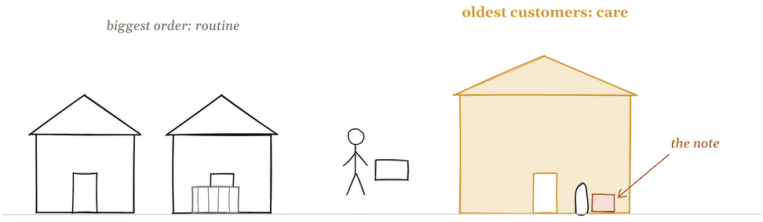
Reading the exception queue by cause showed that 31% of short payments came from pricing disputes. Those disputes started upstream, in quote and pricing approval and in billing. The map review chose billing as the next loop in the chain. Customer credit checks, which now received cleaner payment histories, rose in readiness and were marked to read after it.



Like following muddy water upstream to a broken bank: *most short payments traced back to pricing disputes, so billing became the next loop to fix.*

What this loop teaches

The tail of a loop is not always the largest cases. Here it was a set of customers, defined by the relationship. Reminders were the customer-facing part of the loop, and the standard had to include how customers received them. The contested decision was commercial, between sales and finance, and only their two leaders could settle it. And the largest economic effect was cash released, not cost saved, which a cost-only economics sheet would have missed.

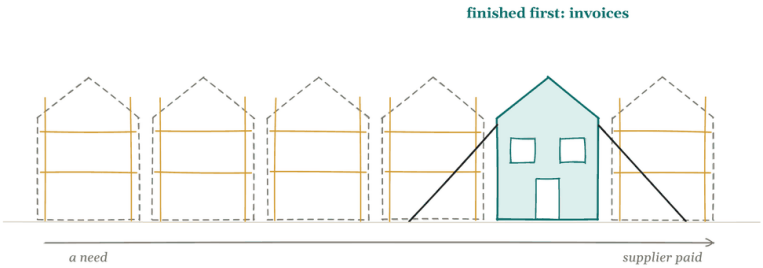


Like a milk round: *the doorsteps that need care are not the biggest orders but the oldest customers, and the note left with the milk matters too.*

LXXXVII

The Procure-to-Pay Chain

Procure to pay is the chain that turns a need inside the company into goods received and a supplier paid. It runs from an employee's request, through supplier onboarding, the purchase order and the receipt of goods or services, to approving the supplier's invoice and releasing payment. It touches every plant and office, procurement, finance and the suppliers themselves. At the composite manufacturer, the supplier invoice loop in this chain was the first loop the company embedded, and it already runs in the core. The chain is therefore where the company learned the method, and where its later loops can lean on a proven neighbour.



Like a row of terraced houses built from one end: *the invoice loop was finished first, and the loops still going up can lean on its walls.*

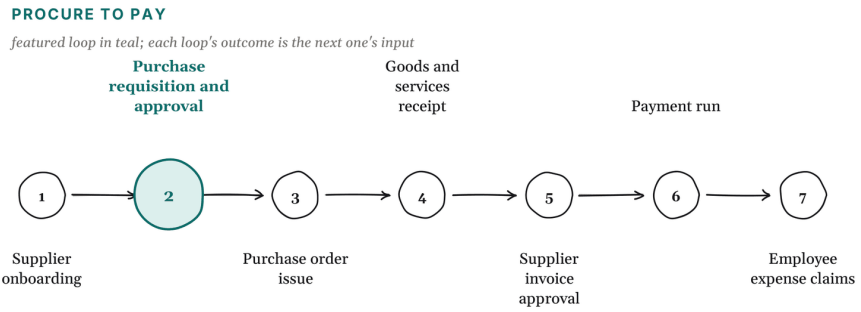


FIGURE 75 The procure to pay chain. Its loops in handoff order, with the featured loop, purchase requisition and approval, in teal.

The loops and the evidence

The table gives, for each loop in the chain, the shape of its cases and a published benchmark. Shapes marked as estimates are reasoned from the nature of the work where no published distribution was found.

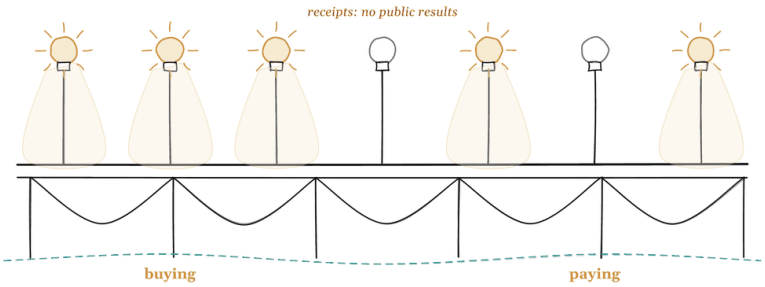
Loop	Shape of cases	Published benchmark
Supplier onboarding	Heavy tail (estimate)	Median of 3 days to set up a supplier in the procurement system (APQC)
Purchase requisition and approval	Heavy tail	Median of 2 days from requisition received to purchase order released (APQC); tail spend is about 20% of spend but about 80% of suppliers (Hackett Group data, 2021)
Purchase order issue	Heavy tail	Median total cost of $55 per purchase order (APQC)
Goods and services receipt	Mixed (estimate)	No public benchmark found
Supplier invoice approval	Heavy tail	Average of $9.40 per invoice; 9% of invoices raise an exception at the best performers against 22% for others (Ardent Partners, 2025); median of 75% of invoice lines matched first time (APQC)
Payment run	Heavy tail (estimate)	Median of 1.5% of annual payments duplicate or erroneous (APQC)
Employee expense claims	Mixed (estimate)	$58 to process one expense report; 19% of reports with errors (GBTA Foundation, 2015, old data)

Named public deployments with reported results cluster at the two ends of the chain: buying and paying. Results are as reported by the companies or their vendors.

Loop	Company	Reported result
Supplier onboarding	Kantar	Onboarding time cut from 60 days to 12 days (vendor-reported)
Purchase requisition	Microsoft, own device requests	23% lower spend on primary devices, about $20 million a year
Purchase order issue	Walmart	68% of tail suppliers approached reached a deal; average savings of 3% (vendor-reported)
Supplier invoice approval	Uber	Average handling time down 70%; 25% to 30% cost saving against manual work

Loop	Company	Reported result
Supplier invoice approval	Applied Industrial Technologies	87% of invoices processed with no manual touch (vendor-reported)
Employee expense claims	Takeda	63% of expense reports approved automatically after review of every report (vendor-reported)

The featured loop is purchase requisition and approval. Its risk does not sit in the largest requests. It sits in a long tail of small ones, where policy exceptions and off-contract buying hide. It also carries a contested decision between the plants and procurement, and its economics turn on speed and contract coverage more than on processing cost.



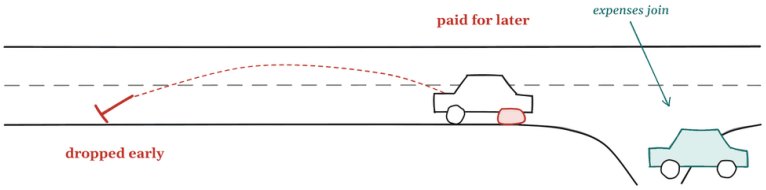
Like a long bridge lit mostly at either end: *named results cluster at buying and paying, while the receipt of goods in the middle stays dark.*

LXXXVIII

The Loops of Procure to Pay

Procure to pay is seven loops in a line, with one side entrance. A need becomes an approved requisition. If the supplier is new, onboarding makes it payable. The approved request becomes a purchase order, the delivery is recorded against it, and the supplier's invoice is matched to both. Approved invoices then leave the company in a payment run. Employee expense claims enter separately and join the chain at that last loop. Each loop hands the next a reference it will check against, so an error made early is usually caught, or paid for, two or three loops later.

Most of these loops share one fact. About 80% of a company's suppliers carry only about 20% of its spend, a pattern called tail spend (Hackett Group data, 2021). So most cases are small, and value sits in a few. Where the loops differ is in where their risk sits. In requisitions it hides among small requests. In onboarding and payment it sits in a few suppliers and a few bank changes. In receipts and expenses the evidence is thin and the shape is only estimated.



Like a motorway with one slip road: *the loops run in a line, expenses merge near the end, and a nail dropped early bursts a tyre further on.*

Purchase Requisition and Approval

At a glance	
Outcome	An employee's need becomes an approved, correctly coded requisition within budget and policy
Shape of cases	Heavy-tailed: many small requests, few large ones
Decisions	1 mechanical · 3 bounded · 1 consequential · 1 contested
Hardest to undo	Approving the spend, once a purchase order has gone to the supplier
Anchor figure	Median of 2 days from requisition received to purchase order released (APQC, accessed 2026)

The loop takes a need from an employee or budget holder, or a system-generated request from inventory replenishment. It returns a requisition with item, quantity, price, supplier and account coding. That goes to purchase order issue. When the preferred supplier is not yet in the vendor master, the company's file of approved suppliers, the request also goes to supplier onboarding.

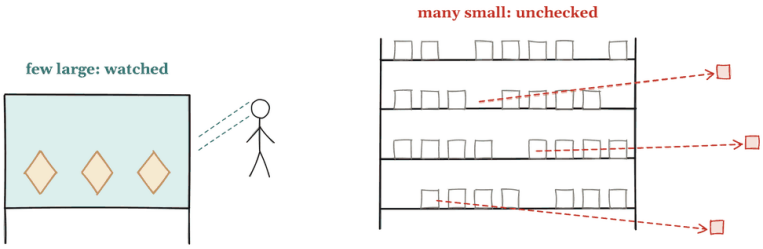
Requisition counts follow the tail-spend split. Low-value requests dominate the count, and a few large ones carry the value. That is why approval matrices tier by amount. The body is the catalogue and contract request that codes itself. The tail is less obvious than it looks. Large requests are few and well watched, while risk often gathers in the many small free-text requests nobody checks.

In the United States, the Sarbanes-Oxley Act, section 404, requires approval by an authorised budget holder independent of the requester. Public bodies face formal tender above the thresholds of EU Directive 2014/24 or the US Uniform Guidance. Routing to catalogue, contract or free text is mechanical. Coding, budget and approver checks are bounded, and so is the trigger for legal, security or privacy review. Approving the spend is consequential. The contested decision is the maverick purchase, one made outside policy. Plants and procurement rarely agree when it is allowed.

Written rules can route, find approvers and enforce limits. A model adds coding and the call on borderline reviews. People hold the spend approval and the exception, because both commit money before anything can be undone.

Grade each requisition on coding, budget, approver, contract use and required reviews. APQC puts the median at 2 days from requisition to released order, and at $55 total cost per order for the wider process (both accessed 2026). No public figure exists for volume or for requisitions returned for recoding.

As read in 2026, OpenAI, Canva, Webflow and Wiz were named early adopters of Zip's intake agents, at pilot stage with no metrics published. Microsoft's own device requisitions cut primary device spend by 23%, about $20 million a year, with agent results not yet reported. A faster loop tends to shrink its own tail, because requesters stop buying around it. The next chapter follows this loop through the whole lifecycle.



Like a jeweller guarding the glass case: *the few big requests are watched closely, while losses slip out through the many small free-text ones on the open shelves.*

Supplier Onboarding

At a glance	
Outcome	A new supplier is screened and active in the vendor master with verified legal, tax and bank details
Shape of cases	Heavy-tailed (estimate): many low-spend suppliers, few high-risk ones
Decisions	2 mechanical · 1 bounded · 1 consequential · 1 contested
Hardest to undo	Clearing a sanctions match and activating the supplier for payment
Anchor figure	Median of 3 days to set up a supplier in the procurement system (APQC, accessed 2026)

Requests arrive from requisition and approval, when a buyer wants a supplier not yet on file, and from sourcing, when a contract has been awarded. The output serves three loops at once. Purchase order issue needs terms and an ordering address. The payment run needs a verified bank account. Invoice approval needs the tax ID and e-invoicing channel.

Most onboarding requests are for small tail suppliers. A few strategic or high-risk suppliers, with sanctions exposure, data access or critical supply, carry most of the risk. This tiering is an estimate built on the spend split. No published distribution of onboarding effort was found. The body should pass quickly on registry checks. The tail deserves the slow, documented review.

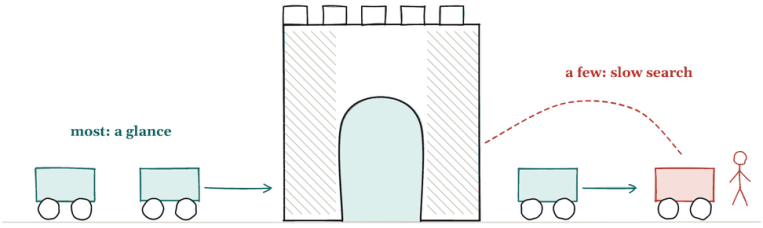
Few loops sit under more law. US sanctions rules from OFAC, the Treasury's sanctions office, impose strict liability, and EU Regulation 269/2014 requires asset-freeze screening. Germany's supply chain act and the EU due diligence directive add human-rights risk analysis. The Uyghur Forced Labor Prevention

Act demands supplier tracing. Tax ID checks on Form W-9 are mandatory in the US. Sarbanes-Oxley separates whoever changes bank details from whoever approves invoices or releases payments.

The duplicate check and the tax and bank validations are mechanical. Choosing the risk tier and questionnaire is bounded by policy thresholds. Clearing or blocking a sanctions or adverse-media hit is consequential and belongs to a compliance owner. Approving a high-risk or single-source supplier with open findings is contested, because business urgency and compliance pull apart. A model can read documents and pre-score risk. People keep the clearance.

APQC reports a median of 3 days to set up a supplier (accessed 2026). Graphite Connect's 2026 vendor survey of Fortune 500 leaders found an average of 18.8 days end to end, and up to 91. No public figure exists for cost per supplier or for vendor-record error rates.

Kantar, in a Coupa-published story, cut onboarding from 60 days to 12 in production. As read in 2026, no named company runs an AI agent for onboarding end to end. Faster onboarding may also lower the bar to adding suppliers, which lengthens the tail it then has to watch.



Like a town gate: *most carts roll through after a glance at their papers, while the few carrying risky cargo are pulled aside for a slow search.*

Purchase Order Issue

At a glance	
Outcome	A confirmed purchase order with agreed price, quantity, date and terms is sent to an active supplier
Shape of cases	Heavy-tailed: tail suppliers dominate the count, few carry the value
Decisions	1 mechanical · 1 bounded · 1 consequential · 1 contested
Hardest to undo	Accepting a supplier's change to price or quantity once goods are moving

At a glance

Anchor figure	Median total cost of $55 per purchase order (APQC, accessed 2026)

This loop joins two inputs, an approved requisition and an active vendor record. Its purchase order becomes the reference for the rest of the chain. Receiving uses it to check deliveries. Invoice approval uses it for matching. Supplier confirmations and delays flow on to the delivery exceptions loop in the order-to-delivery chain.

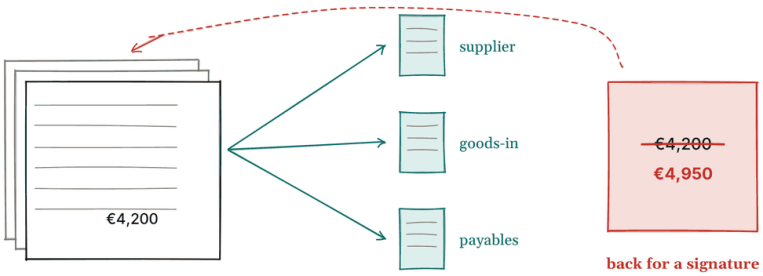
The value skew is the same as for requisitions. Pactum, which builds negotiation software, designs around automating roughly 80% of non-strategic supplier negotiations while people handle the top 20% of vendors. The body is the order raised against an existing contract or catalogue. The tail is the large or unusual order, and every order whose supplier proposes a change after it is sent.

Sarbanes-Oxley bars the buyer who issues an order from also receiving the goods or approving the invoice. It also requires re-approval of changes above tolerance. The EU Late Payment Directive allows terms beyond 60 days only when expressly agreed and not grossly unfair. Converting a requisition against a contract is mechanical, because price and supplier are already fixed. Choosing among spot quotes is bounded by sourcing rules. Accepting supplier-proposed changes is consequential. Microsoft's own procurement agent flags those for buyer approval. Negotiating away from standard terms is contested.

Rules can issue the contract order. A model can compare quotes, read supplier emails and PDFs, and extract proposed changes. People should hold the change and the non-standard term, since both alter what the company owes.

APQC data reported in 2026 by Supply Chain Management Review put cost per order at about $14 at the 25th percentile and $54.50 at the 75th, averaging $42. Ninety per cent of organisations were below $100. An older APQC study from 2014 found top performers placed an order in 8 hours against 11 for others. No public error or change-order rate exists.

Walmart used Pactum's chatbot to negotiate with tail suppliers. In Pactum's figures, 68% of suppliers approached reached a deal, at 3% average savings. Bloomberg reported 75% preferred the bot. Maersk used Pactum for port trucking contracts with no published result. As read in 2026, no named company reports results from Microsoft's or SAP's order agents.



Like a carbon-copy order book: *copies go to the supplier, goods-in and payables, and any figure the supplier crosses out comes back for a buyer's signature.*

Goods and Services Receipt

At a glance	
Outcome	What the supplier delivered is recorded against the purchase order, so invoices can be matched and stock or expense booked
Shape of cases	Mixed (estimate): routine stock receipts plus a smaller set of services and inspected materials
Decisions	1 mechanical · 2 bounded · 1 consequential · 0 contested
Hardest to undo	Releasing an inspected material from quarantine into use
Anchor figure	No reliable published figure

The loop takes open purchase order lines and a delivery or completed service. It records what arrived. That record feeds four places: invoice approval for matching, inventory for use, delivery exceptions for short or damaged goods, and the period close. At close, goods received but not yet invoiced must be accrued.

No published evidence describes the shape of receipt work, so this is reasoning. Physical receipts of standard stock are numerous and alike, with independent small variations, which gives a normal body. Service confirmations, partial deliveries and regulated materials are fewer, and they carry most exceptions. Many companies run goods and services as separate loops, and they may deserve two entries.

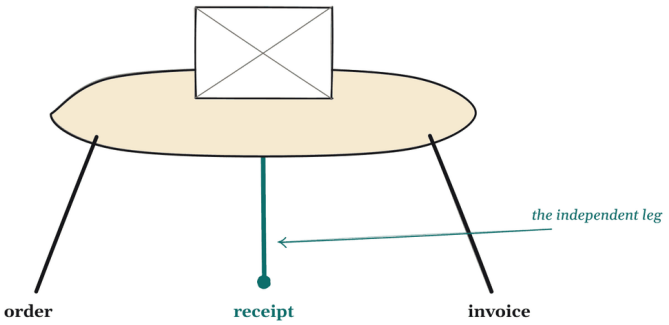
The receipt is the independent evidence in a three-way match, the check of invoice against order and receipt. Sarbanes-Oxley therefore keeps receiving

apart from purchasing and payables. In drug manufacturing, 21 CFR 211.84 quarantines components until the quality unit approves them. Device makers must verify purchased product under 21 CFR Part 820. ISO 9001 asks certified firms to verify purchases before release.

Matching a count or scan to the order is mechanical. Accepting a partial or over-delivery within tolerance is bounded. So is confirming a service against its statement of work, though disputes escalate. Quality acceptance in regulated industries is consequential and belongs to the quality unit. No contested decision is recorded, which may only mean nobody has looked. Rules and scanning take the stock body. Cameras and models can capture labels and lot numbers. People hold quality release and service sign-off, where the evidence is judgement rather than a count.

Grade each receipt on whether it matches what arrived, by quantity, item and condition, and on how quickly it posts. No public benchmark was found for volume, time from delivery to posting, cost per receipt or receipt error rate.

Lineage, with Kargo's camera towers at its Decatur, Alabama warehouse, posted over 64,000 receipts in six months. That is more than 500 pallets a day, with per-receipt time down from minutes to seconds. As read in 2026, this is a partial deployment. Services remain unmeasured everywhere.



Like the third leg of a stool: *the order and the invoice only stand firm once the receipt, an independent record of what arrived, takes its share of the weight.*

Supplier Invoice Approval

At a glance	
Outcome	A supplier invoice is matched, coded, approved and posted as a liability due for payment
Shape of cases	Heavy-tailed: low-value invoices dominate; exceptions and fraud sit in a minority

At a glance	
Decisions	2 mechanical · 1 bounded · 2 consequential · 1 contested
Hardest to undo	Approving an out-of-tolerance or non-PO invoice that then reaches the payment run
Anchor figure	Average of $9.40 per invoice (Ardent Partners, 2025)

This loop has more inputs than any other in the chain. The invoice comes from the supplier by network, portal, email or paper. The order, the receipt and the vendor's validated tax and bank data come from three earlier loops. Approved invoices go to the payment run with due dates and discount terms. Postings, accruals and VAT records go to record to report.

Three tails overlap here. By value, the tail-spend split makes most invoices small. By effort, Ardent Partners found 9% of invoices raising exceptions at the best performers and 22% elsewhere in 2025. By risk, billing schemes were 22% of occupational fraud cases in the 2024 study by the Association of Certified Fraud Examiners (ACFE), with a median loss of $100,000. The body matches cleanly. The tail is where a person's time goes.

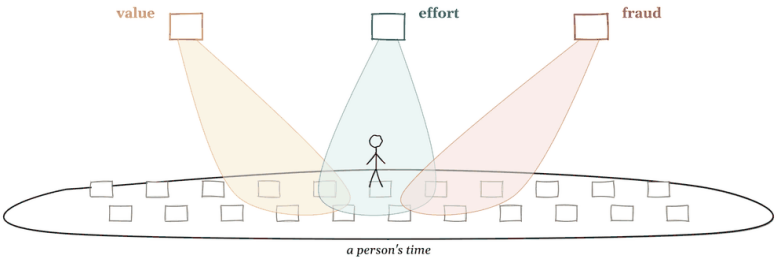
Sarbanes-Oxley requires independent approval, separate from vendor master and payment release. The EU VAT Directive lets a company deduct input VAT only on a valid invoice. EU Directive 2025/516 is phasing in structured e-invoicing to 2030. The Late Payment Directive charges interest when approval delays push payment past terms.

Extraction and matching are mechanical. Coding non-PO invoices is bounded. Two decisions are consequential: approving an invoice without full evidence, and holding a suspected fraud or duplicate. A price or quantity dispute with the supplier is contested between buyer, payables and supplier. Rules take the match. A model extracts, codes and drafts supplier replies. People hold the two consequential calls.

Ardent Partners put the average cost at about $9.40 per invoice in 2025 and the cycle at 17.4 days, against 3.1 for the best. APQC reports a median of 75% of invoice lines matched first time and 12,000 invoices per payables employee a year (accessed 2026).

This is the most documented loop in the chain, as read in 2026. Uber's in-house system reached about 90% extraction accuracy and cut handling time by 70%. In AppZen stories, Applied Industrial Technologies runs 87% of invoices

without a touch, and Georgetown University auto-approves 66%. Basware lifted its own e-invoicing share from 69% to 99%. No named company reports results from Oracle′s or Ramp′s payables agents. As more invoices pass untouched, the remaining queue grows denser with disputes.



Like three spotlights on one stage of invoices: *value, effort and fraud each light a small corner, and a person′s time goes where the beams fall.*

Payment Run

At a glance	
Outcome	Approved invoices due in the period are paid once, on time, to verified accounts, and recorded
Shape of cases	Heavy-tailed (estimate): loss concentrated in a few redirected or duplicate payments
Decisions	1 mechanical · 1 bounded · 2 consequential · 1 contested
Hardest to undo	Releasing a payment to a changed or mismatched bank account
Anchor figure	Median of 1.5% of annual disbursements duplicate or erroneous (APQC, accessed 2026)

The payment run collects approved invoices, approved expense reimbursements paid through payables, verified bank details from onboarding, and the cash position from treasury. It sends funds and remittance advice to suppliers. Bank entries go on to account reconciliation and cash postings to journal entries.

No published evidence shows the value spread of payments, and it is assumed to follow the invoice skew. The risk spread is clearer. The Association for Financial Professionals (AFP) found that 79% of organisations faced attempted or actual payments fraud in 2024. Business email compromise,

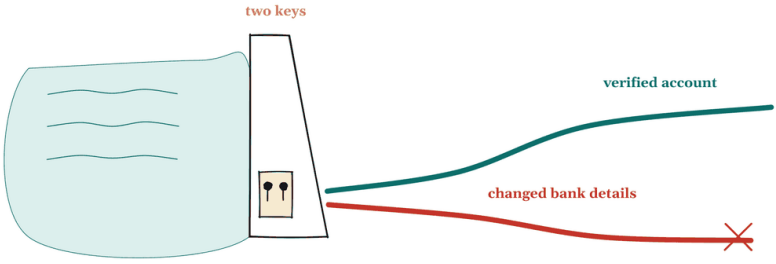
where a fraudster poses as a supplier or executive by email, was the top vector. A handful of redirected payments can outweigh a year of routine ones.

Sarbanes-Oxley requires release by authorised signatories independent of payables and vendor master, with dual sign-off above limits. In the euro area since October 2025, Verification of Payee has the bank check the payee name against the account. On a mismatch the payer must decide. Nacha's 2026 rules require fraud monitoring on ACH, the US bank-transfer network, OFAC screening applies, and UK large companies must publish their days-to-pay.

Selecting invoices due is mechanical. Choosing method and date is bounded by policy on discounts, card rebates and cash. Releasing the proposal to the bank is consequential, and so is holding or releasing a payment flagged for a bank change. Once wrongly released, such a payment is usually unrecoverable. Paying early, stretching or prioritising suppliers when cash is tight is contested between treasury and procurement. This loop has the least room for a model to act alone. Rules build the proposal. A model can score anomalies and duplicates. Release stays with named signatories, because nothing later in the chain can reverse it.

Grade each run on paid once, on time, to the right account. APQC's median of 1.5% duplicate or erroneous disbursements (accessed 2026) is the main error baseline. The AFP's 2022 survey, quoted by a vendor, put medians at $2.00 to $4.00 per check and $0.26 to $0.50 per ACH payment.

As read in 2026, no named company has published an automation or agent result specific to payment runs. Oracle's and Ramp's payment agents name no customers. No public volume or recent cycle-time benchmark exists for the run itself.



Like a dam's sluice that opens only on two keys: *water sent down a changed channel cannot be called back, so the release is held by named signatories.*

Employee Expense Claims

At a glance	
Outcome	An employee's spend is checked against policy and tax rules, approved, reimbursed or cleared, and posted
Shape of cases	Mixed (estimate): routine small claims with a misconduct tail
Decisions	2 mechanical · 1 bounded · 1 consequential · 1 contested
Hardest to undo	Treating a pattern of claims as misconduct
Anchor figure	$58 to process one expense report (GBTA Foundation, 2015, old data)

This is the chain's side entrance. Its inputs are the employee's profile, cost centre and approver from hire to retire, plus card transactions and travel bookings. Approved reimbursements leave through the payment run or through payroll. Expense postings and VAT reclaim data go to journal entries.

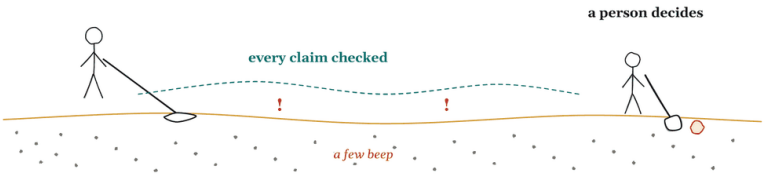
Most claims are small, routine and alike. The risk sits in a tail. Expense reimbursement schemes were 13% of occupational fraud cases in the ACFE's 2024 study, with a median loss of $50,000. Takeda auto-approves 63% of reports after an AI audit of every report, which implies that a minority truly need review. No published distribution of claim values was found.

Tax law sets the first design choice. Under US accountable plan rules, a reimbursement is tax-free only with business purpose, substantiation and return of any excess. A non-compliant claim becomes taxable wages. Sarbanes-Oxley requires an approver other than the claimant. The Foreign Corrupt Practices Act and the Sunshine Act require hospitality to officials and clinicians to be recorded and reviewed. Two EU rules apply to the fraud flags themselves. The AI Act may treat AI that monitors employee behaviour as high-risk, and GDPR Article 22 gives employees the right to human intervention.

Receipt matching and policy checks are mechanical. Auto-approving a low-risk claim is bounded by risk thresholds. A policy exception, such as an upgrade or a missing receipt, is consequential because it carries tax and audit effects. Treating a pattern as misconduct is contested, with HR, legal and works councils involved. A model can audit every line. People must hold any finding that reaches an employee's record.

The newest public figures are old. The GBTA Foundation found in 2015 that a report cost $58 and 20 minutes, and that 19% had errors or missing information.

As read in 2026, AppZen stories show Takeda live in 63 countries, with about 400,000 audits a year and 4,000 auditor hours saved each quarter. Electrolux reported 60% faster approvals and $200,000 less spend in a year. Databricks reported $483,000 saved annually. No named customer has published results from the Concur, Workday or Ramp expense agents. Once employees know every line is audited, the tail of deliberate misclaims may thin, leaving mostly honest errors.

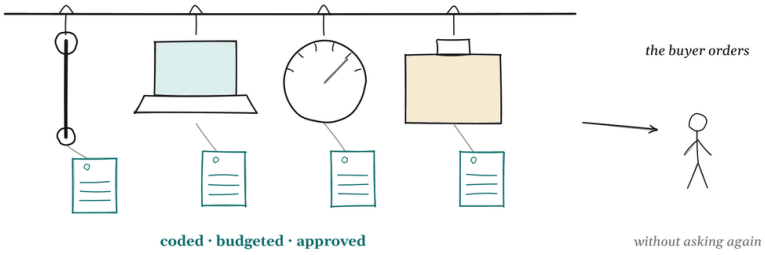


Like a metal detector sweeping a whole beach: *it checks every patch and beeps at a few, but only a person digs and decides what a find really means.*

LXXXIX

Purchase Requisition and Approval

At the composite manufacturer, the loop receives about 2,600 purchase requests a month from employees across three plants and the offices. Each request starts as a need: a spare part, a calibration service, a laptop, a consultant. The loop turns it into an approved requisition, coded to the right account and cost centre, within budget and policy. A buyer can then raise the purchase order without going back to the requester.



Like a dry cleaner's ticket on every item: *each need, from spare part to consultant, leaves tagged with code, budget and approval, so nobody asks again.*

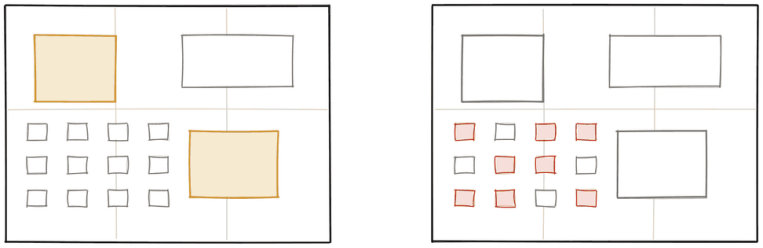
Step 1, Read: the map entry

The chief financial officer's map placed the loop at value four and readiness three. Coding history was rich and the delegation of authority was written down. But the catalogue covered only part of what plants bought, and one dispute between plants and procurement was still open.

Field	Entry
Loop	Purchase requisition and approval
Chain	Procure to pay
Value to the core	4: about 2,600 requests a month; about €210 million of spend a year; plant uptime depends on it
Readiness	3: coding history in place; approval matrix written; catalogue partial; owner willing; one open dispute over urgent plant purchases
Handoffs	In from employees and budget holders; out to purchase order issue and to supplier onboarding

Field	Entry
Risk category	Not high risk; approval limits and segregation of duties are audited
Owner	Head of indirect procurement (agreed)
Distribution read	By case: 64% of requests are catalogue or contract items with clear coding; 27% are free text under €5,000; 9% are above €5,000 or need a specialist review. By value: the 5% of requests above €25,000 carry 58% of value. By exception: free-text requests under €5,000 hold 68% of policy exceptions
Signed	Chief financial officer

The value tail and the risk tail were different sets. Large requests were few and well watched. Small free-text requests were many, and nobody looked at them closely. That was where off-contract buying, split orders and missing reviews accumulated.



Like two heat maps of the same town: *one lights the few big buildings where the money is, the other the many small houses where the trouble is.*

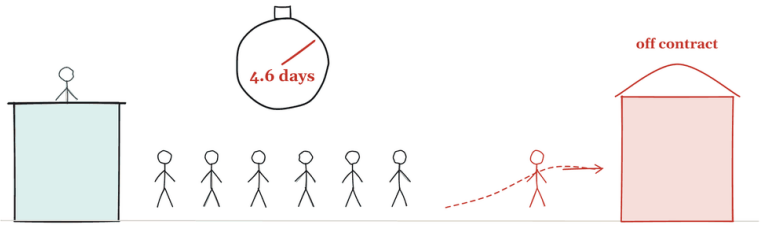
Step 2, Map: the baseline sheet

Mapping with buyers, procurement assistants and a sample of budget holders drew fourteen steps across six lanes: requester, budget holder, procurement, plant management, finance and the specialist reviewers.

Field	Entry
Sample and period	3,700 requests over six weeks
Volume	2,600 requests a month; about 260 a month declared urgent by plants
Fully loaded cost per requisition	€34.00, including approver and requester time
Requisition-to-order time	4.6 working days, median

Field	Entry
Share of spend through contracts	61%
Requests returned for recoding	17%
Urgent requests bought off contract	41% of urgent requests
People	4 procurement assistants, 7 indirect buyers, about 160 budget holders
Method agreed with finance	Yes, planning and analysis lead

Requisition-to-order time was more than double the published median. Requesters who waited four or five days for a spare part learned to buy around the process. That habit fed the off-contract share.



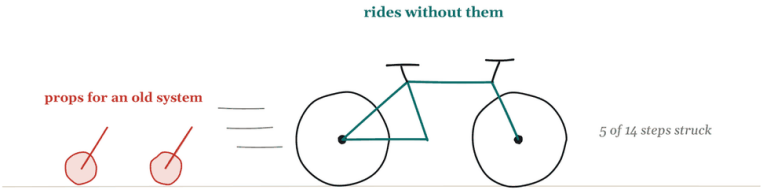
Like timing the queue at a slow counter: *the baseline clocked 4.6 days to an order, and showed people slipping off to buy around it.*

Step 3, Strip: the strip register

Step	Status	Constraint	Still applies?
Printed approval form signed by the plant manager for requests over €500	Struck	Workflow approval did not exist in the old purchasing system	No: approvals are recorded in the ERP since 2020
Buyer re-keys requests sent by email into the ERP	Struck	Requesters had no access to create requisitions	No: all employees can now raise requests
Site controller approves every free-text request under €1,000	Struck	Added after a 2017 incident with a fictitious supplier	No: vendor master controls now block unverified suppliers
Requester emails the buyer to ask for status	Struck	No visible status for requesters	No: status can be sent automatically
Monthly spreadsheet of open requests to plant managers	Struck	Plant managers had no report access	No

Step	Status	Constraint	Still applies?
Budget holder approves the spend	Kept	Commits company money and must be authorised	Yes
Check the request against contracts and catalogue	Kept	Required by the outcome	Yes

Five of fourteen steps were struck, a strip ratio of 36%. Three of the five were approvals or reports that existed only because a system once lacked a feature.



Like taking the stabilisers off a bike: *five of fourteen steps were props added when an old system could not balance alone, and the loop rides fine without them.*

Step 4, Sort: rules, decisions and the contested log

The rule register held 52 rules. Eight contradicted each other. The two that mattered most are shown.

Rule	Source	Contradiction	Decision
Three quotes above €5,000	Group procurement policy	One plant applied the threshold at €10,000 under a local procedure	Head of indirect procurement: €5,000 stands everywhere; requests against framework contracts are exempt
Plant managers approve up to €25,000	Delegation of authority	IT purchases went to the IT director at any amount	Financial controller: approval follows value only; IT categories trigger a security review instead of an extra approval

The decision register held fifteen decisions.

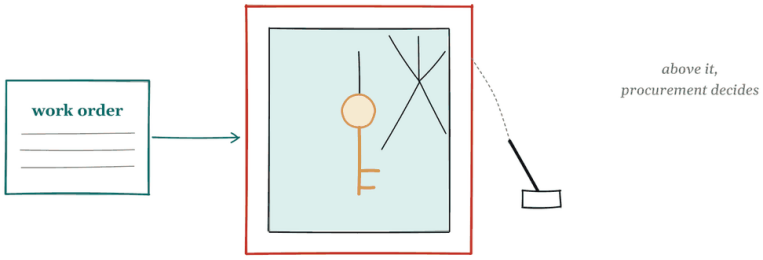
Decision	Class	Limit or junction
Is the item on a catalogue or contract?	Mechanical	
Route to catalogue, contract or free-text buying	Mechanical	

Decision	Class	Limit or junction
Which approver does the matrix require for this amount?	Mechanical	
Release a catalogue item under €500 within budget	Mechanical	Standing approval in the delegation of authority
Assign account, cost centre and tax code	Bounded	Above a confidence threshold set in shadow; new categories to a buyer
Confirm budget on the cost centre	Bounded	Overrun up to 5% flagged to the controller; more to the budget holder
Propose a contract item in place of a free-text request	Bounded	Where the specification matches; the requester may decline with a reason
Does the request need legal, security or privacy review?	Bounded	By category and data-access rules; borderline cases to the reviewer
Approve the spend	Consequential	Budget holder under the matrix; chief financial officer above €250,000
Approve an urgent purchase outside policy from a plant	Contested	See log

A maverick purchase is one made outside policy: from a supplier with no contract, or without the required quotes. Plants called these urgent. Procurement called them leakage. The decision could not be designed until leaders settled it.

Field	Entry
Decision	Do we approve an off-contract purchase when a plant says it is urgent?
Positions	Operations: approve, because a stopped line costs more than any price premium. Procurement: never, because urgency is claimed far more often than lines stop
Logged	Week 1 of Sort
Settled by	Chief operating officer and chief financial officer
Answer	Approve up to €10,000 when the plant manager cites a maintenance work order for a stoppage or safety risk, and the contract supplier cannot deliver in time. Above €10,000, the head of indirect procurement decides within four hours. Every case is reviewed monthly by plant
Date settled	Week 4 of Sort
	Bounded up to €10,000 with a work order; consequential above, with a junction

Field	Entry
Resulting class	



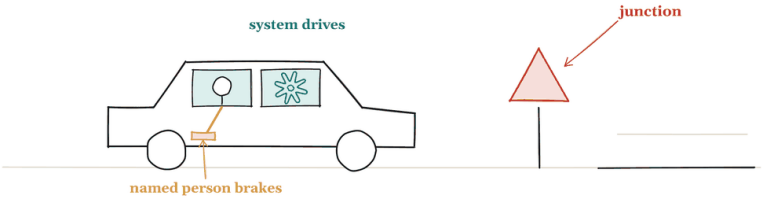
Like a break-glass key box: *a plant may take the urgent key without procurement only when a work order proves a stoppage, and only up to €10,000.*

Step 5, Design: junctions and the standard

Junction	Who decides	Evidence shown	How reversed
Spend above the approver's limit	Next budget holder in the matrix; chief financial officer above €250,000	Request, budget position, quotes, contract status	Requisition returned before any order is raised
Legal, security or privacy review	Legal counsel, information security lead or data protection officer	Category, supplier, data the supplier will access, draft terms	Review reopened; order held
Urgent off-contract purchase above €10,000	Head of indirect procurement	Work order, contract supplier's delivery date, price against contract	Order cancelled or moved to the contract supplier
Switch-off	Head of indirect procurement	Grading dashboard	All requests return to buyers within 30 minutes

Field	Entry
Criteria	Each requisition correctly coded, within budget, approved by the right person, on contract where one exists, and reviewed where rules require; body requests ready for an order within one working day
Rule checks	Every approval checked against the matrix; no requests split to stay under a limit
Model grader	Every requisition graded against the criteria
Human audit	3% at random; 15% of urgent and free-text requests

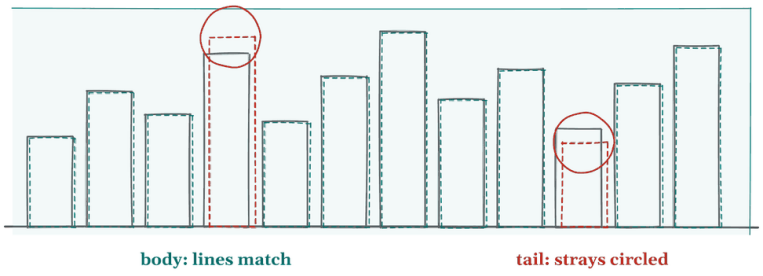
Field	Entry
Requester signals	Redirects declined; complaints about delay
Calibration	Monthly, by the procurement quality analyst



Like a dual-control driving car: *the system drives the routine road, a named person brakes at each junction, and the instructor can take the wheel within minutes.*

Step 6, Shadow: the report

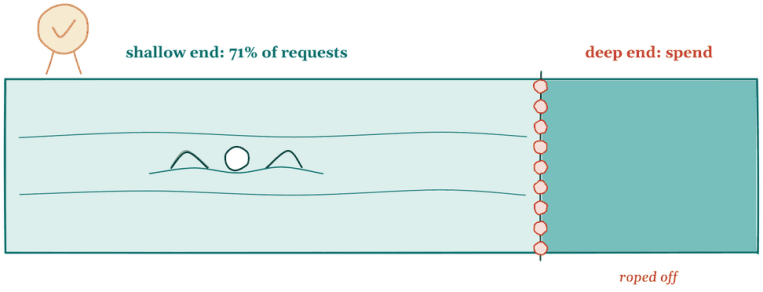
Field	Entry
Period and cases	Six weeks; 3,600 requests
Agreement, body	98.4%
Agreement, tail	84%: free-text requests, urgent plant requests and review triggers
Disagreements	Executor wrong 49%: services coded as materials; laboratory software not flagged for security review. Person wrong 34%: buyers still applying the €10,000 quote threshold. Both defensible 11%. Standard unclear 6%
Changes made	Three coding rules added; software categories mapped to security review; quote threshold briefed to buyers
Known weaknesses	Requests describing a part by drawing number rather than catalogue number
Recommendation	Pass G1. Signed by the head of indirect procurement and the procurement quality analyst



Like tracing paper laid over a master drawing: *the copy matched almost every line of routine requests, and the few strays in the tail were circled and fixed.*

Step 7, Bound: the gate review record

Field	Entry
Scope alone	Catalogue and contract requests within budget; coding above the threshold; proposed contract redirects; review triggers. 71% of requests and 23% of value
Evidence	Four weeks bounded: 99.1% of bounded requisitions met the standard; no split request passed undetected
Permissions	Create and code requisitions, route to approvers, release catalogue items under €500, propose redirects, open reviews. No permission to approve spend, add suppliers or change budgets
Switch-off test	Week 1, before the first live case: work returned to buyers in 20 minutes; nothing lost
Audit consulted	Internal audit: keep the approval trail and any declined redirect with each requisition
Decision	Pass G2. Signed by the financial controller



Like a swimming badge: *four clean weeks earned the system the shallow end alone, most requests, while the deep end of spend approval stays roped off.*

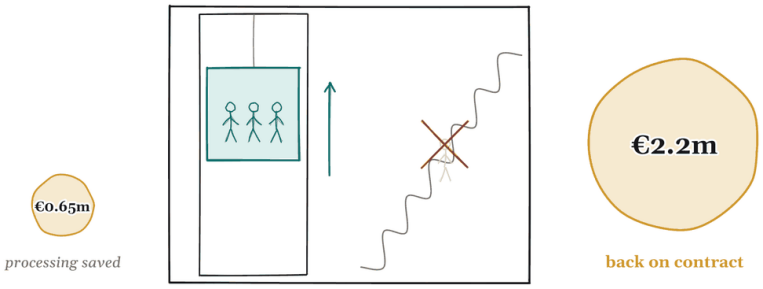
Step 8, Own: charter and economics

Field	Entry
Loop owner	Head of indirect procurement
Junction holders	About 160 budget holders; chief financial officer; legal counsel; information security lead; data protection officer
Exception owner	Senior buyer, exception queue cleared within one working day
Grader	Procurement quality analyst
Adoption owner	Procurement operations manager
Control owner	Financial controller
Changes to work	Procurement assistants from four to one, on exceptions. Two moved to the supplier onboarding loop; one joined category management to extend contract coverage. Buyers spend their time on free-text and urgent requests
Learning path	New buyers work the exception queue for two months and sit in on urgent-purchase junctions

Field	Entry
Period	Eight weeks across a quarter-end
Cost lines	Platform and model; grading and audit; junction time; exception queue; adoption
Cost per accepted requisition	€13.20 overall; body €7.40; tail €26.70
Baseline	€34.00
Requisition-to-order time	4.6 working days to 1.3 overall; body 0.4, tail 2.9
Share of spend through contracts	61% to 76%
Urgent requests bought off contract	41% to 15%
Annualised effect	About €650,000 a year in processing cost. Fifteen points of contract coverage on €210 million moves about €31.5 million onto contract; at a measured 7% premium for off-contract buying, about €2.2 million a year

Field	Entry
Signed	Planning and analysis lead

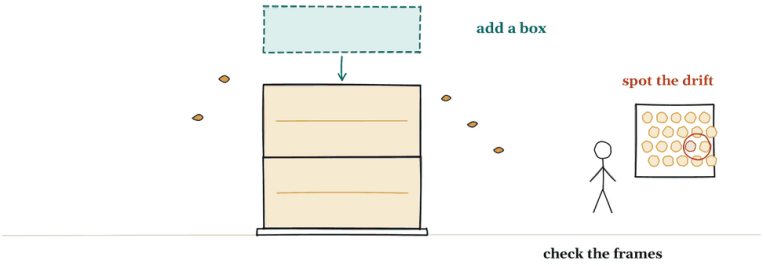
The processing saving was the smallest number on the sheet. Faster orders removed the reason to buy around the process. The contract share then rose on its own.



Like fixing a slow lift: *the repair saved a little on upkeep, but the larger gain came from people no longer taking the side stairs around it.*

Step 9, Core: an operating review

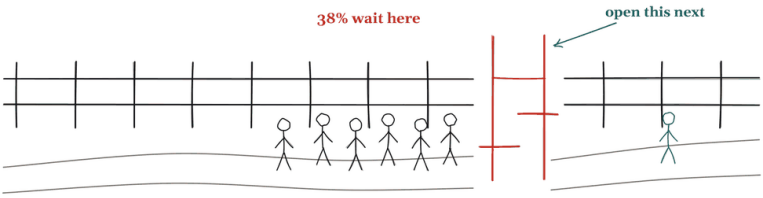
Field	Entry
Month	Second month in Core
Standard met	Body 99.3%; tail 92%
Cost per accepted requisition	€12.80
Exception queue	Oldest case 1 day. Repeated exception: calibration services coded to maintenance; rule added
Junctions	The head of indirect procurement refused one urgent purchase above €10,000 in five, because the contract supplier could deliver in time
Drift	A contract supplier sent its new price file late; redirects quoted old prices for two days until the file was loaded
Requester signals	Plant engineers declined 9% of redirects, mostly for specification reasons; two declined items added to the catalogue
Decisions	Widen bounded scope to free-text requests under €2,000 with coding above the threshold, agreed by the financial controller



Like a beekeeper's monthly hive check: *lift the frames, note what is healthy, spot what has drifted, and add a box when the colony is ready to grow.*

Step 10, Scale: what the edges showed

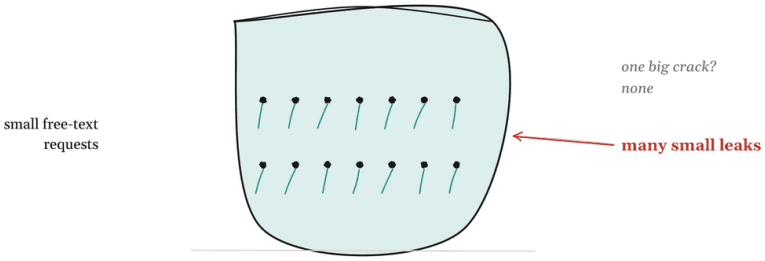
Reading the exception queue by cause showed that 38% of tail cases asked for a supplier not yet in the vendor master. Most were small, one-off suppliers chosen by plants. Those requests stalled at the handoff to supplier onboarding. The map review chose supplier onboarding as the next loop in the chain. Purchase order issue, which now received cleaner requisitions, rose in readiness and was marked to read after it.



Like walkers bunching at one stile on a footpath: *38% of stuck cases waited at the handoff to supplier onboarding, which showed which loop to open next.*

What this loop teaches

The tail of a loop is not always where the money is. Here it was a mass of small free-text requests, each too minor to check and together the main source of leakage. The contested decision was operational, between plants and procurement, and it was settled by tying urgency to evidence that already existed: a work order. The junctions for spend above limits and for specialist reviews kept every commitment of money with a named person. And the economics had to show time and contract coverage. A cost-only sheet would have missed both, and with them most of the effect.

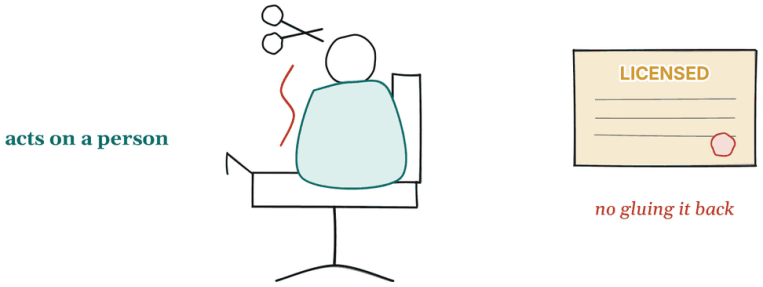


Like a leaking bucket: *the water was lost through many pinholes, not one big crack, so the tail here was the mass of small requests.*

XC

The Hire-to-Retire Chain

Hire to retire is the chain that brings a person into the company, pays and supports them while they work, and closes their employment when they leave. It runs from opening a job requisition, through screening, offer and onboarding, to payroll changes, leave, HR questions and offboarding. Every loop in it acts on a person rather than a document. That makes it the most regulated chain in the book, and the one where a wrong decision is hardest to reverse.



Like a barber's chair: *every cut lands on a person rather than on cloth, so the trade is licensed, and a wrong snip cannot be glued back.*

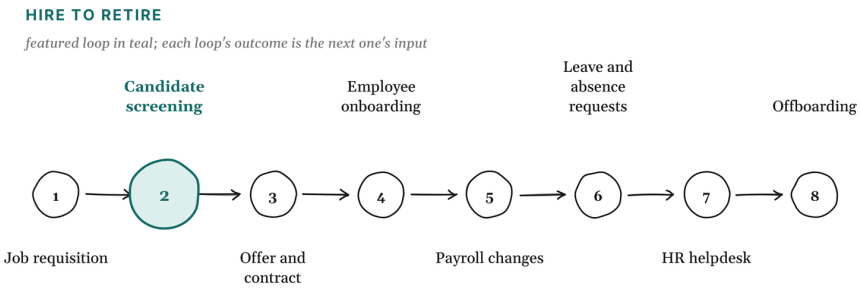


FIGURE 76 The hire to retire chain. Its loops in handoff order, with the featured loop, candidate screening, in teal.

The loops and the evidence

The table gives, for each loop in the chain, the shape of its cases and a published benchmark. Shapes marked as estimates are reasoned from the

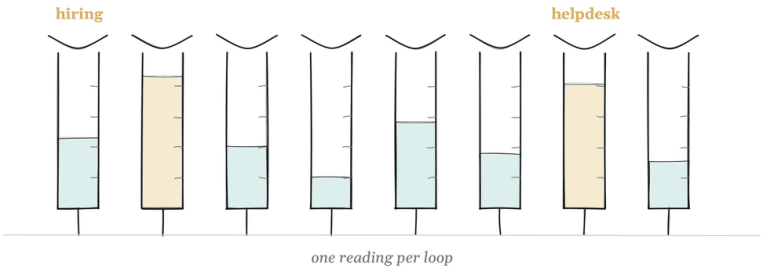
nature of the work where no published distribution was found. Benchmarks in US dollars are left in dollars.

Loop	Shape of cases	Published benchmark
Job requisition	Mixed	Median of 30 days from requisition approval to offer acceptance (APQC); median cost per hire of $1,200 for nonexecutive and $10,625 for executive roles (SHRM, 2025)
Candidate screening	Heavy tail (estimate)	Median of 50 applications per approved requisition (APQC)
Offer and contract	Mixed	Median of 30 days from requisition approval to offer acceptance, the end point of this loop (APQC)
Employee onboarding	Normal (estimate)	No public benchmark found
Payroll changes	Mixed	About one payroll in five contains errors, at an average of $291 per error (EY survey for Paycom, 2022)
Leave and absence requests	Mixed	Absence rate of 3.2% for full-time employees (US Bureau of Labor Statistics, 2024); 721 leave-related payroll errors per 1,000 employees a year (EY survey for Paycom, 2022)
HR helpdesk	Mixed	Median of 68.1% of pay and benefits questions are routine (APQC)
Offboarding	Mixed	58% of former employees said their old passwords still worked (PasswordManager.com survey, 2023)

Named public deployments in the chain are concentrated in frontline hiring and the HR helpdesk. Results are as reported by the companies or their vendors.

Loop	Company	Reported result
Candidate screening	Chipotle Mexican Grill	Application to start from 12 days to 4; application completion from 50% to over 85% (vendor-reported)
Candidate screening	Capita plc	Time to hire down 43%, as quoted by the vendor
Candidate screening	Hungry Jack's	Hiring that took three to four weeks can take as little as 24 hours (vendor-reported)
HR helpdesk	IBM	94% of common questions answered without an adviser; 40% lower HR operating cost over four years

The featured loop is candidate screening for plant operators. It is the regulated loop of the book. The law fixes some of its decision classes before the company sorts anything. Its contested decision is settled by agreeing criteria rather than limits, and its value comes from speed and consistency, not from fewer people.



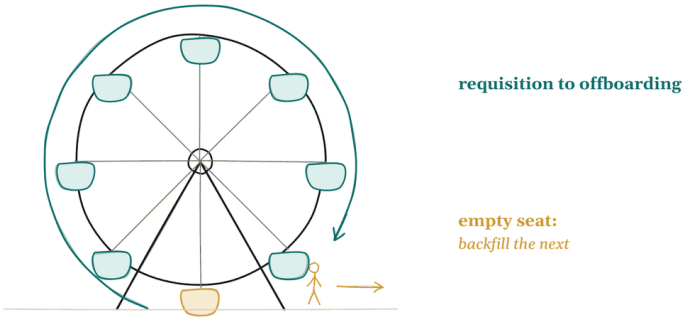
Like a row of rain gauges along a valley: *each loop gets its own reading, and the fullest gauges, frontline hiring and the helpdesk, show where deployments landed.*

XCI

The Loops of Hire to Retire

Hire to retire carries a person through the company. A job requisition opens a role and hands it to candidate screening. Screening passes one person to offer and contract, and the signed contract goes to onboarding. While the person works, payroll changes, leave requests and the HR helpdesk keep pay, time and answers correct. Offboarding closes the record and sends a backfill trigger back to the requisition, so the chain is a circle rather than a line.

The eight entries below share one trait. Every case is a person, and almost every important decision has a legal effect on that person. The entries differ most in where the hard decisions sit. In the hiring loops they sit early, at the choice of who goes forward. In the running loops they sit in a thin layer of bank-detail changes, medical leave and dismissals beneath a large routine body.



Like a Ferris wheel: *a person boards at the requisition, rides through offer, pay and leave, and the seat they leave sends the next one round.*

Job requisition

At a glance	
Outcome	An approved, budgeted, posted requisition with job description, pay range and named hiring manager
Shape of cases	Mixed: a repeatable nonexecutive body, a costly executive tail
Decisions	1 mechanical · 2 bounded · 1 consequential · 0 contested
Hardest to undo	Committing to a redesigned role and its multi-year salary cost

At a glance	
Anchor figure	Median cost per hire of $1,200 nonexecutive and $10,625 executive (SHRM, 2025)

A requisition is the formal request to fill a role. The loop turns an approved headcount plan, or a backfill trigger from offboarding, into a posted vacancy. Its result is a requisition with current criteria and a pay range, handed to candidate screening.

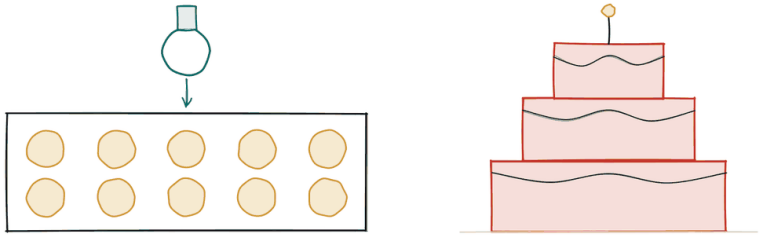
The cases split into two groups. Most requisitions are frontline and nonexecutive roles that repeat with small changes. A few are executive roles, and SHRM's 2025 medians show each one costing about 8.9 times a nonexecutive hire. That gap comes from seniority and search effort, not from chance. The body can run on templates; the executive tail cannot.

Several rules shape the design. The EU pay transparency directive, due for transposition by 7 June 2026, requires the pay range to be disclosed before interview and the pay criteria to be explainable. In Germany, section 99 of the Works Constitution Act lets the works council, the elected employee body, refuse consent to each hire. Checking a role against budget is mechanical. Setting grade and pay range, and wording the ad, are bounded. Replacing a role like-for-like or redesigning it is consequential.

Written rules can confirm budget and cost centre and fill the requisition from the grade structure. A model can draft the job description and the ad. People should hold the redesign decision, because it shapes a team for years. Targeted ad placement by AI counts as high-risk under the EU AI Act, so a person must be able to override it.

Grade the outcome on completeness and approval before posting. APQC puts the median from requisition approval to offer acceptance at 30 days, without stating a data year. SHRM's 2025 report gives a median time-to-fill of about 45 days, rising to 61 days in organisations above 5,000 employees. A recruiter carries a median of 20 requisitions a year.

Capita reports time to hire down 43% using the HiredScore model now sold as Workday's Recruiting Agent, as quoted by Workday (as read in 2026). No one has published an agent that drafts and routes requisitions end to end. No benchmark isolates approval time on its own.



Like a cookie cutter beside a wedding-cake order: *frontline roles stamp out from one template, while each executive role is a custom build costing about nine times more.*

Candidate screening

At a glance	
Outcome	A shortlist of assessed candidates, with a documented select or reject decision for every applicant
Shape of cases	Heavy-tailed (estimate): a few postings draw most applications
Decisions	2 mechanical · 0 bounded · 2 consequential · 1 contested
Hardest to undo	Rejecting a candidate, or acting on a background-check finding
Anchor figure	Median of 50 applications per approved requisition (APQC, data year not stated)

Screening takes a posted requisition and returns a shortlist with interview evidence to the hiring manager. The selected candidate moves to offer and contract with consent for checks. Rejected applicants become records kept for retention and audit.

No published distribution of applications per posting was found, so the shape is an estimate. APQC's median is 50 applications, but popular corporate roles and high-volume frontline hiring draw far more. Attention compounds here. A well-placed posting gets seen and shared, and the pile grows. The risk does not follow volume. It sits in the few rejections that a candidate or regulator later challenges.

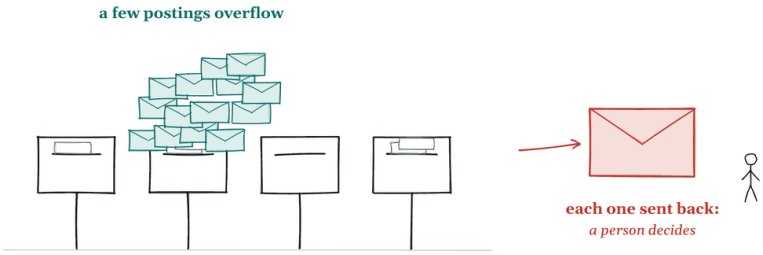
This loop is where the law reaches furthest. The EU AI Act lists AI that filters or evaluates applicants as high-risk. Its obligations apply from 2 August 2026, with a delay to late 2027 proposed in November 2025, as read in 2026. Article 22 of the GDPR bars a rejection based solely on automated processing. New

York City requires a yearly independent bias audit of hiring tools. Colorado, Illinois and the US Fair Credit Reporting Act add notice, consent and human-owned adverse-action steps.

Knock-out checks, such as right to work, licence and location, are mechanical. So is interview scheduling. Rejection and acting on background checks are consequential. Ranking candidates is contested, because most companies have never agreed what a fair ranking means or how much disparate impact, an uneven effect on protected groups, they will accept.

Written rules take the knock-outs and the calendar. A model can summarise applications against agreed criteria and answer candidate questions. People must own every rejection and the shortlist itself.

The best-documented results are frontline. Chipotle cut application-to-start time from 12 days to 4 and lifted application completion from 50% to over 85% with a Paradox assistant (vendor-reported, as read in 2026). Hungry Jack's reports hiring in as little as 24 hours instead of three to four weeks. These results are vendor-reported, and no independent audit was found. Faster screening also draws more applicants, which thickens the tail it was built to manage. The next chapter follows this loop through its whole lifecycle.



Like letterboxes on one street: *a few popular postings overflow with applications, yet the real risk sits in each single envelope sent back, which a person must decide.*

Offer and contract

At a glance	
Outcome	A signed offer and contract on approved terms, checks cleared, start date confirmed
Shape of cases	Mixed: templated offers at grade, a costly negotiated minority
Decisions	1 mechanical · 1 bounded · 2 consequential · 1 contested

At a glance	
Hardest to undo	Withdrawing an offer after checks
Anchor figure	Median of 30 days from requisition approval to offer acceptance (APQC, data year not stated)

This loop turns a selected candidate into a signed employee. It receives the interview record and check consents from screening. It sends the contract and start date to onboarding, and the agreed pay to payroll changes.

Most offers come from a template at a fixed grade. A minority are executive, negotiated or cross-border, and they carry most of the cost and legal risk. The same 8.9-to-1 cost ratio seen in requisitions explains why. Each extra term negotiated adds legal judgement that the template does not hold.

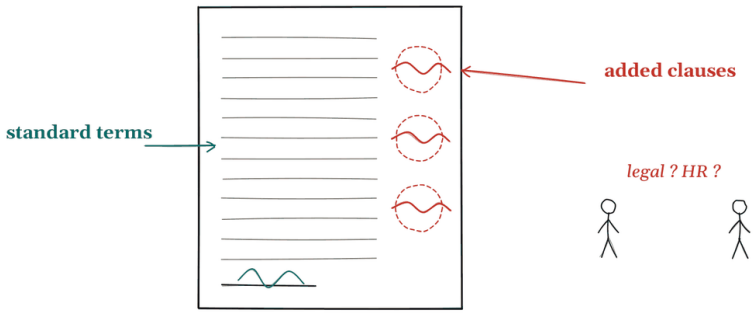
The EU directive on transparent working conditions sets the essential terms an employee must receive in writing within seven days of starting. Pay transparency makes placement within the band explainable on request. Withdrawing an offer after a background check triggers US adverse-action steps and GDPR Article 22. In Germany, the works council must again be informed of the hire.

Generating the letter is mechanical. Choosing a salary inside the band is bounded. Approving an out-of-band offer or sign-on bonus is consequential, because it sets a pay precedent. Non-standard clauses, such as non-competes, relocation or equity, are contested. Their enforceability varies by jurisdiction, and legal and HR rarely share one written position.

Written rules can assemble the offer from the requisition and grade and check it against the band. A model can draft the essential-terms statement and flag clauses that depart from standard. A named approver should sign every out-of-band offer. A person must decide every withdrawal.

APQC's 30-day median ends at offer acceptance, which marks this loop's finish line. SHRM's 2025 medians put cost per hire at $1,200 for nonexecutive and $10,625 for executive roles. No benchmark body publishes an offer acceptance rate or offer-to-start time.

No company has publicly reported an agent that drafts and negotiates offers. Oracle announced an employee contracts analyst agent in February 2025, but no customer story was found (as read in 2026).



Like a printed lease with handwritten clauses in the margin: *the standard form signs itself, but each added clause needs a judgement that legal and HR rarely share.*

Employee onboarding

At a glance	
Outcome	A new employee verified, on payroll, equipped, enrolled in benefits and inducted by day one
Shape of cases	Normal (estimate): templated by role and country
Decisions	1 mechanical · 2 bounded · 1 consequential · 0 contested
Hardest to undo	Signing off identity and right-to-work verification
Anchor figure	No reliable published figure for the loop; 65.3 million US hires in 2024 (BLS JOLTS) gives the scale

Onboarding receives a signed contract and start date and delivers a working employee on day one. It sends tax forms and bank details to payroll changes, a profile to the HR helpdesk, and a joiner request to IT for accounts and equipment.

The cases look alike. Onboarding tasks are largely set by role and country, and the variations come from many small, independent causes. That points to a normal shape, though no published distribution confirms it. The tail is narrow but real: cross-border hires, regulated roles and executives each add steps that a template misses.

The legal core is verification. In the US, an employer representative must examine identity and work-authorisation documents for Form I-9 within three business days of the start. UK employers must check right to work before

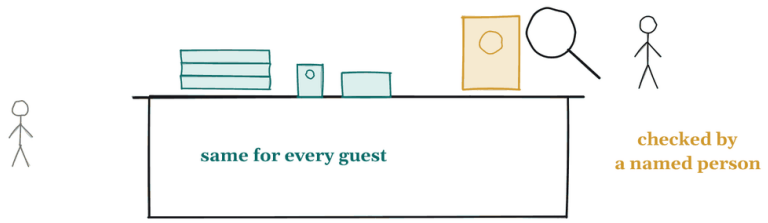
employment begins to keep a statutory excuse, their defence against illegal-working penalties. The GDPR requires a data protection impact assessment before new technology processes employee data at scale. German works councils co-determine any system able to monitor staff.

Creating accounts and issuing standard equipment is mechanical. Granting elevated access and choosing the training plan are bounded. Document verification is consequential because the law names a person to do it.

Rules can run the role-based provisioning and the task checklist. A model can guide the new hire through forms and answer first-week questions. A trained person signs the verification, and an access owner approves sensitive rights.

Grade the outcome on day-one readiness. Here the evidence is thin. No public benchmark exists for onboarding cycle time, cost per new hire or day-one readiness; APQC's time-to-productivity values are member-only. The US Bureau of Labor Statistics counted 65.3 million hires in 2024, a 3.4% monthly hires rate, which shows the volume at stake.

Oracle has announced a new-hire onboarding assistant, but no named deployment with results was found (as read in 2026). A company starting this loop will be building its own baseline.



Like a hotel check-in: *the room, key and towels follow one routine for every guest, but the passport must be examined by the named person at the desk.*

Payroll changes

At a glance	
Outcome	A validated change to pay, deductions, tax or bank details applied before the cut-off
Shape of cases	Mixed: a routine body, a high-risk layer of bank and pay-rate changes
Decisions	1 mechanical · 2 bounded · 2 consequential · 0 contested
	Paying to a changed bank account

At a glance	
Hardest to undo	
Anchor figure	About 1 in 5 US payrolls contains errors, averaging $291 each (EY survey for Paycom, 2022)

Payroll changes is the junction box of the chain. Pay terms arrive from offers, tax and bank details from onboarding, leave from the absence loop, and final pay from offboarding. The loop validates each change and applies it before the payroll cut-off, so the next run pays the person correctly.

Errors are concentrated by type. Per 1,000 employees a year, the EY survey for Paycom counted 1,139 time, attendance and expense errors and 721 leave-related errors, against 159 direct-deposit errors. Volume and risk part ways here. Bank-detail and pay-rate changes are rare, but they are the main route for payroll fraud.

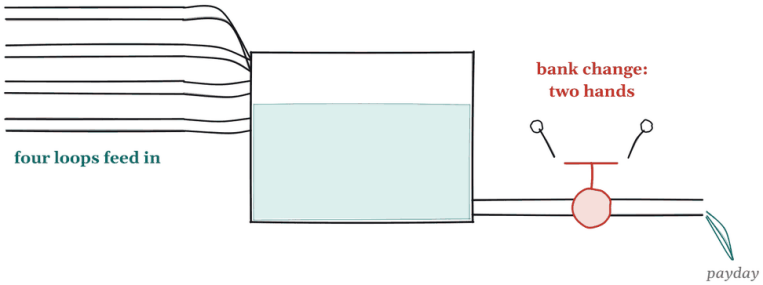
Sarbanes-Oxley section 404 governs the design for US-listed companies. It requires segregation of duties, meaning the person who requests a pay-rate, bank or new-payee change cannot also approve it. IRS rules leave withholding to the employee's own W-4 election. Pay transparency and GDPR health-data rules add explanation and privacy duties.

Applying an employee's own tax election is mechanical. Validating changes against cut-off and policy is bounded, as is a retroactive correction. Changing bank details and approving pay-rate or off-cycle payments are consequential.

Rules can enforce cut-offs, limits and the separation between requester and approver. A model can explain a payslip, spot anomalies before the run and draft corrections. People approve bank changes after verified identity and sign off pay-rate changes.

Grade every change on paid-correctly-first-time. The EY survey, from 2022 and covering US firms of 250 to 10,000 employees, found 15 corrections per pay period per organisation. It also found 229 W-4 errors per 1,000 employees a year. APQC puts the median at 15,000 disbursements per payroll employee.

IBM's AskHR agent runs manager workflows for compensation changes in SAP SuccessFactors and serves payslips. IBM reports 40% lower HR operating cost over four years (as read in 2026). Workday and SAP have announced payroll agents, but no deployment with results has been published. No European error benchmark from 2023 or later was found.



Like a water tank fed by four pipes: *each inflow is checked before payday, and the valve that redirects the outflow needs two separate hands to turn.*

Leave and absence requests

At a glance	
Outcome	A leave request approved or declined against entitlement, law and staffing, and recorded correctly
Shape of cases	Mixed: short routine absences, a heavier tail of statutory leave
Decisions	1 mechanical · 1 bounded · 1 consequential · 1 contested
Hardest to undo	Refusing or interfering with statutory leave
Anchor figure	Absence rate of 3.2% for full-time US workers (BLS Current Population Survey, 2024)

Leave requests usually arrive through self-service or the HR helpdesk. The loop checks each one against balance, law and cover, then records the decision. Approved leave that affects pay goes to payroll changes, and the absence goes to workforce scheduling.

Two populations share the queue. Most requests are short planned or sick absences; the US absence rate was 3.2% in 2024. A smaller group is statutory leave under the Family and Medical Leave Act (FMLA), the US law on protected unpaid leave. About 15% of employees took FMLA-qualifying leave in a year, averaging about 28 business days, according to a US Department of Labor survey from 2018. That group carries most legal and cost risk.

The FMLA sets a five-business-day deadline for the eligibility notice and makes interference a liability. The EU Working Time Directive guarantees four weeks of paid leave. Under EU case law, leave cannot lapse unless the

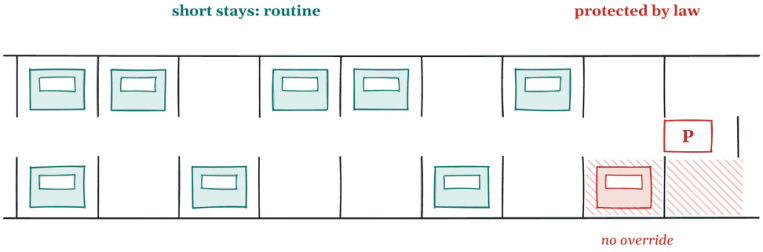
employer actively enabled the worker to take it. Medical certificates are special-category health data under the GDPR.

A routine vacation request within balance is mechanical, and IBM already approves these automatically. Statutory eligibility is bounded by fixed rules and deadlines. Deciding on medical certification is consequential. Refusing or moving leave for business needs is contested, because it sets operations against entitlement and, in Germany, works-council rules.

Rules can run balances, blackout dates and eligibility tests with their deadlines. A model can answer policy questions in the employee's own country. A person must decide any medical or disputed case and any refusal.

Measure leave decisions recorded correctly the first time. The EY survey for Paycom found 721 leave-related payroll errors per 1,000 employees a year in 2022, costing about $220,000. No benchmark body publishes cost per request or approval time.

IBM and Microsoft both handle leave through agents in production (as read in 2026). Microsoft's figures are targets, not results: 44% fewer monthly HR tickets by mid-2026. When requests get easier to file, more short absences are recorded properly, which can raise the apparent volume of the body.



Like a car park: *short-stay bays fill and empty all day without fuss, while a few protected bays carry rules no attendant may override for convenience.*

HR helpdesk

At a glance	
Outcome	An employee question or HR case answered correctly against current policy, first time, and logged
Shape of cases	Mixed: about two-thirds routine, a sensitive minority
Decisions	2 mechanical · 1 bounded · 0 consequential · 1 contested

At a glance	
Hardest to undo	Mishandling a harassment, grievance or whistleblowing contact
Anchor figure	68.1% of pay and benefits inquiries are routine (APQC median, data year not stated)

The helpdesk takes questions from every loop in the chain. It answers what it can and routes the rest: corrections to payroll, leave requests to the absence loop, and sensitive cases to employee relations.

APQC's median shows 68.1% of pay and benefits inquiries are routine. The rest are complex or sensitive. The split follows the subject matter. Policy questions repeat because many employees share the same policy, while personal disputes are each unique.

Few laws dictate the routine answer. The weight falls on privacy and fairness: GDPR health-data rules and the impact assessment, Article 22 on automated decisions, and German co-determination over any system that could monitor staff. If the helpdesk begins to decide on terms of work, the EU AI Act's high-risk rules apply.

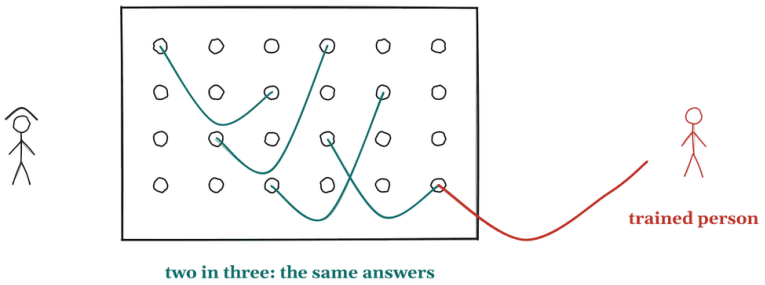
Answering from the knowledge base and issuing a standard employment letter are mechanical. Routing a case by topic and sensitivity is bounded. Handling harassment, grievance or whistleblowing contacts is contested. Companies rarely agree how quickly such a contact must reach a trained person, or what an automated first reply may say.

This loop suits a model better than most. A model can retrieve the right country's policy, issue letters and complete simple transactions. Written rules should send any sensitive keyword straight to a person, with no automated judgement in between.

Grade first-time resolution and correct routing. Containment, the share of questions resolved without a human adviser, is the main operating figure. A ScottMadden and APQC study from 2017 put staffing at about 600 employees per service-centre employee. The Hackett Group reported in 2025 that top HR organisations run 29% cheaper than peers, a relative figure only.

IBM reports 94% containment, more than 2.1 million conversations a year and 75% fewer tickets since 2016 (as read in 2026). Microsoft has rolled out its agent to over 200,000 employees, with ticket reductions so far stated as

targets. No public benchmark from 2023 or later gives cost per case. As containment rises, the cases left for people grow harder on average, so adviser skills must rise with it.



Like a telephone switchboard: *two calls in three are answered from the same sheet, and a sensitive call is patched straight through to a trained person.*

Offboarding

At a glance	
Outcome	A leaver with access revoked, assets returned, final pay settled on time and records closed
Shape of cases	Mixed: routine quits, a riskier tail of dismissals and layoffs
Decisions	1 mechanical · 1 bounded · 1 consequential · 1 contested
Hardest to undo	A dismissal, and late revocation of access
Anchor figure	63.2 million US separations in 2024 (BLS JOLTS)

Offboarding starts with a resignation, retirement or termination decision. It ends with access revoked, equipment back, final pay settled and the record closed. It sends final pay to payroll changes, a leaver request to IT and a backfill trigger to the job requisition, which closes the circle.

The US Bureau of Labor Statistics counted 63.2 million separations in 2024. About 62% were quits, which are largely routine. Layoffs and discharges made up 20.2 million and carry most legal risk. Mass layoffs form their own tail, because crossing a size threshold switches on notice and consultation duties.

Thresholds and deadlines dominate the rules. The US WARN Act requires 60 days′ notice of a covered mass layoff, and the EU directive on collective redundancies requires consultation first. In Germany, a dismissal without

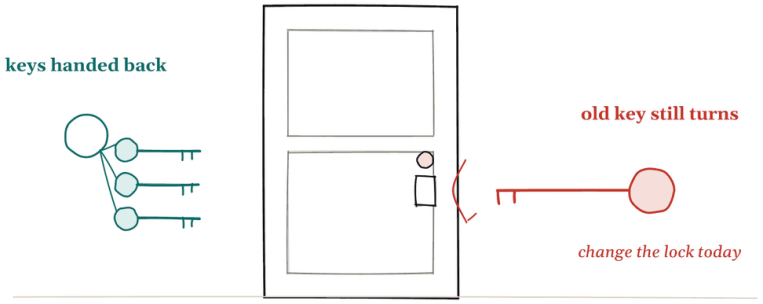
hearing the works council is void. California makes final wages due immediately on involuntary termination, with waiting-time penalties. For platform workers, the EU Platform Work Directive requires a human to decide any account termination.

Revoking access on the leave date is mechanical. Calculating final pay is bounded. Approving a settlement is consequential. The dismissal itself is contested: few companies have agreed what evidence an automated system may contribute. It is high-risk under the EU AI Act and falls under GDPR Article 22.

Rules should fire revocation and asset recovery from the termination record without waiting for anyone. A model can assemble the final-pay calculation for the right jurisdiction and draft the leaver's letters. People hold the dismissal and every settlement.

Grade offboarding on access closed and final pay paid by the legal deadline. The mechanical step is where failure is common. In a 2023 survey of 1,000 US workers by PasswordManager.com, 58% said former employers' passwords still worked, and 47% had used them. That figure comes from a vendor survey, not a benchmark body.

No named deployment of an offboarding agent with results was found, and no benchmark exists for cycle time or cost per leaver (as read in 2026).

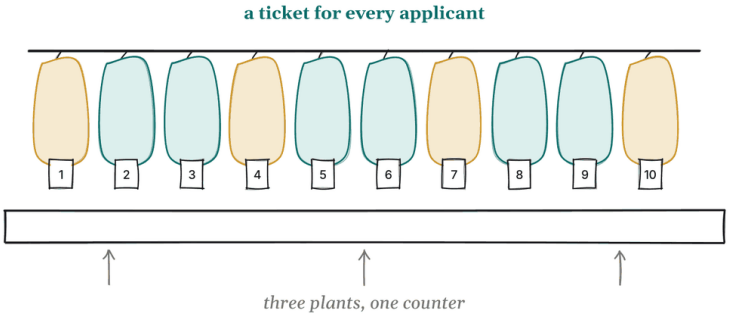


Like moving out of a flat: *return the keys, settle the last bill and change the lock that day, because old keys that still turn are the common failure.*

XCII

Candidate Screening

At the composite manufacturer, the loop receives applications for about 450 plant operator hires a year across three plants in three countries. It checks each application against the published requirements, keeps candidates informed, books interviews and gives recruiters the evidence they need. Its outcome is a documented select or reject decision for every applicant, and a shortlist handed to the plant's hiring manager.



Like a theatre cloakroom with three entrances: *every coat handed in gets a numbered ticket, so every applicant ends with a recorded answer.*

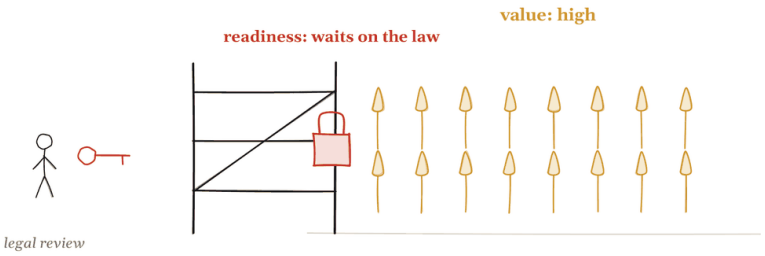
Step 1, Read: the map entry

The chief human resources officer's map placed the loop at value four and readiness three. Volume was high and repetitive, and slow contact was losing candidates. Readiness was held back by the law. Recruitment and selection is a high-risk use of AI under the EU AI Act, and the company took legal advice before going further.

Field	Entry
Loop	Candidate screening, plant operators
Chain	Hire to retire
Value to the core	4: about 6,300 applications a year, 14 for every hire; vacancies at the plants covered by overtime
Readiness	3: applicant tracking system in place; requirements written but different at each plant; owner willing; legal and works council review required
Handoffs	In from job requisition; out to offer and contract

Field	Entry
Risk category	High risk: EU AI Act, Annex III point 4(a). GDPR Article 22 limits decisions based solely on automated processing
Owner	Head of talent acquisition (agreed)
Distribution read	By case: 70% of applications complete, with documents issued in the plant's country; 21% incomplete or with documents issued abroad; 9% need judgement from the start, such as internal candidates, former employees and requests for adjustments at interview
Signed	Chief human resources officer

The legal advice set two boundaries before mapping began. Annex III point 4(a) covers AI used to analyse and filter applications or to evaluate candidates. Even simple eligibility checks fall inside it, so the company treated the whole deployment as high risk. The high-risk duties were due to apply from August 2026, and a delay had been proposed. The company chose to meet them regardless of the final date.



Like a ripe field behind a locked gate: *the crop is worth picking now, but readiness waits on the key that legal review holds.*

Step 2, Map: the baseline sheet

Mapping with recruiters, recruiting coordinators and two plant hiring managers drew fourteen steps across five lanes: candidate, coordinator, recruiter, plant hiring manager and plant HR.

Field	Entry
Sample and period	1,560 applications over twelve weeks, across all three plants
Volume	525 applications a month; about 38 hires a month
	€67, including systems

Field	Entry
Fully loaded cost per decided application	
Team time per hire	22 hours of recruiter and coordinator time
First contact	6 working days from application
Application to offer	19 days, median
Application to start	41 days, median, including notice periods
Candidates withdrawing before interview	22%
People	4 recruiters, 2 recruiting coordinators
Method agreed with finance	Yes, HR finance business partner

Cost per hire worked out at about €930, below the published median for nonexecutive roles. Speed was the weakness. One candidate in five dropped out before anyone interviewed them.



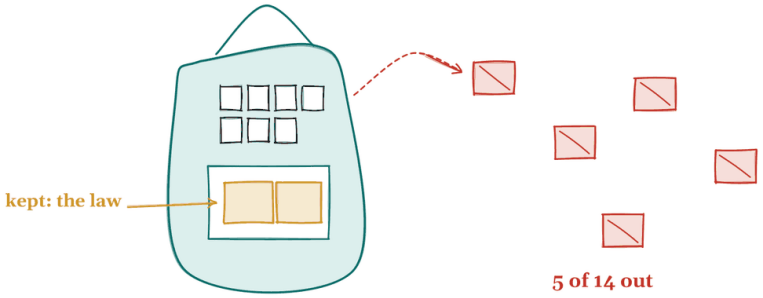
Like a bakery queue where one customer in five gives up before being served: *the price was fair, and the wait was the problem.*

Step 3, Strip: the strip register

Step	Status	Constraint	Still applies?
Re-key paper applications from the plant gate into the tracking system	Struck	Plants had no online form for walk-in applicants	No: a tablet form has run at every gate since 2024
Phone each applicant to book a screening call	Struck	Candidates had no self-service booking	No: the calendar tool supports it
Print CVs for the plant hiring manager	Struck	Hiring managers once had no system access	No
	Struck		

Step	Status	Constraint	Still applies?
Copy certificates to a shared drive for plant HR		Plant HR could not see attachments in the tracking system	No: access granted in 2023
Recheck right-to-work documents at offer stage	Struck	A 2019 audit finding at one plant	No: the check at application now keeps the evidence on file
Recruiter decides each rejection	Kept	Rejection affects the candidate and the law requires human involvement	Yes: becomes a junction
Verify right to work before the first shift	Kept	Required by law in all three countries	Yes

Five of fourteen steps were struck, a strip ratio of 36%.



Like emptying a backpack before a hike: *five of fourteen items packed for old trips come out, and the two the law requires stay in.*

Step 4, Sort: rules, decisions and the contested log

Legal counsel and the works council joined Sort from the first week. A works council is the elected body that represents employees, and it must be informed before such a system is used. The rule register held 33 rules. Five contradicted each other. The two that mattered most are shown.

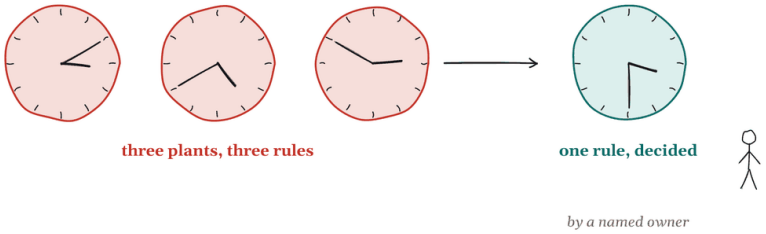
Rule	Source	Contradiction	Decision
Forklift licence required at application	Plant A job profile	Plants B and C hired without one and trained in the first three months	Head of operations: licence within three months of start, at all plants
Two years of shift experience	Plant B job profile	Not required at the other plants; no evidence it predicted retention	Head of talent acquisition, with legal: requirement removed

The decision register held eleven decisions.

Decision	Class	Limit or junction
Is a valid right-to-work document present for the plant's country?	Mechanical	Result shown to a recruiter, never acted on alone
Are the required certifications present and in date?	Mechanical	Result shown to a recruiter, never acted on alone
Book a screening call or interview slot	Mechanical	
Answer a candidate's question about the role, shifts or pay range	Bounded	Approved answers only; anything else goes to a recruiter
Request missing documents	Bounded	Two reminders, then to a recruiter
Prepare the evidence pack against the agreed criteria	Bounded	Evidence only; no score, no ranking
Reject a candidate, including one who fails a knock-out check	Consequential	Recruiter
Shortlist candidates for interview	Consequential	Recruiter, after settlement
Review a decision at the candidate's request	Consequential	Senior recruiter not involved in the first decision
Rank and shortlist candidates with the executor	Contested	See log

Field	Entry
Decision	May the executor rank applicants and propose a shortlist?
Positions	Plant managers: yes, to cut time to interview. Works council: no automated ranking of people. Legal: ranking is evaluation of candidates under the AI Act, and criteria differ by plant, so no ranking could be checked for fairness
Logged	Week 1 of Sort
Settled by	Chief human resources officer and chief operating officer, after legal advice and consultation with the works council
Answer	One set of five written criteria for all three plants. The executor shows evidence against each criterion, in order of application date, with no score. Recruiters shortlist. To be revisited after twelve months of outcome monitoring
Date settled	Week 5 of Sort
Resulting class	Consequential, with the junction held by the recruiter

The settlement was about criteria first. Until the plants agreed what a good operator candidate looked like, there was nothing fair for anyone to rank against.



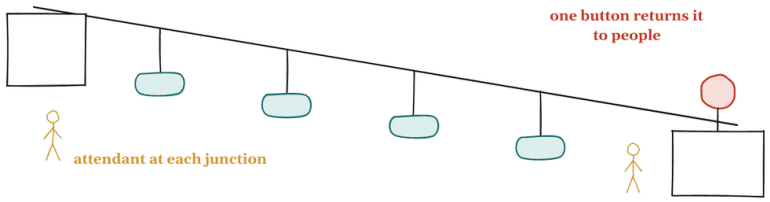
Like three town clocks showing different times: *someone with authority must choose the true time before every plant can run to it.*

Step 5, Design: junctions and the standard

Junction	Who decides	Evidence shown	How reversed
Reject a candidate	Recruiter	Application, check results with the documents behind them, evidence pack	Candidate reinstated and contacted the same week
Shortlist for interview	Recruiter	Evidence packs against the five criteria	Candidate added to the next interview round
Human review at the candidate's request	Senior recruiter not involved before	Full file, the first decision and the candidate's statement	Decision changed and candidate told in writing
Foreign qualification of uncertain equivalence	Plant HR lead	Certificate, issuing body, recognition guidance	Candidate reinstated
Switch-off	Head of talent acquisition	Grading dashboard and outcome monitor	All applications return to coordinators within 30 minutes

Field	Entry
Criteria	Every knock-out check correct and backed by the document; every evidence pack accurate against the five criteria; first contact within one working day; no rejection without a recruiter's decision on record
Rule checks	No status change to rejected without a recruiter's identity; every message from an approved template
Model grader	Every check and every evidence pack graded against the criteria

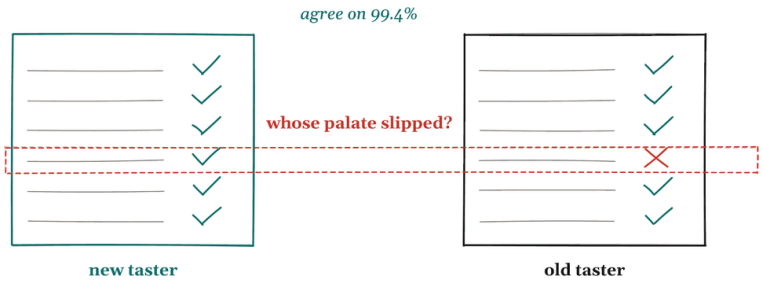
Field	Entry
Human audit	5% at random; every rejection after a failed knock-out check for the first three months
Candidate signals	Complaints; requests for human review; withdrawals
Outcome monitoring	Monthly pass rates at each stage, by sex, age band and nationality, where the company may lawfully record them. Any group below 80% of the highest group's rate triggers investigation. This is an internal trigger, not a legal test
Candidate rights	Candidates told before applying that an AI system supports screening; any candidate may ask for a human review and give their view
Calibration	Monthly, by the HR quality analyst; outcome report to the works council each quarter



Like a ski lift: *it carries every chair on its own, but an attendant stands where people step on or off, and one red button stops the line.*

Step 6, Shadow: the report

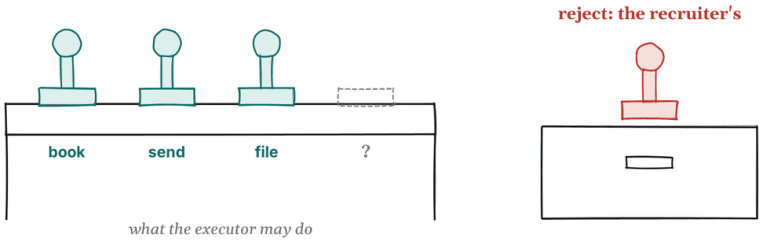
Field	Entry
Period and cases	Eight weeks; 1,080 applications run in parallel with the team
Agreement, body	99.4%
Agreement, tail	92%: documents issued abroad and incomplete applications
Disagreements	Executor wrong 47%: certificates issued in another EU country under a different name; scanned residence permits with two expiry dates. Person wrong 33%: one plant still applying the old forklift rule. Both defensible 13%. Standard unclear 7%
Changes made	Equivalence table for foreign certificates added; residence permits with two dates sent to a recruiter; recruiters briefed on the new licence rule
Known weaknesses	Handwritten certificates; candidates applying for two plants at once
Recommendation	Pass G1. Signed by the head of talent acquisition and the HR quality analyst



Like two tasters scoring the same wines blind: *they agree on almost every glass, and where they differ, the check is whose palate slipped, the new taster's or the old.*

Step 7, Bound: the gate review record

Field	Entry
Scope alone	Knock-out checks for right to work and certifications, with every result going to a recruiter; booking screening calls and interviews; requests for missing documents; approved answers to questions; evidence packs. These tasks took 46% of team hours in the baseline
Evidence	Four weeks bounded: 99.1% of bounded work met the standard; no rejection recorded without a recruiter
Permissions	Read applications; write check results and evidence packs; send approved messages; book calendar slots. No permission to change a status to rejected. No score field. No access to the data used for outcome monitoring
Switch-off test	Week 1: work returned to coordinators in 20 minutes; candidates mid-conversation received a holding message
Audit consulted	Data protection officer: logs kept and retained under the company's policy. Works council: informed before first use and shown the outcome report format
Decision	Pass G2. Signed by the head of HR risk and compliance



Like a clerk's rack of rubber stamps with one missing: *the executor can book, send and file, but the reject stamp stays in the recruiter's drawer.*

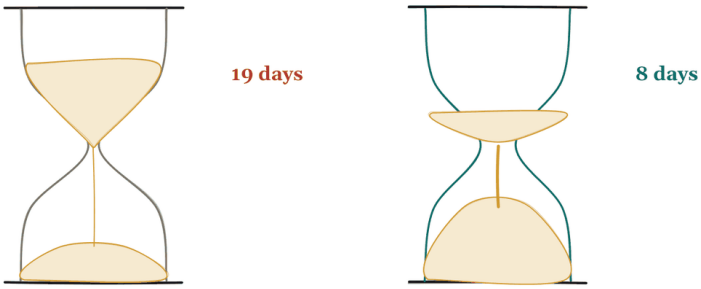
Step 8, Own: charter and economics

Field	Entry
Loop owner	Head of talent acquisition
Junction holders	4 recruiters; 1 senior recruiter for human reviews; 3 plant HR leads for foreign qualifications
Exception owner	Recruiting coordinator lead; no application untouched for more than one working day
Grader	HR quality analyst
Adoption owner	HR operations manager
Control owner	Head of HR risk and compliance
Changes to work	No roles removed. Recruiters spend the freed time on interviews and on hard-to-fill maintenance technician roles. One coordinator moved to the onboarding loop, and the role was backfilled
Learning path	New recruiters shadow human reviews for one month before deciding rejections alone

Field	Entry
Period	Twelve weeks, including the spring hiring peak
Cost lines	Platform and model; grading and audit; outcome monitoring and legal review; recruiter junction time; exception queue; adoption
Cost per accepted application	€61 overall; body €39; tail €113
Baseline	€67 per decided application
Team time per hire	22 hours to 12.5
First contact	6 working days to under 1
Application to offer	19 days to 8
Application to start	41 days to 27
Withdrawals before interview	22% to 11%
Annualised effect	About 4,300 team hours redeployed to harder hiring. Fourteen fewer vacant days on 450 hires is 6,300 vacancy days, worth about €440,000 a year in overtime cover

Field	Entry
Signed	HR finance business partner

The cost per accepted application fell by under a tenth. The value of the loop lies in speed and in every candidate being treated by the same criteria, not in headcount.

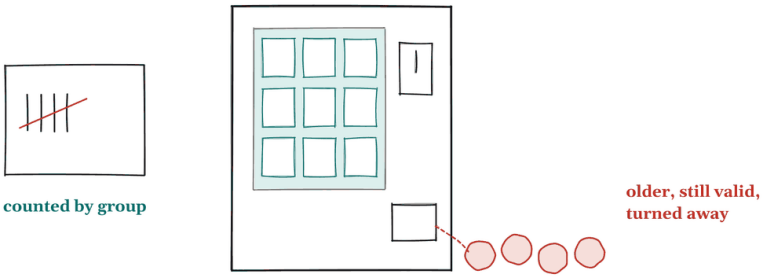


Like widening the neck of an hourglass: *the sand costs about the same, it simply runs through faster, and application to offer fell from nineteen days to eight.*

Step 9, Core: an operating review

Field	Entry
Month	Third month in Core
Standard met	Body 99.5%; tail 94%
Cost per accepted application	€58
Exception queue	Oldest case 1 day. Repeated exception: candidates applying to two plants; rule added to merge the applications
Junctions	Recruiters overrode one failed knock-out result in fourteen, mostly for documents the executor read correctly but which were renewed after submission
Outcome monitoring	Candidates over 50 at one plant passed the certification check at 74% of the highest group's rate. The cause was older certificates in a superseded format being read as expired. Reading corrected; 14 candidates re-reviewed and 9 invited to interview
Candidate signals	9 requests for human review; 2 decisions changed

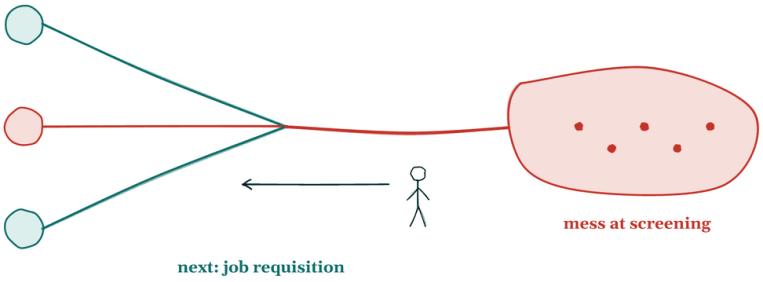
Field	Entry
Decisions	Report to the works council brought forward; bounded scope unchanged until the next quarter's monitoring



Like a vending machine that keeps returning older, still-valid coins: *only counting which coins came back showed the reader needed fixing.*

Step 10, Scale: what the edges showed

The exception queue and the contested log both pointed upstream. Most disagreements began with job profiles written differently at each plant, and the five criteria had to be agreed before screening could be fair. The map review chose job requisition as the next loop in the chain. Onboarding, which now received complete certificates and right-to-work evidence, rose in readiness and was marked to read after it.

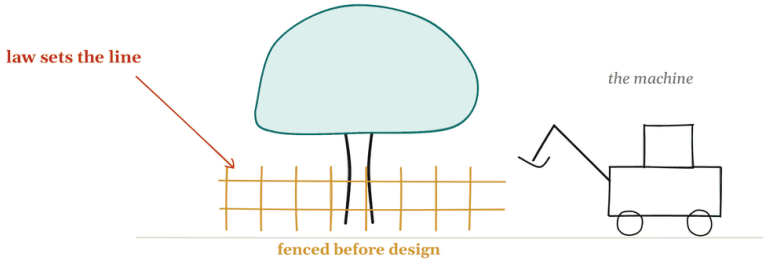


Like walking upstream from a muddy pond: *the mess at screening came from job profiles written differently at each plant, so the next fix goes to the source.*

What this loop teaches

In a regulated loop, the law sets some decision classes before Sort begins. Rejection was consequential because it affects a person, and no rule could make it bounded. The contested decision was not about a limit. It was about

what the company valued in a candidate, and only its leaders could write that down. The standard had to watch outcomes across groups of candidates, not only the accuracy of single cases. And the economics showed speed and consistency, which a sheet counting only cost would have undervalued.

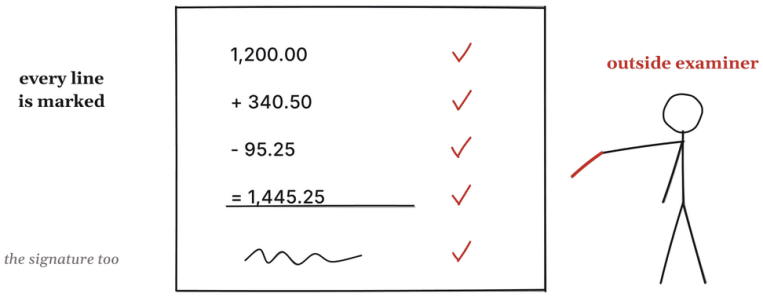


Like fencing a protected oak before the diggers arrive: *in a regulated loop the law marks some decisions, such as rejecting a person, before any design begins.*

XCIII

The Record-to-Report Chain

Record to report is the chain that turns the company's transactions into figures that leaders, lenders and auditors can rely on. It runs from posting journal entries, through reconciling accounts and settling balances between the group's own companies, to closing the period, producing management and statutory reports and answering the auditors. Its customers are internal, but its standard is set from outside. External auditors read the records themselves, so every reconciliation, entry and sign-off is part of what the chain delivers.



Like an exam marked by an outside examiner who grades the working line by line: *every entry, match and signature is part of the answer, not just the total.*

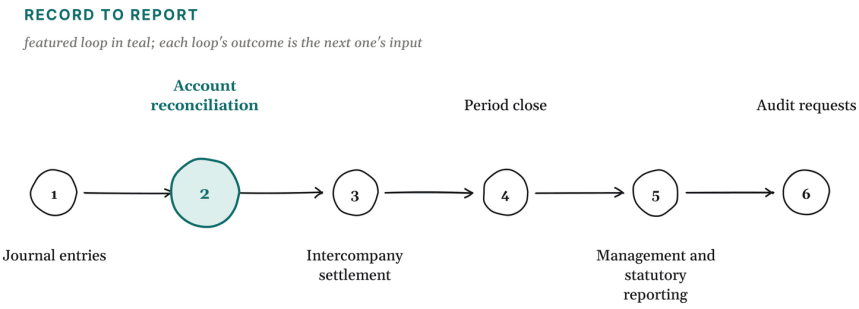


FIGURE 77 The record to report chain. Its loops in handoff order, with the featured loop, account reconciliation, in teal.

The loops and the evidence

The table gives, for each loop in the chain, the shape of its cases and a published benchmark. Shapes marked as estimates are reasoned from the nature of the work where no published distribution was found.

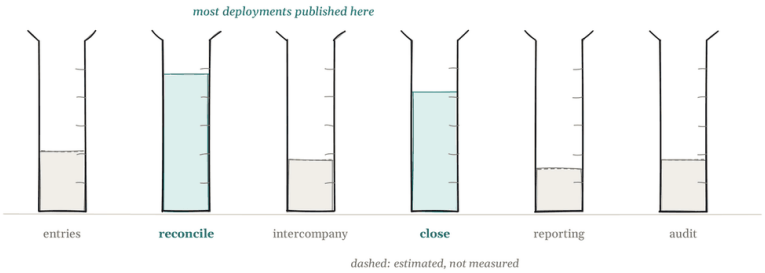
Loop	Shape of cases	Published benchmark
Journal entries	Mixed (estimate)	Median of $3.50 per journal entry, with 90% of entries processed automatically (APQC)
Account reconciliation	Heavy tail (estimate)	20 to 50 hours a month spent on cash reconciliation (Ledge survey, 2025); 86% of shared-services leaders still reconcile in spreadsheets (SSON, 2026)
Intercompany settlement	Heavy tail (estimate)	Median of 40% of intercompany transactions fully automated and 25% fully manual (APQC)
Period close	Mixed (estimate)	Median of 6.4 calendar days for the monthly close, 4.8 for top performers (APQC, 2017, the latest public figure); median of 8 days in finance shared services centres (APQC)
Management and statutory reporting	Heavy tail	Median of $0.42 per $1,000 of revenue to produce financial reports (APQC); about 3% of companies restate their accounts each year (PCAOB, 2025)
Audit requests	Heavy tail (estimate)	Average of 15,581 hours a year on compliance control programmes, or 16 hours per control tested (KPMG, 2025)

Named public deployments in the chain are concentrated in reconciliation and the close. Results are as reported by the companies or their vendors.

Loop	Company	Reported result
Journal entries	US Foods	Five times more journal processing automated (vendor-reported)
Account reconciliation	Konica Minolta	Daily bank reconciliation 75% faster; about 99% of more than 45,000 monthly lines matched automatically (vendor-reported)
Account reconciliation	Microsoft, own finance organisation	About 20 minutes saved per receivables account reconciled; about 22% lower handling cost (pilot)
Period close	ConferenceDirect	Close cut from 10 days to under 5 (vendor-reported)

The featured loop is account reconciliation. It shows a body of high-volume matching and a tail of judgemental accounts, a control that must stay with a

named reviewer, and a contested decision about what counts as support. Its economics show up in close days and aged items as much as in cost.



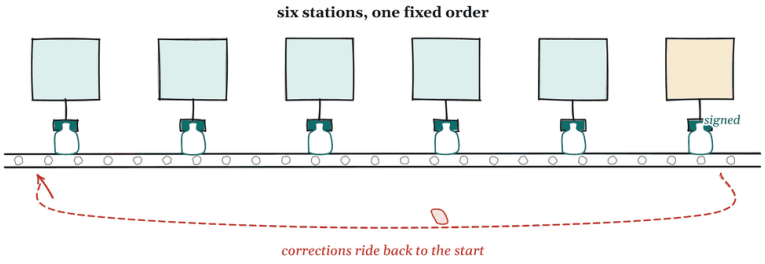
Like a row of rain gauges: *each loop is read against a published mark, and the measured readings cluster where companies have published, in reconciliation and the close.*

XCIV

The Loops of Record to Report

Record to report turns the company's transactions into figures that outsiders can rely on. Its six loops run in a fixed order. Journal entries put transactions into the general ledger, the company's master book of accounts. Account reconciliation proves each balance against independent evidence. Intercompany settlement agrees and clears what the group's own companies owe each other. Period close locks the month, management and statutory reporting turns the closed ledger into packs and filings, and audit requests hand the evidence to the external auditor. Work also flows backwards: reconciliations, the close and the audit all send correcting entries back to the start.

The entries share one feature that sets this chain apart. Its standard is written outside the company, by securities law, accounting standards and the auditor. In the United States, the Sarbanes-Oxley Act (SOX) makes managers maintain and certify their controls over financial reporting. The Public Company Accounting Oversight Board (PCAOB) writes the standards auditors use to test them. As a result every loop ends in a named person's signature. Where the loops differ most is volume. Journal entries and bank matching run in the tens of thousands a month, while reporting and the audit are a few large events a year.



Like a bottling line with a return belt: *six stations run in a fixed order, each stamps its work, and faulty bottles ride back to the start.*

Journal entries

At a glance	
Outcome	A balanced, supported, approved entry posted to the right ledger, period and accounts

At a glance	
Shape of cases	Mixed: automated recurring body, manual period-end tail
Decisions	1 mechanical · 1 bounded · 2 consequential · 1 contested
Hardest to undo	Approving and posting a manual or top-side entry
Anchor figure	Median cost of $3.50 per journal entry (APQC, accessed 2026)

This loop produces a posted journal entry: a balanced record of debits and credits, with support and an approval, sitting in the correct period. Most entries arrive from other chains. Billing, cash application, supplier invoices and payroll send subledger postings and accruals, which are amounts owed but not yet invoiced or paid. Reconciliation and the close send correcting and period-end requests. Posted balances go on to reconciliation, intercompany settlement, the close and the auditors.

By count the loop is a machine. APQC's median is 90% of entries processed automatically and 10% manually. They are recurring interfaces that apply one template every time. Risk sits elsewhere. PCAOB standard AS 2401 tells auditors to focus on period-end and unusual entries, because that is where fraudulent ones cluster. The body is large and safe; the tail is small, manual and where the danger lives. That shape is an inference from these two sources, not a measured distribution.

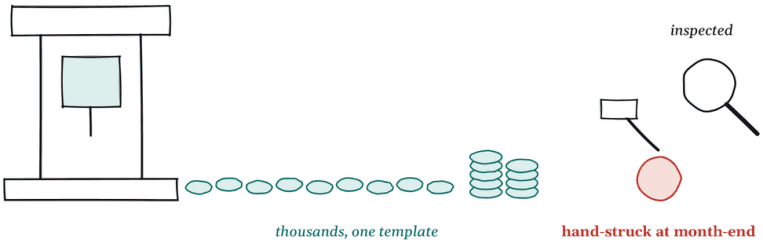
Generating a recurring entry from source data is mechanical. Choosing accounts, cost centres and period for a manual entry is bounded by the chart of accounts. Estimating an accrual and approving a manual entry are consequential, because SOX segregation of duties bars a preparer from approving their own work. The contested decision is the non-standard adjustment whose accounting treatment is in dispute. Few companies have an agreed route for it.

Written rules can keep the interface body. A model can draft manual entries, suggest accounts and write the explanation that a reviewer reads. People hold approval and posting, and any top-side entry, meaning one booked directly at group level outside the subledgers. Those entries are the least reversible and the first an auditor tests.

The standard is a balanced, supported, correctly approved entry. APQC puts the median at $3.50 per entry and $0.83 per line, with seven lines per entry.

The Hackett Group reported in 2025 that leading finance teams automate 99% of entries against 85% for peers. In a 2026 SSON survey for Redwood Software, 84% of shared-services leaders named journal entries their largest source of manual effort.

As read in 2026, US Foods runs journal processing on BlackLine in production and reports a fivefold expansion in automated processing. Payoneer links transaction matching directly to automated entries and is exploring agents. No named company runs an agent that prepares and posts manual entries end to end. No public cycle-time or first-time-right figure exists. As the body automates further, the manual tail becomes a larger share of what auditors sample.



Like a coin press: *recurring entries stamp out identically by the thousand, while the few hand-struck coins at month-end are where forgeries hide and inspectors look.*

Account reconciliation

At a glance	
Outcome	A balance substantiated by independent evidence, with differences explained and a reviewer's certification
Shape of cases	Heavy-tailed: mass matching, a few judgemental accounts
Decisions	1 mechanical · 1 bounded · 2 consequential · 1 contested
Hardest to undo	Writing off an aged reconciling item
Anchor figure	86% of shared-services leaders still reconcile in spreadsheets (SSON for Redwood, 2026)

A reconciliation proves that a ledger balance is true. It compares the balance with an independent source such as a bank statement or subledger, and explains every difference. The loop receives posted balances from journal

entries, bank and cash data from the payment and collection loops, and agreed or disputed balances from intercompany settlement. It returns correcting entries to the journal loop and passes certified reconciliations to the close and to the auditors.

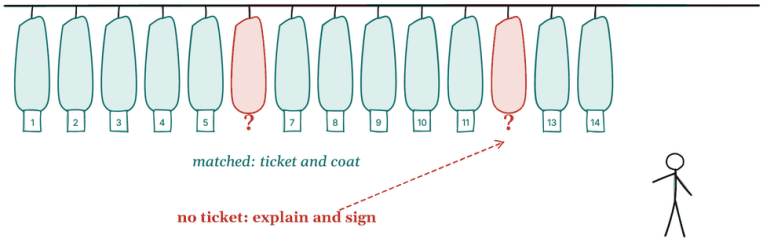
The cases split sharply. High-volume, low-value matching against bank and point-of-sale data clears at very high rates. A few accounts, such as reserves, accruals, suspense and intercompany balances, need reading and argument. Each of those balances rests on an estimate or on another party's books, so effort concentrates there. No published study measures effort by account. The heavy tail is inferred from vendor match rates.

Matching transactions is mechanical. Explaining an unmatched item is bounded, since the causes are few: timing, error or a missing entry. Proposing a correcting entry or write-off is consequential, and so is certifying as reviewer. SOX and AS 2201 treat that sign-off as a key control whose reviewer the auditor tests for competence. The contested decision is whether a reserve or accrual balance is adequate, which auditors often challenge.

Rules and matching engines take the body. A model can draft explanations for exceptions and assemble support. People certify each reconciliation and approve write-offs, which change the ledger and are hard to reverse once the period is closed.

Grade each reconciliation on evidence, explained differences and signatures. A 2025 Ledge survey of 100 finance professionals found 20 to 50 hours a month spent on cash reconciliation alone. In SSON's 2026 survey, 69% named reconciliations a major source of manual effort.

As read in 2026, Konica Minolta reports bank reconciliation 75% faster in production, with about 99% of over 45,000 monthly lines auto-matched (vendor-reported). Microsoft's own finance pilot saved about 20 minutes per receivables account reconciled. Delaware North, Bristol Myers Squibb and LAZ Parking are early or partial adopters of BlackLine's agents. There is no public median for hours, cost or first-time-right per reconciliation. The next chapter follows this loop through the whole lifecycle.



Like a cloakroom at closing time: *most tickets match their coats at a glance, while a few unclaimed coats need someone to explain them and sign.*

Intercompany settlement

At a glance	
Outcome	Group charges priced at arm's length, agreed by both sides, netted and settled so they eliminate cleanly
Shape of cases	Heavy-tailed: a few large flows and disputes
Decisions	1 mechanical · 1 bounded · 2 consequential · 1 contested
Hardest to undo	Setting or adjusting a transfer price
Anchor figure	Median of 40% of intercompany transactions fully automated, 25% fully manual (APQC, accessed 2026)

When one company in a group sells to or charges another, both books must show the same amount. This loop makes sure they do. It receives intercompany entries and recharges from the journal loop, sales and invoices between group entities from lead to cash, and cut-off dates and exchange rates from the close. It sends agreed or disputed balances to reconciliation, matched balances and eliminations to the close, and netting instructions to the payment run. Netting means offsetting what entities owe each other so only the difference moves as cash.

APQC's medians show the shape. Intercompany entries are 5% of journal entries but 50% of journal lines, so a few entries carry many lines. In a 2023 BlackLine survey, over 75% of respondents said intercompany value exceeded five times revenue. 97% had resolved multi-million-dollar variances. A small number of large flows and disputes carry most of the risk.

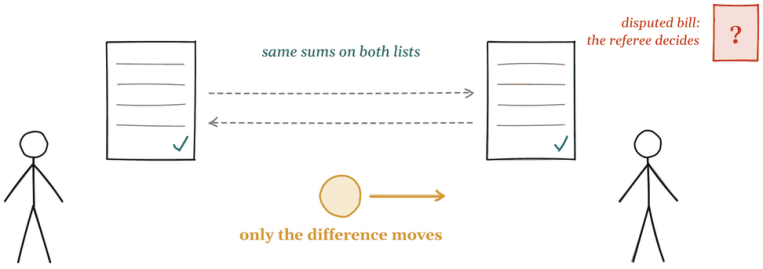
Matching receivables and payables between counterparties is mechanical. Resolving a mismatch over timing, currency or a missing charge is bounded,

but needs evidence from both sides. Setting the transfer price, the price one group company charges another, is consequential. The OECD guidelines and national rules such as US IRC section 482 require it to match what independent parties would agree. Forcing an elimination of an unresolved difference is consequential too, since IFRS 10 and ASC 810 require intragroup balances to be removed in full. The contested decision is which entity bears a charge. Two local finance teams have different incentives, and the group controller often settles it.

Rules can match and net. A model can find trading pairs and propose causes for mismatches. People hold transfer prices and write-offs, which carry tax exposure across jurisdictions.

The standard is a balance that eliminates without a plug and a price that survives a tax audit. APQC's median is 12,000 intercompany transactions per business unit a year.

No named company has published an agent running intercompany matching or settlement. HighRadius offers one, as read in 2026. No public figure exists for cycle time or cost per transaction.



Like two flatmates squaring their IOUs: *both lists must show the same sums, only the difference changes hands, and a disputed bill goes to a referee.*

Period close

At a glance	
Outcome	A locked period with a reviewed trial balance and consolidated ledger, handed to reporting on the deadline
Shape of cases	Mixed: even spread across companies, a few tasks set the pace
Decisions	1 mechanical · 1 bounded · 2 consequential · 1 contested
Hardest to undo	Locking the period and signing off the close

At a glance	
Anchor figure	Median of 8 days for a shared services centre to close the month (APQC, accessed 2026)

The close turns a month of activity into a locked period. Its output is a trial balance, the list of every account and its balance, reviewed and consolidated across the group. Inputs are accruals and adjustments from journal entries, certified reconciliations, matched intercompany balances and cut-off confirmations from the subledgers. The closed trial balance goes to reporting and evidence goes to the auditors. Late adjustments from variance review return to the journal loop.

Across companies the close is spread fairly evenly. A 2025 Ledge survey found 18% closing in one to three business days, 32% in four to five, 23% in six to seven and 27% in more than seven. Within one close, a few dependent tasks set the finish date. 56% of Ledge respondents named dependence on other departments and regions as the main blocker. That concentration is an inference, not a measurement.

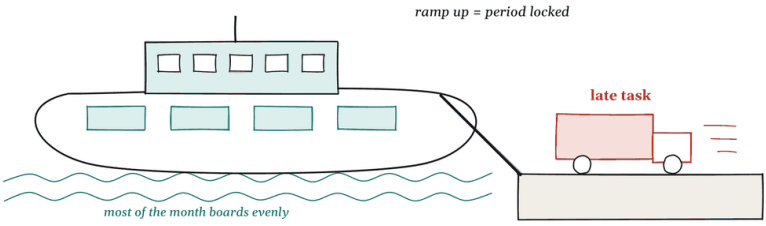
Scheduled jobs such as depreciation and allocations are mechanical. Chasing late tasks is bounded by the calendar and checklist. Investigating a flux, an unexpected change in a balance from one period to the next, is consequential, because it decides whether results are misstated. So is locking the period, which feeds the quarterly SOX section 302 certification by the chief executive and chief financial officer. The contested decision is materiality: whether a late error is booked now or next period.

The calendar can run on rules. A model can sequence tasks, chase owners and draft flux explanations. People hold the lock, since reopening a closed period is costly and visible.

APQC's public figure from 2017 put the monthly close at 6.4 calendar days at the median and 4.8 for top performers. Hackett reported in 2025 that leading finance teams close 35% to 57% faster than peers. SSON found in 2026 that only 2% have a fully automated close.

As read in 2026, ConferenceDirect reports its close cut from 10 days to under 5 in production (vendor-reported). Bristol Myers Squibb is piloting agentic workflows toward a touchless close. Booking Holdings has standardised operations across several ERPs without a published close result. No company

has published an agent running the close end to end. A faster close moves the bottleneck to whichever dependency was hidden behind the old one.



Like a ferry that cannot sail until the last late lorry is aboard: *a few late arrivals set the departure, and once the ramp is up, reopening it is costly.*

Management and statutory reporting

At a glance	
Outcome	Approved management packs, statements and filings delivered on deadline
Shape of cases	Heavy-tailed: rare but severe errors
Decisions	1 mechanical · 1 bounded · 2 consequential · 1 contested
Hardest to undo	Signing and filing statements and certifications
Anchor figure	About 3% of companies a year restate their accounts (PCAOB, 2025)

This loop turns the closed ledger into documents for executives, the board, regulators and investors. It receives the closed trial balance from the close, eliminations and related-party disclosures from intercompany settlement, and the auditor's review or opinion. Its outputs go to the auditor as drafts, to planning as actuals, and outside the company as filings and board packs.

The volume is low and the deadlines are fixed. The SEC gives large accelerated filers 60 days after year-end to file the annual report. Errors are rare but severe. The PCAOB found major restatements, the kind that reissue prior statements, at about 3% of companies a year from 2005 to 2024. Counts swing with single events: Ideagen Audit Analytics counted about 1,470 restatements in 2021, during the SPAC warrant restatements, against 430 in 2023. Consequence comes in a few large events rather than spread evenly.

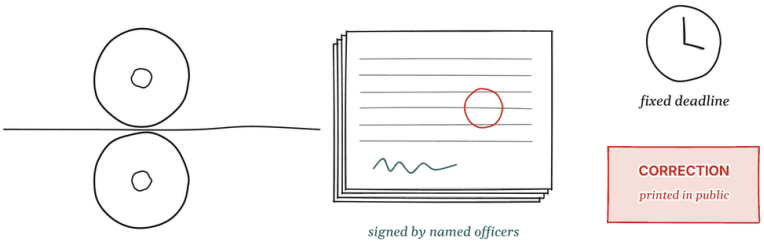
Populating statements from the ledger is mechanical. Drafting variance commentary is bounded by known drivers. Deciding disclosures, segments and non-GAAP measures is consequential under regulator scrutiny. Signing is consequential and personal: SOX section 302, the EU Accounting Directive

and the UK Companies Act all place it on named officers or directors. The contested decision is whether an error is material enough to restate.

Templates and tie-out rules can produce the numbers. A model can draft commentary and flag variances. People decide disclosure and sign, and a filed statement can only be corrected by a public restatement.

Grade on accuracy, completeness and deadline. APQC's median cost is $0.42 per $1,000 of revenue. In the EU the CSRD, the sustainability reporting directive, adds a sustainability statement to the management report. Management approves it and it is subject to assurance, so the loop's scope is growing.

No named company has published an agent producing statutory or management reports, as read in 2026. Public cycle-time medians are members-only. There is no first-time-right figure for management packs. If drafting becomes cheap, more variance commentary will be produced, and the reviewer's attention becomes the scarce resource.



Like a newspaper's printing press: *a few editions on fixed deadlines, signed off by named editors, and a printed error can only be fixed by a public correction.*

Audit requests

At a glance	
Outcome	Every auditor request answered with complete, accurate evidence by the agreed date
Shape of cases	Heavy-tailed: routine pulls, a few judgemental areas
Decisions	1 mechanical · 1 bounded · 2 consequential · 1 contested
Hardest to undo	Signing the management representation letter
Anchor figure	

At a glance

15,581 average SOX programme hours a year, 16 per control tested (KPMG, 2025)

The auditor asks; this loop answers. Each request is a provided-by-client item, a sample, a walkthrough or evidence that a control ran. Evidence comes from every other loop in the chain: entry populations and approvals, signed reconciliations, close checklists, draft statements and transfer-pricing files. The loop returns audit adjustments to the journal loop, control weaknesses to reconciliation and the close, and the opinion to reporting.

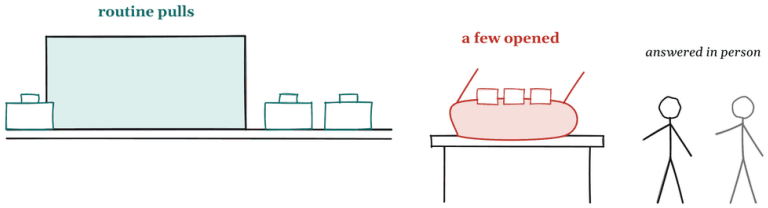
Cost is concentrated across companies. FERF's 2026 audit fee survey found the top quarter of S&P 500 companies paid 58.5% of all S&P 500 audit fees. The median fee was $7.96 million. Within one audit, most requests are routine pulls, while a few areas such as estimates, revenue and control deficiencies absorb most auditor hours. That inner shape is an estimate, since no study measures it.

Pulling standard evidence is mechanical. Assembling support for a sample is bounded by the auditor's scope. Responding to a finding or proposed adjustment is consequential, as is signing the representation letter in which officers attest that the information given is complete. The contested decision is disagreement with the auditor over treatment or the severity of a deficiency. It escalates to the audit committee and can affect the opinion.

A model can retrieve and package evidence quickly. AS 1105 requires the auditor to test company-produced information, so every pack needs a traceable source. AS 2401 requires the auditor to question people directly, and those answers cannot be delegated. People hold responses to findings and the signature.

Grade each response on completeness, accuracy and date. KPMG's 2025 survey put the average SOX programme at 546 key controls and $2.3 million a year.

Workday offers a financial audit agent, as read in 2026, but no named company has published a deployment. No benchmark exists for requests per audit or time per request. Better reconciliations and entries upstream should shrink this loop's tail more than any work inside it.

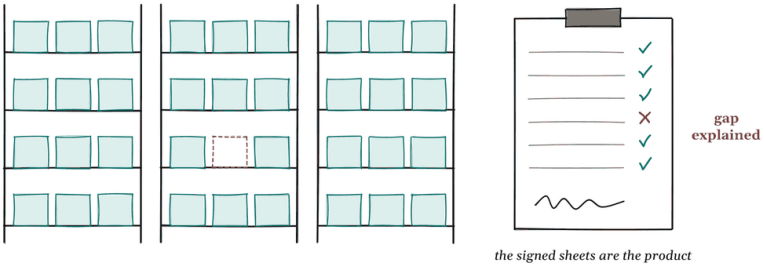


Like a customs check: *most bags pass the scanner with their papers, a few are opened and argued over, and the traveller must answer the officer in person.*

XCV

Account Reconciliation

At the composite manufacturer, the loop substantiates about 1,400 balance-sheet accounts every month, including 38 bank accounts across three countries. Each balance is compared with an independent source, and every difference is explained, aged and either cleared or escalated. A named reviewer then certifies the result. The outcome is a certified record for each account, ready for the close and for the auditors. Because auditors read those records directly, the record set is the product.



Like a warehouse stocktake: *every shelf count is checked against the book, each gap is explained, and the signed count sheets are what the auditor takes away.*

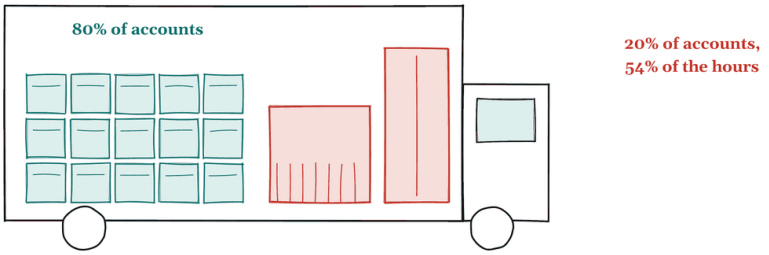
Step 1, Read: the map entry

The chief financial officer's map placed the loop at value four and readiness three. The bank feed and the cash application loop already supplied clean data. The judgemental accounts were the open question.

Field	Entry
Loop	Account reconciliation
Chain	Record to report
Value to the core	4: about 1,400 reconciliations a month; on the critical path of a 9-working-day close
Readiness	3: bank and subledger data structured; reconciliation policy written; reserve and accrual support held in plant spreadsheets; one known dispute between the controller and the business units
Handoffs	In from journal entries, cash application and supplier invoices; out to the close, journal entries for corrections, intercompany settlement and audit requests

Field	Entry
Risk category	Not high risk under the EU AI Act; reconciliation review is a key financial control
Owner	Head of general accounting (agreed)
Distribution read	By account: 1,120 accounts (80%) are bank, subledger tie-outs and low-activity balances; 280 accounts (20%) are reserves, accruals, suspense and intercompany. By effort: the 280 judgemental accounts take 54% of reconciliation hours
Signed	Chief financial officer

The body is the matching work. The 38 bank accounts carry about 21,400 lines a month, and most clear by rule. The tail is small in count but heavy in effort. A warranty reserve or a suspense account needs judgement that a bank match does not.



Like a removal van being loaded: *four-fifths of the boxes are light and labelled, but the piano and wardrobe, a fifth of the load, take over half the lifting.*

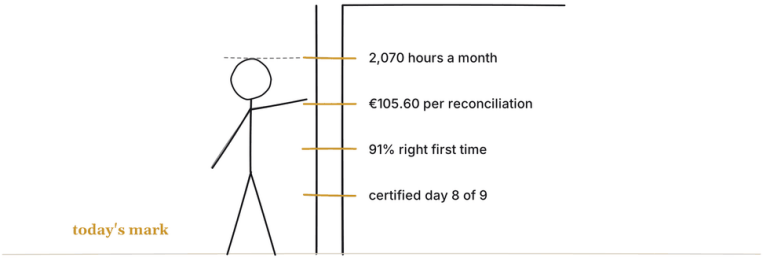
Step 2, Map: the baseline sheet

Mapping with the preparers and reviewers drew fourteen steps across six lanes: ledger, bank, plant finance, preparers, reviewers and internal control.

Field	Entry
Sample and period	4,200 reconciliations and 64,000 bank lines over three month-end closes
Volume	1,400 reconciliations a month; 21,400 bank lines a month
Hours a month	2,070: 1,660 preparing, 410 reviewing. Cash reconciliation alone 48 hours
Fully loaded cost per accepted reconciliation	€105.60; body €60.70; tail €319.60
Met the standard first time	91%: body 94%, tail 79%, graded retrospectively

Field	Entry
Bank lines matched by existing rules	82%
Open reconciling items at month-end	3,140, of which 610 older than 90 days (€4.8 million gross)
Reconciliations certified by	Working day 8 of a 9-day close
People	11 preparers, 4 reviewers
Method agreed with finance	Yes, planning and analysis lead

Cash reconciliation hours sat at the top of the published range. The close itself ran longer than the published medians.



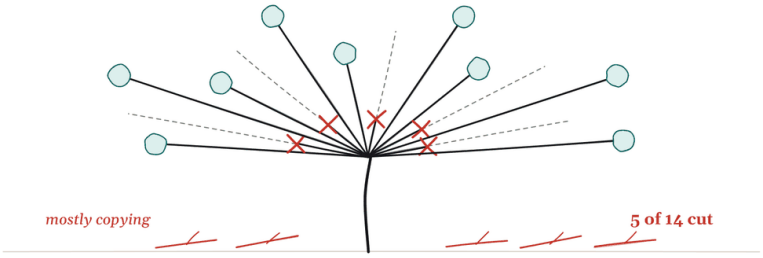
Like pencil marks on a doorframe: *the baseline marks today's hours, cost, first-time rate and aged items, so any later change can be read against it.*

Step 3, Strip: the strip register

Step	Status	Constraint	Still applies?
Copy ledger balances from the trial balance into spreadsheet templates	Struck	Templates were built before the reconciliation module was licensed	No
Re-key bank statement lines for two foreign bank accounts	Struck	Those banks had no electronic feed until 2022	No
Prepare a full monthly template for zero-balance accounts with no activity	Struck	Policy written when every account was reviewed monthly	No: policy now allows quarterly review when there is no activity
Email plant finance each month for accrual schedules	Struck	Schedules lived on local drives	No: schedules can be read from a shared folder
Print and initial reconciliations for an audit binder	Struck	Auditors asked for paper until 2016	No
	Kept	Required by the outcome	Yes

Step	Status	Constraint	Still applies?
Explain and age each reconciling item			
Reviewer certifies the reconciliation	Kept	Key control with a named person	Yes: becomes a junction

Five of fourteen steps were struck, a strip ratio of 36%. Most of what was struck was copying, and copying is where preparers made their errors.



Like pruning dead wood from a shrub: *five of fourteen branches go, mostly copying steps kept from old limits, and with them the places mistakes grew.*

Step 4, Sort: rules, decisions and the contested log

The rule register held 36 rules. Five contradicted each other. The two that mattered most are shown.

Rule	Source	Contradiction	Decision
Reconciling items older than 60 days are cleared or escalated	Group reconciliation policy	Plants carried items for up to 180 days	Group financial controller: escalate at 60 days; propose clearing or write-off at 90
Unexplained differences under €1,000 may be carried	Group policy	One plant used €5,000	Group financial controller: €1,000 for all accounts; zero for bank accounts

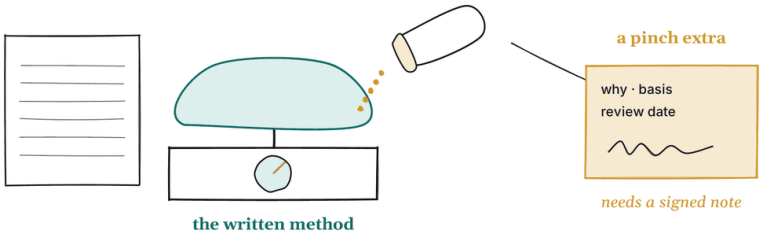
The decision register held eleven decisions.

Decision	Class	Limit or junction
Match bank lines to ledger entries	Mechanical	
Tie a subledger total to its control account	Mechanical	
Roll forward a low-activity account against its schedule	Mechanical	

Decision	Class	Limit or junction
Explain and classify an unmatched item as timing, error or missing entry	Bounded	From the finite list, with evidence attached
Draft a correcting entry for a classified error	Bounded	Draft only; posting goes through the journal entries loop
Check that a reserve or accrual carries the support the standard requires	Bounded	Settled rule, see log
Write off an aged reconciling item	Consequential	Reviewer up to €5,000; group financial controller above
Clear a suspense item to its final account	Consequential	Reviewer
Judge whether a reserve or accrual balance is adequate	Consequential	Reviewer; overlays above the limit go to the group financial controller
Certify the reconciliation	Consequential	Named reviewer
Decide what counts as adequate support for a reserve or accrual	Contested	See log

Field	Entry
Decision	What evidence makes a reserve or accrual balance adequately supported?
Positions	Group financial controller: every balance needs a calculation from a stated method and ledger or operational data. Business units: plant managers′ knowledge of customers, claims and stock is evidence, and a note from them should be enough
Logged	Week 1 of Sort
Settled by	Chief financial officer and chief operating officer
Answer	A balance is supported when it rests on a stated method applied to current data. A judgemental overlay on top is allowed with a signed memo from the plant finance manager giving the event, the basis and a review date. Overlays above €250,000, or above 20% of the method amount, go to the group financial controller
Date settled	Week 4 of Sort
Resulting class	Support check bounded; adequacy consequential with the reviewer; large overlays consequential with the group financial controller

A judgemental overlay is an amount added to a calculated reserve because managers know something the data does not yet show. Before the settlement, overlays had no common form. After it, each one had a memo, an owner and a date.



Like settling a family argument over a recipe: *measure by the written method on fresh ingredients, and any pinch added by feel needs a signed note saying why.*

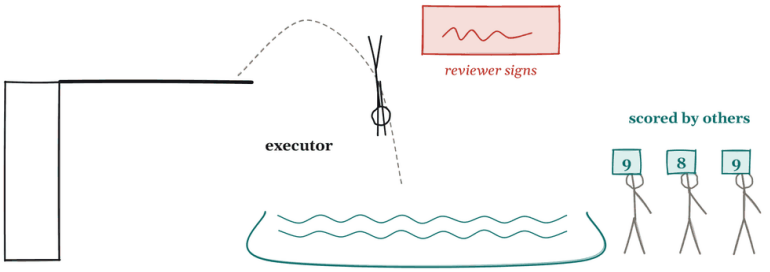
Step 5, Design: junctions and the standard

Segregation of duties normally means that one person prepares a reconciliation and another reviews it. With an executor preparing, that separation moves. What must now be kept apart is the executor and the standard it is held to. The head of general accounting runs the executor. The standard is signed by the head of internal control and graded by a finance quality analyst outside general accounting. The reviewer junction remains, and the executor has no permission to certify.

Junction	Who decides	Evidence shown	How reversed
Certify a reconciliation	Named reviewer	Balance, source, every reconciling item with age and evidence, grader result	Withdraw certification and return to the preparer queue
Write off an aged item over €5,000	Group financial controller	Item history, attempts to clear, proposed entry	Reverse the entry in the next period
Reserve overlay above the limit	Group financial controller	Method amount, overlay memo, prior overlays for the plant	Overlay removed and the balance reset to the method amount
Clear a suspense item	Reviewer	Source transaction and proposed account	Reclassify by correcting entry
Switch-off	Head of general accounting	Grading dashboard and close calendar	All preparation returns to preparers within 30 minutes

Field	Entry
Criteria	Balance agrees to an independent source; every reconciling item explained, classified, evidenced and aged; no unexplained difference above tolerance; items over 60 days

Field	Entry
	escalated; judgemental accounts carry the support the settled rule requires; prepared by working day 3 and certified by working day 5
Rule checks	Every reconciliation balances; every item has an age and a class; every overlay has a memo and a review date
Model grader	Every reconciliation graded against the criteria
Human audit	3% of body reconciliations at random; every tail reconciliation is reviewed by its certifier
Audit signals	Auditor sample exceptions; control deficiencies raised
Calibration	Monthly, by the finance quality analyst

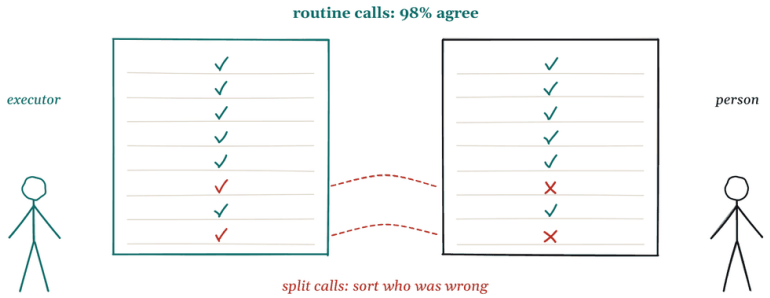


Like a diving contest: *the coach prepares the diver, independent judges hold up the scores, and only the referee signs the result, never the diver.*

Step 6, Shadow: the report

Field	Entry
Period and cases	Two month-end closes; 2,800 reconciliations; 42,800 bank lines
Agreement, body	98.4%
Agreement, tail	81%: accruals from plant spreadsheets and suspense accounts
Disagreements	Executor wrong 44%: three plant accrual schedules in different layouts; revaluation timing on two foreign bank accounts. Person wrong 33%: items over 60 days still marked as timing differences; support reused from earlier months. Both defensible 15%. Standard unclear 8%, about what counts as current data for the warranty reserve
Changes made	Accrual schedule layout fixed across plants; revaluation rule added; "current data" defined as no older than the prior quarter

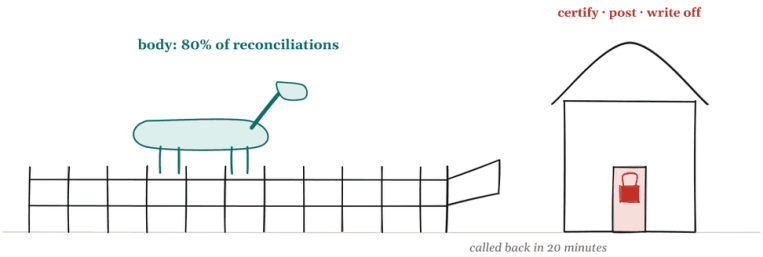
Field	Entry
Known weaknesses	Suspense items with no reference to a source document
Recommendation	Pass G1. Signed by the head of general accounting and the finance quality analyst



Like two umpires scoring the same match in parallel: *they agree on nearly every routine call, and the few split calls show who needs correcting.*

Step 7, Bound: the gate review record

Field	Entry
Scope alone	Preparation of the 1,120 body reconciliations up to the reviewer's queue; drafts of support checks for tail accounts. 80% of reconciliations and 41% of balance-sheet value
Evidence	Two closes bounded: 99.1% of bounded preparations met the standard
Permissions	Read the ledger, bank feeds, subledgers and plant schedules; prepare reconciliations; classify items; draft correcting entries into the journal entries queue. No permission to certify, post, write off or edit the standard
Switch-off test	Before the first live close: preparation returned to preparers in 20 minutes; close calendar unaffected
Audit consulted	Internal audit: store the evidence pack and the version of the standard with each certified reconciliation, and log that the reviewer opened the evidence before certifying
Auditor informed	External auditor told in writing of the change before the year-end walkthrough
Decision	Pass G2. Signed by the head of internal control



Like opening a field gate for a horse: *it may graze the fenced lower field, the house stays locked, and the farmer has shown he can bring it in fast.*

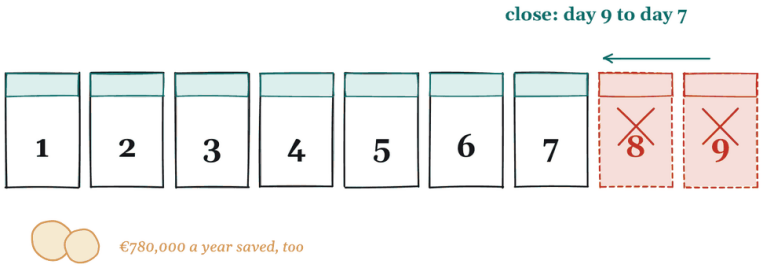
Step 8, Own: charter and economics

Field	Entry
Loop owner	Head of general accounting
Junction holders	4 reviewers; group financial controller; 3 plant finance managers for overlay memos
Exception owner	Reconciliation lead, open items over 60 days reviewed weekly
Grader	Finance quality analyst
Adoption owner	Finance operations manager
Control owner	Head of internal control
Changes to work	Preparers: eleven to five, working tail accounts and the reconciling-items queue. Three moved to the period close loop, two to intercompany settlement; one retired and was not replaced. Reviewers stay at four, with their time moved to the tail
Learning path	New accountants prepare tail accounts for three months and sit in on overlay junctions

Field	Entry
Period	Three month-end closes, including a quarter-end
Cost lines	Platform and model; grading and audit; reviewer time; exception queue; adoption
Hours a month	2,070 to 960: body 230, tail 730
Cost per accepted reconciliation	€50.40 overall; body €18.00; tail €189.20

Field	Entry
Baseline	€105.60; body €60.70; tail €319.60
Met the standard	Body 94% to 99.3%; tail 79% to 93%
Bank lines matched automatically	82% to 97%
Items older than 90 days	610 (€4.8 million) to 140 (€0.9 million)
Reconciliations certified by	Working day 8 to working day 5
Close	9 working days to 7
Annualised effect	About €780,000 a year in reconciliation cost. Two fewer close days and fewer aged items for auditors to test
Signed	Planning and analysis lead

The cost saving was real. The two close days mattered more to the chief financial officer, because they brought results to leaders earlier each month.

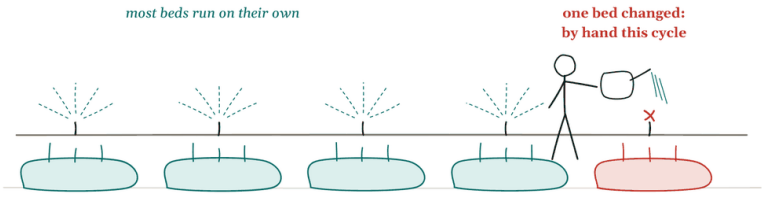


Like a tear-off calendar with two days gone: *the cost saving was real, but leaders valued most that results now arrive two days sooner each month.*

Step 9, Core: an operating review

Field	Entry
Month	Second month in Core
Standard met	Body 99.5%; tail 94%
Cost per accepted reconciliation	€48.90
Exception queue	Oldest item 41 days. Repeated exception: goods received but not invoiced mismatches in the clearing account; reported to the supplier invoice loop owner

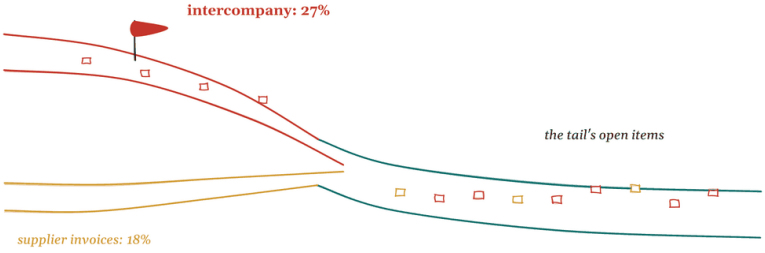
Field	Entry
Junctions	Reviewers returned one tail reconciliation in nine for more support. The group financial controller cut two plant overlays out of seven
Drift	One plant changed its accrual spreadsheet without notice; its accruals went to people for one close while reading was corrected
Audit signals	The external auditor's interim walkthrough accepted reviewer certification as the key control and asked for the standard's version log
Decisions	Extend the bounded support check to the warranty reserve at one plant, agreed by the head of internal control



Like a garden's sprinkler round: *most beds water themselves, but one bed changed without notice is hand-watered for a cycle while the sprinkler is reset.*

Step 10, Scale: what the edges showed

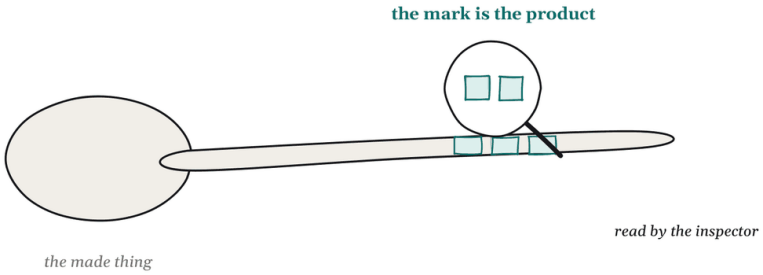
Reading the open items by cause showed that 27% of tail items came from intercompany balances that the three plants booked differently. Another 18% came from clearing accounts fed by the supplier invoice loop. The map review chose intercompany settlement as the next loop in the chain. The period close, now fed by earlier certified reconciliations, rose in readiness and was marked to read after it.



Like tracing litter on a riverbank upstream: *most of the tail washed down the intercompany stream, so that tributary is the next one to clean.*

What this loop teaches

When auditors read every record, the record is the outcome, not a by-product. The body here was bank and subledger matching, and the executor took it on quickly. The tail was a few hundred judgemental accounts. There, the hard work was agreeing what evidence counts, and only the chief financial officer and chief operating officer could settle it. Segregation of duties did not disappear. It moved, so the executor and the standard sit with different people, and a named reviewer still certifies. And the economics were read in close days and aged items, which a cost-only sheet would have missed.

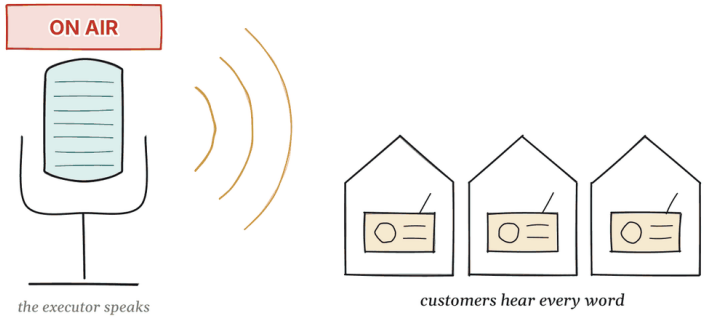


Like the hallmark on a silver spoon: *when an inspector reads every piece, the stamped record is part of what is sold, not a by-product of making it.*

XCVI

The Issue-to-Resolution Chain

Issue to resolution is the chain that takes a customer's problem and closes it. It starts with a question, a fault or a request, and runs through support, returns, complaints, warranty claims, field visits and the incidents behind them. Almost every loop in it speaks to customers directly. That changes the standard. A loop here is judged on what the customer experienced as well as on internal accuracy, and the company answers for every word its executor sends.



Like a radio station on air: *listeners hear every word as it is spoken, and the station answers for each one, so the loop is judged by what customers heard.*

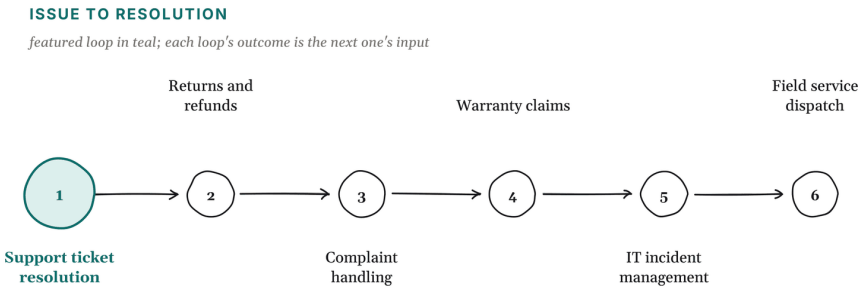


FIGURE 78 The issue to resolution chain. Its loops in handoff order, with the featured loop, support ticket resolution, in teal.

The loops and the evidence

The table gives, for each loop in the chain, the shape of its cases and a published benchmark. Shapes marked as estimates are reasoned from the nature of the work where no published distribution was found.

Loop	Shape of cases	Published benchmark
Support ticket resolution	Mixed	Median of 68.4% of service desk contacts resolved at first contact (MetricNet, 2023); $8.01 per contact on live channels against $0.10 through self-service (Gartner, 2019)
Returns and refunds	Mixed (estimate)	16.9% of US retail sales returned (NRF, 2024); median of $25 to manage returns per sales order fulfilled (APQC)
Complaint handling	Power law	56.6% of complaints closed by UK regulated firms were upheld; 44.7% closed within three days (FCA, 2025)
Warranty claims	Power law (estimate)	Median warranty claims of 0.65% of revenue (APQC); 1.33% of product sales for US manufacturers (Warranty Week, 2025)
IT incident management	Power law	Median of $88 per desktop support ticket and 67.2% resolved at first contact (MetricNet, 2023); 175 minutes on average to resolve a customer-facing incident (PagerDuty, 2024)
Field service dispatch	Mixed (estimate)	$196 per field support ticket in IT support (MetricNet and HDI, 2011 data, now old)

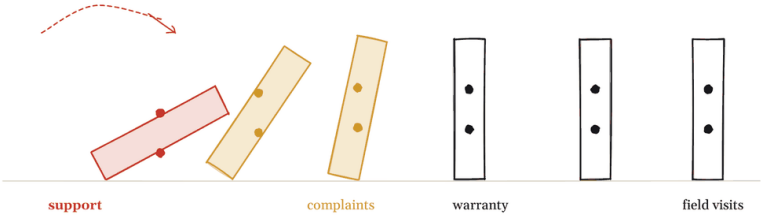
Named public deployments with reported results sit mostly in support and warranty. Results are as reported by the companies or their vendors.

Loop	Company	Reported result
Support ticket resolution	Klarna	2.3 million conversations in the first month, two-thirds of service chats; resolution time under 2 minutes against 11; 25% fewer repeat inquiries (vendor-reported)
Support ticket resolution	Salesforce, own help site	Over 1 million conversations by July 2025; 84% or more of questions resolved; 4% handed to a person by design
Warranty claims	Master Group	Claim payment time from up to 90 days to under 24 hours; backlog from over 2,000 claims to fewer than 10 (vendor-reported)
Returns and refunds	bstore	Post-return net promoter score of 95 from about 6,000 customers surveyed (vendor-reported)

Klarna's first figures measured volume handled. In 2025 its chief executive said the AI-first approach had produced lower quality, and the company began hiring human agents again so that a person was always available.

The featured loop is support ticket resolution. It is the chain's front door, so its errors become complaints, warranty disputes and wasted field visits further

along. It also shows the difference between handling a contact and resolving a problem.



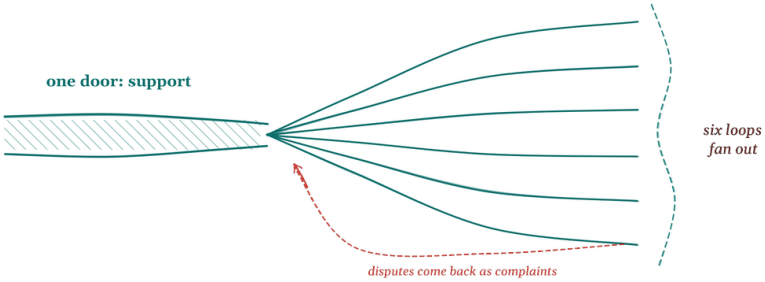
Like a row of dominoes: *support is the first tile, so an error there knocks on into complaints, warranty disputes and wasted field visits further down the chain.*

XCVII

The Loops of Issue to Resolution

Issue to resolution is the chain that hears a customer's problem and closes it. Work enters at one door, support ticket resolution, and fans out from there. A refund request goes to returns and refunds. Dissatisfaction that meets a regulator's definition goes to complaint handling. A product fault inside its cover goes to warranty claims. A cluster of similar contacts that points to a broken system goes to IT incident management, and a fault that cannot be fixed remotely goes to field service dispatch. Field visits then feed repair records back into warranty, and disputed decisions anywhere in the chain come back as complaints.

The six loops share one condition. Nearly all of them speak to customers or act on their equipment, so the company answers for every word and every action its executor takes. They differ most in where the tail comes from. In support and returns the tail is a small share of awkward cases inside a large routine flow. In complaints, warranty and IT incidents it is concentrated: one practice, one defect or one outage can generate most of the cost in a year.



Like a river delta: *every case enters through one channel, support, then splits toward returns, complaints, warranty, incidents or field visits, and disputes flow back as complaints.*

Support ticket resolution

At a glance	
Outcome	The customer's question or problem is answered or fixed, recorded and confirmed, with nothing left to chase
Shape of cases	Mixed: a large routine head and a small, costly escalated tail
Decisions	2 mechanical · 2 bounded · 1 consequential · 0 contested

At a glance	
Hardest to undo	Closing an account or denying service
Anchor figure	Median net first contact resolution of 68.4% at service desks (MetricNet, 2023)

Contacts arrive by chat, email, phone or web form from the customer channels of order to delivery and lead to cash. Most close here. The rest leave with a reference attached: refunds to returns, complaints to complaint handling, in-warranty faults to warranty claims, site visits to field dispatch and outage patterns to IT incident management.

Cases come in a mixed shape, and the support tier explains it. Each escalation adds a more expensive person. MetricNet and HDI priced a Level 1 ticket at $22 against $85 at Level 3 and $471 when a vendor is involved, in 2011 data now old. The body is order status, how-to questions and password resets; the tail is the escalated fault.

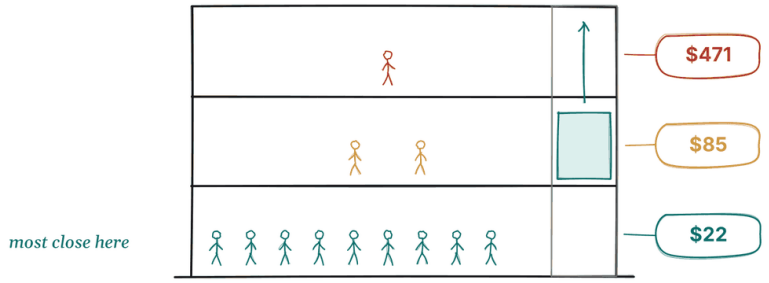
Four rules fix the design. The EU AI Act requires customers to be told when they deal with an AI system, and treats AI that allocates tickets to agents by their behaviour as high-risk. GDPR Article 22 gives a customer the right to human review of a solely automated decision with significant effect. In Moffatt v. Air Canada (2024), a tribunal held the airline to what its chatbot had said. Routing and answering from policy are mechanical. Goodwill credits and escalation are bounded. Closing an account, denying service or treating a customer as vulnerable is consequential.

Written rules should carry routing and every policy sentence, matched to approved text. A model reads the free text: intent, language, the fault behind a vague description. People hold the junctions whose effects are hardest to reverse, and every reply keeps a path to a person.

First contact resolution, the share of contacts solved without a second contact, had a median of 68.4% at service desks in MetricNet′s 2023 data. Gartner put a live-channel contact at $8.01 against $0.10 for self-service in 2019, yet only 9% of customers fully resolved an issue through self-service. Repeat contacts and reopened tickets show whether the answer held.

As read in 2026, Klarna′s assistant handled 2.3 million chats in its first month, two-thirds of the total, cutting resolution time from 11 minutes to under 2. In 2025 its chief executive said quality had dropped, and Klarna hired people again. Salesforce resolves 84% or more of help-site questions and hands 4% to

a person by design. No current public cost per consumer ticket exists, and no distribution of ticket value. Once the routine head moves to the executor, people see harder tickets on average. The next chapter follows this loop through its whole lifecycle.



Like a department store's floors: *most customers are served on the ground floor, and each floor the lift climbs brings a more expensive person, from $22 to $471.*

Returns and refunds

At a glance	
Outcome	An eligible return is authorised and refunded, exchanged or credited inside the legal deadline, with a credit note booked and the item dispositioned
Shape of cases	Mixed: routine eligible returns, with losses concentrated in fraud and fast-depreciating goods
Decisions	2 mechanical · 2 bounded · 1 consequential · 1 contested
Hardest to undo	Flagging a customer for return fraud or restricting a serial returner
Anchor figure	16.9% of US retail sales returned (NRF and Happy Returns, 2024)

A return request arrives from support with an order reference, a reason and often photos. Order to delivery supplies the shipment record that proves what was sent and when. The loop sends a credit note and refund instruction to billing in lead to cash, and returned stock with a disposition to inventory. Faulty goods worth recovering go to warranty claims, and disputed refusals go to complaints.

Most returns are low in value and plainly eligible. Losses gather in two places. Fraud made up about 13.6% of accepted returns in 2023, on NRF and Appriss

figures of $101 billion out of $743 billion. Value also decays unevenly: Optoro reported apparel losing 20% to 50% of its value in 8 to 16 weeks, and electronics 4% to 8% a month, in data now old. The tail comes from who returns what, not from how many.

The law sets much of the loop. The EU Consumer Rights Directive grants 14 days to withdraw from a distance sale and caps the refund wait at 14 days. The UK Consumer Rights Act gives 30 days to reject faulty goods. In the US, a refund the merchant refuses can become a card dispute under Regulation Z or E. Eligibility and refund method are mechanical. A returnless refund, where the customer keeps an item not worth shipping back, is bounded, as is disposition. A fraud flag is consequential under GDPR Article 22. Exception refunds outside policy are contested, because goodwill and fairness between customers pull against each other.

Rules take the window, exclusions and payment route. A model can weigh item value, handling cost and customer history to propose returnless refunds and grade condition for disposition. A person decides any restriction on a customer.

APQC reports a median of $25 to manage returns per sales order fulfilled, without stating the edition year. Narvar estimated $25 to $30 to process each return in 2024. No published median exists for time from request to refund; the 14-day legal ceiling is the only firm number.

As read in 2026, Klarna's assistant handles returns conversations, with no returns-specific figure. Walmart and Amazon decide returnless refunds on purchase history, value and processing cost, with no quantified result. bstore reported a post-return net promoter score of 95 from about 6,000 customers using Loop Returns. No public case shows a language model approving returns end to end with reported accuracy. Generous automated thresholds tend to become known, and can raise the very return and fraud rates they were set against.



Like a teacher slipping one child an extra sweet: *kind to that child, unfair to the queue watching, which is why refunds outside policy are a contested decision.*

Complaint handling

At a glance	
Outcome	A complaint is acknowledged, investigated and answered with a final response and any redress inside the regulatory deadline, with escalation rights stated
Shape of cases	Heavy-tailed: concentrated by product, firm and mass-claim event
Decisions	1 mechanical · 1 bounded · 2 consequential · 1 contested
Hardest to undo	Rejecting a complaint in a final response
Anchor figure	56.6% of complaints closed by UK regulated firms were upheld (FCA, 2025)

Complaint handling produces a final response: the complaint upheld or not, any redress paid, and the customer told how to escalate. Redress means money or action that puts the customer right. Cases arrive from support when a contact meets the complaint definition, from returns and warranty when a decision is disputed, and from regulators or ombudsmen with a deadline attached. Redress and complaint data go to record to report for regulatory returns. Patterns that suggest a product defect go to warranty and recall.

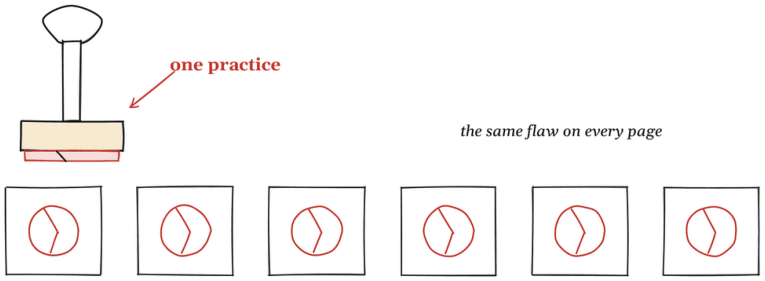
This loop has a heavy tail, meaning a few sources carry most of the volume. In the CFPB's 2024 data, 85% of complaints concerned credit or consumer reporting, and about 79% named three firms. At the UK Financial Ombudsman, motor hire purchase alone made up about a quarter of new cases in 2024/25. The mechanism is replication. One practice applied to many customers, such as motor finance commission, produces complaints by the hundred thousand.

Deadlines are fixed by rule. The FCA requires a final response within 8 weeks, or 15 business days for payment services. The CFPB expects a response within 15 days and a final one within 60. US medical device makers must evaluate every complaint for reportability to the FDA. Setting the clock is mechanical. Deciding whether a contact is a complaint at all is bounded, and misreading it is a conduct breach. Upholding and the amount of redress are consequential. Whether a pattern calls for a wider remediation programme or a regulatory report is contested, since legal and reputational costs pull apart.

Rules can classify against the definition's wording and run the clocks. A model can draft the investigation summary and spot clusters across cases. People sign the final response, because the ombudsman reviews it.

The FCA reported 1,783,169 complaints closed in the second half of 2024, 44.68% within three days and about 94.7% within eight weeks. Redress averaged about £132 per closed complaint, counting payments only. The sharper quality measure sits downstream: the ombudsman upheld 34% of escalated cases against firms in 2024/25. The CFPB received about 3.19 million complaints in 2024.

As read in 2026, the only named deployment is Salesforce, which hands 4% of help-site conversations to people by design so dissatisfied customers keep a human path. No public case shows an AI agent deciding uphold or reject on regulated complaints, and no handling-cost benchmark exists. Good root-cause work should thin the tail over time, since fixing one practice removes a whole cluster.



Like a cracked rubber stamp: *the one flaw is pressed onto every page, so a single practice applied to many customers produces complaints by the hundred thousand.*

Warranty claims

At a glance	
Outcome	A claim is validated against cover, approved or rejected with a reason, paid or credited, and any supplier share recovered
Shape of cases	Heavy-tailed (estimate): cost driven by a few failures and industries
Decisions	1 mechanical · 1 bounded · 2 consequential · 2 contested
Hardest to undo	Declaring a failure pattern a safety defect
Anchor figure	Median warranty claims of 0.65% of revenue (APQC, edition year not stated)

Claims reach this loop from three directions. Support passes faults reported inside the warranty period. Field dispatch passes repair records with parts, labour and a failure code. Lead to cash supplies the registration, serial number and sale date that establish cover. Out of the loop go dealer payments and supplier chargebacks to procure to pay, reserve entries to record to report, and failure trends to product quality. Rejected claims the customer disputes become complaints.

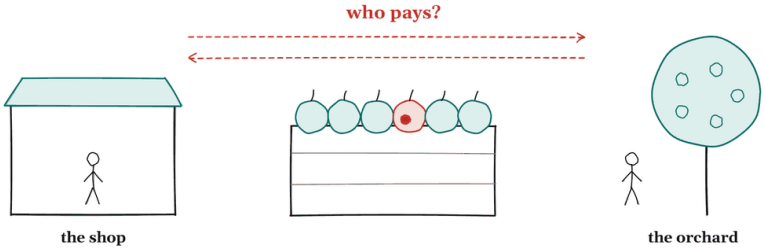
Claim cost is concentrated. Warranty Week found vehicles accounted for 68% of the $29.2 billion US manufacturers paid in 2024. Single failure events move whole company rates: SolarEdge went from 4.6% to 15.6% of sales, and Brunswick from 0.4% to 2.0%. The cause is that one defect repeats across every unit built with the faulty part. Per-claim severity is not published, so this shape is an estimate.

In the US, the Magnuson-Moss Warranty Act sets a 40-day limit where a dispute mechanism is used. The TREAD Act requires vehicle makers to report claim data quarterly and file a defect report within five business days of finding a safety defect. In the EU, a two-year legal guarantee runs beside any commercial warranty. Reserve estimates need signed controls. Cover is mechanical. Checking labour times and parts against rate tables is bounded. Approval and fraud suspicion are consequential. Supplier recovery and recall escalation are contested, because root cause becomes a negotiation or a legal finding.

Rules handle cover and rate tables. A model adds reading of technician notes and failure descriptions, and pattern detection across claims. People decide rejections, fraud and anything that may be a defect.

Warranty Week put US manufacturers′ claims at 1.33% of product sales in 2024, against a 22-year average of 1.42%. Fraud was estimated at 3% to 15% of warranty cost in an old column whose study is unnamed. A 2024 WarrCloud study found auto dealers spending 47% more time per claim than in 2020. No cross-industry cost or cycle time per claim is published, and no rejection-rate benchmark.

As read in 2026, Master Group, a Canadian HVAC distributor, cut claim payment from up to 90 days to under 24 hours with Syncron and its backlog from over 2,000 claims to fewer than 10. GAF reported shorter guarantee issuance with Tavant, without a figure. No named case shows a generative AI agent approving claims with reported accuracy. Cleaner, faster claim data should let engineering catch defects earlier, shrinking the next tail.



Like a rotten apple in a crate: *the shop refunds the customer at once, then argues with the orchard over who caused it, which is a negotiation, not a rule.*

IT incident management

At a glance	
Outcome	Service is restored within the agreed level, the incident closed with a cause, and any required notification made
Shape of cases	Heavy-tailed: routine tickets beside rare, very costly outages
Decisions	1 mechanical · 2 bounded · 1 consequential · 1 contested
Hardest to undo	A regulatory or market disclosure, once filed
Anchor figure	175 minutes on average to resolve a customer-impacting incident (PagerDuty, 2024)

The loop restores normal service. Input comes from support tickets that point to a failure, from monitoring alerts, and from the employee service desk. Root-cause work goes to problem and change management. On-site fixes go to field dispatch. Status updates and service credits go back to support and complaints. Material incidents go to record to report for disclosure.

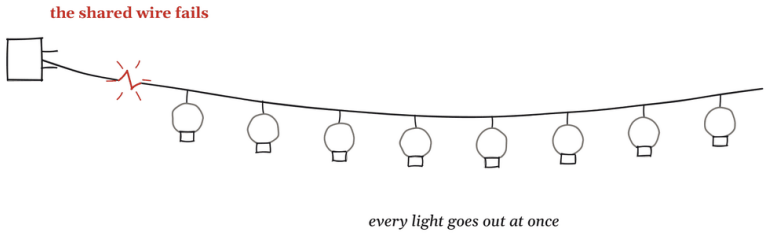
Most incidents are routine desktop and user issues; a few outages carry most of the cost. Uptime Institute found 76% of operators reported only minimal outages and 3% severe ones, yet 20% of significant outages cost over $1 million. Shared dependencies explain the tail. When a common system fails, every service on it fails at once.

Reporting rules turn severity into a legal question. NIS2 requires an early warning within 24 hours of a significant incident and a notification within 72. DORA sets classification and reporting duties for EU financial firms. The SEC requires disclosure within four business days of deciding a cyber incident is material. GDPR Article 33 sets 72 hours for many data breaches. Correlating alerts is mechanical. Setting priority and applying a runbook, a written set of recovery steps, are bounded. Declaring a major incident is consequential. Whether an incident is reportable or material is contested, a legal judgement that accountable people must own.

Rules and monitoring tools can open incidents and run known recovery steps inside guardrails. A model helps summarise alerts and match them to past fixes. SOX controls require that whoever makes an emergency change does not approve it. People declare major incidents and own disclosure.

MetricNet's 2023 data gives a median of 0.39 desktop tickets per user a month, 13.4 business hours to resolve, $88.07 per ticket and 67.2% resolved at first contact. PagerDuty's 2024 survey found 25 high-priority customer-facing incidents a year, averaging about $794,000 each. Manual handling averaged 3 hours 58 minutes, against 2 hours 40 minutes fully automated.

As read in 2026, the dataset holds no named public deployment of an AI agent resolving incidents with reported figures. No current public cost per incident ticket exists apart from cost per contact. As automation absorbs routine incidents, people may see major outages less often, so their practice on the tail needs deliberate care.



Like fairy lights wired in series: *most nights one bulb flickers and is swapped, but when the shared wire fails, every light on the string goes out at once.*

Field service dispatch

At a glance	
Outcome	A qualified technician with the right parts arrives in the promised window, fixes the fault on the first visit where possible, and closes a signed job record
Shape of cases	Mixed (estimate): routine jobs with a costly minority of repeat and safety-critical visits
Decisions	1 mechanical · 3 bounded · 1 consequential · 0 contested
Hardest to undo	Declaring the asset safe and the job complete
Anchor figure	$196 per field support ticket in IT support (MetricNet and HDI, 2011 data, now old)

Field dispatch puts a person on site. Faults arrive from support with asset, location and symptoms, and from IT incidents needing hardware or network work. Inventory reports which parts sit in which depot or van. The completed job feeds warranty claims when covered, billing when chargeable, parts replenishment, and payroll for hours and travel.

No published distribution of job value or complexity exists, so the shape is reasoned. Most jobs are routine installs and repairs of similar length. A minority of repeat visits, safety-critical work and outages on high-value equipment likely carry most of the cost and risk. Independent jobs of similar length produce a stable body. The tail comes from assets whose failure stops a customer's operation.

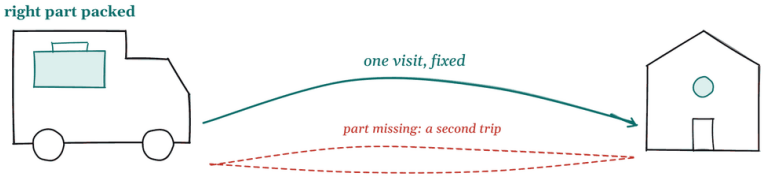
Rules here protect workers as much as customers. The EU AI Act treats software that assigns jobs on individual behaviour, or monitors technicians, as high-risk. The Platform Work Directive gives platform workers human review

of significant automated decisions. The Working Time Directive caps hours, and GDPR limits continuous location tracking. OSHA's lockout and tagout rule requires a trained person to cut off and verify an energy source before servicing. Assigning technician, slot and route is mechanical. Deciding between remote fix and visit, re-planning the day and choosing parts are bounded. Declaring the asset safe is consequential.

An optimiser should own routing within working-time limits. A model adds remote diagnosis and predicted parts lists. A dispatcher holds the trade-offs when the day breaks, and a qualified technician signs off safety.

The only sourced cost is $196 per field ticket, set against $22 for a Level 1 ticket in the same old benchmark, so an avoided visit is the main prize. MetricNet's 2023 median technician utilisation of 53.5% in desktop support is a rough proxy for dispatch efficiency. No recent first-time fix rate or repair-time benchmark was found.

As read in 2026, the dataset holds no named deployment of AI dispatch with reported results. Better remote diagnosis removes easy visits first, so the jobs that remain will be harder and longer on average.

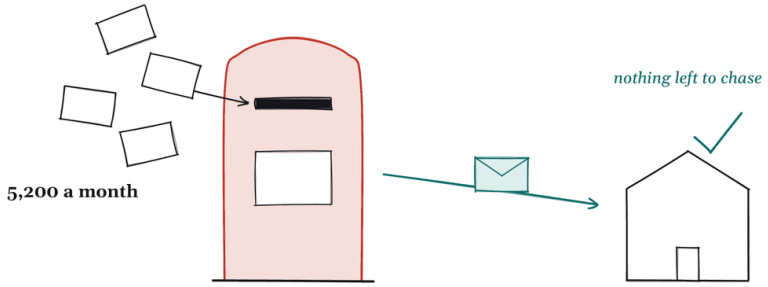


Like a plumber's van packed before setting off: *one trip with the right part fixes the fault, while a missing part means a second costly visit.*

XCVIII

Support Ticket Resolution

At the composite manufacturer, the loop receives about 5,200 tickets a month by email, web form, chat and phone. They come from distributors and from end users of its pumps and valves. They ask about order status, installation, faults and spare parts. The outcome is a problem answered or fixed, a recorded resolution, and a customer who does not have to chase.



Like a busy postbox: *thousands of letters drop in each month, and each is finished only when the reply arrives and the sender has no reason to write again.*

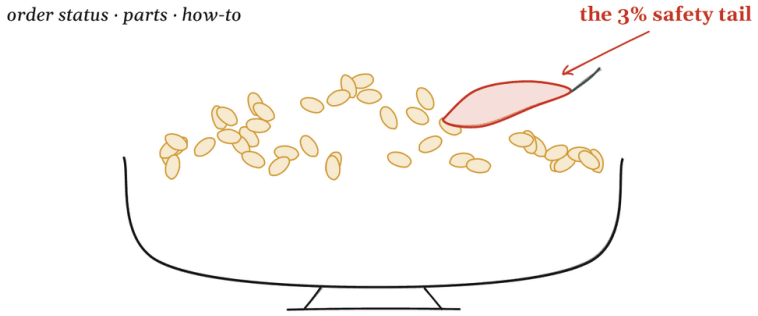
Step 1, Read: the map entry

The chief commercial officer's map placed the loop at value four and readiness three. Volume was high and the work visible to every customer. Readiness was held back by product documentation spread across three plants.

Field	Entry
Loop	Support ticket resolution
Chain	Issue to resolution
Value to the core	4: about 5,200 tickets a month; every customer relationship touched
Readiness	3: ticket system in place; order and shipment data readable; installation manuals in three formats; owner willing
Handoffs	In from customer channels and order to delivery; out to returns, complaints, warranty claims and field dispatch
Risk category	Not high risk if tickets are routed to queues, not allocated to individual agents on their behaviour; customers must be told when they deal with an AI system

Field	Entry
Owner	Head of customer service (agreed)
Distribution read	By case: order status 41%; spare parts 23%; installation questions 18%; faults 14%; documents and certificates 4%. Within faults, about 3% of all tickets touch safety-related equipment such as pumps for hazardous fluids or valves on gas lines
Signed	Chief commercial officer

Most tickets were cheap and repetitive. The tail was the fault tickets. A single fault on a gas valve could carry a warranty promise, a safety risk and a field visit.



Like a bowl of peanuts with one chilli in it: *almost every handful is ordinary, but the few safety-related faults carry warranty, safety and field-visit risk.*

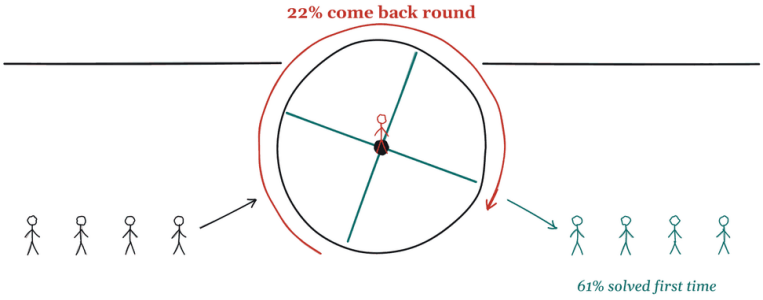
Step 2, Map: the baseline sheet

Mapping with service agents and the technical specialists drew fourteen steps across six lanes: customer, service desk, technical support, order management, quality and field service.

Field	Entry
Sample and period	7,400 tickets over six weeks
Volume	5,200 tickets a month
Fully loaded cost per resolved ticket	€29.80: €142,500 a month across 4,780 tickets resolved to standard
First-contact resolution	61%
Repeat contacts	22% of tickets were a second or later contact on the same problem

Field	Entry
Reopened tickets	7%
Complaints	38 a month
First reply	9 working hours on average
People	22 service agents, 5 technical specialists
Method agreed with finance	Yes, planning and analysis lead

First-contact resolution sat seven points below the published median. One ticket in five was a customer asking again.



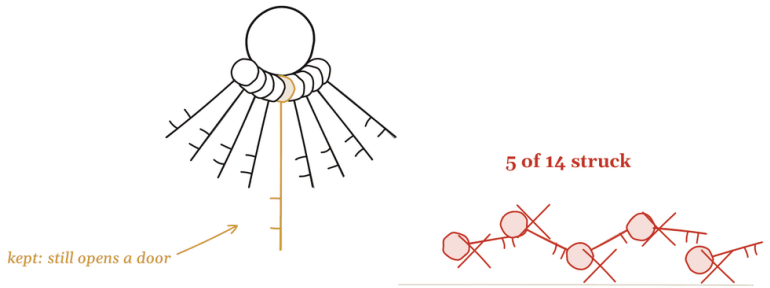
Like a revolving door where one person in five comes straight back round: *the baseline found 22% of tickets were customers asking again.*

Step 3, Strip: the strip register

Step	Status	Constraint	Still applies?
Phone the plant to confirm a shipping date	Struck	Plant schedules were not visible to service	No: shipment data has been in the ledger since 2022
Copy the ticket into a spreadsheet for the weekly report	Struck	The old ticket system had no reporting	No
Forward spare-part questions to the plant parts clerk	Struck	Part numbers lived only in plant files	No: the parts catalogue is now central
Send a printed installation manual by post on request	Struck	Manuals were not online	No
Supervisor approves every goodwill credit	Struck	Credits had no written limit	No: a limit is now set
Quality engineer decides on warranty cover	Kept	Cover depends on failure analysis	Yes: becomes a junction

Step	Status	Constraint	Still applies?
Confirm the answer with the customer before closing	Kept	Required by the outcome	Yes

Five of fourteen steps were struck, a strip ratio of 36%.



Like a key ring cleared of keys to locks changed years ago: *five of fourteen steps went, while the one that still opens a real door stayed.*

Step 4, Sort: rules, decisions and the contested log

The rule register held 58 rules. Nine contradicted each other. The two that mattered most are shown.

Rule	Source	Contradiction	Decision
Goodwill credit up to €100 per ticket	Service policy	Agents at one plant gave up to €300	Head of customer service: up to €150 per ticket and €400 per customer a quarter, reported weekly
Delivery dates quoted from the order confirmation	Order policy	Agents quoted the plant's latest estimate	Order management lead: quote the current shipment date and say when it changed

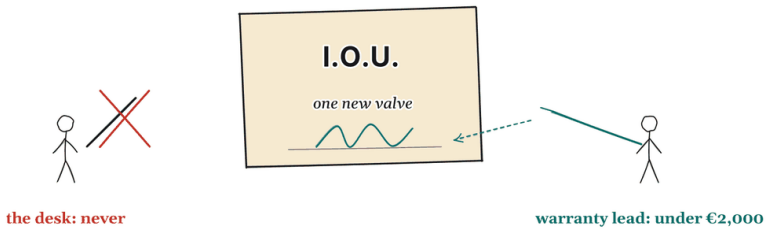
The decision register held sixteen decisions.

Decision	Class	Limit or junction
Classify the ticket and route it to a queue	Mechanical	
Answer order status from shipment data	Mechanical	
Identify a spare part from serial number and catalogue	Mechanical	
Answer an installation question from approved documentation	Bounded	Approved text only; above a confidence threshold set in shadow

Decision	Class	Limit or junction
Grant a goodwill credit	Bounded	Up to €150 per ticket, €400 per customer a quarter
Hand a ticket to returns or complaints	Bounded	Written criteria
Treat a fault as safety-related	Consequential	Product safety engineer
Book a field visit	Consequential	Field service manager
Promise warranty cover before inspection	Contested	See log

Field	Entry
Decision	May the service desk tell a distributor that a failed part will be replaced under warranty before the part is inspected?
Positions	Sales: yes for key distributors, to keep their customers running. Quality: never before failure analysis
Logged	Week 1 of Sort
Settled by	Chief commercial officer and chief operating officer
Answer	The executor never promises cover. It opens a claim and states the inspection timeline. An advance replacement before inspection is allowed for parts under €2,000, decided by the warranty lead
Date settled	Week 2 of Sort
Resulting class	Consequential, with a junction held by the warranty lead

The ruling followed a known precedent. In Moffatt v. Air Canada (2024), a tribunal held the airline to a refund policy its chatbot had wrongly described. A company is bound by what its executor tells customers.

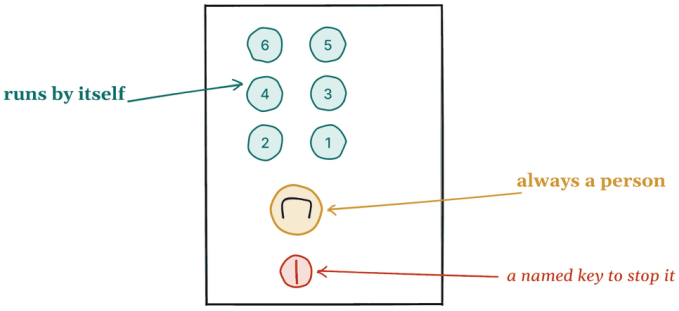


Like an IOU slip: *whoever writes it binds the shop, so the desk never promises cover, and only the warranty lead may approve an advance part under €2,000.*

Step 5, Design: junctions and the standard

Junction	Who decides	Evidence shown	How reversed
Safety-related fault	Product safety engineer	Ticket text, product, installation, fluid or gas, fault history for the serial number	Downgrade to a normal fault with a note to the customer
Field visit	Field service manager	Fault description, remote checks done, contract terms, nearest technician	Cancel the visit before dispatch
Advance replacement under warranty	Warranty lead	Serial number, purchase date, failure description, distributor's claim history	Invoice the part if inspection finds no warranty fault
Customer asks for a person	Service agent on duty	Full ticket history	Not applicable: the customer always gets a person
Switch-off	Head of customer service	Grading dashboard	All tickets return to the agents within 30 minutes

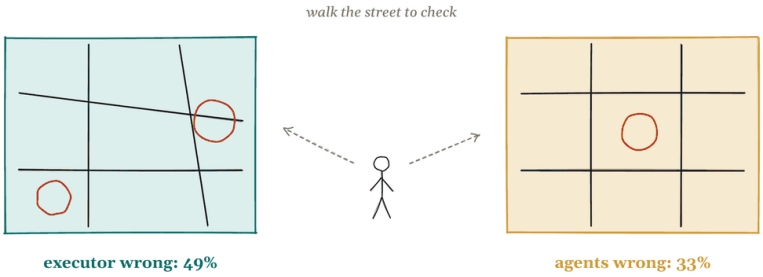
Field	Entry
Criteria	Each ticket answered correctly and closed with a recorded resolution; every policy statement matches the approved policy library word for word; no promise of warranty cover, refund or visit; a person offered in every reply and reached within one working hour
Rule checks	Every policy sentence checked against the library; every credit checked against both limits; every fault ticket checked against the safety word list
Model grader	Every reply graded against the criteria
Human audit	3% at random; 20% of fault tickets
Customer signals	Repeat contacts within 14 days; reopened tickets; complaints; requests for a person
Calibration	Monthly, by the service quality analyst



Like a lift's control panel: *the lift runs by itself, a help button always reaches a person, and only the named manager's key can switch it off.*

Step 6, Shadow: the report

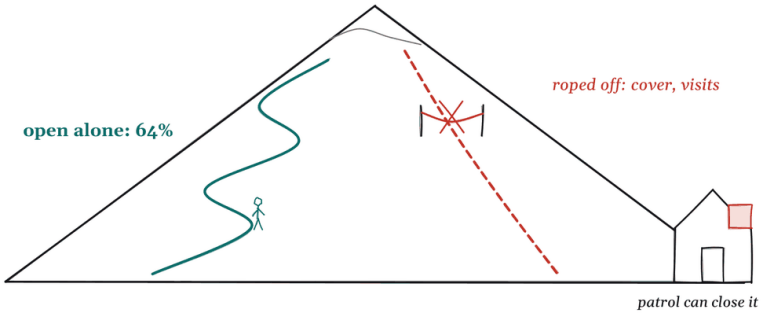
Field	Entry
Period and cases	Six weeks; 7,300 tickets
Agreement, body	97.6%
Agreement, tail	81%: faults, mixed tickets and installations in hazardous settings
Disagreements	Executor wrong 49%: superseded part numbers for an older valve series; installation steps from the wrong plant's manual. Person wrong 33%: agents quoting old delivery dates and giving credits above the limit. Both defensible 13%. Standard unclear 5%
Changes made	Superseded parts mapped; manuals tagged by plant and series; safety word list widened
Known weaknesses	Tickets that raise two problems at once
Recommendation	Pass G1



Like two maps of one town held side by side: *wherever they differ, someone walks the street, and both turn out wrong, the executor more often than the agents.*

Step 7, Bound: the gate review record

Field	Entry
Scope alone	Order status; spare parts in the central catalogue; installation questions above the threshold; goodwill credits within limits. 64% of tickets
Evidence	Four weeks bounded: 98.8% of bounded tickets met the standard; no policy statement outside the library; 6% of customers asked for a person and all reached one within the hour
Permissions	Read orders, shipments and the parts catalogue; send replies; issue credits within limits; hand tickets to returns, complaints, warranty and field service. No permission to promise cover, book visits, or change orders or prices
Switch-off test	Week 1: tickets returned to the agents in 20 minutes; nothing lost
Audit consulted	Legal: every reply states that the customer is dealing with an AI system and how to reach a person
Decision	Pass G2. Signed by the head of quality and compliance



Like a ski map for a new skier: *gentle runs open alone, steep ones roped off, and the patrol has shown it can close the mountain in minutes.*

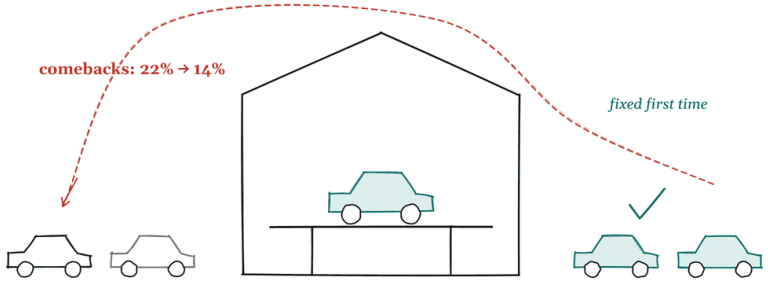
Step 8, Own: charter and economics

Field	Entry
Loop owner	Head of customer service
Junction holders	Product safety engineer; field service manager; warranty lead; agents on duty
Exception owner	Service desk lead, no ticket waiting more than four working hours
Grader	Service quality analyst
Adoption owner	Service operations manager
Control owner	Head of quality and compliance
Changes to work	Service agents from 22 to 11, now on faults, mixed tickets and customers who ask for a person. Specialists from 5 to 6, one on safety triage. Four agents moved to a distributor success team, three to the warranty desk, two to field dispatch; two left by choice and were not replaced
Learning path	New agents work the person queue for two months and sit in on safety junctions

Field	Entry
Period	Eight weeks
Cost lines	Platform and model; grading and audit; agents and specialists; junction time; adoption
Tickets	5,200 to 4,720 a month, as repeat contacts fell
Cost per resolved ticket	€23.90 overall; body €9.80; tail €88.40
Baseline	€29.80
First-contact resolution	61% to 79%
Repeat contacts	22% to 14%
Reopened tickets	7% to 3%
Complaints	38 to 29 a month
Annualised effect	About €394,000 a year in service cost

Field	Entry
Signed	Planning and analysis lead

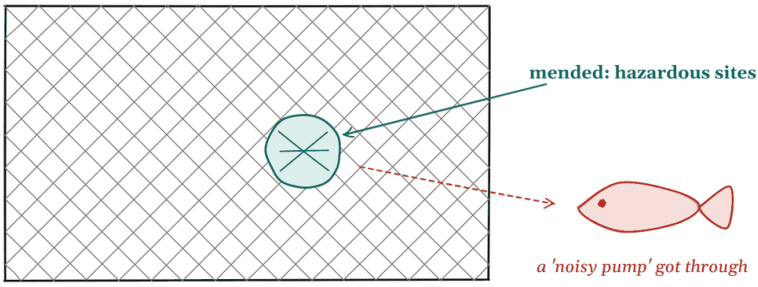
The saving per ticket was modest. Fewer tickets arrived because fewer customers had to ask twice, and that was worth more than faster answers.



Like a garage whose cars stop coming back: *fixing them right the first time shortened the queue, which saved more than working faster on each car.*

Step 9, Core: an operating review

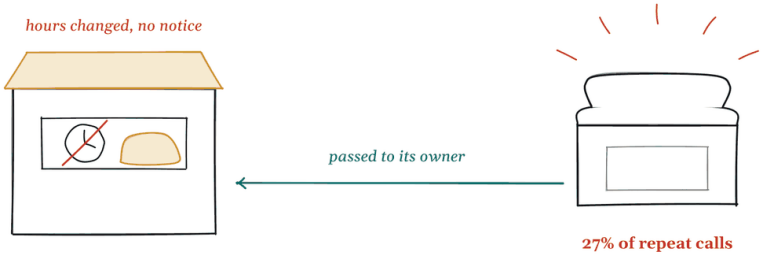
Field	Entry
Month	Second month in Core
Standard met	Body 98.9%; tail 94%
Cost per resolved ticket	€23.40
Exception queue	Oldest ticket 3 hours. Repeated exception: tickets naming a distributor's own part code; mapping added
Junctions	The product safety engineer cleared 2 of 31 flagged faults as not safety-related. Audit found one missed: a "noisy pump" in a chemical plant, read as an installation question. A rule was added for any fault at a site handling hazardous fluids
Drift	A new valve series entered the catalogue; its tickets went to people for a day while the executor's parts list was updated
Customer signals	Requests for a person steady at 7%; median wait 38 minutes
Decisions	Widen bounded scope to document and certificate requests, agreed by the head of quality and compliance



Like a fishing net mended after one fish slipped through: *the audit found a 'noisy pump' at a chemical plant missed, so a rule now catches every hazardous site.*

Step 10, Scale: what the edges showed

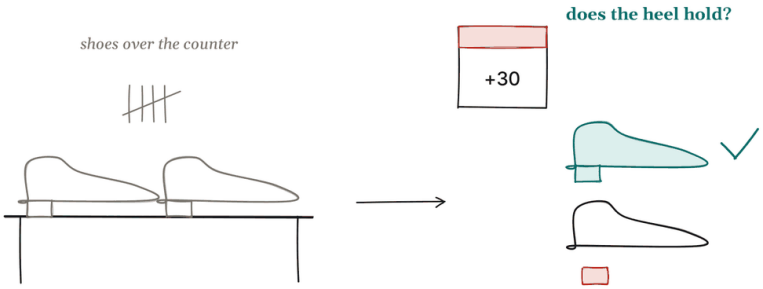
Reading repeat contacts by cause showed that 27% came from shipping dates changed without notice. That cause sat upstream, in order to delivery, and was passed to its owner. Inside the chain, fault tickets now arrived at the warranty desk with serial numbers and clean failure descriptions. The map review chose warranty claims as the next loop.



Like a phone that keeps ringing because a bakery moved its opening hours without notice: *the fix is a word with the bakery, not more people answering.*

What this loop teaches

A customer-facing loop is measured by what customers do next. Repeat contacts, reopened tickets and complaints showed whether a problem was solved. Conversations handled showed only activity. Klarna's early result counted volume, and its later turn back to human agents showed what a volume count misses. The company also answers for its executor's words, so policy statements came only from approved text. Warranty, safety and field visits stayed with people. And every customer could reach a person, which kept the trust that let the rest run alone.

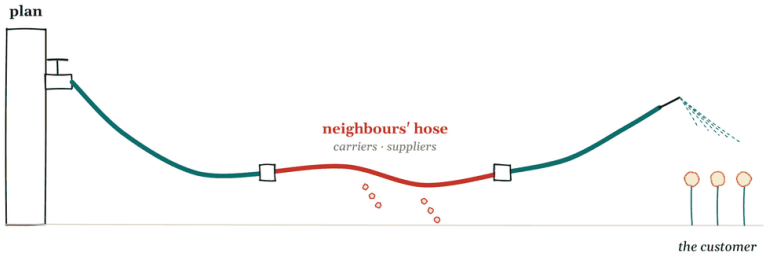


Like a cobbler judged by heels that still hold a month later, not by shoes across the counter: *repeat contacts and complaints show whether a problem was solved.*

XCIX

The Order-to-Delivery Chain

Order to delivery is the chain that puts the right goods in front of the customer on the promised date. It runs from planning demand and replenishing stock, through picking, packing and shipping each order, to handling what goes wrong on the road and at the supplier's dock. It touches planning, the plants, the warehouses, transport and customer service. Much of its data comes from outside the company, from carriers, forwarders, brokers and suppliers. That makes the quality of other people's data part of every loop's standard.



Like garden hoses joined from tap to flowerbed, some borrowed from neighbours: *the goods arrive on the promised date only if the borrowed sections hold too.*

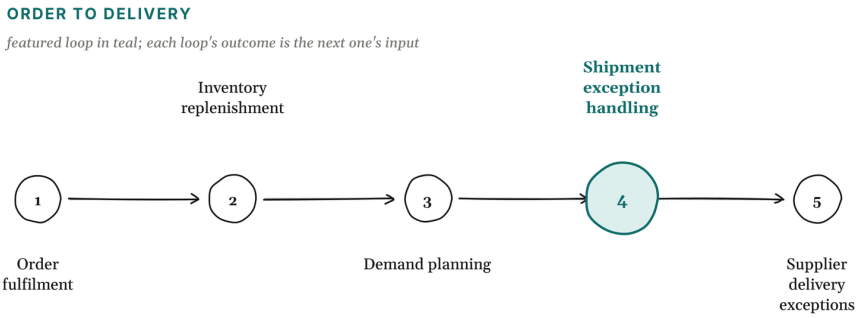


FIGURE 79 The order to delivery chain. Its loops in handoff order, with the featured loop, shipment exception handling, in teal.

The loops and the evidence

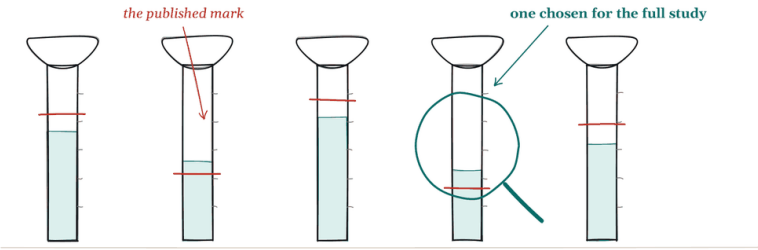
The table gives, for each loop in the chain, the shape of its cases and a published benchmark. Shapes marked as estimates are reasoned from the nature of the work where no published distribution was found.

Loop	Shape of cases	Published benchmark
Demand planning	Mixed	Median forecast accuracy of 85% of units shipped (APQC); 75% of statistical forecasts adjusted by hand on average, with small adjustments adding little or harming accuracy (Fildes and Goodwin, 2007, the latest public figure)
Inventory replenishment	Power law (estimate)	Median inventory carrying cost of $19 per $1,000 of revenue (APQC)
Order fulfilment	Mixed (estimate)	Median of 88% perfect orders: on time, complete, undamaged, with correct documents (APQC); median of $8.48 to manage a sales order (APQC); median of 15 days from order to delivery (APQC and Esker, 2026)
Shipment exception handling	Mixed	6.2% of shipments with an exception by volume, 7.7% at the median organisation; 10 to 15 minutes of manual investigation per exception (AfterShip tracking data, 2026, vendor estimate)
Supplier delivery exceptions	Power law (estimate)	Median of 90% of supplier deliveries on time (APQC)

Named public deployments in the chain cluster in planning, replenishment and order intake. Results are as reported by the companies or their vendors.

Loop	Company	Reported result
Demand planning	Amazon	10% better long-term national forecasts for deal events; 20% better regional forecasts for millions of popular items (company-reported)
Inventory replenishment	Amazon	Delivery times cut by almost a day (company-reported)
Inventory replenishment	Wm Morrisons	Shelf gaps reduced by up to 30% (vendor-reported, 2017)
Order fulfilment	Topa Bathroom Products	Over 90% of incoming orders processed with no manual input (vendor-reported)
Shipment exception handling	Dr. Squatch	Exception resolution time improved by more than 58%; most frequent "where is my order" tickets down 25% (vendor-reported)

The featured loop is shipment exception handling. Its executor works mostly on data owned by other companies. Half of its alerts resolve themselves, while a small minority are costly and cannot be undone. And its economic effect sits in freight premiums and on-time delivery rather than in handling cost.



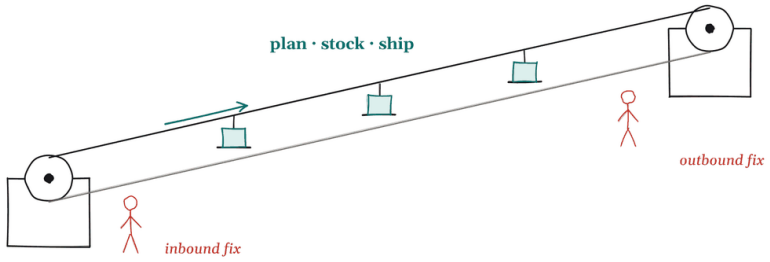
Like rain gauges set out in five gardens: *each loop's level is read against a published mark, and one garden is chosen for the full study.*

C

The Loops of Order to Delivery

Order to delivery turns expected demand into goods on a customer's doorstep. Its loops hand work along a fixed line. Demand planning produces a forecast, replenishment turns it into stock at each location, and order fulfilment ships from that stock. Two exception loops then repair what breaks: shipment exception handling on the way out, supplier delivery exceptions on the way in. The supplier loop feeds revised dates back to replenishment and fulfilment, so the chain closes on itself.

The five entries share one trait. Much of the data they act on belongs to someone else, whether a carrier, a supplier or a customer's purchasing system. They differ most in where the hard cases sit. In planning and replenishment the tail is a small set of items or overrides that carry most of the value. In the two exception loops the tail is a minority of events that cost money and cannot be reversed.



Like a ski lift: *chairs travel one way from plan to stock to shipping, a mechanic at each station frees what sticks, and the cable closes on itself.*

Demand planning

At a glance	
Outcome	A consensus forecast by product, location and period, with accuracy measured, signed off and received downstream
Shape of cases	Mixed: routine items plus a few large overrides that carry the value
Decisions	1 mechanical · 2 bounded · 1 consequential · 1 contested
Hardest to undo	Changing the forecasting method and accuracy metric

At a glance

Anchor figure	Median forecast accuracy of 85% of units shipped (APQC, accessed 2026)

Demand planning produces the number the rest of the chain builds on. It takes order and shipment history from fulfilment, pipeline and promotions from the sales side, and stock-out history from replenishment. The forecast goes to replenishment by product and location, to purchasing as material requirements, and to finance as volume assumptions for budgets. It is signed off in sales and operations planning (S&OP), the monthly meeting where the functions agree one plan.

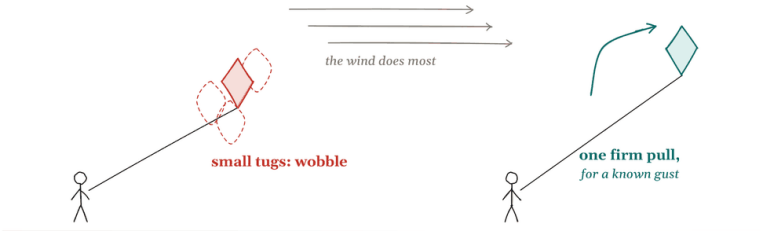
The cases here are forecasts, and they split cleanly. Most items are best left to the statistical baseline. A few large overrides, made by people who hold real information, add most of the value. Fildes and Goodwin studied over 60,000 forecasts at four UK companies in 2007. On average 75% were adjusted by hand. Large adjustments tended to help, small ones added little or harmed accuracy, and upward adjustments often went the wrong way. At one retailer, those upward adjustments more than doubled the average percentage error, from 32% to 65%.

No law demands a signature on a forecast as such; the signatures come from accounting. Where forecasts drive inventory reserves or guidance, SOX Section 404 requires documented management review. Under IAS 2, expected sales underpin the write-downs that management must judge. Generating the baseline is mechanical. Cleaning history and applying market overrides are bounded. Changing the method or the accuracy metric is consequential, because every downstream commitment rests on it. The consensus number itself is contested: sales wants a target, operations wants realistic demand, and the meeting is a negotiation, not a calculation.

The model should own the baseline for the body of items. People should spend their time on the few large, information-bearing overrides, and a written rule should block small ones. The S&OP chair holds the junction on the consensus number.

APQC puts median forecast accuracy at 85% of units shipped. Its Metric of the Month column reports a 90% median by value, with top performers at 95%. No public benchmark exists for the cost of planning or for the length of the S&OP cycle.

As read in 2026, Amazon reports 10% better long-term national forecasts for deal events and 20% better regional forecasts for millions of popular items. Walmart's in-house neural network predicts demand at several horizons, with no results published. HEINEKEN announced a Blue Yonder implementation in 2024 with no results reported. The override evidence is nearly twenty years old. A good model should reduce small overrides over time, which leaves people with fewer but harder calls.



Like flying a kite: *the wind carries most of the work, constant small tugs only make it wobble, and one firm pull for a known gust helps.*

Inventory replenishment

At a glance	
Outcome	Each location holds the right quantity of each item for the target service level, and receives a released replenishment order or transfer
Shape of cases	Heavy-tailed (estimate): few items carry most volume and value
Decisions	1 mechanical · 1 bounded · 2 consequential · 1 contested
Hardest to undo	Writing off expiring or obsolete stock
Anchor figure	Median inventory carrying cost of $19 per $1,000 of revenue (APQC, accessed 2026)

Replenishment decides how much stock each store, warehouse or line should hold, and orders it. The consensus forecast arrives from demand planning. Stock depletion arrives from fulfilment, and revised supplier dates arrive from the supplier exception loop. Out go purchase requisitions to procurement, available stock to fulfilment, and stock-out signals back to planning. Those signals matter, because an empty shelf records no sales and hides real demand.

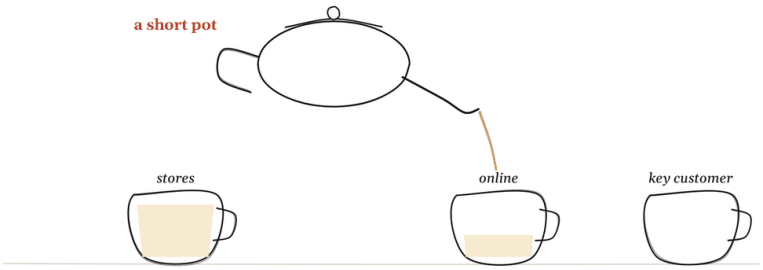
The volume is large. Blue Yonder reported that Morrisons made more than 13 million ordering decisions a day across 491 stores and 26,000 items in 2017. Practice assumes a heavy tail: a small share of items carries most volume and value, while a long tail sells slowly or now and then. No sourced item-level distribution was found, so the shape is an estimate. The body is millions of formula calls; the tail is high-value items and slow movers that no formula predicts well.

Accounting rules force the human points. Under SOX Section 404, stock adjustments and count variances above thresholds need review by someone other than the preparer. IAS 2 requires management to judge write-downs of excess stock. Computing reorder points is mechanical. Overriding a proposed order for a promotion or local event is bounded. Changing safety-stock policy for a class of items, and writing off stock, are consequential. Allocating scarce supply across stores, channels or customers is contested. Each function holds a different priority.

Rules carry the formula. A model adds item-level forecasts and spots records that look wrong. People hold policy changes, write-offs and the allocation of scarce stock, which are the moves that are hard to reverse.

APQC gives a median carrying cost of $19 per $1,000 of revenue. IHL Group put the global cost of out-of-stocks and overstocks at $1.77 trillion in 2023, 7.2% of retail sales. The record itself is weak: Auburn University and GS1 US found 63% inventory accuracy with barcodes against 95% with RFID tags in 2018. No public benchmark exists for replenishment cycle time, cost per order or a recent out-of-stock rate.

As read in 2026, Amazon says its forecasting model positions stock across more than 400 million items and has cut delivery times by almost a day. Walmart's inventory agents detect, diagnose and correct issues, with no figures published. Morrisons cut shelf gaps by up to 30%, a vendor figure from 2017. A loop that stocks more accurately also censors less demand, so the next forecast it receives gets better.



Like the last pot of tea at a busy table: *a routine fills the cups every day, but sharing a short pot between stores, channels and customers is a choice.*

Order fulfilment

At a glance	
Outcome	An order accepted, picked, packed and shipped complete, on time, undamaged and with correct documents, plus accurate confirmation and invoice trigger
Shape of cases	Mixed (estimate): standard orders plus a minority of exception orders
Decisions	1 mechanical · 2 bounded · 2 consequential · 0 contested
Hardest to undo	Releasing an order flagged for export-control, sanctions or hazardous-goods review
Anchor figure	Median of 88% perfect orders (APQC, accessed 2026)

A booked, credit-cleared order arrives from the lead-to-cash chain. Stock that can be promised arrives from replenishment. The loop sends the shipped consignment, with its tracking number, to shipment exception handling. It sends the goods-issue event to billing, the stock depletion to replenishment, and the order history to demand planning.

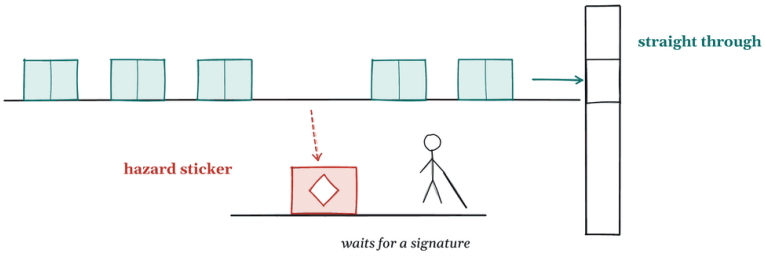
APQC puts the median perfect-order rate at 88%. A perfect order is on time, complete, undamaged and correctly documented. So roughly one order in eight fails somewhere. The likely shape is a large body of near-identical orders and a minority of backorders, substitutions, hazardous goods and export-controlled lines. That minority absorbs most of the handling effort. No published distribution of order value or complexity exists, so this shape is an estimate.

Several laws fix the hard points. The US Export Administration Regulations require an exporter who sees a red flag to inquire into end use and end user before proceeding. The EU Dual-Use Regulation holds an order while the authority decides whether a licence is needed. For hazardous goods, a person must sign the shipper's certification. On consumer delays, the US FTC Mail Order Rule and Article 18 of the EU Consumer Rights Directive limit the seller to consent, delay or refund. Validating an order against master data is mechanical. Choosing the fulfilling site and carrier, and choosing between delay and refund, are bounded. Allocating short stock across competing orders is consequential, since it decides which customer is disappointed.

Written rules can take validation and the legal options on delay. A model adds the choice of site and carrier, and reads orders from email and PDF. People hold the export and hazmat releases, which carry legal liability, and the allocation of short stock to key accounts.

The standard is the perfect order. APQC and Esker reported a median of 84% in 2026, with on-time in-full delivery at 90% and 15 days from order to delivery. Acknowledging an order took a median of one business hour. APQC's open measures put the cost of managing a sales order at $8.48 and pick-to-ship time at 28 hours.

As read in 2026, Walmart runs an ensemble of agents that picks the fulfilling store or warehouse for each order, with people in the loop for disruptions. It reports some delivery windows under 30 minutes. Topa Bathroom Products, a private company in a vendor story, processes over 90% of incoming orders with no manual input. No named public company has published measured results for touchless order entry.



Like a post office counter: *most parcels slide straight through the slot, but the one with a hazard sticker waits on the counter for a signed release.*

Shipment exception handling

At a glance	
Outcome	Each delayed, lost, damaged, misaddressed or customs-held shipment resolved by reroute, reship, refund or claim, with accurate status and remedy for the customer
Shape of cases	Mixed: half self-resolve, a minority is costly and terminal
Decisions	1 mechanical · 2 bounded · 2 consequential · 0 contested
Hardest to undo	Declaring a shipment lost and refunding or replacing it
Anchor figure	6.2% of shipments with an exception by volume (AfterShip, 2026, vendor data)

This loop catches what goes wrong after the goods leave. The shipped consignment arrives from fulfilment. Exception events arrive from carriers and tracking platforms: delays, failed deliveries, returns to sender, customs holds. The loop sends reships back to fulfilment and refunds to returns. It sends customer notices to support, which cuts "where is my order" contacts, and carrier claims to supplier invoice approval.

AfterShip's data on 963 million shipments shows the shape. About half of all exceptions are transient and resolve themselves. About 15% need action before the parcel moves again, and about 12% are terminal, lost or returned. Most alerts are therefore noise to sort, and the effort sits in a costly minority. The body should be filtered fast. The tail needs a person who can spend money.

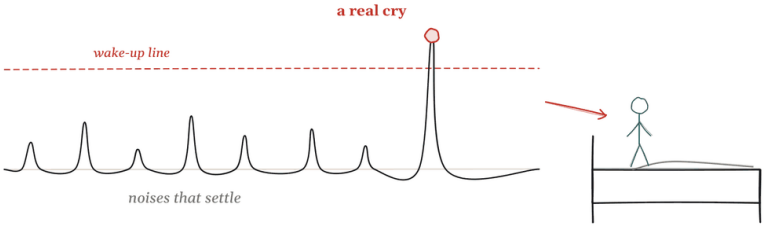
Liability law shapes the tail. The Carmack Amendment makes US carriers liable for loss and damage, but someone must file the claim within the contract period, at least nine months. Under the CMR Convention for international road transport, the receiver must reserve damage at delivery or within seven days. The FTC Mail Order Rule and the EU Consumer Rights Directive force a refund choice on late consumer orders. Certain hazmat incidents require immediate telephone notice. Classifying an event as noise, watch or action is mechanical. Notifying the customer and choosing to reroute or reship are bounded. Declaring a parcel lost and filing a claim are consequential.

Event codes and thresholds can sort most alerts. A model investigates the rest, groups them by cause and drafts the customer message. People approve

refunds and claims, because those commit money and close the carrier's liability window.

AfterShip puts the exception rate at 6.2% of shipments by volume and 7.7% at the median company, with 10 to 15 minutes of manual work each. project44 says one rolled ocean container, bumped to a later sailing, can add seven days or more. It also says expedited trucking afterwards costs 200 to 300% more. These are vendor figures, and no benchmark body publishes a cost per exception.

As read in 2026, Dr. Squatch uses AfterShip's agent, which acts only after human approval. It reports resolution time improved by more than 58% and its most frequent "where is my order" tickets down 25%. Walmart's delivery agents rebalance routes when conditions change, with no exception figures published. The next chapter follows this loop through the whole lifecycle. A loop that warns customers early should shrink the contact volume it was built to absorb.



Like a baby monitor at night: *half the noises settle by themselves, and the parent gets up only for the few cries that cannot wait.*

Supplier delivery exceptions

At a glance	
Outcome	Each unconfirmed, late, short or non-conforming supplier delivery confirmed, re-dated, expedited, substituted or rejected, with an accurate revised supply date
Shape of cases	Heavy-tailed (estimate): a few critical items cause most stoppages
Decisions	1 mechanical · 1 bounded · 2 consequential · 1 contested
Hardest to undo	Rejecting or accepting a non-conforming delivery
Anchor figure	Median of 90% supplier deliveries on time (APQC, accessed 2026)

This loop works the inbound side. Open purchase orders arrive from procurement, along with short or damaged receipts. Confirmations, change requests and shipping notices arrive from suppliers. The loop sends revised dates and shortage alerts to replenishment, and changed delivery promises to fulfilment. Agreed price changes and debit notes go to invoice approval, and the supplier's record goes to supplier evaluation.

With a median of 90% on time, roughly one order line in ten arrives late. Most late lines do little harm. A few critical, single-sourced or long-lead items cause most line stoppages and most of the expediting cost. That is the mark of a heavy tail. Dependence concentrates on a few suppliers, and a late part from one of them stops everything downstream. No published distribution of delay impact exists, so the shape is an estimate.

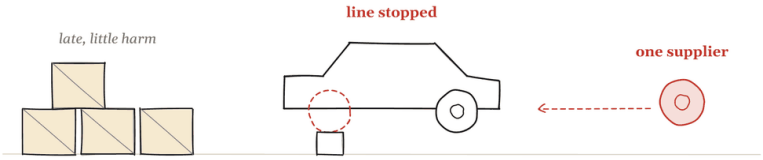
Trade and quality law decide where goods stop. Under the US Uyghur Forced Labor Prevention Act, detained goods are released only on clear and convincing evidence, which a person must assemble. The EU Deforestation Regulation holds covered goods that lack a due diligence statement. In US pharmaceutical manufacturing, the quality unit must approve or reject each lot of components. SOX Section 404 separates authority to change order terms from receiving and payment. Chasing unconfirmed orders is mechanical. Accepting a revised date is bounded by planning tolerances. Expediting or substituting, and accepting or rejecting non-conforming goods, are consequential. Applying penalties to a supplier is contested, because leverage today trades against supply tomorrow.

Queries and templates can chase late orders. A model reads supplier emails and estimates the downstream impact of each change. People hold quality decisions, premium freight and the supplier relationship.

APQC gives the 90% median for supplier on-time delivery. project44's figures for rolled containers apply inbound too: seven days or more of delay and a 200 to 300% premium for expedited trucking, both vendor claims. No public benchmark exists for cost per expedite, time to resolve or the confirmation rate on orders.

As read in 2026, Walmart uses AI to choose ports and routes for imports during disruptions, with no results published. Microsoft and Oracle both offer supplier follow-up agents, but neither has named a customer with results. Public reporting on this loop is thin, and further cases may exist. Suppliers

that see faster, data-backed follow-up may confirm earlier, which would thin the tail of unconfirmed orders.

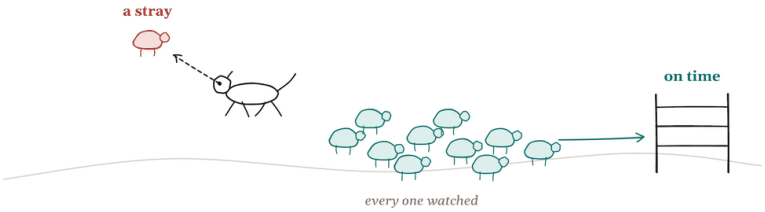


Like a car waiting for its last wheel: *crates of late parts do little harm, but the one part from a single supplier stops the whole line.*

C I

Shipment Exception Handling

At the composite manufacturer, the loop watches about 3,800 outbound shipments a month to distributors, equipment makers, utilities and project sites. An exception is any event that threatens delivery as promised: a delay, damage, a wrong address, a failed delivery or a customs hold. The loop detects each one, decides whether it needs action, acts or escalates, and tells the customer. Its outcome is goods delivered on the promised date or a recovery the customer agreed to.



Like a sheepdog with a flock heading for a gate: *watch every animal, spot the stray, fetch it or call the shepherd, and get them in on time.*

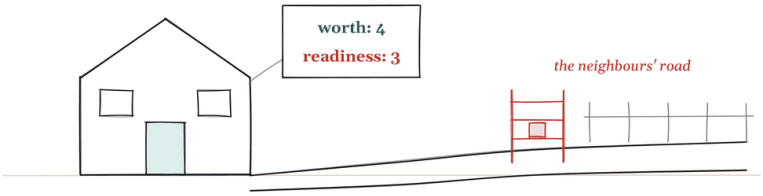
Step 1, Read: the map entry

The chief operating officer's map placed the loop at value four and readiness three. Readiness was lower than value because the loop's data did not belong to the company.

Field	Entry
Loop	Shipment exception handling
Chain	Order to delivery
Value to the core	4: about 300 exceptions a month; on-time delivery and expedited freight directly affected
Readiness	3: 11 of 14 carriers send event feeds; forwarders and brokers mostly email; rules partly written; owner willing
Handoffs	In from order fulfilment and from carrier, forwarder and broker events. Out to support tickets, returns, carrier claims, billing for replacements and freight invoices in procure to pay
Outside data sources	14 road and parcel carriers, two freight forwarders, three customs brokers

Field	Entry
Risk category	Not high risk; carriage conventions set deadlines for damage reservations and claims
Owner	Head of logistics (agreed)
Distribution read	By case: 51% resolved themselves within 48 hours; 24% were delays needing only a notice to the customer; 14% needed action before the shipment could move; 11% were terminal, lost, damaged beyond use or returned to sender. By destination: project sites took 12% of shipments but 31% of exceptions
Signed	Chief operating officer

The split matched published vendor data closely. About half of exceptions are transient, about 15% need action, and about 12% are terminal. Most alerts were therefore noise to sort, not work to do. The tail was the action and terminal cases, plus any shipment to a project site with a fixed installation date.



Like a house valuation card: *the house scores high on worth but lower on readiness, because the only road to it belongs to the neighbours.*

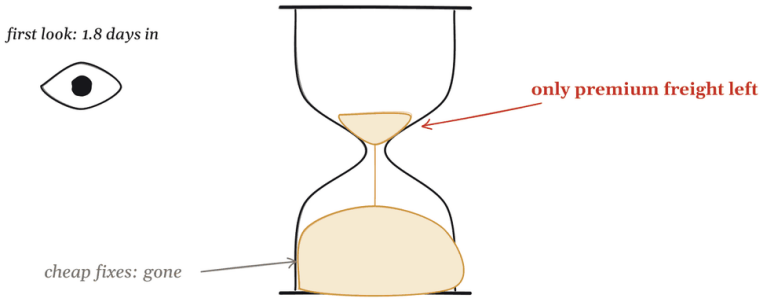
Step 2, Map: the baseline sheet

Mapping with the transport desk drew fourteen steps across seven lanes: carrier, freight forwarder, customs broker, transport desk, warehouse, customer service and sales.

Field	Entry
Sample and period	5,300 shipments and 420 exceptions over six weeks
Volume	3,800 shipments a month; about 300 exceptions, 7.9%
Fully loaded cost per exception resolved	€64, including phone time with carriers and brokers
Time from carrier event to detection	1.8 days average

Field	Entry
Delayed shipments notified before the customer asked	22%
On-time delivery to the promised date	89.5%
Expedited shipments	46 a month; average premium €1,150
Damage and loss claims filed within the contract period	61%
"Where is my order" tickets	640 a month of the 5,200 support tickets
People	6 logistics coordinators on the transport desk
Method agreed with finance	Yes, supply chain controller

Detection was the weak point. A delay found two days late leaves only expensive ways to recover it. Published vendor figures put the premium for expedited trucking after a delay at 200% to 300%.



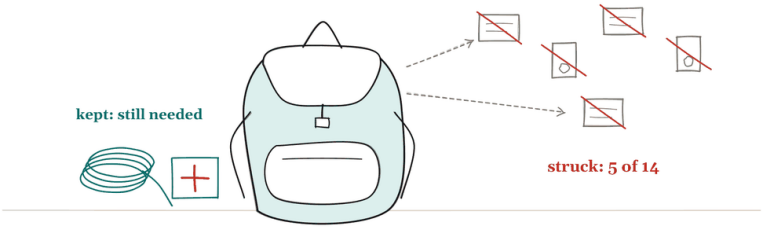
Like glancing at an hourglass already two-thirds run: *delays found 1.8 days late leave only the expensive ways to recover them.*

Step 3, Strip: the strip register

Step	Status	Constraint	Still applies?
Download tracking spreadsheets from carrier portals each morning and re-key them into a shipment log	Struck	Carriers sent no event feeds when the process was designed	No: 11 of 14 carriers now send feeds
Phone the carrier for every flagged event	Struck	Events carried no reason codes	No: feeds now carry reason codes
Copy the account manager on every exception email	Struck	Sales once complained of surprises	No: replaced by notices for project sites only

Step	Status	Constraint	Still applies?
Weekly printed exception report for the head of logistics	Struck	No system access for managers in 2016	No
Re-key customs broker emails into the order record	Struck	Brokers had no shared mailbox	No: the executor can read the broker mailbox
Head of logistics approves any reshipment	Kept	Reshipment commits stock and freight and cannot be undone once dispatched	Yes: becomes a junction
Reserve damage with the carrier within seven days	Kept	CMR Convention, Article 30	Yes

Five of fourteen steps were struck, a strip ratio of 36%. That put the loop between automation and redesign.



Like repacking a rucksack before a trail: *five of fourteen items left from old trips come out, while the rope and first-aid kit stay because they still matter.*

Step 4, Sort: rules, decisions and the contested log

The rule register held 33 rules. Five contradicted each other. The two that mattered most are shown.

Rule	Source	Contradiction	Decision
Notify the customer of any delay over 24 hours	Customer service policy	The transport desk notified only after three days, to avoid false alarms	Head of customer service: notify when the predicted date passes the promised date by more than one working day; project sites at any delay
Expedite at company cost only when the delay is the company's fault	Logistics policy	Sales promised free expediting on any delay	Supply chain controller: company cost up to €800 premium per shipment; above that, a junction

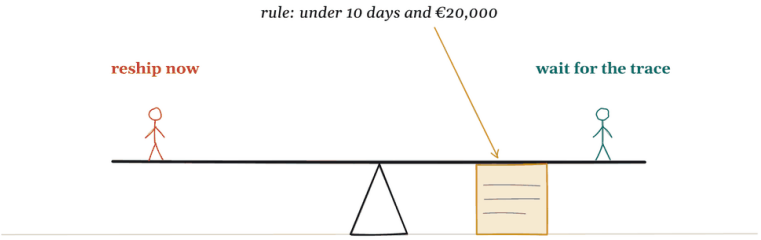
The decision register held twelve decisions.

Decision	Class	Limit or junction
Match a carrier event to its shipment, order and customer	Mechanical	
Classify an event as noise, watch or action by reason code	Mechanical	
Hold a transient alert and recheck before acting	Bounded	Waiting window per reason code, set in shadow
Notify the customer of a delay	Bounded	Approved wording only; new date always stated
Correct a ship-to address	Bounded	Only to an address already on the customer record
Send issued customs documents to a broker on request	Bounded	Existing documents only
Book an expedite to recover a delay	Bounded	Up to €800 premium
Reship a shipment	Consequential	Transport desk lead
Declare a shipment lost	Consequential	Transport claims lead
File a damage or loss claim with the carrier	Consequential	Transport claims lead
Release a customs hold that raises an export control question	Consequential	Trade compliance manager
Reship to a project site before the carrier's trace ends	Contested	See log

A trace is the carrier's search for a missing shipment. It usually takes five working days.

Field	Entry
Decision	Do we reship to a project site before the carrier's trace has ended?
Positions	Sales: reship at once to protect the installation date. Logistics and finance: wait, to avoid double shipments and to protect the claim
Logged	Week 1 of Sort
Settled by	Chief operating officer and chief commercial officer
Answer	Reship before the trace ends only when installation is within ten working days and the shipment is worth under €20,000. Otherwise wait for the trace

Field	Entry
Date settled	Week 4 of Sort
Resulting class	Consequential in every case; the settlement fixes the evidence the junction holder sees

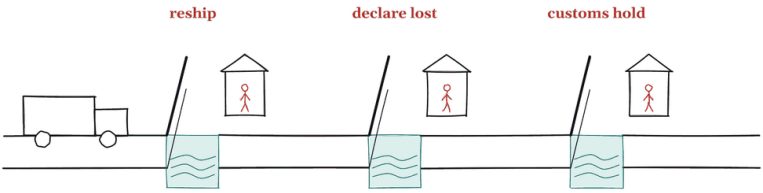


Like a seesaw held by a wedge: *sales pushed to reship now, logistics to wait for the trace, and a written rule fixes which side wins when.*

Step 5, Design: junctions and the standard

Junction	Who decides	Evidence shown	How reversed
Reship	Transport desk lead	Tracking history, trace status, stock at the plant, installation date, shipment value	Cancel before goods leave the plant; after dispatch, only by a return order
Declare lost	Transport claims lead	Last scan, trace result, carrier confirmation, value	If found, the shipment is reinstated and the replacement is recovered through returns
Expedite over €800 premium	Transport desk lead	Delay cause, quote, promised date, customer tier	Not booked until approved
Customs hold with an export control question	Trade compliance manager	Broker request, product classification, end user	Shipment stays held until decided
Delay notice to a project site	Account manager	Draft notice, new date, installation date	Not sent until approved
Switch-off	Head of logistics	Grading dashboard	All exceptions return to the transport desk within 30 minutes

Field	Entry
Criteria	Every event classified within two hours of arrival; customer notified within four working hours when the predicted date passes the promised date; notices use approved wording and state the new date; damage reserved within seven days; claims filed within the contract period
Rule checks	Every notice checked against the approved templates; every claim deadline checked daily
Model grader	Every classification, action and notice graded against the criteria
Human audit	5% at random; 20% for project sites
Customer signals	"Where is my order" tickets; complaints about notices
Calibration	Monthly, by the logistics quality analyst

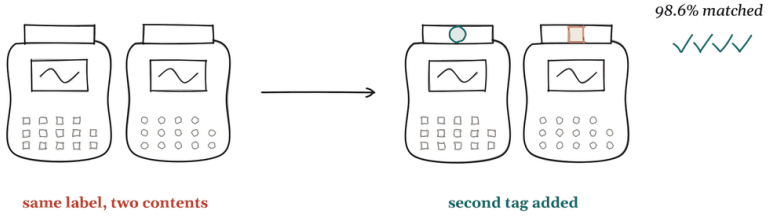


Like drawbridges on a river road: *each costly crossing opens only when its named keeper has seen the evidence, and each has a known way back.*

Step 6, Shadow: the report

Field	Entry
Period and cases	Six weeks; 5,300 shipments; 420 exceptions
Agreement, body	98.6%
Agreement, tail	84%: action cases, terminal cases and project sites
Disagreements	Executor wrong 47%: one forwarder used the same status code for a customs hold and for missing pallets; broker emails in free text. Person wrong 34%: coordinators missed the seven-day damage window and notified late. Both defensible 13%. Standard unclear 6%
Changes made	Forwarder codes split by a second field the forwarder agreed to send; broker emails routed through a fixed form; waiting windows tuned for four reason codes
	Three small carriers with no event feed

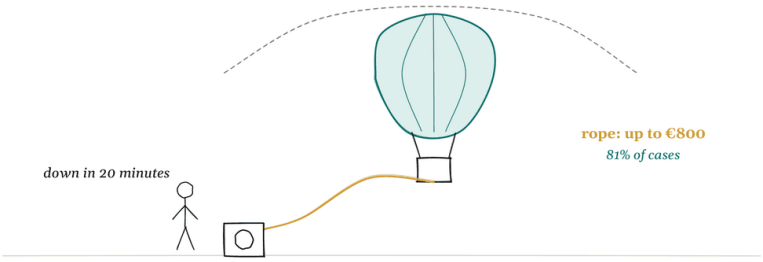
Field	Entry
Known weaknesses	
Recommendation	Pass G1



Like two cooks comparing dishes: *nearly all matched, and most misses traced to two jars sharing one label, fixed by adding a second tag.*

Step 7, Bound: the gate review record

Field	Entry
Scope alone	Classification and waiting for all exceptions; delay notices outside project sites; address corrections from the customer record; sending issued customs documents; expedites up to €800 premium. 81% of exceptions and 46% of exception value
Evidence	Four weeks bounded: 99.0% of bounded actions met the standard; no notice sent outside approved wording
Permissions	Read carrier, forwarder and broker data; update the predicted date on the order; request traces; send templated notices; book expedites up to €800. No permission to create replacement orders, declare loss or file claims
Outside data	Data clauses signed with 11 carriers, both forwarders and two of three brokers: events within two hours, reason codes, 30 days' notice of any format change. Shipments with the other parties stay with people
Switch-off test	Week 1: work returned to the transport desk in 20 minutes; nothing lost
Audit consulted	Internal audit: keep every customer notice with the order record
Decision	Pass G2. Signed by the supply chain controller



Like a tethered hot-air balloon: *it rises freely as far as its rope allows, and the crew showed it can be winched down in twenty minutes.*

Step 8, Own: charter and economics

Field	Entry
Loop owner	Head of logistics
Junction holders	Transport desk lead; transport claims lead; trade compliance manager; 22 account managers for project sites
Exception owner	Transport desk lead, action cases picked up within two working hours
Context keepers	Carriers, forwarders and brokers under the data clauses; the master data team for customer addresses; the head of procurement for carrier contracts
Grader	Logistics quality analyst
Adoption owner	Customer service operations manager
Control owner	Supply chain controller
Changes to work	Six coordinators to three on action cases and project sites. One moved to carrier claims, long under-worked; one to the supplier delivery exceptions loop; one to onboarding carrier data feeds
Learning path	New coordinators work the action queue for two months and sit in on reship and loss junctions

Field	Entry
Period	Eight weeks, including a peak shipping month
Cost lines	Platform and model; grading and audit; junction time; action queue; carrier data onboarding; adoption
	€22 overall; body €8; tail €64

Field	Entry
Cost per accepted exception	
Baseline	€64
Detection	1.8 days to 3 hours
Notified before the customer asked	22% to 91%
On-time delivery	89.5% to 93.1%
Expedited shipments	46 a month at €1,150 average premium to 29 at €980
Claims filed in time	61% to 97%; recoveries from carriers €96,000 to €178,000 a year
"Where is my order" tickets	640 to 470 a month
Annualised effect	About €151,000 in handling cost; €294,000 in expediting premiums avoided; €82,000 in extra carrier recoveries. About €527,000 in all
Signed	Supply chain controller

Handling cost was the smallest of the three gains. Earlier detection let the desk recover delays by normal rerouting rather than by premium freight. It also lifted on-time delivery by 3.6 points, which no cost line captures.

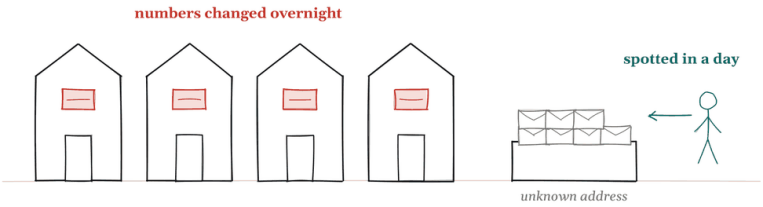


Like weighing a harvest: *the sack you came for, handling cost, is lighter than the freight premiums avoided, and the on-time deliveries never fit a sack.*

Step 9, Core: an operating review

Field	Entry
Month	Second month in Core
Standard met	Body 99.5%; tail 92%
	€21

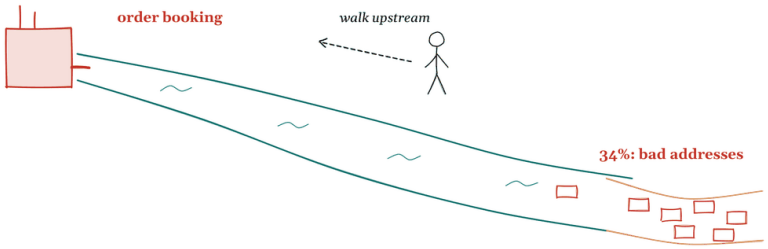
Field	Entry
Cost per accepted exception	
Exception queue	Oldest case 6 hours. Repeated exception: project sites closed at weekends refusing Friday deliveries; rule added to check site hours
Junctions	Transport desk lead declined one reship in six to wait for the trace; four of those shipments were found
Drift	A carrier moved to a new feed format without the agreed notice. Reason codes arrived blank and classifications slid towards watch. The grader caught the drop in a day; that carrier's events went to people for two days. The head of procurement raised the breach at the carrier review
Customer signals	A utility complained that a notice called a nine-day customs hold a minor delay; templates changed to state the new date without adjectives
Decisions	Widen bounded scope to notices for project sites with no installation date, agreed by the supply chain controller



Like a street renumbered overnight: *letters pile up as unknown address, and the postman's next-day count spots the pile before the round goes wrong.*

Step 10, Scale: what the edges showed

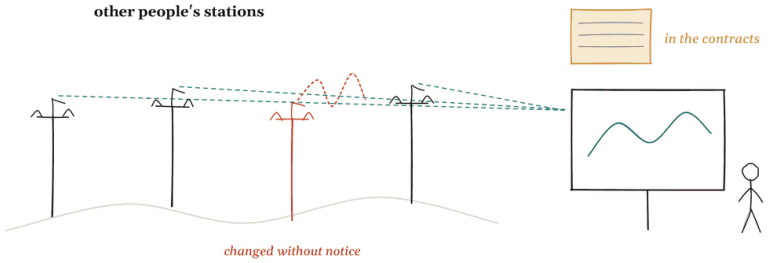
Reading the action queue by cause showed that 34% of action cases were wrong or incomplete delivery addresses. Those errors began upstream, in order booking and order fulfilment. Damage cases clustered at one plant's packing line. At the other edge, carrier surcharges for failed deliveries reached the supplier invoice loop in procure to pay. That loop now receives the exception reference with each surcharge, so it can match or dispute it. The map review chose order fulfilment as the next loop in the chain.



Like litter on a beach traced up the river: *a third of the action cases were bad addresses from order booking, so that loop is fixed next.*

What this loop teaches

A loop's context can belong to other companies. Here the executor's view of the world was only as good as fourteen carriers' feeds, so the data clauses in their contracts were part of the design. Drift came from outside too, when a partner changed its format. Most alerts needed nothing, and the skill was in waiting well. The costly actions, reshipping and declaring a loss, stayed with people because they cannot be taken back. And the largest effects were freight premiums avoided and deliveries kept on time, which a handling-cost sheet would have missed.

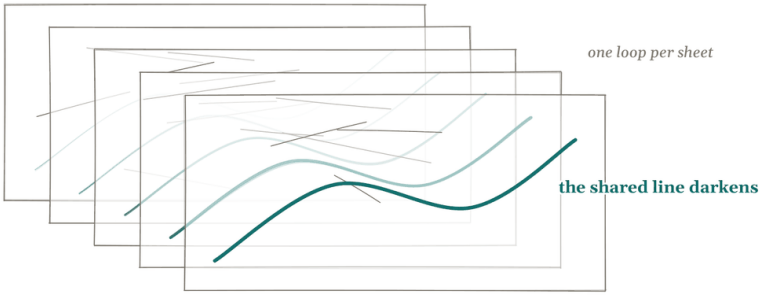


Like a forecaster relying on other people's weather stations: *the forecast is only as good as their readings, so how they report is written into contracts.*

CII

What Thirty-Nine Loops Have in Common

A single entry helps a company start one loop. Read together, the thirty-nine entries say something about embedding AI that no single loop could. This chapter reads across them.



Like tracing-paper sheets laid one on another: *where many loops share a line, it darkens into a pattern no single sheet could show.*

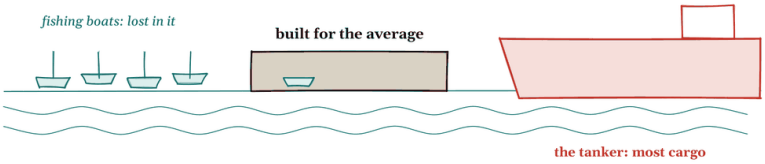
Most loops are not normal

Of the thirty-nine loops, twenty have a heavy-tailed distribution of cases, where a small share of cases carries most of the value, risk or effort. Eighteen are mixed: a large routine body sits beside a tail driven by a few customers, suppliers, defects or legal categories. Only one loop has cases spread evenly enough for its average case to describe it.

That finding shapes everything downstream. In a heavy-tailed loop, the average case does not exist, so a design built for the average serves no one. It also explains the pattern of results in the deployments. The reported gains come almost entirely from the body: the cash applied with no touch, the routine tickets resolved, the invoices that matched first time. The tail stays with people, and it should.

The mechanisms repeat across chains. A few large customers make the tail in collections and contract review. A few large suppliers make it in invoices and onboarding. A few defects or outages make it in warranty and incident management. Legal categories, such as special-category personal data in

hiring, make a tail of risk rather than value. Knowing the mechanism tells a company where its own tail will be before it pulls a single case.



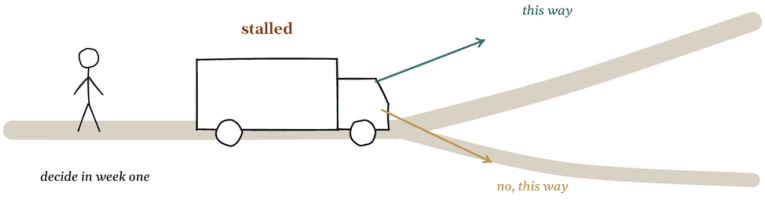
Like a harbour built for the average ship: *too big for the many fishing boats, too small for the one tanker that carries most of the cargo.*

Contested decisions are the rule, not the exception

The entries sort 186 decisions across the thirty-nine loops into the four classes. Forty-nine are mechanical, fifty-three bounded, fifty-six consequential and twenty-eight contested. Twenty-seven of the thirty-nine loops contain at least one contested decision: a question the organisation has never agreed on, which no rule can settle.

The contested decisions are remarkably alike across companies. Whether to extend credit to a strategic but risky customer. Whether to pay suppliers early or stretch them when cash is tight. Whether a late-found error is booked now or next period. Which customers get scarce stock. Whether a failure pattern is escalated to a safety recall. Each is a decision where two senior people would answer differently, and both would cite policy.

This is where most programmes stall without knowing why. An executor cannot be given a decision the company has not made. The library's most practical use may be this: before Sort begins, read the contested decisions listed in the entry, and put them in front of someone with authority in the first week rather than the fourth month.

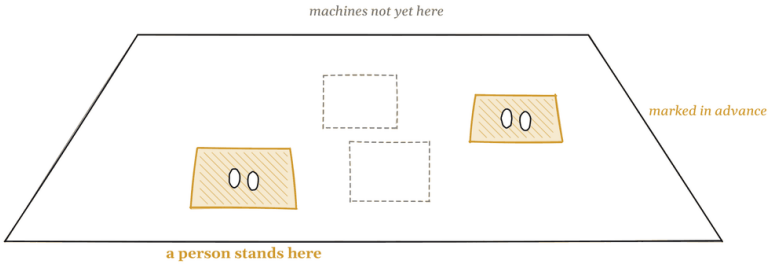


Like a removal van stopped at a fork while two passengers point different ways, both citing the map: *nothing moves until someone with authority picks the road.*

The law puts people in predictable places

The rules that force design choices cluster. Financial reporting controls, such as the Sarbanes-Oxley Act in the United States, run through record to report and parts of lead to cash and procure to pay. Rules on automated decisions about people, such as Article 22 of the European Union's data protection regulation, run through credit checks, hiring and payroll. The EU AI Act treats AI used to recruit and manage workers as high risk. Works councils in some countries must agree to systems that monitor employees.

Read across the chains, these laws mark the junctions in advance. A company does not need to discover in Shadow that a person must review a credit refusal or a rejected candidate. The entry tells it at Read. Dates and thresholds in these laws change, so each entry marks them as read in 2026, and a company should check them at publication of its own design.



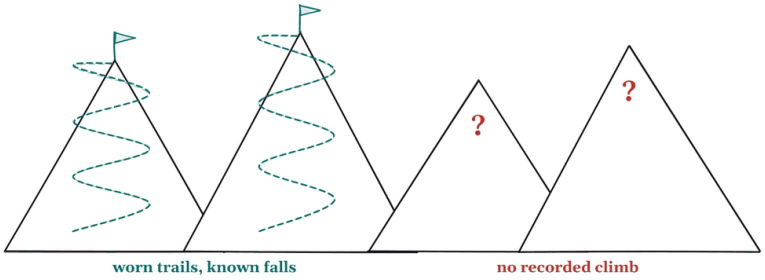
Like safety boxes painted on a workshop floor before the machines arrive: *the law has already marked where a person must stand and check.*

The evidence is uneven

The library cites 189 published benchmarks and 74 named deployments. Fifty-seven of the deployments are reported in production, twelve as partial and five as pilots. Most results are reported by the deploying company or its vendor, and few have been audited.

The evidence is also concentrated. Cash application, invoice processing, customer support and lead qualification have many public cases. Ten loops have no named public deployment at all, including the payment run, intercompany settlement, statutory reporting and offboarding. That gap is not proof that these loops cannot be embedded. It usually means the loop is high in consequence and low in glamour, so companies move carefully and say little.

A company choosing its first loop can use this unevenness. A loop with many public cases offers benchmarks and known failure modes. A loop with none offers an advantage to whoever learns it first, at a higher cost of learning.



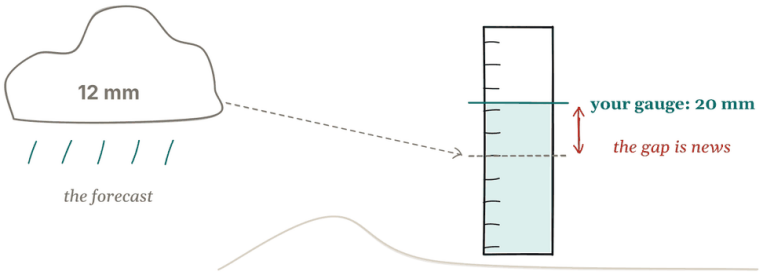
Like a mountain range where some peaks have worn trails and guidebooks and others no recorded climb: *safer to follow, or yours to learn first.*

How to choose and adapt an entry

Start from your own map, not from the library. List your loops by outcome, then find each one's nearest entry. Most companies recognise twenty to thirty of the thirty-nine at once, with local names.

Then read the entry as a set of predictions to test against your own cases. Its shape of cases predicts what your distribution will look like. Its decisions predict what your Sort will find. Its laws predict where your junctions will be. Its benchmarks predict where your baseline will sit. Where your loop differs, the difference is information. A loop whose tail is larger than the entry predicts usually has a mechanism the entry did not name, such as one dominant customer or a legacy contract.

Finally, keep the entry current. A running loop changes the distribution it reads. Once the body runs alone, suppliers and customers adapt, and the tail changes shape. The entry you started from describes the loop before you embedded it. The library is a starting point for the company's own record set, not a substitute for it.

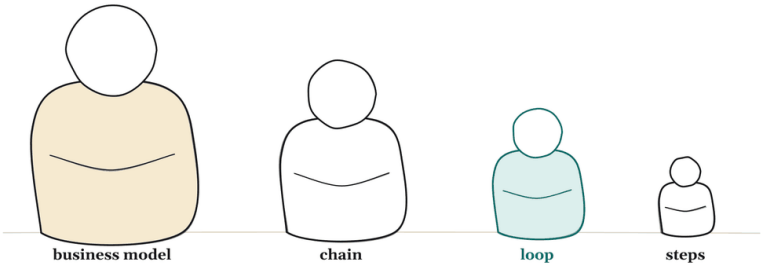


Like a regional forecast checked against your own rain gauge: *the entry predicts, your cases test it, and where they differ you learn about your valley.*

The shape of the library

Taken together, the entries describe a company's work at one useful level: the loop. Above it sits the chain, where loops hand work to each other and where one loop's tail becomes the next loop's exception. Above that sits the business model, which says what the company sells and to whom. Below the loop sit the steps and decisions that the lifecycle maps, strips and sorts.

Christopher Alexander insisted that every pattern is completed by smaller ones and helps complete larger ones. The same is true here. An embedded loop is complete only when its handoffs work, and a chain is embedded only when its loops are. The library is arranged by chain for that reason. **Start with one loop, but read the chain it belongs to.**



Like nesting dolls: *the steps sit inside the loop, the loop inside its chain, the chain inside the business model, and each is whole only with the others.*

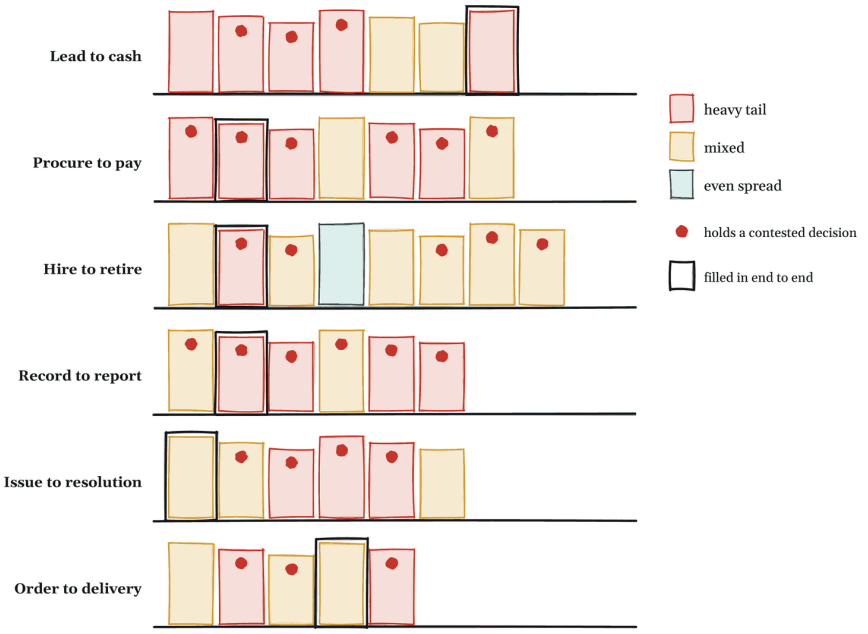


FIGURE 80 The library on one shelf. Thirty-nine loops in six chains, coloured by the shape of their cases; a dot marks a loop with a contested decision, and a frame marks the six filled in end to end.

PART ELEVEN

The Mental Models Library

CIII

How the Models Library Works

A mental model is a compression with a name. It takes a pattern that recurs, states it in a sentence, and lets a practitioner recognise it the next time it appears. Earlier parts of this book introduced many such models where they first mattered. This part gathers them into a library a practitioner can consult.

The library is organised by the type of loop in which each model matters most, not by discipline of origin. A practitioner rarely asks which models come from statistics. They ask what to watch for in the loop in front of them. A loop where a few customers carry most of the money needs different models from a loop that screens job candidates, and both need different models from a loop that runs through outside carriers.



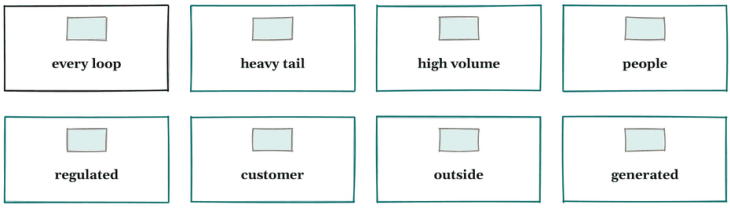
Like a field guide arranged by habitat rather than by family: *you look up what lives where you are standing, not where the species came from.*

Eight types of loop

The first chapter holds the models that apply to every loop, whatever its shape: the body and the tail, the contested decision, the junction, requisite variety, the good regulator and the others that make up the core of the discipline.

Seven chapters follow, one for each type of loop. Heavy-tailed loops are those where a few cases carry most of the value or risk. High-volume routine loops are dominated by a large body of similar cases. Loops that decide about people affect individuals′ work, pay or credit. Regulated and audited loops are read line by line by auditors or regulators. Customer-facing loops are judged by the customer. Loops that depend on outside parties run through suppliers, carriers or banks. Loops whose steps are generated let an executor find its own path to an outcome.

The last chapter covers loops in a chain, because every loop hands work to another, and a loop's tail is often the next loop's exception.



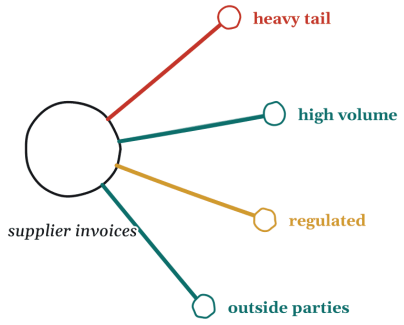
Like a hospital's wards: *the same body of medicine, but each ward keeps the instruments its patients need most close at hand.*

How to use the library

Most loops belong to more than one type. A supplier invoice loop is heavy-tailed, high in volume, audited and dependent on suppliers at once. Appendix O places each of the thirty-nine reference loops of Part Ten by type, so a reader can find the nearest reference loop and read the chapters its marks point to.

Each entry follows the same order. It states the principle in plain words, says where the model comes from, shows how it appears in that type of loop, and says how to use it at a step of the lifecycle or in an instrument. It ends with the common misreading, because a model misapplied does more harm than a model never learned.

No single model is enough. As Charlie Munger argued, judgement comes from a latticework of models from many disciplines, each covering the others' blind spots. A practitioner who reads a loop through one model sees only what that model shows.



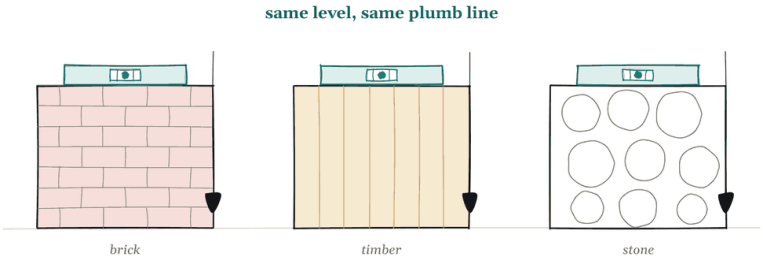
Like a key ring for a building: *one loop opens with several keys, because it belongs to several types at once.*

CIV

Models for Every Loop

Some models belong to no single kind of work. They hold for a supplier invoice and a support ticket, for a candidate and a cash receipt. They describe what every loop is made of rather than what it is about. Every loop has an outcome, a spread of cases, rules and decisions, people placed at its edges, a cost, and some way of knowing whether it worked.

That is how to recognise this territory in a company: wherever those parts exist, these models apply. All 39 reference loops qualify, from Lead qualification and Supplier invoice approval to Candidate screening, Period close, Warranty claims and Shipment exception handling. These models matter most because the specialist models in later chapters rest on them. A team that reads its distribution, sorts its decisions and closes its loop has firm ground. A team that skips them will misapply everything built on top.



Like a builder's spirit level and plumb line: *the same two tools check every wall, whatever the house is made of, as these models check every loop.*

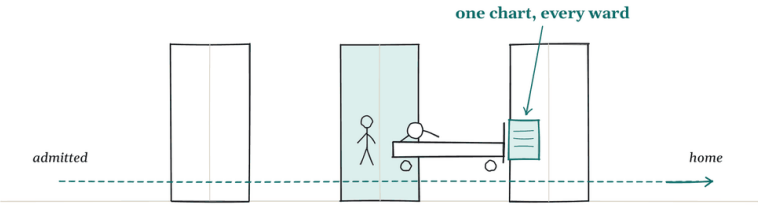
The Work Loop

Work is easiest to change when it is cut at the right size. Too small, and you automate a task while the waits around it stay. Too large, and nobody can own it. The right unit is one outcome, end to end, with the rules, decisions, people and costs that produce it.

Its ancestor is process reengineering. Michael Hammer and James Champy, in Reengineering the Corporation (1993), defined a business process as a set of activities that together create an output of value to a customer. The work loop itself is native to embedding practice. It lives in the Work Loop Canvas and on the Work Loop Map.

In Supplier invoice approval, the outcome is an invoice matched, coded, approved and posted as a liability due for payment. Work arrives from Purchase order issue and Goods and services receipt and leaves for the Payment run. No single team owns it all.

Use it at Read and Map. State the outcome so an outsider could accept or reject a result, then draw the loop's edges where work arrives and leaves. The common misreading is that a loop is one team's process, when that process is usually only a fragment of one.



Like a patient's chart clipped to the bed: *it travels from admission to discharge through every ward, so one outcome is followed end to end.*

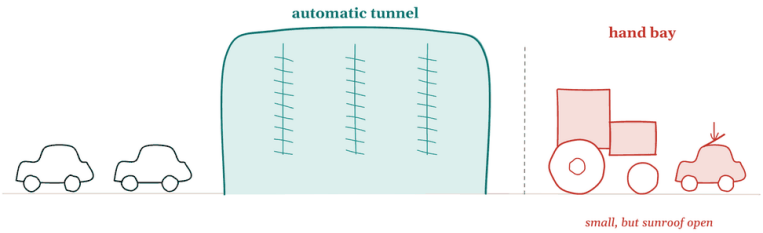
Body and Tail

In most loops, a large share of cases look alike, while a small share carry most of the value, risk or difficulty. The first group is the body and the second is the tail. They need different treatment. An average taken across both hides the difference.

The pattern has a long history. Vilfredo Pareto documented skewed distributions of income in his Cours d'économie politique (1896). Nassim Nicholas Taleb, in The Black Swan (2007), showed how rare, large cases dominate the results in heavy-tailed domains.

The manufacturer's invoices split cleanly. Invoices under €5,000 were 81% of the count and 14% of the spend. The 3% over €50,000 carried 46% of the spend. Shadow running told the same story from the other side. The executor agreed with the accepted result on 97.8% of body invoices and on 81% of tail invoices.

Use it at Read, then carry it through every later step. Report agreement, economics and drift separately for each part, and place junctions mostly in the tail. The common misreading is to draw the line by value alone, when a small first invoice from a new supplier with changed bank details belongs in the tail.



Like a car wash: *most cars roll through the automatic tunnel, while a tractor goes to the hand bay, and so does a small car with its sunroof open.*

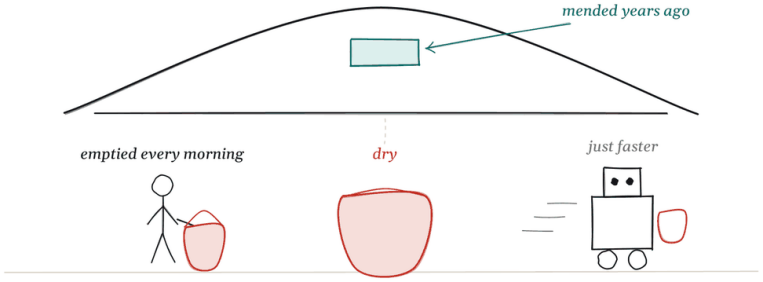
Compensations

Many steps in a process exist only because of a constraint that may have gone. A batch step waits for a system that once ran overnight. A second review survives from a year when errors were frequent. Over time, nobody can tell these workarounds from the real work.

Michael Hammer named the trap in "Reengineering Work: Don't Automate, Obliterate", published in the Harvard Business Review (1990). Automating old processes, he argued, mostly paves the cow paths. In embedding practice, the idea lives in the strip register and its summary measure, the strip ratio.

The manufacturer's invoice loop had a strip ratio of 43%. Almost half its mapped steps were workarounds for constraints that no longer applied. Attaching an executor to the loop as drawn would have run those steps faster. It would have left every wait between them in place.

Use it at Strip. For each step, ask which constraint it serves and whether that constraint still holds. Strike it if the constraint has gone, and record the constraint if not. The common misreading is that a capable executor will route around the workarounds by itself, when it simply runs them faster.



Like a pail still set under a roof leak mended years ago: *someone empties it every morning, and a machine would only empty it faster.*

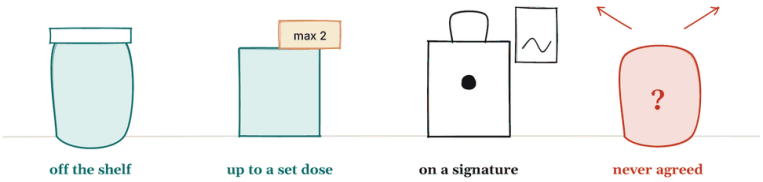
The Four Classes of Decision

Treating every decision alike is the fastest way to get embedding wrong. Some follow a written rule completely. Some need judgement inside a signed limit. Some carry a cost of error so high, or an action so final, that a person must decide. A few have never been agreed at all.

Herbert Simon drew the first line in The New Science of Management Decision (1960), separating programmed decisions from non-programmed ones. The four classes, mechanical, bounded, consequential and contested, are native to embedding practice. They live in the decision register.

The Payment run holds all four. Selecting the invoices due for this run is mechanical. Choosing the payment method within policy on discounts and cash is bounded. Releasing the proposal to the bank is consequential, because cash once sent rarely comes back. Deciding which suppliers to pay early when cash is tight is contested, because treasury and procurement pull different ways.

Use it at Sort, writing each decision as a question with its class and reason. A decision is mechanical only if its rule is written and uncontradicted. The common misreading is to class decisions by comfort, so that everything turns consequential because nobody wants to let go.



Like medicines: *some sold off the shelf, some only up to a set dose, some only on a doctor's signature, and a few the doctors have never agreed on.*

The Contested Decision

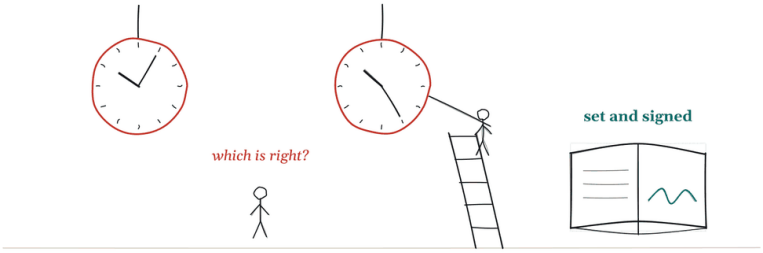
Some questions have no answer inside the company's rules, because the company never chose one. Two senior people would answer them differently, and both would cite policy. Better analysis cannot settle it. Someone with authority has to decide it and write it down.

Kurt Gödel showed in 1931 that any consistent formal system rich enough for arithmetic contains true statements it cannot prove. The lesson carries over by

analogy: some questions can only be settled from outside the system. The instrument, the contested-decision log, is native to embedding practice.

In Period close, an error found late forces a choice: book it now or next period. Controllers and finance chiefs often weigh that materiality call differently. At the manufacturer, the log held one entry, on disputed invoices from single-source suppliers. The chief financial officer and the chief operating officer settled it in thirty minutes.

Use it at Sort by asking, in the room, which decisions two senior people would answer differently. The lifecycle does not pass to Design while any entry stays open. The common misreading is that a capable executor can work out the answer, when it can only be given a rule the company already has.



Like two station clocks that disagree: *a sharper porter cannot say which is right, so the stationmaster sets one and signs the log.*

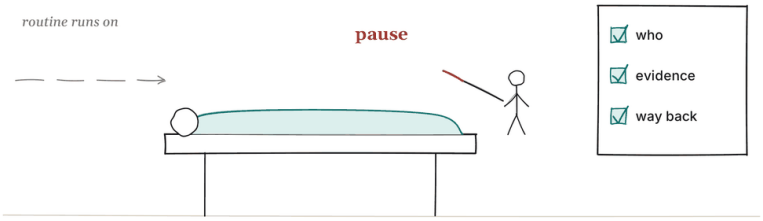
The Junction

An executor can run routine cases on its own, but some decisions must come back to a person. Where they return depends on what an error would cost and whether it can be undone, not on how comfortable anyone feels.

Its ancestor is the cord Toyota workers could pull to stop the line, described by Taiichi Ohno in Toyota Production System (1978). Thomas Sheridan and William Verplank set out graded levels of shared human and machine control in the same year. The junction itself is native to embedding practice and lives in the junction register.

In Candidate screening, law places a junction as well as cost. GDPR Article 22 restricts decisions with legal or similarly significant effects that rest solely on automated processing, as read in 2026. A rejection is such a decision. So a recruiter takes each rejection, and the executor lays the evidence in front of them.

Use it at Design. Each junction names who decides, what evidence they see and how an error is reversed. The common misreading is that any human approval is a junction, when an approval with no designed evidence, no named holder and no way back is a rubber stamp.



Like the pause before the first cut in surgery: *routine work runs on, but at the step that cannot be undone a named person checks the evidence and decides.*

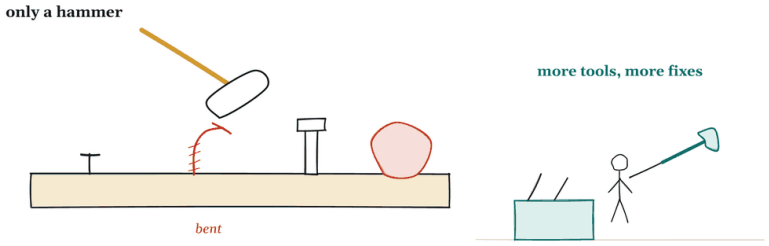
Requisite Variety

A controller can only handle as many kinds of trouble as it has kinds of response. Five rules cannot meet fifty kinds of case. The missing responses must come from somewhere, or the cases they would have met will fail.

W. Ross Ashby stated this as the law of requisite variety in An Introduction to Cybernetics (1956). His own short form was that only variety can destroy variety. In a loop, the variety comes from three places: written rules, a learned model and people.

Warranty claims make the law easy to see. Flat-rate labour tables and part lists handle the body of claims well. They have no response to a pattern that looks like fraud, or to a failure nobody has seen before. A learned model adds variety for messy claim descriptions. People at the edges add judgement for the cases nobody anticipated.

Use it at Design to check that each part of the distribution meets enough variety: rules for the body, the model for messy inputs, people for the tail. The common misreading is that a capable model removes the need for people, when the unforeseen tail still outruns it.



Like a toolbox holding only a hammer: *the nails go in, but screws and bolts need other tools, and the strange fitting needs a craftsman.*

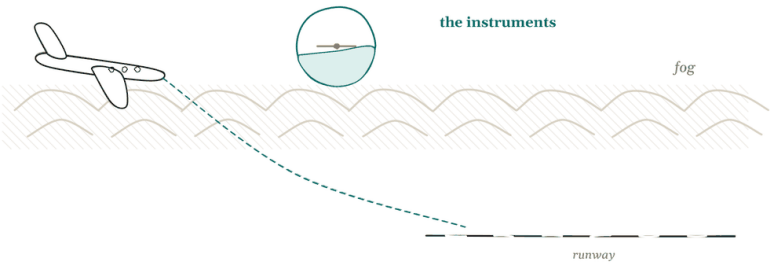
The Good Regulator

You cannot steer well what you do not represent. A controller with no picture of the system it controls succeeds only by luck. The better the picture, the better the control.

Roger Conant and W. Ross Ashby proved this in 1970, in a paper whose title states the result: "Every Good Regulator of a System Must Be a Model of That System". In embedding practice, the model is the loop's record set. It includes the map, the canvas, the registers and the standard.

Inventory replenishment is a regulator in plain sight. The reorder point is computed from a forecast, a lead time and a service level. Suppose the lead time on record is two weeks and suppliers now take five. Every order will then be wrong, however quickly it is placed.

Use it at Core and in the operating review. When outcomes slip, check the model before blaming the executor, and update the registers when the work has moved. The common misreading is that the registers and the standard are paperwork, when they are the regulator itself.



Like a pilot landing in fog: *the instruments are a model of the plane and the ground, and without a true model every landing is luck.*

The Closed Loop

A system that acts without checking its results can only hope. A system that acts, measures the result, compares it with a goal and adjusts can correct itself. The difference grows as the executor takes more of the decisions.

Walter Shewhart applied the cycle to factory quality in Economic Control of Quality of Manufactured Product (1931). Norbert Wiener turned feedback into a general science of control in Cybernetics (1948).

Support ticket resolution shows how a loop can look closed while running open. A ticket marked resolved is not proof of a customer helped. The standard therefore counts reopened tickets and repeat contacts. That way, a fast answer that failed comes back to the grader instead of vanishing.

Use it from Shadow onward. Grade every outcome, review results on a fixed rhythm, and turn findings into changed rules, limits or context. A loop without grading, review or a named owner is open. The common misreading is that a dashboard closes a loop, when nothing closes until someone adjusts.



Like watering a plant: *a timer sprinkler waters even in the rain, while a gardener feels the soil, compares, and waters only when it is dry.*

Cost per Accepted Outcome

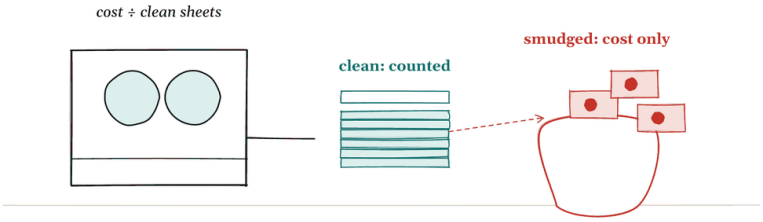
Cheap work that must be redone is not cheap. The honest unit cost divides everything a loop spends by the results that were actually good. Results that fail the standard add to the cost and nothing to the count.

Philip Crosby made the cost of poor quality visible to executives in Quality Is Free (1979). The measure itself is native to embedding practice. It lives on the economics sheet, signed by finance rather than by the programme.

At the manufacturer, the baseline was €11.80 per invoice. After embedding, it was €4.10 per accepted invoice, measured over eight weeks that included a

quarter-end. Split by part, body invoices cost €2.30 each and tail invoices €14.60. The average alone would have hidden that the tail barely moved.

Use it at Own. Take the baseline before the executor starts, and report the figure separately for body and tail. The common misreading is to divide by cases processed, which makes a loop that moves fast and creates rework look cheap.



Like a print shop that prices each clean sheet: *the ink, paper and time are divided only by the sheets that came out right, never by the smudged ones.*

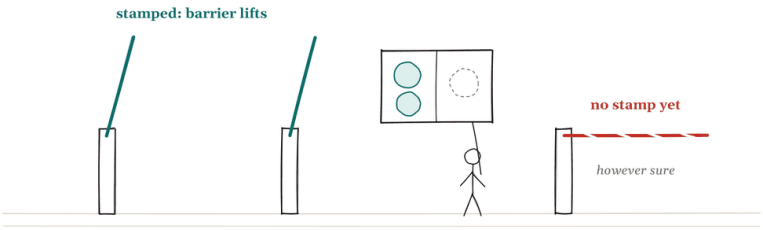
Scale Follows Signatures

Confidence is a poor reason to give a system more scope. A pilot that impressed a steering committee proves little about the cases it never met. A loop should widen only when named people have accepted, on record, the documents that say what it may do.

Robert Cooper's Stage-Gate process, set out in Business Horizons (1990), opened each stage of product development only on a gate decision. In embedding practice, the principle lives in the Scale Gates and the gate review record.

The five gates run from Mapped to Core. At G1, Shadow, the outcome, compensations, rules, context and standard must be signed. At G2, Bounded, the decisions, junctions and exceptions are signed too. For Candidate screening, a high-risk use under Annex III of the EU AI Act as read in 2026, those signatures also become evidence for a regulator.

Use it at every gate, and above all at Scale: no wider scope without the signatures the gate requires. A signature needs a name, a date and a record that outlives the meeting. The common misreading is that signatures slow the work, when they let the next widening happen without reopening every argument.



Like crossing borders: *each barrier lifts for the stamp in the passport, never for how sure the traveller feels, so the journey widens one stamp at a time.*

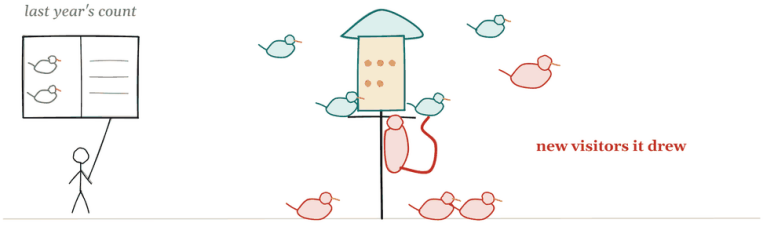
Reflexivity

A loop that acts on the world changes the cases it will meet next. Chase customers sooner and they change how they pay. Answer applicants faster and different people apply. The reading that justified a design goes stale precisely because the design worked.

George Soros called this reflexivity in The Alchemy of Finance (1987): participants′ views change the reality those views describe. Robert K. Merton had described a close cousin, the self-fulfilling prophecy, in The Antioch Review (1948).

Collections and cash application is reflexive by nature. Earlier reminders and better matching change payment habits. The late payers who remain are fewer and harder than before. A body-and-tail split drawn from last year′s ledger no longer fits this year′s cases.

Use it at Scale and in the map review. Re-read the distribution after a loop has run for a while, before widening its scope or choosing the next loop. The common misreading is that a stable result proves a stable world, when the loop may be holding steady on cases it has itself reshaped.



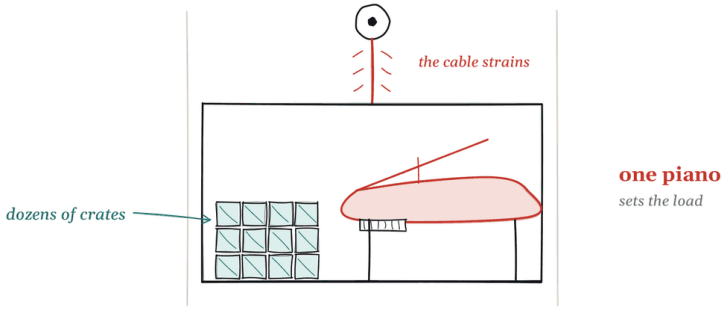
Like a garden bird feeder: *once it is filled, new kinds of birds and a squirrel arrive, so last year′s count of the garden no longer describes it.*

CV

Heavy-Tailed Loops

Some loops look busy everywhere and matter in only a few places. Most of their cases are small and routine, and a handful carry most of the value, the loss or the effort. You can recognise one by sorting a quarter of cases by value and finding that the top few percent hold half the money. Another sign is that the team can name its difficult accounts without looking them up. Typical examples among the reference loops are Supplier invoice approval, Customer credit check, Collections and cash application, Supplier onboarding, Warranty claims and Inventory replenishment.

These loops punish designs that treat every case alike. An executor tuned on the average will do well on the many small cases and badly on the few that decide the result. The models in this chapter help a business engineer see where the weight sits, name more than one kind of tail, and set limits that hold when the large case finally arrives.



Like a goods lift carrying crates and one grand piano: *the many crates are routine, but the piano alone decides whether the cable holds.*

Power Laws and Extremistan

In some kinds of work, one case can be larger than thousands of others put together. Adding more cases does not smooth the picture.

Vilfredo Pareto observed this shape in the distribution of income in 1896. Benoit Mandelbrot showed in 1963 that cotton prices moved with the same wild jumps. Nassim Nicholas Taleb, in The Black Swan (2007), named the two worlds Mediocristan, where no single case matters much, and Extremistan, where one case can matter most.

Supplier invoice approval sits in Extremistan by value. At the manufacturer, 81% of invoices were under €5,000 and carried 14% of spend. The 3% over €50,000 carried 46%. A clerk's day is spent in the body, while the money is in the tail.

Use the model at Read, before anything is designed. Sort a few months of cases by value and compare each band's share of cases with its share of value. If a small band carries a large share, plan the loop as two loops: a body the executor can take and a tail that people must hold. The common misreading is that a power law is rare and exotic, when most money-carrying loops have one.



Like a town of cottages and one tower: *add a hundred more cottages and little changes, but the one tower holds more floor space than all of them.*

The Matthew Effect

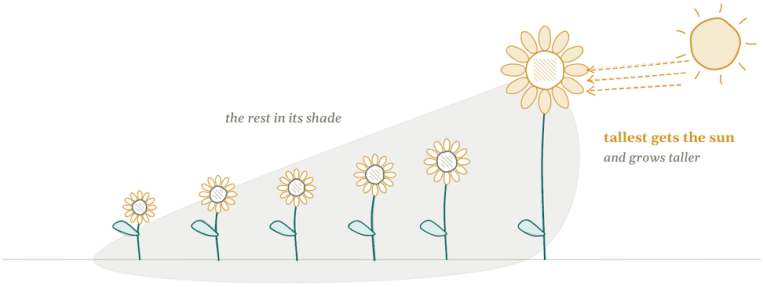
Advantage tends to feed on itself. The customer who buys more gets better terms and so buys more again. Over time, the gap between the largest cases and the rest widens rather than closes.

The sociologist Robert K. Merton named this the Matthew effect in a 1968 paper in Science, after the verse about those who have being given more. He showed how already famous scientists received more credit for the same work. Albert-László Barabási and Réka Albert showed in 1999 that this preferential attachment produces power laws in networks.

In Collections and cash application, a few large customers become more important each year. Their balances grow, their remittances become more complex, and their disputes carry more value. The same happens with strategic suppliers in Supplier onboarding. Each new contract deepens a relationship that was already the largest.

Use the model when you Map a loop, by asking whether the cases come from relationships that compound. If they do, expect the tail to get heavier over time and plan to re-read the distribution at each review. The common misreading is

that today's concentration is a fixed fact, when it is usually a trend still running.



Like sunflowers in a crowded bed: *the tallest catches the most sun, grows taller still, and leaves the rest a little deeper in its shade each week.*

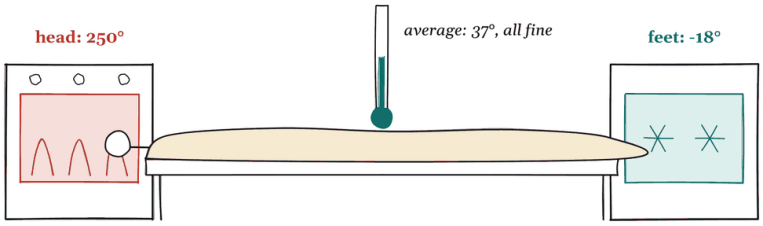
Averages Mislead

A loop can be fine on average and failing where it matters. The average invoice may be small and simple, while the invoices that move the quarter are large and messy. A plan built for the average case fits almost no real case.

Sam Savage called this the flaw of averages in his 2009 book of that name. He showed that plans built on average inputs go wrong when the inputs vary. In heavy-tailed data the error grows worse, because the average is pulled by a few giants and describes neither the body nor the tail.

The manufacturer saw it in shadow running. Overall agreement between the executor and the clerks looked excellent. Split by band, agreement was 97.8% on invoices under €5,000 and 81% on invoices over €50,000. The headline figure had been carried by the many small cases.

Use the model whenever a number goes into the standard or the economics sheet. Report every measure for the body and the tail separately, and refuse a single blended figure for a heavy-tailed loop. The common misreading is that a large sample makes the average safe, when in Extremistan it can make it less representative.



Like lying with your head in an oven and your feet in a freezer: *on average you are at body temperature, and no part of you is comfortable.*

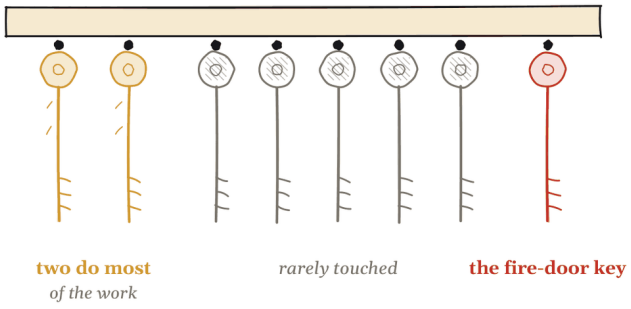
The Pareto Principle and Its Misuse

A small share of causes often produces a large share of effects. A few suppliers account for most spend. A few failure modes cause most returns. Finding those few is often the quickest route to a result.

Joseph Juran turned Pareto's income observation into a management rule in his Quality Control Handbook of 1951. He called it the vital few and the trivial many, and later regretted the word trivial. The figure of 80 and 20 became the popular shorthand, though Juran never claimed it as a law.

In Warranty claims, a few defect families usually drive most of the cost. That makes it tempting to automate the many small claims and leave the rest alone. Yet the trivial many still need handling, and a cheap claim can carry a safety signal.

Use the principle at Sort to choose where attention goes first, and then check the actual split in your own data. The numbers rarely come out at 80 and 20, and they need not add to one hundred. The common misreading is that the vital few deserve all the care and the many deserve none, when the many still need rules, an owner and a standard.



Like a key rack by the door: *two keys do most of the work, yet the dusty fire-door key is the one you need on the worst night.*

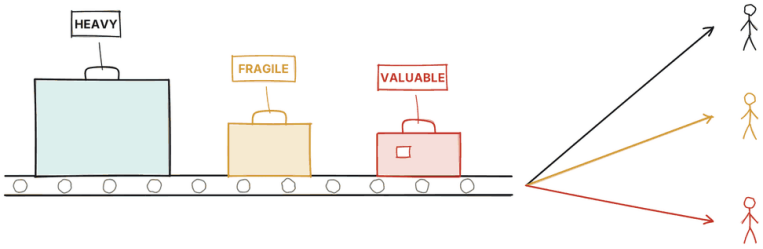
Segmenting by Value, Difficulty and Risk

Cases differ in more than size. A case can be large and easy, small and hard, or small and dangerous. Handling them well means sorting on each of these separately before deciding what goes where.

H. Ford Dickie described ABC analysis at General Electric in 1951, grading inventory by annual value so that the costly items got the closest control. Peter Kraljic added a second axis in Harvard Business Review in 1983, sorting purchases by profit impact and by supply risk. Embedding practice adds difficulty as a third dimension.

In Customer credit check, a large order from a long-standing customer is high in value and low in difficulty. A small first order from a new company is low in value and high in risk. A single value threshold would wave the second through. Inventory replenishment shows the same split between fast movers and rare, critical spares.

Use the three dimensions at Read and carry them into Bound. The executor's limits are then written on several conditions at once, as the manufacturer did with value, tolerance and supplier history. The common misreading is that segmenting by value alone is enough, when risk often hides in the small cases.



Like baggage tags at check-in: *a case can be heavy, fragile or valuable, and each tag sends it down a different chute to a different handler.*

Concentration Risk

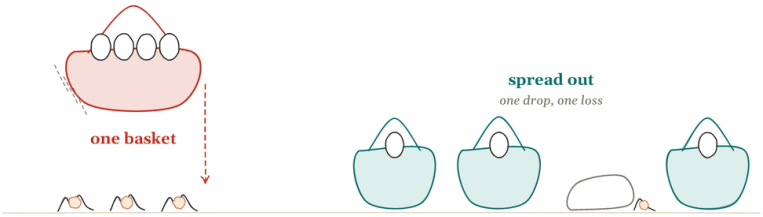
When much depends on a few, the loss of one of them is a large loss. One customer, one supplier or one claim can carry enough exposure to hurt the whole business. That is a fact about the portfolio, not about any single case.

Albert Hirschman measured concentration in foreign trade in 1945, and Orris Herfindahl applied a similar index to the steel industry in 1950. Banking regulators turned the idea into limits. The Basel Committee's large exposures

framework of 2014 caps a bank's exposure to one counterparty, as read in 2026.

At the manufacturer, the contested-decision log held one entry, on disputed invoices from single-source suppliers. Holding payment protected cash. It also risked a plant stopping for want of a part. That dispute existed because one supplier carried a whole production line.

Use the model at Bound and Own. Write portfolio limits into the decision register, not just case limits, so an executor cannot approve many medium cases that add up to one large exposure. Name the owner of each concentration in the charter. The common misreading is that concentration risk is a treasury matter, when operating loops create it every day.



Like carrying every egg in one basket: *one stumble breaks the lot, while eggs spread across several baskets lose only one when a basket drops.*

The Tail Has Many Definitions

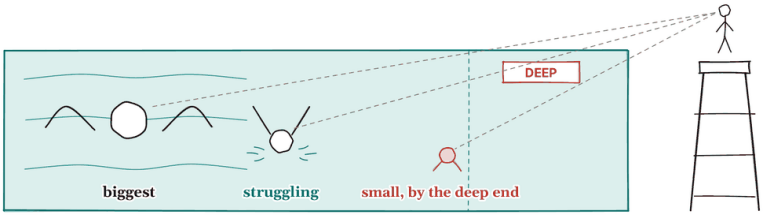
The tail is not one group of cases. The largest invoices are a tail by value, the oddest a tail by difficulty, and those that could hide fraud a tail by risk. They overlap less than people expect.

This model comes from embedding practice and lives in the reading of the distribution at Read. The reading names the body and the tail on each dimension that matters. It also names cases that are new, since a case the loop has never seen belongs to the tail whatever its size.

Supplier invoice approval shows the gap clearly. The large invoice from a supplier paid for years sits in the value tail and the difficulty body. A €900 first invoice with changed bank details sits in the value body and the risk tail. Each needs a different control.

Use the model when writing the junction register. Give each tail its own junction and its own holder, because whoever judges a large commercial dispute is rarely the right judge of a bank-detail change. Cases in two tails at

once need the most attention. The common misreading is that the tail means the big cases, when the most dangerous cases are often small.



Like a lifeguard scanning a pool: *the biggest swimmer, the struggling one and the small child by the deep end each need watching for a different reason.*

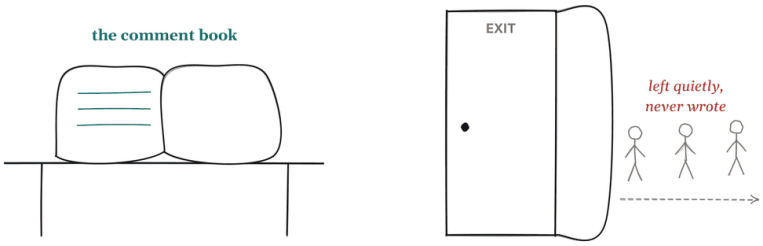
The Unseen Case

The cases that matter most can be the ones missing from the data. A sample built from cases that were easy to find leaves out those that were hard. A loop can look safe because its failures never reached the record.

Abraham Wald made the point in 1943 while studying damage on bombers returning from missions. The holes showed where a plane could be hit and survive, so the armour belonged where the returning planes had none. Frédéric Bastiat had argued in 1850 that good judgement weighs what is not seen as well as what is.

In Warranty claims, the cases on file are the ones customers chose to submit. Failures customers absorbed, or claims abandoned halfway, never appear. A model trained on filed claims learns little about them. The same applies to fraud in Supplier invoice approval, where the record holds the fraud that was caught.

Use the model at Shadow, by testing on the live stream of cases rather than a prepared set. Ask, for each data source, which cases could never have reached it. Record the gaps in the standard. The common misreading is that a large dataset covers the tail, when collection itself can remove it.



Like a restaurant's comment book: *it holds the views of diners who stayed to write, not of those who left quietly and never came back.*

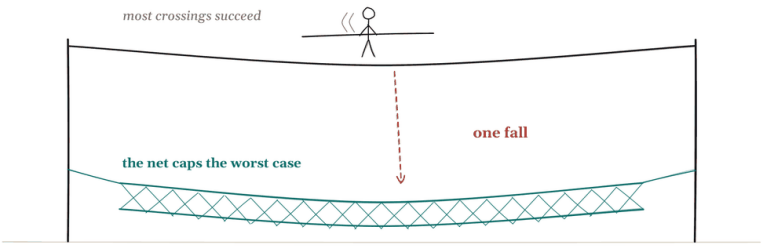
Expected Value and the Worst Case

Two decisions with the same average payoff can carry very different dangers. One loses a little, often. The other rarely loses, but its losses are ruinous. In heavy-tailed loops, the average cannot tell them apart.

Daniel Bernoulli made the point in 1738 with the St Petersburg paradox, a game whose average payoff is infinite but which no sensible person would pay much to play. Philippe Artzner and colleagues set out in 1999 which risk measures behave sensibly. Expected shortfall, the average loss in the worst cases, became a standard alternative to the plain average in banking.

In Customer credit check, approving a slightly risky customer is usually profitable on average. Yet one large default can wipe out a year of margin on small accounts. The average says approve. The worst case says cap the limit and route it to a person.

Use both numbers on the economics sheet. Put the expected cost per accepted outcome beside the largest loss a single error could cause, and set the executor's limit so that worst loss is one the business can absorb. The common misreading is that a positive average makes a decision safe, when ruin can sit inside a good average.



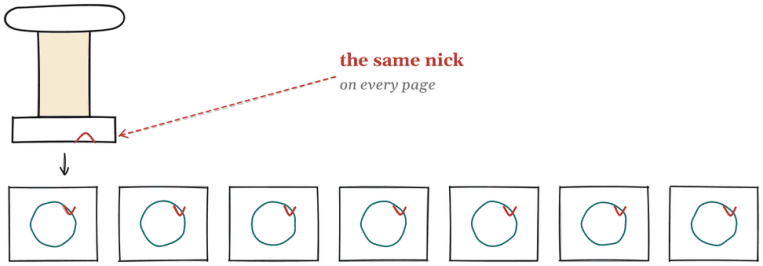
Like a tightrope walk: *most crossings succeed, so the average looks fine, but a single fall is final unless a net caps how far you can drop.*

CVI

High-Volume Routine Loops

Some loops are defined by repetition. Thousands of cases arrive each month, and most of them look alike: an invoice that matches its order, a payment that matches its invoice, a password reset, a standard order from a known customer. You can recognise these loops by their queues, by teams measured on cases per hour, and by a month-end backlog that everyone treats as weather. Typical reference loops are Supplier invoice approval, Collections and cash application, Billing and invoicing, Order booking, Support ticket resolution and Employee expense claims.

In these loops the large body of similar cases is where an executor earns its keep, and where most of the money moves. The danger is not that a case is hard. It is that small errors repeat thousands of times, that waiting hides inside volume, and that the measures meant to steer the loop start to steer it wrong. The models below are about flow, variation and measurement at scale.



Like a rubber stamp with a nick in its face: *every page carries the same small flaw, because in routine work one error repeats thousands of times.*

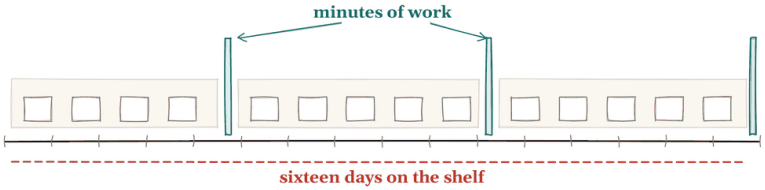
Flow Efficiency

Most of the time a routine case spends in a loop, nobody is working on it. It sits in an inbox, waits for a batch, or waits for someone to come back from a meeting. Divide the minutes of real work by the days from arrival to outcome, and the share is usually tiny.

The measure comes from lean practice. Niklas Modig and Pär Åhlström gave it its current name and its central place in *This Is Lean* (2012). Earlier, George Stalk and Thomas Hout had shown in *Competing Against Time* (1990) how little of elapsed time adds value.

In high-volume loops the waits dominate. At the manufacturer, twenty-three minutes of work on a supplier invoice took sixteen days, a flow efficiency well under 1%. The minutes were spread across scanning, coding, matching and a Friday approval round. The days were spent between them.

Measure flow efficiency at Map, from the walk-through and the baseline, before anyone builds anything. It tells you whether the prize is in the steps or in the waits between them. The common misreading is that a faster executor improves flow, when it only shortens the minutes and leaves the days untouched.



Like a parcel in transit: *handled for minutes and parked on shelves for days, so a faster sorting machine barely changes when it arrives.*

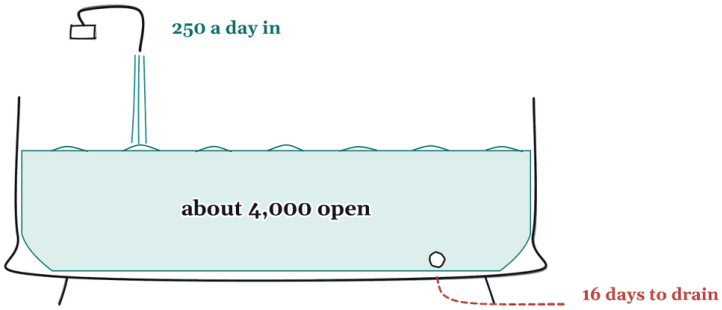
Little's Law

The number of cases sitting in a loop at any moment equals the rate at which they arrive multiplied by the time each one spends inside. If cases take twice as long, twice as many pile up.

John Little proved the relationship for queues in general in "A Proof for the Queuing Formula: L = λW" (1961). It holds for almost any stable system, whatever the order in which cases are served.

It turns a vague backlog into arithmetic. The manufacturer receives about 7,400 invoices a month, roughly 250 a day. At sixteen days each, around 4,000 invoices are open at any moment. Each open invoice is a supplier who may call, a discount that may lapse and a cost that sits unrecorded at month-end. Cut the time to four days and the open stock falls to about a thousand, with no change in volume.

Use it in the economics sheet and the baseline to price waiting, and in Design to size the exception queue that the tail will create. The common misreading is that a backlog proves a team is short of people, when it often proves only that cases wait too long between steps.



Like a bath with the tap running: *how full it gets is the flow in times how long each litre stays, so clear the drain before buying a bigger tub.*

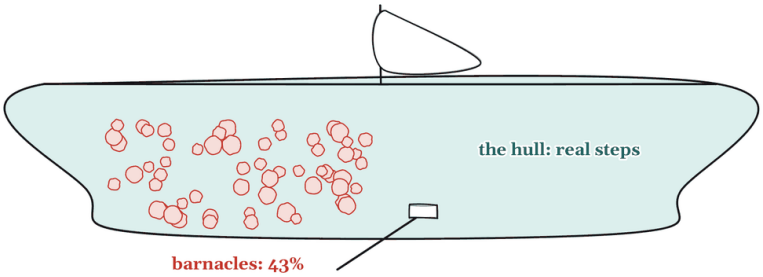
The Strip Ratio

Count the steps on a process map, then count how many exist only to work around something that no longer binds: an overnight batch, a paper signature, a second check added after an old error. The share you can strike tells you what kind of job lies ahead.

This model is native to embedding practice. It lives in the strip register, signed at Step 3, Strip. It builds on the lean idea of separating work that creates value from work that does not, but it asks a narrower question about each step: what constraint does this step compensate for, and does it still apply?

Routine loops gather compensations faster than most, because every past error at volume left a check behind. The manufacturer's invoice loop had a strip ratio of 43%. Printing and scanning PDFs were among the struck steps.

Use the ratio to decide the shape of the work. Under about a fifth, let an executor run what remains. Over about a third, treat the project as a redesign and strip before you build. The common misreading is that a high ratio blames the people, when the compensations were usually sensible answers to constraints that later disappeared.



Like scraping barnacles off a hull: *if a little comes away, sail on; if the crust is two-fifths of the boat, it needs the dry dock.*

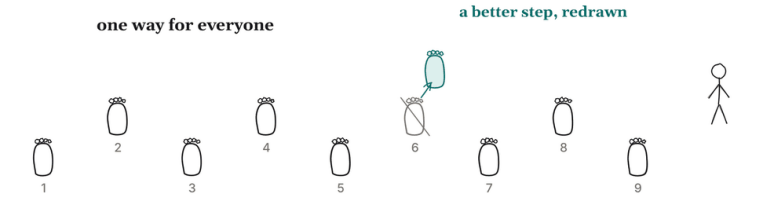
Standard Work

When thousands of similar cases pass through a loop, they should be handled the same way each time. Write down the current best way to handle the routine case, then make sure everyone uses it. Improve it on purpose, not by drift.

The idea comes from the Toyota Production System. Taiichi Ohno described it in *Toyota Production System: Beyond Large-Scale Production* (1978, English edition 1988). Standard work was never meant to freeze a method. It was meant to give improvement a fixed baseline to improve from.

In Order booking, most orders from known customers at contract prices need no human touch. Yet in many companies two clerks still book the same order differently, with different checks and different notes. An executor trained on that history learns both habits and the inconsistency between them.

At Design, the standard for the routine case should describe one accepted outcome and one way of grading it. When an exception reveals a better method, change the standard and the rule register together, through the operating review. The common misreading is that standard work removes judgement, when its real job is to free judgement for the cases that need it.



Like dance steps chalked on the studio floor: *everyone takes the routine case the same way, and when a better step is found, the chalk is redrawn for all.*

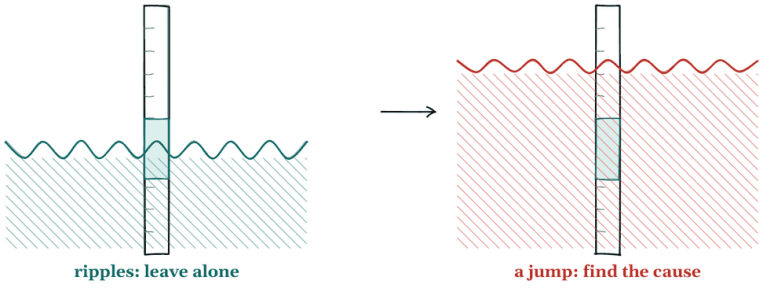
Common and Special Causes

Every high-volume loop wobbles. Some days more cases fail, some days fewer, for reasons nobody can name. That ordinary noise belongs to the system. Occasionally something new happens and moves the numbers for a reason you can find. The two call for opposite responses.

Walter Shewhart drew the distinction in *Economic Control of Quality of Manufactured Product* (1931) and built the control chart to separate them. W. Edwards Deming named them common and special causes and spread the idea in *Out of the Crisis* (1986).

In Collections and cash application, the daily share of unmatched payments moves up and down by a few points with no single cause. Then a large customer changes its remittance format, and unmatched cash jumps and stays high. Chasing the first wobble wastes effort. Ignoring the second leaves cash unapplied for weeks.

Plot the share of outcomes meeting the standard as a control chart in the operating review. React to points outside the limits by finding the cause, and improve the system to narrow the limits themselves. The common misreading is that every bad day deserves an explanation, which leads teams to tamper with a stable loop and make it worse.



Like a river gauge post: *daily ripples inside the band are just the river, but a rise that jumps the band and stays has a cause upstream.*

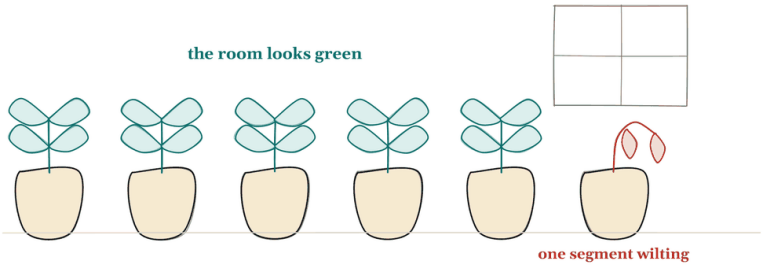
Drift

A loop can get slowly worse without anything breaking. Suppliers change their layouts, customers change how they write, a model is updated, and the people who handle exceptions change. No alarm sounds. The share of good outcomes simply slides.

This is native to embedding practice, and it is watched through the standard and the operating review. It borrows from machine learning research on concept drift, described by Gerhard Widmer and Miroslav Kubat in "Learning in the Presence of Concept Drift and Hidden Contexts" (1996).

Volume makes drift both dangerous and visible. In Billing and invoicing, national e-invoicing mandates change the formats and clearance channels invoices must pass. Germany's requirement to receive B2B e-invoices from 1 January 2025 is one example, as read in 2026. An executor tuned to last year's formats can fail on one segment while the overall figure barely moves.

Catch drift in Core by grading every outcome, checking the grader with a weekly human sample, and reading body and tail separately each month. When drift appears, use the switch-off on the affected segment. The common misreading is that drift is a model problem, when it usually starts in the inputs or in the people around the executor.



Like a row of houseplants where one by the far window slowly wilts: *the room still looks green, and only checking each pot finds the dying one.*

The Ironies of Automation

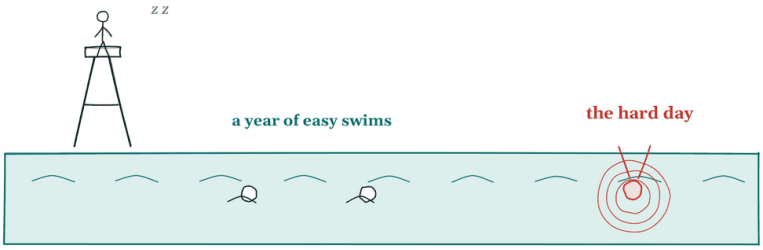
Automate the easy cases, and the people left behind get only the hard ones. They see fewer cases, practise less, and lose the feel for what normal looks like. Then, on the rare day the system fails, they are asked to take over at full volume with skills that have faded.

Lisanne Bainbridge set this out in "Ironies of Automation" (1983), written about process control and aircraft cockpits. Her point was that automation does not remove the human role. It changes it, and often makes it harder.

In Support ticket resolution, an assistant may absorb most routine contacts, as the published deployments in the reference loop suggest. The human agents then handle only escalations. When the assistant is switched off for an

incident, those agents must resolve password resets and billing questions at a pace they have not worked for months.

Plan for this at Own and test it with the switch-off. Rotate people through some routine cases, keep the switch-off drills real, and size the team for the day the executor stops. The common misreading is that the remaining human role is the easy part, when it is the hardest and least practised job in the loop.



Like a lifeguard at a pool where the swims are easy: *a year without a rescue, and the hard one arrives when the skills are least practised.*

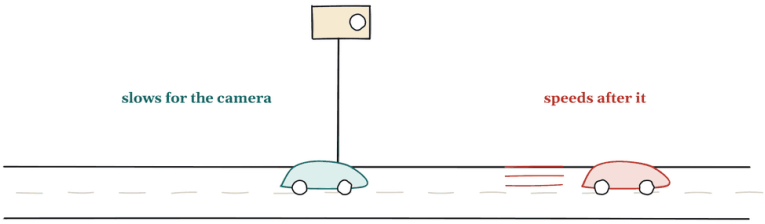
Goodhart's Law

A number that once told you how a loop was doing stops telling you once people are rewarded for moving it. They find ways to move the number that do not move the outcome. In routine work, with thousands of cases and simple counts, those ways are easy to find.

Charles Goodhart made the observation about monetary targets in "Problems of Monetary Management: The U.K. Experience" (1975). Marilyn Strathern gave it its familiar form in 1997: when a measure becomes a target, it ceases to be a good measure.

Support ticket resolution shows it plainly. Reward closure speed, and tickets are closed early and reopened later. Reward first-contact resolution, and hard tickets are passed on rather than solved. An executor optimised against the same count will learn the same tricks, only faster.

Define success in the standard as an accepted outcome, and include the customer's view through reopened cases, repeat contacts and complaints. Then report cost per accepted outcome in the economics sheet, not cost per case processed. The common misreading is that a better metric solves the problem, when any single number will bend once it carries a reward.



Like a speed camera: *drivers brake for the box and speed up after it, so once the reading is the target it stops describing the road.*

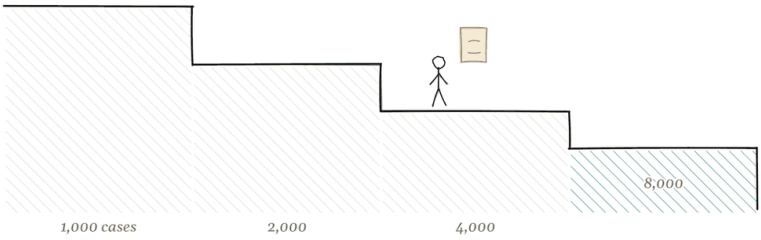
The Learning Curve

The more often a team does the same work, the cheaper each repetition becomes. The gain is steady and predictable. Every time cumulative output doubles, the cost of a unit falls by roughly the same share.

Theodore Wright measured it in aircraft assembly and published it as "Factors Affecting the Cost of Airplanes" (1936). The Boston Consulting Group later generalised it as the experience curve.

High-volume loops climb the curve quickly, because doublings arrive in weeks rather than years. At the manufacturer, each exception the controller resolved became a candidate rule or context fix. The exception queue shrank as volume accumulated. The same curve applies to the lifecycle itself: the first loop was the most expensive, and later loops ran faster and cheaper.

Plot cost per accepted outcome against cumulative cases in the economics sheet, and feed every resolved exception back to the rule register. At Scale, track the falling time and cost per loop as the company's real asset. The common misreading is that the curve comes free with volume, when it comes only from deliberately capturing what each case teaches.



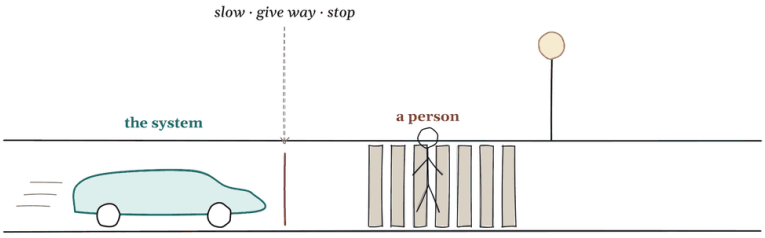
Like walking down a staircase where each step is a doubling of cases: *the cost drops a stair each time, provided every lesson is written down.*

CVII

Loops That Decide About People

Some loops produce an outcome that lands on a named person. A candidate is shortlisted or rejected, a sole trader gets credit or a demand for prepayment, a pay change goes through, leave is granted, an employment ends. You can recognise these loops by asking who would complain if the decision were wrong. If the answer is an individual rather than a supplier's accounts team, the loop belongs here. In the reference set, the typical examples are Candidate screening, Customer credit check, Payroll changes, Leave and absence requests, HR helpdesk and Offboarding.

These loops are where an error costs most per case and where the law already says a good deal about how decisions must be made. The person affected cannot see the machinery and usually cannot walk away. So the models in this chapter are less about speed and more about fairness, evidence and a way back. They tell the business engineer which decisions the law has already classified, what each person is owed, and how a well-meant system can drift into treating groups unequally.



Like a zebra crossing: *wherever a machine's path meets people, it must slow down, give way and be able to stop, because what it hits is a person.*

High Risk by Law

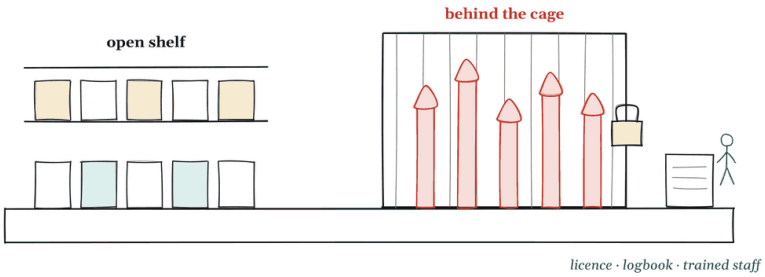
In most loops the company decides how careful to be. In some, the law has already decided. Certain uses of automated systems are named in advance as risky to people's rights, and they carry extra duties however well they work.

The clearest statement is the EU Artificial Intelligence Act, Regulation (EU) 2024/1689, as read in 2026. Its Annex III lists high-risk uses. Point 4 covers recruitment, selection and decisions affecting work relationships. Point 5(b)

covers scoring the creditworthiness of natural persons. Colorado's SB 24-205 takes a similar approach for consequential employment decisions.

The classification follows the use, not the vendor's label. In Candidate screening, a tool that filters applications is high-risk, so the deployer owes trained human oversight, logs and notice to workers. In Customer credit check, credit on companies falls outside point 5(b), but sole traders fall inside it.

Use it at Read and Sort. For each decision in the decision register, record whether a listed use applies and cite the rule in the rule register. The common misreading is that buying a compliant product makes the deployment compliant, when most of the deployer's duties stay with the deployer.



Like a fireworks counter: *the law, not the shopkeeper, lists which goods go behind the cage and need a licence, a logbook and trained staff.*

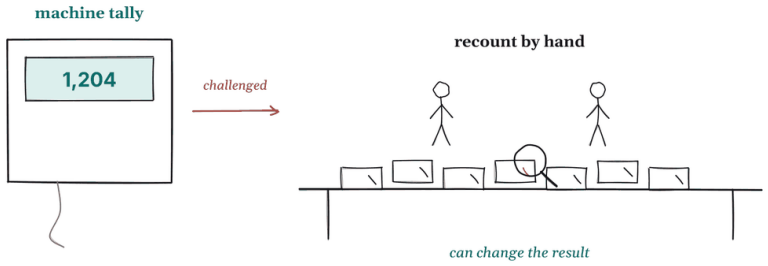
The Right to Human Review

When a decision with real effect on someone is made by a system, that person can ask for a human to look at it again. A person who can only confirm what the system said is not a review.

The right comes from Article 22 of the General Data Protection Regulation, Regulation (EU) 2016/679, as read in 2026. A decision based solely on automated processing, with legal or similarly significant effect, is barred unless an exception applies. Even then the person can obtain human intervention. The Court of Justice widened the reach in SCHUFA (C-634/21, 2023): a score that strongly drives a lender's decision counts as the decision.

In Offboarding, a dismissal sits under Article 22 and, in Germany, under a works-council hearing as well. In Candidate screening, a rejection has legal effect on the applicant. Both must reach a person with authority and the facts to reverse.

Use it at Design and Own. In the decision register, name the reviewing role, its authority and the evidence it sees. The common misreading is that a human somewhere in the loop satisfies the right, when what counts is a human who can actually decide the case.



Like a recount by hand: *when the machine's count is challenged, people with authority go back to the actual papers and can change the result.*

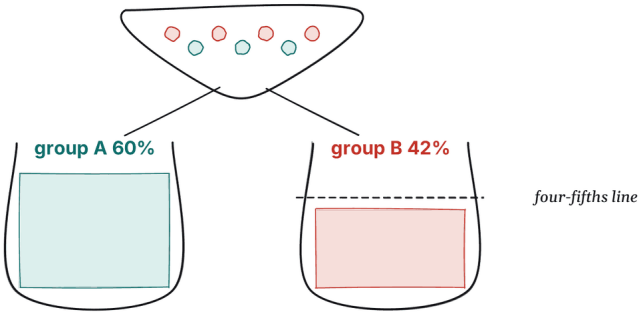
Adverse Impact and the Four-Fifths Rule

A selection step can be neutral in its wording and still pass one group far less often than another. The quickest test is to compare pass rates. Divide the lower group's rate by the higher group's rate and look at the ratio.

The Uniform Guidelines on Employee Selection Procedures of 1978, issued by the US Equal Employment Opportunity Commission and other agencies, made this a working rule. A selection rate for any group below four-fifths of the highest group's rate is generally taken as evidence of adverse impact. The idea rests on Griggs v. Duke Power Co. (1971), where neutral tests with unequal effects were held to need a business justification. New York City's Local Law 144 now requires published impact ratios for automated hiring tools, as read in 2026.

In Candidate screening, suppose a ranking agent advances 60 per cent of one group and 42 per cent of another. The ratio is 0.70, under the line.

Use it at Shadow and keep it in the standard, recomputed whenever the model or criteria change. The common misreading is that passing the ratio proves a tool fair, when it is only a screen that tells you where to look.



Like two jars filled by the same sorting machine: *if one group's jar fills to under four-fifths of the other's level, look hard at the machine.*

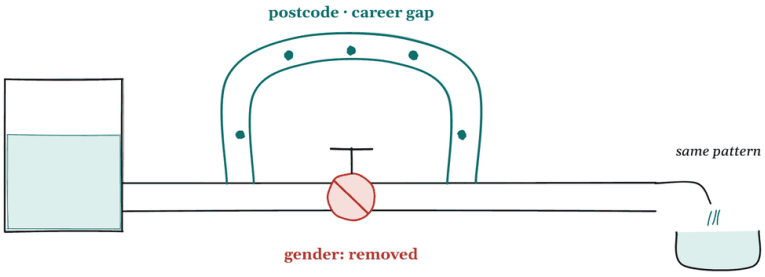
Proxy Discrimination

Deleting a sensitive field from the data does not remove what it predicts. Other fields often carry the same signal. A model trained to find patterns will find the substitute and use it, without anyone asking it to.

Solon Barocas and Andrew Selbst set out the mechanism in "Big Data's Disparate Impact" (2016). Anya Prince and Daniel Schwarcz named it proxy discrimination in 2020. In 2018 Reuters reported that Amazon had dropped an experimental recruiting tool that marked down CVs containing the word "women's".

In Candidate screening at the pump and valve maker, gender is never an input. But a gap in employment history, part-time spells or certain hobbies can stand in for it. In Customer credit check, a sole trader's postcode can stand in for ethnicity. The model never sees the protected trait, yet its outputs can still split along it.

Use it at Strip and Sort. For every input in a people decision, ask what it could be standing in for, and keep the answer in the rule register. Then test outcomes, not inputs, with the four-fifths ratio. The common misreading is that a model blind to a trait cannot discriminate on it.



Like shutting a valve while the water runs round the bypass: *remove the sensitive field and the pattern flows on through the inputs that stand in for it.*

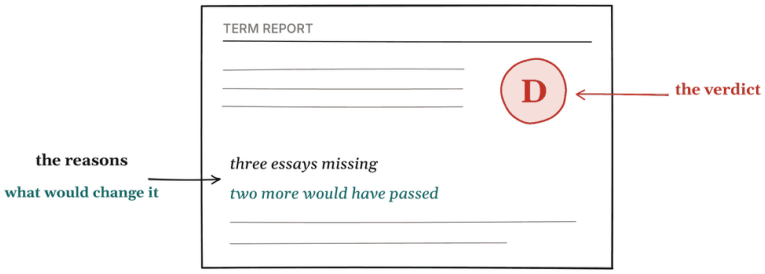
The Explanation Owed

A person turned down is owed more than the verdict. They are owed the main reasons, in words they can act on. A useful explanation says what drove the outcome and what would have changed it.

The US Equal Credit Opportunity Act of 1974 and its Regulation B require a creditor to state the principal reasons for adverse action. The GDPR adds a right to meaningful information about the logic of automated decisions, as read in 2026. Sandra Wachter, Brent Mittelstadt and Chris Russell proposed counterfactual explanations in 2018. These are statements of the smallest change that would have flipped the result.

In Customer credit check, Regulation B requires reasons for a declined small-business applicant, given or disclosed. Larger firms and trade credit can ask in writing within 60 days. The loop must keep the factors behind each decision so a credit manager can state them.

Use it at Design. Make "reasons stated" a required output in the decision register for every consequential people decision, and test it during Shadow. The common misreading is that an explanation means describing how the model works, when the person needs to know why their own case went as it did.



Like a school report: *a grade alone tells the pupil nothing; the teacher's comments say what drove it and what would have changed it.*

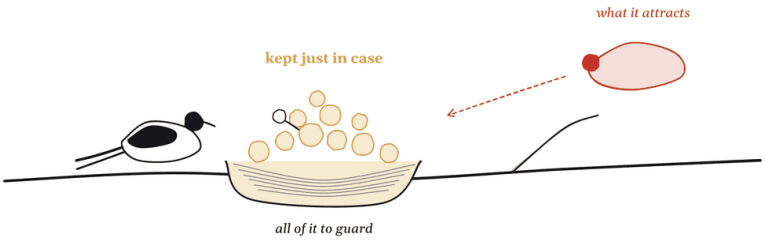
Data Minimisation

Collect only what the decision needs, and keep it only as long as the decision needs it. Every extra field is something to protect, to justify and, sooner or later, to explain.

The OECD's Guidelines on the Protection of Privacy of 1980 set a collection limitation principle. Article 5(1)(c) of the GDPR makes data minimisation binding, as read in 2026. Article 9 adds stricter conditions for special-category data such as health.

In Leave and absence requests, a medical certificate carries health data. An agent that approves routine holiday needs only the balance, the dates and the team calendar. It never needs the diagnosis behind a sick note. At the pump and valve maker, the HR helpdesk agent answers policy questions for all 4,200 people. It does not need their payslips to tell them how parental leave works.

Use it at Strip, when the context an executor sees is decided, and record it in the data protection impact assessment that Article 35 requires before go-live. The common misreading is that more data always makes a better model, when in people loops it mostly makes a larger liability.



Like a magpie's nest: *every shiny thing hoarded just in case becomes one more thing to guard, and the extras are what draw the thieves.*

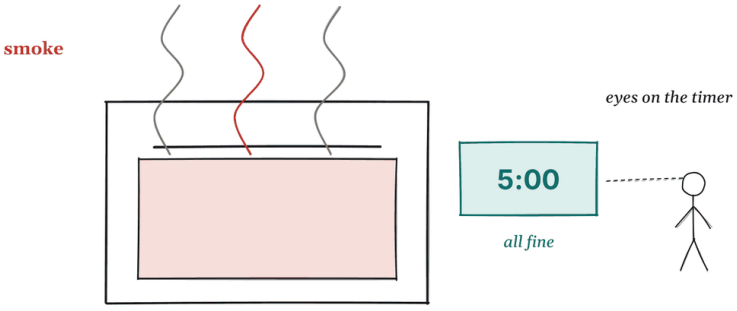
Automation Bias

People who check a system's output tend to accept it. The more reliable the system looks, the less closely they check. Over time the reviewer stops reviewing and starts approving.

Raja Parasuraman and Victor Riley described this misuse of automation in 1997. Linda Skitka, Kathleen Mosier and Mark Burdick measured it in 1999. Their participants followed wrong automated advice and missed events the aid did not flag. Article 14 of the EU AI Act requires that those who oversee a high-risk system stay aware of the tendency to over-rely on it, as read in 2026.

In Candidate screening, a recruiter shown a ranked shortlist with confidence scores will rarely reopen a rejected file. The human review exists on paper. In practice the agent decides. The same drift appears in Payroll changes when an approver signs off a batch the system has already marked as clean.

Use it at Shadow and Own. Watch the override rate. If reviewers almost never disagree, the system is near perfect or the review has gone hollow. Seed known errors to tell which. The common misreading is that adding a human reviewer removes the risk, when the reviewer's attention is the very thing the system erodes.



Like a cook who trusts the oven timer while smoke pours from the door: *the confident display wins over what the reviewer's own eyes would catch.*

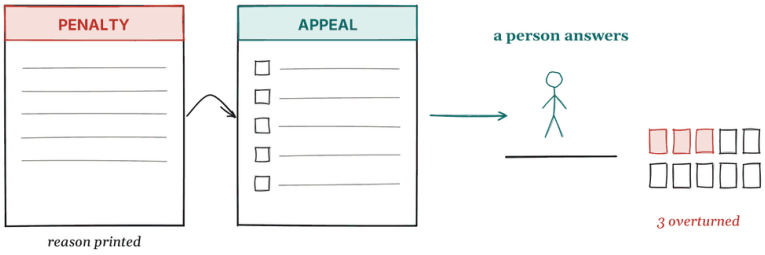
Contestability

A good people loop is built so that its decisions can be challenged. The person needs a channel, a reason to argue with, someone who must answer, and an outcome that can be reversed. The challenges, taken together, also tell the company where the loop is wrong.

Article 22(3) of the GDPR gives a right to contest automated decisions, as read in 2026. Marco Almada argued in 2019 that contestability should be designed into a system from the start, not bolted on. The Platform Work Directive, (EU) 2024/2831, requires written reasons and human review when a platform worker's account is suspended or ended.

In Leave and absence requests, refusing leave for business reasons is a contested decision. In Offboarding, a dismissal is the extreme case. Each needs a route for the employee to push back, reaching someone other than whoever decided.

Use it at Design and Own. Route every challenge into the contested-decision log, track how many succeed, and feed the pattern back into the rule register at each review. The common misreading is that a rising number of appeals means the system is failing, when a loop nobody can challenge simply hides its errors.



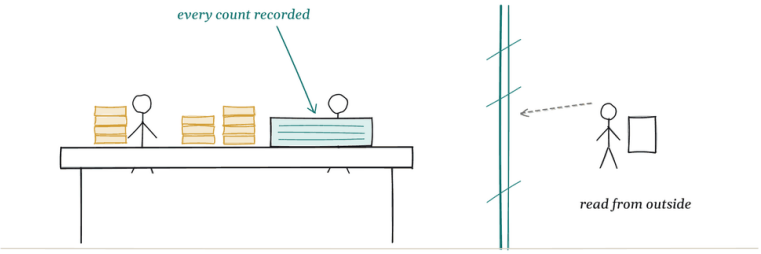
Like a parking ticket with an appeal form on the back: *the reason is printed, the appeal reaches a person, and the overturned tickets show where wardens err.*

CVIII

Regulated and Audited Loops

Some loops are read twice. The people who run them read the records to get the work done, and then an auditor, a regulator or a finance controller reads the same records to decide whether the work can be trusted. These are the loops of record to report, payments and controls over financial reporting. You can recognise one by its sign-offs, its evidence requests and its calendar of quarter-ends. Typical examples among the reference loops are Journal entries, Account reconciliation, Period close, Audit requests, Payment run and Intercompany settlement.

In these loops a correct outcome is not enough. The company must also be able to show, months later and to a sceptical outsider, who did what, on what evidence, under which control. An executor that decides changes who does the work, but not who must answer for it. The models in this chapter are the ones auditors already think with, and the ones a business engineer needs in order to keep a loop auditable once it is embedded.



Like a casino counting room behind glass: *the staff count the cash, and an outsider can watch and later check every count they wrote down.*

Segregation of Duties

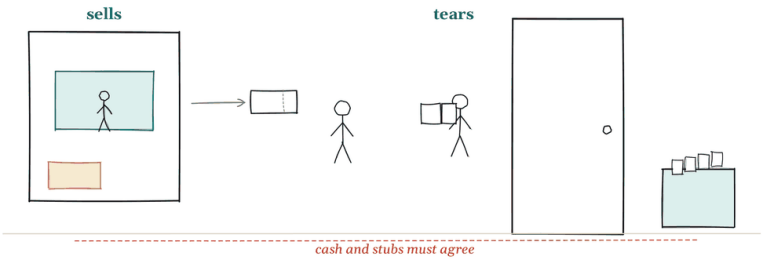
No single pair of hands should be able to start a transaction, approve it and hide it. When one person can create a supplier, enter that supplier's invoice and release the payment, an error or a fraud can travel from start to finish with nobody else seeing it. Split the steps, and a mistake must get past someone who had no part in making it.

Its modern form comes from the COSO *Internal Control: Integrated Framework* (1992, updated 2013). The Sarbanes-Oxley Act of 2002 made it a

matter of law for US-listed companies through management's annual report on internal control.

In a Payment run, the person who maintains supplier bank details must not be the person who releases the payment file. In Journal entries, a preparer cannot approve and post their own manual entry. At the manufacturer, the executor that codes invoices has no access to the supplier master.

Use it at Sort, tagging each rule in the rule register with the duties it separates, and at Design, when the executor's permissions are drawn. The common misreading is that an executor has no duties to segregate, when in fact it is the clerk with the widest reach of all.



Like a cinema where one person sells the tickets and another tears them at the door: *cash and stubs must agree, so neither can cheat alone.*

The Audit Trail

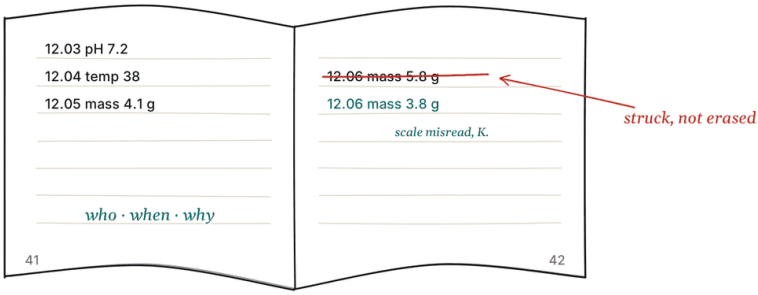
Every change to a record should leave a trace that says what changed, who or what changed it, when, and on what basis. A correction is a new entry that points to the old one, never an erasure.

Luca Pacioli's *Summa de arithmetica* (1494) described the journal and ledger that let a merchant follow any figure back to its source. Modern standards such as PCAOB AS 1105 *Audit Evidence* (2010), require auditors to test whether the information a company produces is complete and accurate. That test is only possible with a trail.

In Account reconciliation, the trail shows which items matched automatically, which a person explained, and who signed the balance. An executor adds a new kind of entry. Its trail must also record the model version, the rule it applied and the evidence it read, because an auditor will ask why a case was decided, not only that it was.

Write the trail requirement into the standard at Design, and test it during Shadow by asking someone outside the team to reconstruct ten decisions from

the log alone. The common misreading is that system logs are an audit trail, when most record activity rather than reasons.



Like a lab notebook with numbered pages in ink: *a mistake is struck through and corrected beside it, never erased, so anyone can see what changed and why.*

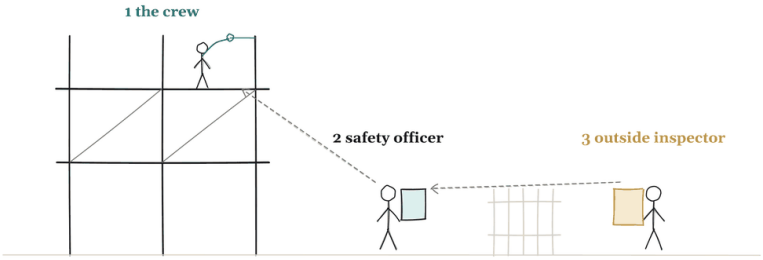
Three Lines of Defence

Risk is managed best when three groups each do a different job. The people who run the work own its risks and control them day to day. A second group sets the policies and watches whether they hold. A third group, independent of both, tells the board whether the whole arrangement works.

The model was set out by the Institute of Internal Auditors in its position paper *The Three Lines of Defense in Effective Risk Management and Control* (2013). It was revised as *The IIA's Three Lines Model* (2020), which stresses cooperation over separate walls.

In Period close, the accounting team runs the close checklist and signs each task. The financial controls function sets which reconciliations count as key controls and reviews exceptions. Internal audit tests a sample each year and reports to the audit committee. An embedded loop must fit all three lines, or one will reject it late.

At Own, name the first-line loop owner, the second-line control owner and the third-line contact in the loop charter. The common misreading is that the executor sits in the first line alone, when the second line must also be able to read its standard and its grading.



Like a building site: *the crew works safely, a site safety officer checks them daily, and an outside inspector checks whether that checking works.*

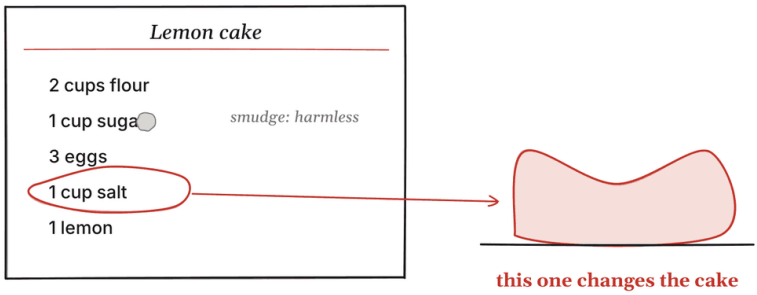
Materiality

An error matters when it is large enough, or of the right kind, to change the decision of someone who relies on the figures. A rounding difference of a few euros in a ledger of millions changes nothing. A small misstatement that turns a loss into a profit changes everything.

The accounting definition is in the IASB's *Definition of Material*, amendments to IAS 1 and IAS 8 (2018). In the United States, SEC Staff Accounting Bulletin No. 99 (1999) warned that a percentage rule of thumb is not enough, because a small amount can still be material for qualitative reasons.

In Period close, a materiality threshold decides which variances must be investigated and which late errors force the books to be reopened. In Intercompany settlement, a similar threshold decides when an unmatched difference may be written off. The threshold is set by people, not by the executor.

Record materiality thresholds as rules in the rule register at Sort, each with its source and owner, and use them to set the edges of bounded scope at Bound. The common misreading is that materiality is a single number, when qualitative factors such as fraud or a broken covenant override any size.



Like a typo on a recipe card: *a smudged letter changes nothing, but a cup of salt instead of a teaspoon ruins the cake, so only that error is fixed.*

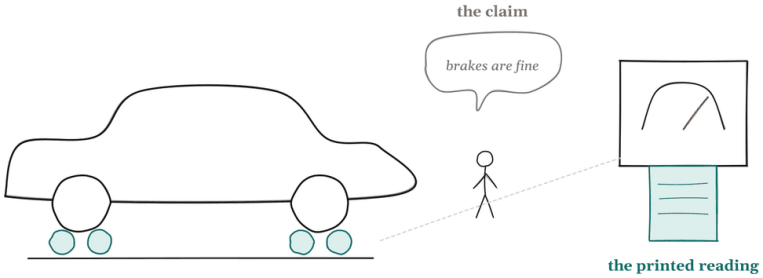
Evidence Over Assertion

A statement that something is true is a claim. Evidence is what lets someone else check the claim without trusting the person who made it. In audited work, "the reconciliation was reviewed" counts for little. A signed, dated review with the reconciling items attached counts for a great deal.

Audit standards call management's statements "assertions" and require auditors to gather evidence for each one. The international version is ISA 500 *Audit Evidence* (IAASB, 2009). The same stance runs through PCAOB AS 1105 (2010), which ranks evidence by its source and by how directly it was obtained.

In Audit requests, the auditor asks for evidence that a control operated, not for a description of the control. An executor can assemble that evidence faster than anyone, but its own summary is still an assertion. The packet must point to the source records, so the auditor can test them directly.

At Design, write the standard so that each criterion names the evidence that proves it, and make the junction register say what evidence the junction holder sees. The common misreading is that an executor's confident explanation is evidence, when it is only a well-written assertion.



Like a roadworthiness test: *the owner says the brakes are fine, but the garage runs the car on the rollers and prints a reading anyone can check.*

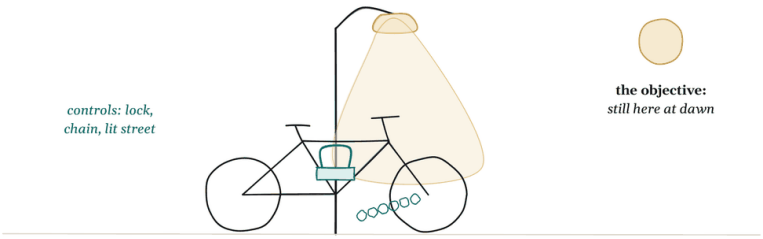
Control Objectives and Controls

Every control exists to make something true. The thing that must be true is the objective, for example "only valid suppliers are paid". The control is one way of achieving it, such as a call-back before bank details change. Several controls can serve one objective, and a control with no objective is only a habit.

The distinction was made explicit by ISACA's *Control Objectives for Information and Related Technologies*, known as COBIT (1996). Service-organisation reports under the AICPA's SSAE 18 (2017) still list control objectives first and the controls that meet them second.

In Supplier invoice approval, the objective "no invoice is paid twice" was once met by a clerk scanning for duplicates. Once the loop is embedded, the same objective can be met by a mechanical duplicate check on every invoice plus grading. The objective stays fixed while the control changes.

At Sort, list each control in the rule register beside the objective it serves, and agree with the control owner that objectives are signed and controls may be redesigned. The common misreading is to preserve every existing control unchanged, when auditors care whether the objective is still met.



Like keeping a bicycle overnight: *the goal is the bike still there at dawn, and the lock, the chain and the lit street are different ways to make sure.*

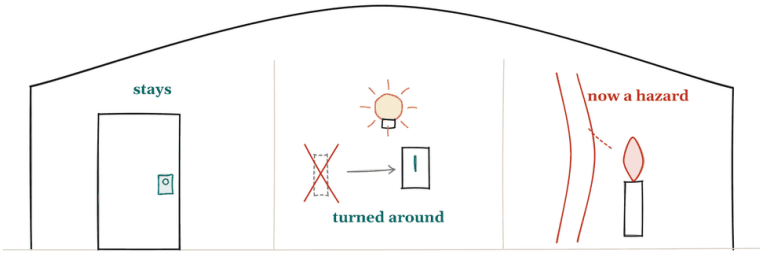
Controls That Survive and Controls That Invert

Controls written for people do not all carry over to an executor that decides. Some work exactly as before. Some keep their purpose but must be turned around to work. A few become harmful if they are left as they are.

This model comes from embedding practice and lives in the junction register and the standard. It is defined in the Control and Acceptance area of this book. It applies the previous model directly: because the objective is fixed, the control is free to change.

A call-back on changed bank details survives unchanged. Segregation of duties inverts. It becomes a separation between the executor and the standard it is graded against, with junctions at the consequential points. A four-eyes review becomes a grader plus a junction, and sampling becomes grading every outcome. "Review everything" becomes dangerous. Applied to thousands of decisions a day in Account reconciliation, it guarantees that reviewers stop reading.

At Design, sort every existing control into one of the three groups, and take the sorted list to the control owner before Bound. The common misreading is that a control which inverts has been weakened, when it usually covers more cases than the original did.



Like a house moving from candles to electric light: *the door lock stays, the nightly candle round becomes a switch, and candles left burning everywhere become a hazard.*

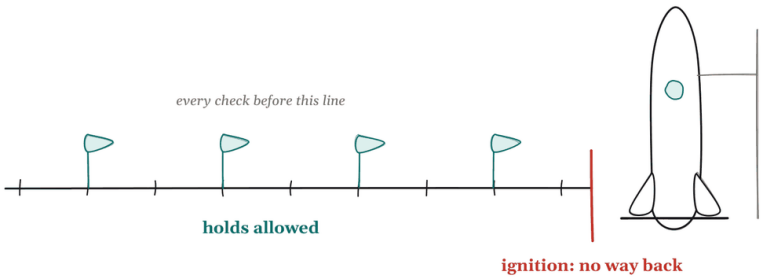
The Irreversible Action

Most mistakes in a loop can be undone. A wrong code can be reversed and a held invoice released. A few actions cannot be pulled back once taken, and they deserve a different kind of care. Every check that matters must sit before that point, because nothing placed after it can help.

This model comes from embedding practice and lives in the junction register, where junctions are placed by reversibility and cost. A familiar ancestor is Jeff Bezos′s distinction between one-way and two-way doors, in his letter to Amazon shareholders for 2015 (2016).

The Payment run is the clearest case. Once funds leave, recovery depends on the receiving bank and on the payee′s goodwill. Under the EU Instant Payments Regulation (2024), euro credit transfers settle in seconds, which removes the old overnight window to catch an error, as read in 2026. Locking a period at the end of Period close is another one-way door.

At Design, mark every irreversible action on the Work Loop Canvas and give each a junction with evidence and a named holder. The common misreading is to place junctions where people feel uneasy, when they belong where the action cannot be undone.



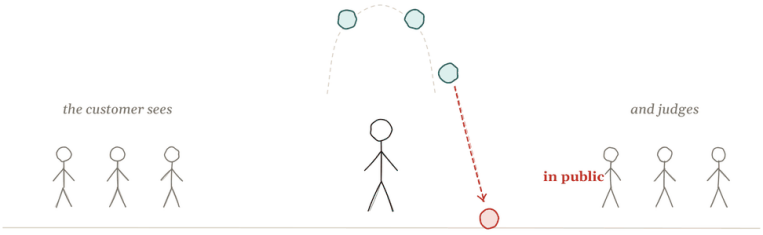
Like a rocket countdown: *holds are allowed at any point before ignition, so every check sits before that moment, because after it nothing comes back.*

CIX

Customer-Facing Loops

Some loops end in front of a customer. The outcome is an answer, a refund, a reply to a complaint, a call from a seller or a reminder to pay. You can recognise them by one test: the person who receives the result can see it, judge it and respond. Typical reference loops are Support ticket resolution, Returns and refunds, Complaint handling, Lead qualification, Collections and cash application and Warranty claims.

In a back-office loop, a weak outcome is caught by the grader or the review before anyone outside notices. Here the customer notices first, and the customer's reaction becomes part of the result. Words become commitments, a closed case can return as a second call, and a single badly handled minute can outweigh a fast process. The models in this chapter matter most here because they deal with that exposure: who else is grading, what the executor may say and promise, and which moments the customer will remember.



Like a street juggler: *practice happens out of sight, but every ball dropped in the act lands in front of the crowd who decide whether to stay.*

The Customer Grades Too

A company can write a careful standard, grade every outcome against it and still lose the customer. The customer applies a standard of their own. It rests on what they expected, and nobody inside the company wrote it down.

Service researchers made this gap explicit. A. Parasuraman, Valarie Zeithaml and Leonard Berry, in their 1985 model of service quality, defined quality as the distance between what a customer expects and what they perceive they received. Their later SERVQUAL instrument (1988) tried to measure that distance directly.

In Support ticket resolution the gap is easy to miss. An executor can answer a delivery question exactly as the policy says and pass the grader. The customer, who wanted the parcel rather than the policy, rates the contact poorly and asks again.

Use it when writing the standard at Design. Add at least one signal that comes from the customer rather than from the company: a reopened ticket, a follow-up complaint, a short rating. Read it next to the grader's verdict, band by band. The common misreading is that a high grader pass rate means satisfied customers, when it only means the company's own criteria were met.



Like a contest with two judges: *the company's grader holds up a pass, while the customer holds up their own score against what they expected.*

Liability for What the Executor Says

Whatever the executor tells a customer, the company has said. It cannot later argue that the words came from a system rather than from staff.

The clearest test so far is Moffatt v. Air Canada, decided by the British Columbia Civil Resolution Tribunal in 2024. A chatbot on the airline's website told a passenger he could claim a bereavement fare after travelling. The airline's actual policy said otherwise. The tribunal rejected the idea that the chatbot was responsible for its own statements, and ordered the airline to pay the difference.

Returns and refunds is exposed in the same way. An executor that tells a customer an opened item can still be returned within thirty days has made that the company's position for that customer, whatever the policy page says.

Use it in the rule register. Every policy statement the executor may make should trace to a signed rule, and the grader should check each answer against the rule text, not against whether it sounded helpful. The common misreading is that a disclaimer saying answers may be inaccurate shifts the risk back onto the customer.



Like a price painted in a shop window: *if the sign-writer paints the wrong price, the shop still has to honour it at the till.*

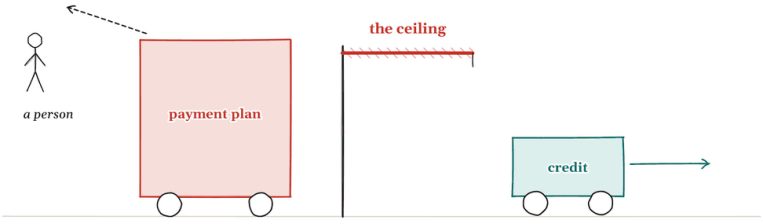
Promise Limits

An executor that talks to customers needs two kinds of limit. One covers what it may do. The other covers what it may offer, because an offer made in writing is already a kind of action.

This model is native to embedding practice. It lives in the decision register, where offers are sorted like any other decision, and in the limits set at Bound. It extends the ordinary approval limit, familiar from any delegation of authority, from money moved to words used.

Collections and cash application shows why. Sending a payment reminder is mechanical. Agreeing a payment plan or putting an account on hold is consequential and sits with the credit manager. An executor that writes "we can split this over three months" has agreed the plan, whoever was meant to approve it.

Write the list of offers the executor may make, each with a ceiling: a goodwill credit up to a set amount, a replacement but not a refund, a callback but not a delivery date. Anything above the ceiling goes to a junction. The common misreading is that limiting the executor's system permissions also limits what it can promise in a message.



Like the height bar before a low bridge: *offers that fit under the bar go through, and anything taller is stopped and sent round to a person.*

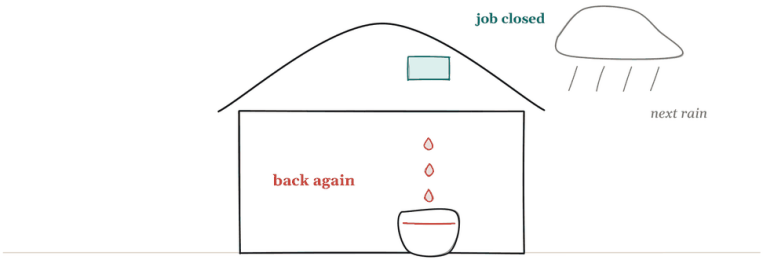
Repeat Contact as the True Measure

A case marked closed is not the same as a problem solved. The surest sign that it was not solved is that the same customer comes back about the same thing.

Bill Price, who led customer service at Amazon, made this the centre of his book with David Jaffe, The Best Service Is No Service (2008). He argued that most contacts signal a failure somewhere upstream, and that contacts per order mattered more than speed. Matthew Dixon, Karen Freeman and Nicholas Toman reached a related finding in Harvard Business Review in 2010: customer effort, including having to call again, predicted disloyalty better than delight did.

Support ticket resolution is where the trap is set. An executor can close tickets quickly and drive the closure rate up while the same customers return within days. The dashboard improves and the work grows.

Measure repeat contact on the same issue within a fixed window when you map the baseline, and carry it into the standard and the economics sheet. A repeat counts as a cost of the first contact. The common misreading is that a contact the executor deflected is a contact resolved.



Like a roof patch: *the roofer marks the job done, but the real test is whether the bucket under the ceiling fills again at the next rain.*

Service Recovery

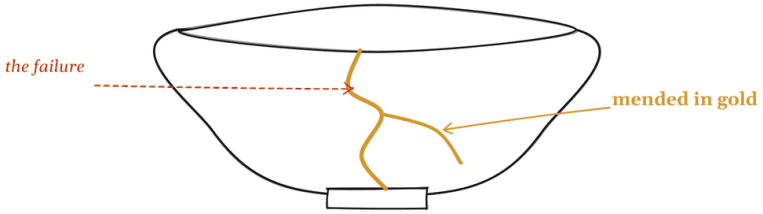
Failures will reach customers, however good the loop. What the company does next often matters more to the relationship than the failure itself. A quick, fair and visible repair can keep a customer who would otherwise have left.

Christopher Hart, James Heskett and Earl Sasser set this out in "The Profitable Art of Service Recovery" in Harvard Business Review in 1990. They argued that recovery should be planned and that front-line staff need the authority to fix things on the spot. Later research, such as Stefan Michel and

Matthew Meuter in 2008, found that recovery rarely leaves customers happier than if nothing had gone wrong.

In Complaint handling, recovery is the loop's real output. An executor can acknowledge the complaint at once, gather the facts and propose redress within its limits. It shortens the slowest part of the old process, which was usually the wait before anyone replied.

Design the recovery route as part of the loop, with remedies listed in the decision register and their ceilings set at Bound. The common misreading is that a good recovery makes up for a failure, so the failure rate matters less.



Like a bowl mended with gold: *a crack repaired quickly and visibly can keep the bowl on the table, though a bowl that never cracked is still better.*

The Peak-End Rule

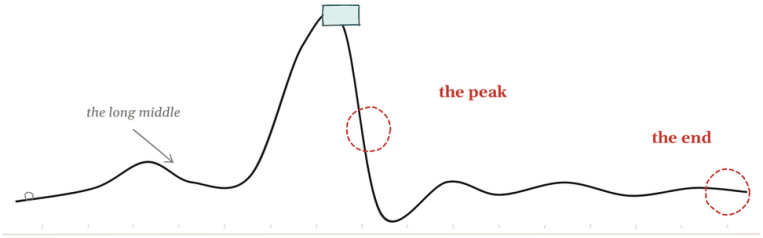
People do not remember an experience as an average of its moments. They remember mostly its most intense moment and how it ended. Its length weighs surprisingly little.

Daniel Kahneman and colleagues showed this in a 1993 paper in Psychological Science, "When More Pain Is Preferred to Less: Adding a Better End". Volunteers who held a hand in cold water for longer, with a slightly milder final stretch, preferred that trial to a shorter one. The pattern became known as the peak-end rule, and the small weight given to length as duration neglect.

Returns and refunds has an obvious peak and an obvious end. The peak is often the moment the customer is told an item is not eligible. The end is the refund arriving, or not. An executor that handles the refusal coldly and closes with a template will be remembered for those two moments, however smooth the rest.

Use it at Design and in grading. Give the refusal and the closing message particular care, and have the human audit read them first. The common

misreading is that only the end matters, so the middle of the experience can be neglected.



Like a rollercoaster ride: *people remember the steepest drop and the way it stopped, hardly how long they queued or rode.*

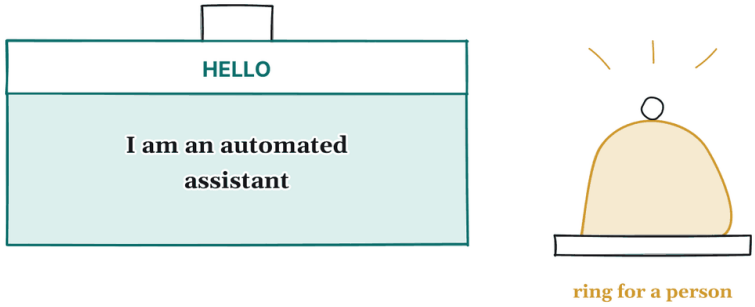
Disclosure

Customers are entitled to know when they are dealing with a machine. Telling them early is simpler than being found out late, and it lets them ask for a person when they need one.

This is now law in several places. Article 50(1) of the EU AI Act requires that people interacting with an AI system be told so, unless it is obvious. It applies from 2 August 2026, as read in 2026. California's Bolstering Online Transparency Act of 2018 already required bots used to sell to disclose that they are bots.

Lead qualification meets this first. An executor that answers a prospect's pricing questions on a website, or writes the first outreach email, is exactly the system the article describes. Disclosure becomes a design decision made once, not a review of each conversation.

Record the obligation in the rule register for every customer-facing loop. At Design, place the disclosure in the first message and add a visible path to a person, which a junction can serve. The grader can then check both on every case. The common misreading is that disclosure is a banner that satisfies the rule, when its purpose is to give the customer a real choice.



Like a name badge that says what you are dealing with, worn beside a bell that brings a person whenever the customer rings it.:

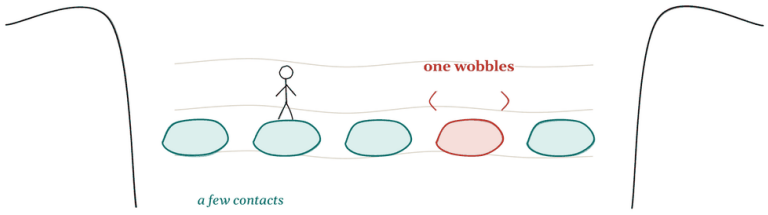
Moments of Truth

A customer meets a company in a few short contacts. Each one is a test. If it fails, the quality of everything behind it is invisible to that customer.

Jan Carlzon, who turned around the airline SAS in the early 1980s, called these contacts moments of truth in his book of that name (1987). He argued that the front line must be able to act in those moments, without passing every question up the hierarchy.

In Collections and cash application, a payment reminder is a moment of truth with a customer who is still buying. At the manufacturer, a reminder sent to a distributor who is already disputing a short delivery tells that distributor nobody has read the file. The reminder was correct and the moment was lost.

Use it at Map. Mark every point in the loop where the result reaches the customer, and check that the executor has the context it needs at each one. A reminder should not go out while a dispute is open. The common misreading is that every step in a loop is a moment of truth, when only the few the customer actually sees are.



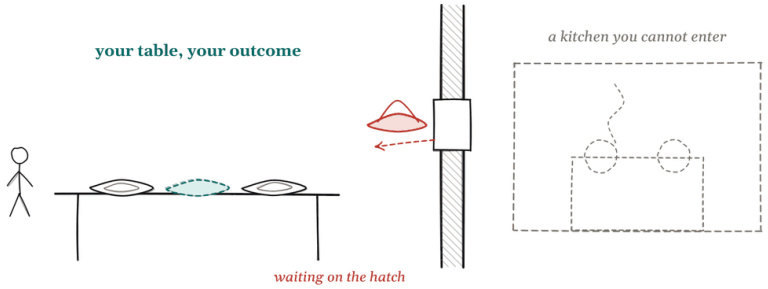
Like stepping stones across a river: *the customer touches only a few stones, and one that wobbles drops them in, however solid the rest.*

CX

Loops That Depend on Outside Parties

Some loops cannot finish inside the company. The case starts with a carrier's scan, a supplier's confirmation or a bank's file, and the action ends in someone else's system. You can recognise these loops by their waiting: the queue holds cases marked "awaiting supplier", "chasing carrier" or "pending bank". Typical examples among the reference loops are Shipment exception handling, Supplier delivery exceptions, Supplier onboarding, Payment run, Goods and services receipt and Inventory replenishment.

In these loops the company owns the outcome but not the facts. The executor reads data it did not create, acts through parties it does not manage, and pays for their mistakes as if they were its own. The models below matter most here because each one names a way an outside party can quietly decide how well the loop runs. Together they turn a vague worry about partners into questions a loop owner can answer before the executor goes live.



Like a host whose dinner depends on a neighbour's kitchen: *the host owns the meal, but the dish arrives through a hatch the host cannot see behind.*

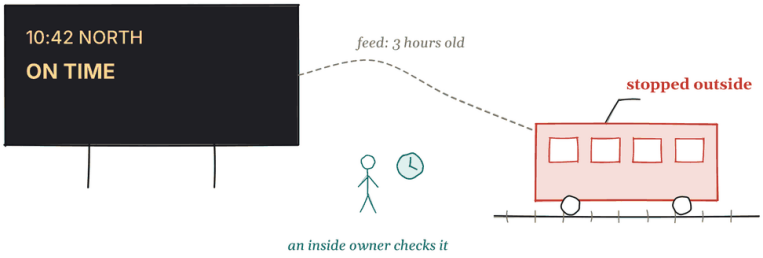
Context Keepers Outside the Company

An executor is only as good as what it reads. In these loops much of that reading is kept by someone on the other side of a contract. The carrier keeps the tracking events. The supplier keeps the order confirmation. The bank keeps the account check. Nobody inside can fix those sources, yet someone inside must answer for them.

The idea comes from embedding practice. It lives in the executor's list of context sources and keepers, extended with a column for who outside holds the source and who inside answers for it.

Shipment exception handling shows why. The executor tells a customer where an order is, and it can only repeat what the carrier's last scan says. If a carrier posts delay events hours late, the executor gives confident, wrong answers. The fault looks like the model's. It belongs to a feed nobody inside was watching.

Use it at Design. For every outside source, name an inside owner, record how fresh the data usually is, and decide what the executor does when it is stale. The common misreading is that an outside source needs no keeper because the company cannot change it.



Like a station board fed by another company's signals: *when the feed runs late the board shows confident nonsense, so someone inside must watch the feed.*

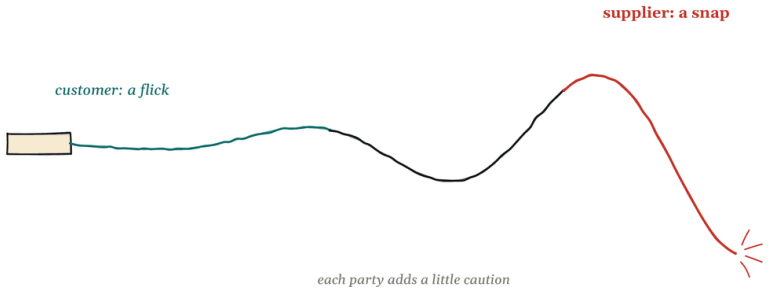
The Bullwhip Effect

Small changes at one end of a chain become large swings at the other. Each party sees only the orders of the party next to it, adds a little caution, and passes the result on. A gentle ripple in customer demand arrives at the supplier as a crashing wave.

Jay Forrester showed the pattern in simulations of production and distribution systems in Industrial Dynamics in 1961. Hau Lee, V. Padmanabhan and Seungjin Whang named it the bullwhip effect in 1997. They traced it to four causes: forecasts built on orders rather than demand, batching, price swings and rationing games.

Inventory replenishment is where an executor can make it worse. A replenishment agent that reorders faster than people did, on the same order signals, amplifies the same noise more quickly. Suppliers then see sharper swings and answer with longer lead times, which feed back into the next order.

Use it at Map, when tracing handoffs between loops. Ask what signal each loop reorders on, and feed it real demand rather than the previous loop's orders. The common misreading is that the bullwhip is a supplier problem, when the company's own ordering rules are usually the first amplifier.



Like cracking a whip: *a small flick at the customer's end of the chain grows into a violent snap by the time it reaches the supplier.*

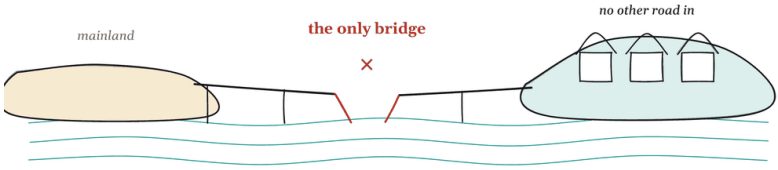
Single-Source Risk

When one outside party carries a step, its failure becomes yours. A loop with one carrier, one bank link or one supplier for a critical part looks efficient on a normal day. On a bad day it has no other road.

Peter Kraljic set out the purchasing side in the Harvard Business Review in 1983. He sorted supplies by their profit impact and supply risk, and singled out the items that are both critical and hard to source. Yossi Sheffi's The Resilient Enterprise (2005) documented how a 2000 fire at a Philips chip plant hit Ericsson, which had no alternative source, far harder than Nokia, which moved quickly to other suppliers.

Embedding adds a new single source: the data feed. Supplier delivery exceptions may run on a single visibility platform. If that platform changes its format or goes down, the executor stops reading every inbound delivery at once.

Use it in the junction register and at Bound. For each outside party, record what the loop does if it disappears for a day, and rehearse that path with people. The common misreading is that a second supplier on paper removes the risk, when the executor is still wired to only one.



Like an island village with one bridge: *it is the quickest road every day until the bridge goes, and then there is no road at all.*

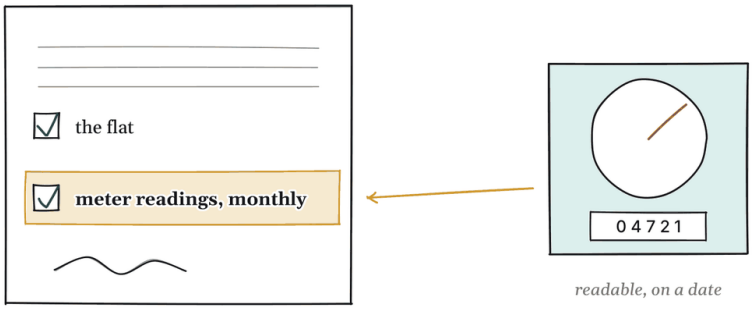
Contract the Data, Not Only the Service

Agreements with partners usually describe the service: the goods, the delivery window, the price. An executor also needs the data that comes with the service, in a form it can read and on a timetable it can rely on. If the contract is silent, the partner owes nothing when the data is late, partial or locked in a PDF.

The model comes from embedding practice and lives in the executor's list of context sources. It has a long precedent in data protection. GDPR Article 28 requires a controller to bind its processors by contract to specific duties over personal data. The same habit suits operational data.

The manufacturer met this early. Contract prices for about a third of its suppliers existed only in PDF agreements, so the invoice executor could not read them. Procurement loaded prices for the top two hundred suppliers before shadow running. Supplier onboarding is where the gap can be closed at the source.

Use it at Strip and Design. Add structured confirmations, notice periods and data formats to onboarding terms and renewals. The common misreading is that data quality is the partner's goodwill rather than a term you negotiate.



Like a flat lease that also fixes the meter readings: *write in what you will be shown and when, not only what you will get.*

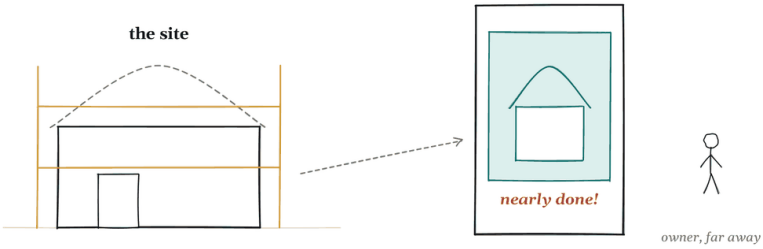
The Principal-Agent Problem

When you hire someone to act for you, their interests are close to yours but not the same. They also know things you cannot see. Left alone, they will act sensibly for themselves, and that is not always sensibly for you.

Stephen Ross framed the principal's problem in economic terms in 1973. Michael Jensen and William Meckling gave it its lasting form in 1976, as the costs of aligning and monitoring an agent whose effort the principal cannot fully observe.

Outside parties in these loops are agents in exactly this sense. A supplier eager to win an order confirms a delivery date it is unsure it can meet. A carrier marks a parcel "attempted" rather than admit a missed stop. In Supplier delivery exceptions, the executor reads these signals at face value unless someone has taught it which ones to doubt.

Use it when filling the decision register. Note where a partner's report is also a claim in its own interest, and set the executor to check that claim against an independent fact, such as the goods receipt. The common misreading is that bad partner data is careless, when it is often rational.



Like a builder sending progress photos to an owner abroad: *the report says nearly done because that keeps the payments coming, whatever the site shows.*

Lead-Time Variability

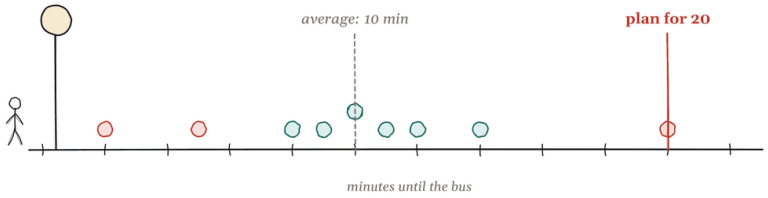
How long something takes on average matters less than how much that time wanders. A supplier who always takes six weeks is easy to plan around. One who takes four weeks or ten forces the company to hold extra stock or accept shortages, even if the average is the same.

Wallace Hopp and Mark Spearman made the point central in Factory Physics in 1996. Their laws of variability show that variation in arrivals and processing must be paid for in inventory, capacity or time. The standard safety-stock

formula in operations textbooks carries lead-time variance as its own term for the same reason.

The pattern dominates Supplier delivery exceptions. APQC's open benchmark puts median supplier on-time delivery at 90 percent, as read in 2026. That still leaves one delivery in ten off its date. An executor that re-dates exceptions quickly reduces surprise, but it cannot remove the spread.

Use it at Read. Plot the spread of each key supplier's lead times, not just the mean, and treat the wide ones as tail. The common misreading is that a faster average lead time solves the problem when the variance is unchanged.



Like a bus that comes every ten minutes on average but sometimes in two and sometimes in twenty: *you plan around the spread, not the average.*

Trust, Then Verify

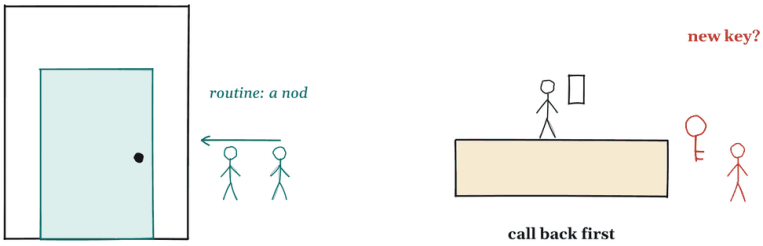
Partners must be trusted for the work to move at all. Checking every routine message from every supplier would stop the loop. The skill is to trust by default on what is cheap to undo, and to verify without exception on what is not.

The phrase adapts an old Russian proverb, "trust, but verify", which Ronald Reagan used at the signing of the Intermediate-Range Nuclear Forces Treaty in 1987. In embedding practice the idea lives in the bounded limits and the junctions. Reversibility decides where checks go.

Payments make this concrete. In the Payment run loop, a request to change a supplier's bank details is the classic route for invoice fraud. The Association for Financial Professionals' 2025 survey found that 79 percent of organisations faced attempted or actual payments fraud in 2024. A bank detail change arriving by email should never pass on trust, however familiar the sender looks.

Use it at Bound and Shadow. Let the executor accept routine confirmations alone, and route every change to bank details or remittance through a call-

back to a number already on file. The common misreading is that verification signals distrust of the partner, when it protects both sides.



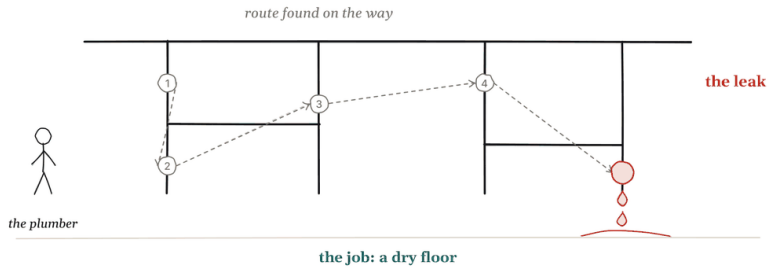
Like a doorman: *residents walk in with a nod, but anyone asking for a new key is checked by a call to the number already on file.*

CXI

Loops Whose Steps Are Generated

Some loops have a clear outcome but no fixed path to it. A customer reports a problem nobody has seen before, a server fails in a new way, a shipment is stuck at a border. The executor reads the case, decides what to look up, tries something, checks the result and decides again. Each case produces its own sequence of steps. You can recognise these loops because the people who run them describe their work as investigating rather than processing, and two good handlers rarely take the same route.

Typical examples among the reference loops are Support ticket resolution, IT incident management, Shipment exception handling, Supplier delivery exceptions, Complaint handling and Audit requests. With no procedure to audit, the design moves up a level: to what counts as done, what the executor may touch, and how its results are checked. The models in this chapter keep an executor that plans its own steps both useful and answerable.



Like a plumber called to a strange leak: *the job is a dry floor, and the route to the leak is found pipe by pipe on the day.*

Outcome, Fences and Trace

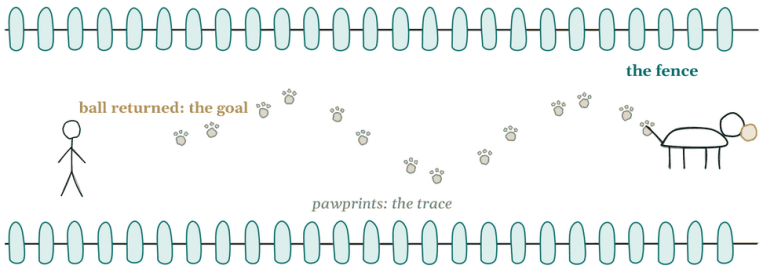
When you cannot write the steps in advance, you specify three other things instead. The first is the outcome, meaning what a finished case looks like. The second is the fences: the rules, permitted actions and junctions the executor may not cross. The third is the trace, a record of every step it actually took.

The model is native to embedding practice, where it fills the generated-steps node of the loop's canvas. Its ancestor is management by objectives, set out by

Peter Drucker in The Practice of Management in 1954. Drucker asked managers to agree results and leave the method to the person doing the work.

In Shipment exception handling, the outcome is a shipment resolved by reroute, reship, refund or claim, with the customer told accurately. The fences let the executor reroute within a cost limit but not declare a parcel lost. The trace shows which carrier records it read before choosing.

At Design, write the outcome into the standard, the fences into the rule register and the actions list, and require the platform to keep the trace for every case. The common misreading is that a trace replaces a design, when it only lets you inspect one.



Like a dog fetching a ball in a fenced garden: *you set the goal and the fence, it picks the route, and its muddy pawprints record it.*

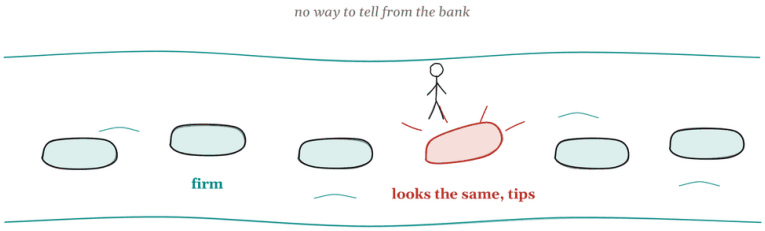
The Jagged Frontier

An AI executor is not simply good or bad at a type of work. It is excellent at some tasks and poor at neighbouring ones that look just as hard. The boundary cannot be seen from outside, and its poor answers often sound as confident as its good ones.

The phrase comes from a 2023 field experiment by Fabrizio Dell'Acqua and colleagues with consultants at Boston Consulting Group. On tasks inside the frontier, consultants using AI produced clearly better work. On a task just outside it, they did worse than colleagues without AI, because they accepted answers they should have checked.

Loops with generated steps meet the frontier case by case, since every case calls for a different chain of reasoning. In Support ticket resolution, an executor may answer order-status questions flawlessly and still misstate a refund policy. A Canadian tribunal held Air Canada liable for exactly that kind of chatbot answer in Moffatt v. Air Canada in 2024.

At Shadow, read agreement by case type and by band, never as one overall figure. Keep with people whatever types fall outside the frontier until the evidence moves. The common misreading is that a strong demonstration shows where the frontier lies.



Like stepping stones across a river: *firm stones and loose ones look identical from the bank, and the loose one tips only when trusted.*

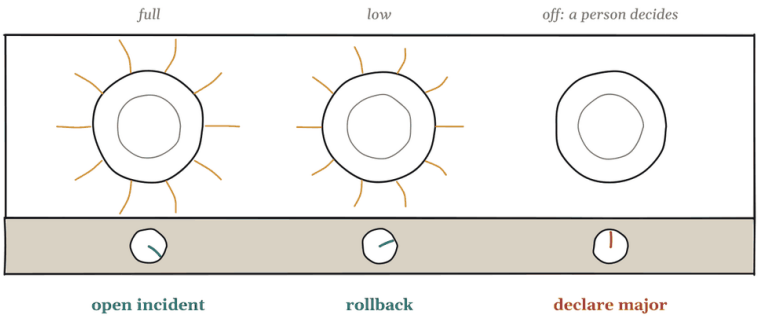
Levels of Autonomy

Handing work to a machine is not one switch between manual and automatic. There are settings in between. The machine can list options, recommend one, act once a person approves, act and then report, or act and say nothing unless asked. Each setting moves a different share of judgement away from people.

Thomas Sheridan and William Verplank set out a ten-level scale of this kind in 1978, in an MIT report on controlling undersea teleoperators. In 2000, Raja Parasuraman, Sheridan and Christopher Wickens added that the level should be chosen separately for each function of a system, not once for the whole.

That refinement matters most when steps are generated. In IT incident management, an executor might correlate alerts and open incidents on its own, apply a known rollback and report it at once, and only propose that a major incident be declared. One loop runs at three levels at the same time.

Record the level for each decision in the decision register. Raise it only through the Scale Gates, with shadow evidence for every step up. The common misreading is to give a whole loop one autonomy level, as if every decision in it carried the same risk.



Like a cooker with a knob for each ring: *one loop can run one decision on full, another on low, and keep a third switched off.*

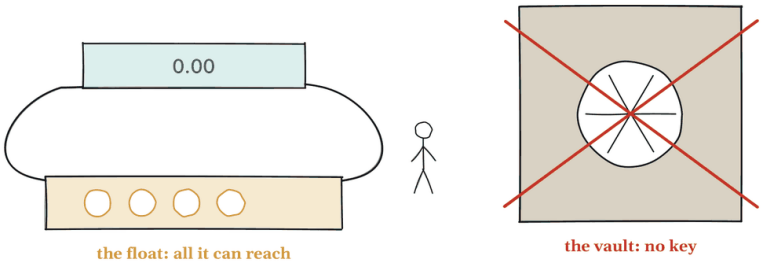
Least Privilege

Give every worker, human or machine, exactly the access its job needs and nothing more. Then a mistake, a misunderstanding or a deliberate misuse can only reach as far as that access reaches. The damage is bounded by the rights held, not by good intentions.

Jerome Saltzer and Michael Schroeder named the principle of least privilege in "The Protection of Information in Computer Systems", published in the Proceedings of the IEEE in 1975. It has been a basic rule of computer security ever since.

It bites harder when an executor plans its own steps, because nobody can list in advance every action it might attempt. An executor assembling evidence for Audit requests needs to read ledgers, populations and reconciliations. It has no need to post journal entries. Holding that right would let one wrong plan alter the very records the auditor is testing.

At Bound, compare the rights the executor holds in each system with the signed actions list, and remove everything that is not on it. Repeat the check whenever a model or tool is added. The common misreading is that broad rights are harmless as long as the instructions say not to use them.



Like a cashier's till holding only a float: *whatever goes wrong at the counter, the loss can never exceed what the drawer holds.*

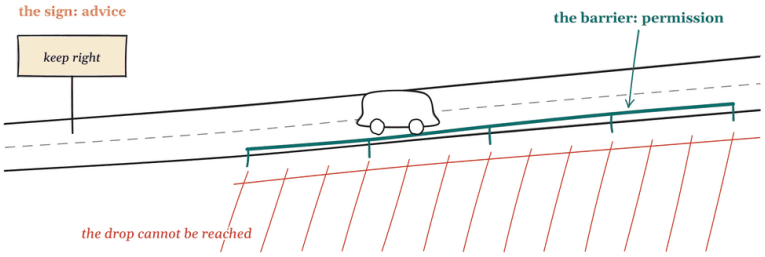
Permissions as Design

The rights an executor holds are not an administrative detail settled after the design. They are the design in its most binding form. Instructions describe what the executor should do. Permissions decide what it can do, and when the two disagree, permissions win.

The legal scholar Lawrence Lessig made the general argument in Code and Other Laws of Cyberspace in 1999. He showed that the architecture of a system regulates behaviour as firmly as law, often more firmly, because it needs no enforcement. Embedding practice applies the idea to the executor's actions list.

In Returns and refunds, policy may say that a refund outside the return window goes to a person. Written into the instructions, that sentence is advice. A permission set that lets the executor refund only inside the window, and only to the original payment method, makes the policy impossible to break. The exception reaches a person because no other path exists.

Design the permission set alongside the junctions, and treat each permission as a decision someone signs at Bound. Where a rule really matters, enforce it in permissions rather than in prompts. The common misreading is that permissions belong to IT while the business writes the rules.



Like a crash barrier on a mountain road: *a sign only asks drivers to keep right, while the barrier makes the drop impossible to reach.*

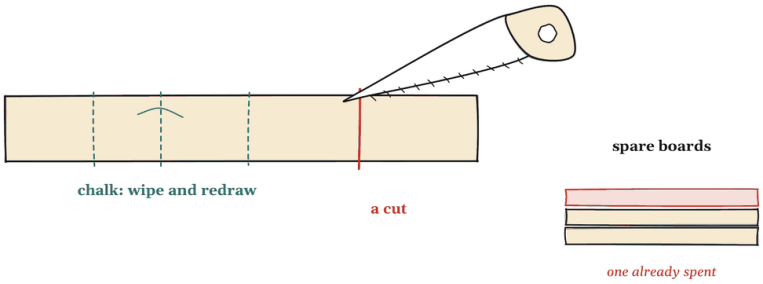
The Reversibility Budget

Some actions can be taken back cheaply, some at a cost, and some not at all. An executor that finds its own path will sometimes take a wrong step. The question is how much irreversible action it may spend before a person has to look.

The model is native to embedding practice and lives in the actions list and the economics sheet. Each action carries a cost of undoing it, and each loop receives an allowance of irreversible exposure per case and per period. Jeff Bezos drew the same line between one-way and two-way doors in Amazon's 2015 letter to shareholders.

In Supplier delivery exceptions, chasing a late supplier costs nothing to retract. Accepting a revised date can be revisited. Booking premium air freight to avoid a line stoppage at the manufacturer spends money that will not come back. A sensible budget lets the executor chase freely and expedite only up to a small sum per case, under a monthly ceiling.

Set the budget at Bound from shadow evidence, enforce it in the platform, and read its use at the operating review. The common misreading is that a reversible action is free, when undoing it still costs time and goodwill.



Like a carpenter's chalk and saw: *chalk lines can be wiped and redrawn freely, but every saw cut spends one of a few spare boards.*

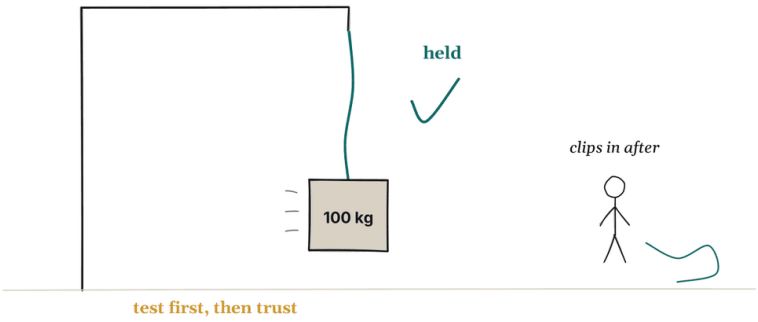
Evaluation Before Trust

Trust in a machine should follow evidence of how it performs, not run ahead of it. People tend to trust too little at first and far too much after a few good results. Both errors come from impressions rather than measurement.

John Lee and Katrina See made the argument in "Trust in Automation: Designing for Appropriate Reliance", in Human Factors in 2004. The aim, they wrote, is calibrated trust, matched to what the system can actually do. Raja Parasuraman and Victor Riley had already described the costs of misuse and disuse in 1997.

With generated steps there is no procedure to inspect, so graded outcomes are the only evidence. Klarna's experience in Support ticket resolution shows the risk. Its AI assistant handled two thirds of customer chats in its first month in 2024. The company later recruited human agents again, after its chief executive said quality had suffered.

Write the standard and build the grader before the executor touches a live case. Let autonomy grow only as graded results accumulate, at Shadow and then in Core, with a human sample checking the grader. The common misreading is that trust is earned once at launch, rather than renewed every month.



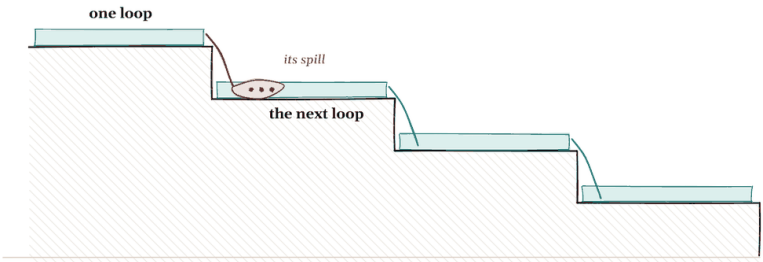
Like a new climbing rope drop-tested with a heavy weight: *the climber clips in only after the rope has held, and it is tested again each season.*

CXII

Loops in a Chain

Some loops exist mainly to pass work along. Their outcome is the next loop's input, so a sales order becomes a pick list, a receipt becomes a match, and an invoice becomes a payment. You can spot them on the map by their lines. They sit inside the long chains, such as lead to cash, procure to pay and order to delivery. Typical examples are Order booking, Purchase order issue, Goods and services receipt, Supplier invoice approval, Billing and invoicing and Order fulfilment.

In a chain, one loop's tail becomes the next loop's exception. A careless receipt shows up as a failed match two loops later. Improving one loop can therefore move the problem rather than solve it. The models in this chapter keep attention on the whole chain: where it is slowest, how its parts talk to each other, and where work waits and piles up.



Like terraced fields watered from the top: *whatever one terrace lets spill, mud and all, lands in the terrace below as its problem.*

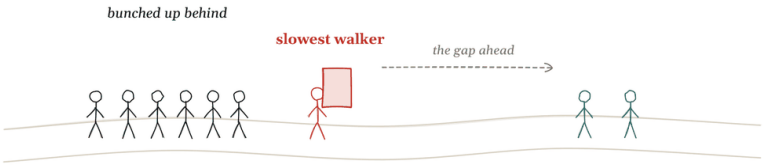
The Theory of Constraints

A chain can move no faster than its slowest link. Speeding up any other link adds nothing. It only makes the line in front of the slow one longer.

Eliyahu Goldratt set this out in The Goal (1984), a novel about a struggling plant. He gave five steps. Find the constraint, get the most from it, make everything else serve it, then add capacity to it. When it moves, start again.

In lead to cash, suppose orders spend most of their time waiting for Customer credit check. An executor that halves the time of Quote and pricing approval looks like progress. Yet the order reaches cash no sooner, because it now waits longer for a credit decision.

Use it when you Read the map and choose the next loop. Ask where cases in the chain wait longest, and start there. The common misreading is that the constraint is the busiest team, when it is the step that sets the pace of the whole chain.



Like a line of walkers on a narrow path: *the whole group arrives when the slowest walker does, and hurrying the others only bunches them behind.*

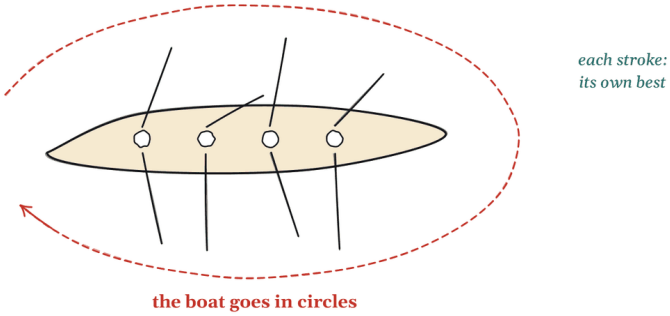
Local Optima

Each loop can do its own job well and the chain can still do badly. The best result for one part is often not the best result for the whole.

W. Edwards Deming called this suboptimisation in The New Economics (1993). He argued that a system works only when every part is managed for the aim of the whole. Goldratt made the same point in The Goal (1984). Local measures, he showed, push each department to look efficient while the plant loses money.

Consider a buyer in Purchase order issue judged on unit price. She picks the cheapest supplier, and her number improves. The cost reappears in Supplier delivery exceptions, where late and short deliveries now need chasing. No single loop report shows the trade.

At Design, write each loop's standard so that it includes what the next loop needs. Then put the chain's cost on the economics sheet beside the loop's own cost per accepted outcome. The common misreading is that local optima come from bad behaviour, when they come from honest people meeting the measures they are given.



Like a rowing crew where each rower perfects their own stroke out of time with the rest: *every oar works hard and the boat turns in circles.*

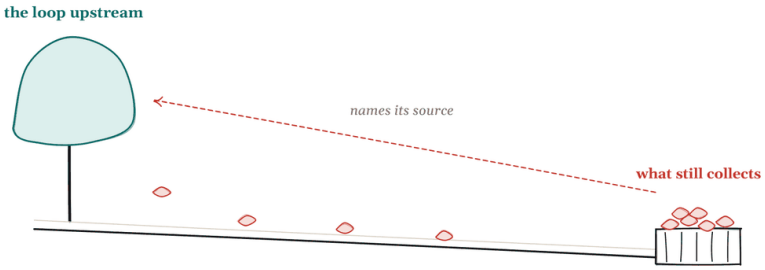
The Edges Name the Next Loop

Once a loop runs well, the cases it cannot handle stop being random. What is left tends to come from somewhere else. Read those leftovers, and they point to the loop that should go next.

This model is native to embedding practice. It lives in the map review, which reads the edges of every embedded loop before choosing what to read next. It also lives in the junction register, where each case sent to a person carries its reason.

At the manufacturer, supplier invoice approval was embedded first. Afterwards, 38% of its remaining exceptions came from missing goods receipts. The invoice loop could not fix that. The cause sat upstream, in Goods and services receipt, and the map review named that loop as a strong candidate.

Tag every exception with the loop it came from, and count the tags at each operating review. When one upstream source dominates, take it to the map review. The common misreading is that a loop′s leftover exceptions show its own limits, when they often show its neighbour′s.



Like leaves clogging a drain grate: *once the grate is clear of litter, what still collects there names the tree upstream to prune next.*

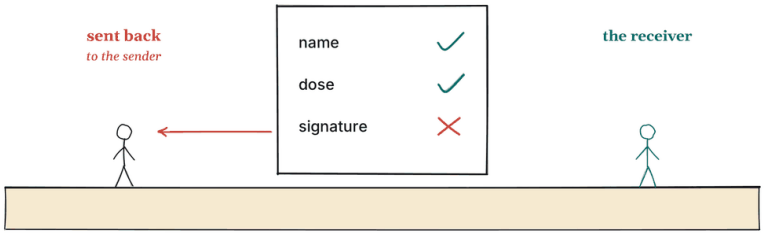
Handoffs as Contracts

When work passes between two loops, both sides need the same picture of what is being passed. Write it down as a promise. Spell out what the sender guarantees and what the receiver may assume. Say what happens when the promise is broken.

Bertrand Meyer called this design by contract in Object-Oriented Software Construction (1988). Each software routine states its preconditions and its guarantees. If a caller breaks a precondition, the fault is the caller's, and the routine does not have to guess.

Order booking hands Billing and invoicing the order lines, prices and billing triggers. If a price is missing, billing either stalls or invents one. A contract would say which fields must be present and valid at release. It would also say that an incomplete order goes back to booking.

Write the contract at Design, on the handoffs node of the Work Loop Canvas. Add its checks to the standard of both loops. The common misreading is that a contract slows the handoff, when it moves the check to the one place where the fault can be fixed cheaply.



Like a prescription at the pharmacy counter: *it is filled only when name, dose and signature are all present, and a missing field goes back to the doctor.*

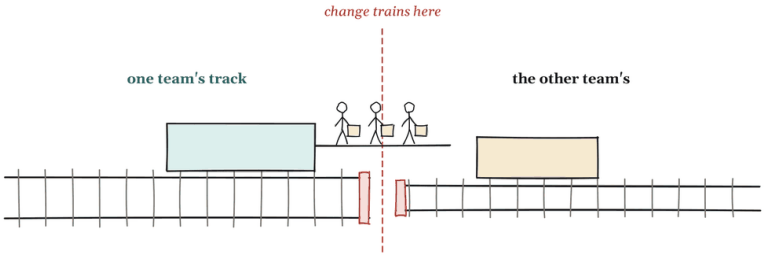
Conway's Law

Organisations tend to build systems that copy their own structure. Where two teams rarely talk, their systems meet awkwardly. The same happens to processes. The work tends to change hands wherever the company changes reporting lines.

Melvin Conway made the observation in "How Do Committees Invent?" (1968), published in Datamation. He wrote that a design mirrors the communication structure of the group that produced it. Software teams have tested and quoted the idea ever since.

At the manufacturer, order fulfilment ranked first on value but could not go first. Its rules were scattered across three plants. Each plant had its own way of handling the same order. The loop was really three loops, shaped by three management lines rather than by the order.

Use it at Map. Draw the lanes as they really are, and mark each handoff that only exists because of a reporting line. At Own, give the chain a single owner where its loops must scale together. The common misreading is that new software will fix the seams, when it usually copies them.



Like a railway that changes gauge at each border: *every country laid its own track, so passengers change trains wherever the map changes owner.*

Queues Compound

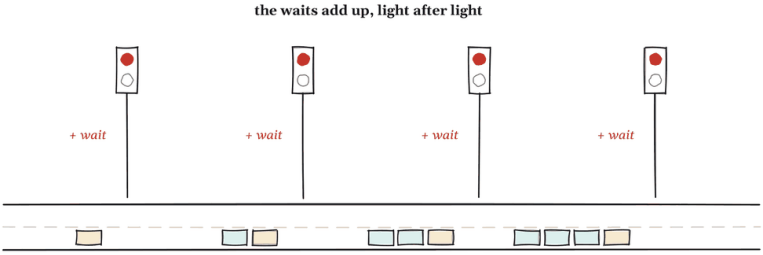
Every handoff has a queue in front of it, and the queues add up along a chain. As a team gets close to full, its waits grow much faster than its workload. Uneven arrivals make this worse, and one loop's uneven output is the next loop's uneven input.

John Little proved in 1961 that the number of cases in a system equals the arrival rate times the average time each spends inside. In the same year, John Kingman showed how waiting time rises steeply as utilisation nears one

hundred percent. Wallace Hopp and Mark Spearman, in Factory Physics (1996), traced how variability passes from station to station.

In procure to pay, a single invoice can wait at requisition approval, at the purchase order, at receipt and at matching. It then waits for the Payment run. If payments run weekly, an invoice approved just after a run waits nearly seven days more.

Use it at Strip. Add up the waits across the chain, not only inside one loop, and remove batch steps first. The common misreading is that a faster executor shortens the chain, when most of the elapsed time sits in waits between loops.



Like a road with traffic lights that are not in step: *the car stops at every one, and the small waits add up to most of the journey.*

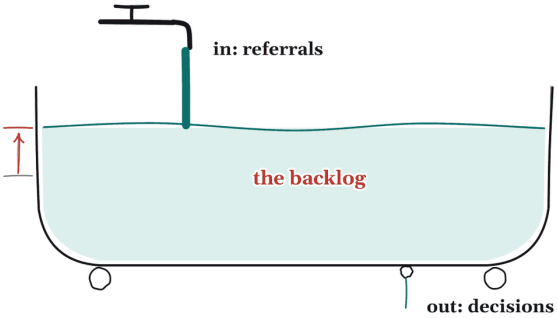
Stocks and Flows

Some things pile up and some things move. A backlog is a pile, and it changes only through what flows in and what flows out. If more arrives than leaves, the pile grows, however hard people work.

Jay Forrester built this view in Industrial Dynamics (1961). He showed that delays between stocks and flows make supply chains swing far more than demand does. John Sterman later taught the method to managers in Business Dynamics (2000).

In Supplier invoice approval, the exception queue is a stock. Cases flow in from the executor's referrals and flow out through people's decisions. At quarter-end, referrals rise while the same people are busy closing the books. The queue then grows for weeks, and payments to suppliers slip.

At Bound, set the executor's limits with the outflow in mind. Every case it refers must be one the junction staff can clear. At each operating review, watch the level of every queue between loops, not only the rate of cases handled. The common misreading is that a steady rate means a healthy loop, when a backlog can grow quietly behind it.



Like a bathtub: *the water level changes only by what the tap pours in and the drain lets out, so a faster tap fills it however busy the drain is.*

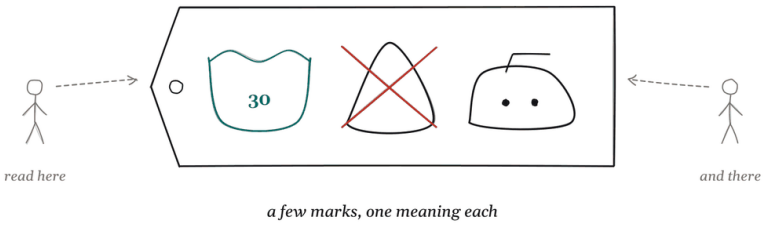
PART TWELVE

Across the Discipline

CXIII

Notation

A discipline needs a shared way of drawing its instruments, so that a loop drawn in one company can be read in another. This chapter defines the notation used for the Work Loop Canvas, the Work Loop Map and the process map. It is deliberately small. Every symbol stands for a concept defined in the areas, and nothing is drawn that does not carry meaning.



Like the care symbols on a clothing label: *a handful of small marks, each with one fixed meaning, read the same way in any country.*

States

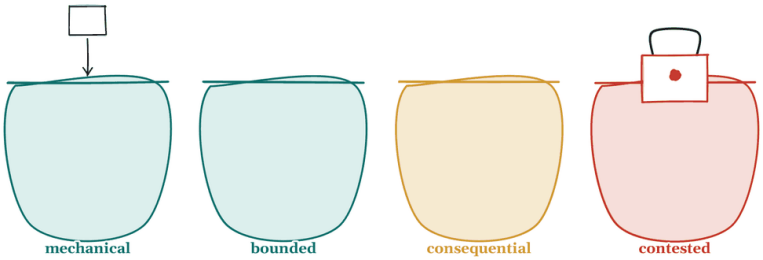
Every node of a canvas, and every document in the record set, is in one of three states. Blank means nothing has been written. Drafted means a proposal exists but nobody has signed it. Signed means a named person has accepted it, with a date. On drawings, blank is an outline in ink, drafted is filled in amber, and signed is filled in teal. Lines between signed nodes are drawn solid in teal. Lines where either end is only drafted are dashed in amber. The rule is simple: colour is earned by signature, not by effort.



Like a cheque: *blank, then filled in, then signed, and only the signed one can be cashed. Colour on a drawing is earned the same way.*

Decision classes

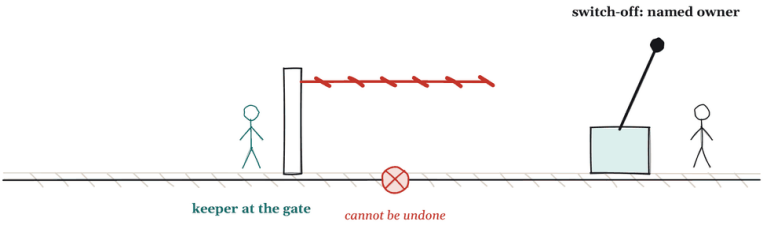
The four decision classes are drawn as bins. Mechanical and bounded decisions are teal, consequential decisions amber, and contested decisions red, with a lock on the bin while any contested decision is open.



Like a row of sorting bins: *routine items drop straight in, some wait for a check, and one bin stays padlocked while a dispute about it is open.*

Junctions and switches

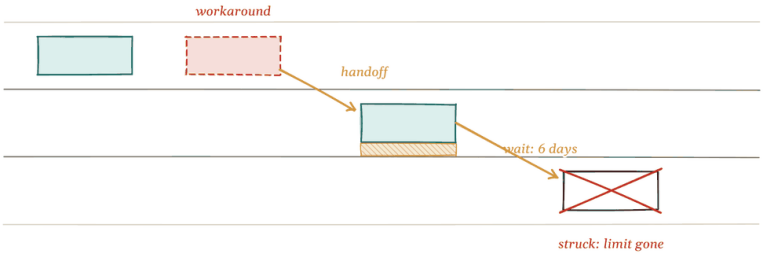
A junction is drawn as a gate with a person beside it. A red valve on a line means an action that cannot be undone and therefore requires a junction. The switch-off is drawn as a lever, and the person who holds it is named beside it.



Like a level crossing: *a keeper stands at the gate where a mistake cannot be undone, and one named person holds the lever that stops everything.*

Process maps

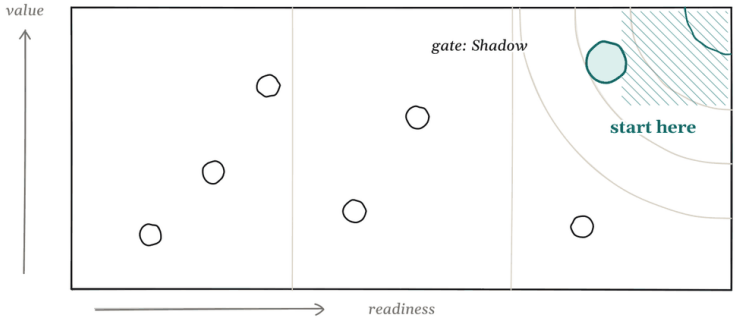
On process maps, steps are rectangles in the lane of whoever does them. A workaround, which is a step that compensates for a constraint, is drawn with a red dashed outline. Rework has an amber dashed outline. Waits are drawn as hatched amber blocks beneath the step they follow, with their duration. Handoffs between lanes are drawn as solid amber arrows. Struck steps are crossed through in red, with the constraint written beside them.



Like a running track seen from above: *each runner keeps a lane, the baton passes between lanes, and anyone stopped or detouring shows up at once.*

The map

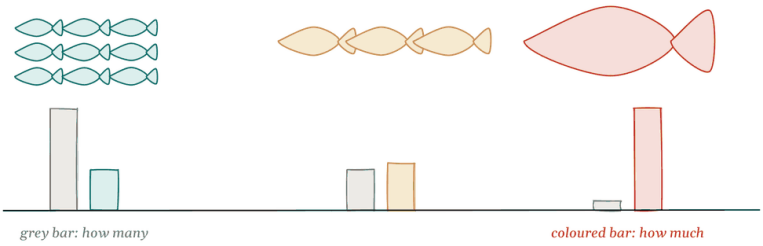
On the Work Loop Map, each loop is a circle placed by value to the core (vertically) and readiness (horizontally). The upper right region, where to start, is tinted teal. Handoffs between loops are lines. The loop currently in a lifecycle is drawn larger, with its gate written beside it.



Like a hiking map with pins: *each loop sits where its value and readiness put it, the near summit is shaded as the place to start.*

Distributions

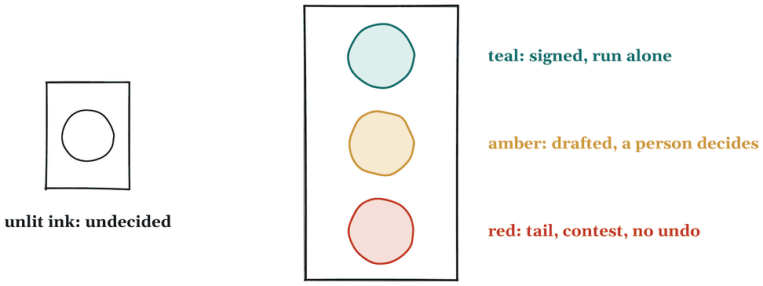
Distributions are drawn as paired bars for each band, grey for the share of cases and coloured for the share of value: teal for the body, amber for the middle and red for the tail.



Like a day's catch on the quay: *count the fish and weigh the fish, because a few big ones in the tail can outweigh the whole pile of small ones.*

Colour

The notation uses four colours, each with one meaning. Teal means signed, safe to run alone, or the body. Amber means drafted, waiting, or needing a person's decision. Red means a workaround, a contest, an irreversible action or the tail. Ink means everything not yet decided. Drawings should never use colour for decoration.



Like a traffic light: *green means go alone, amber means a person must decide, red means stop, and an unlit lamp means nothing is decided yet.*

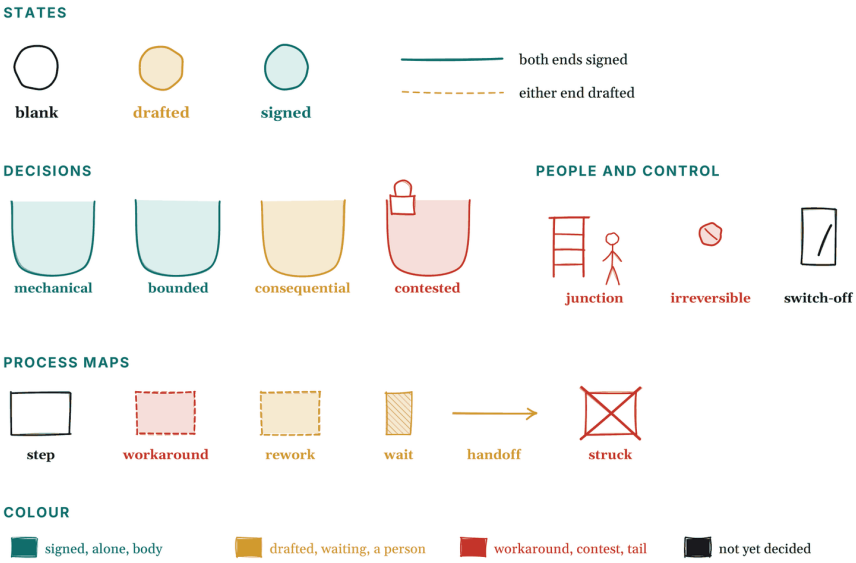


FIGURE 81 The notation. Three states earned by signature, four decision classes, the junction and the switch, the marks used on process maps, and the four colours with one meaning each.

CXIV

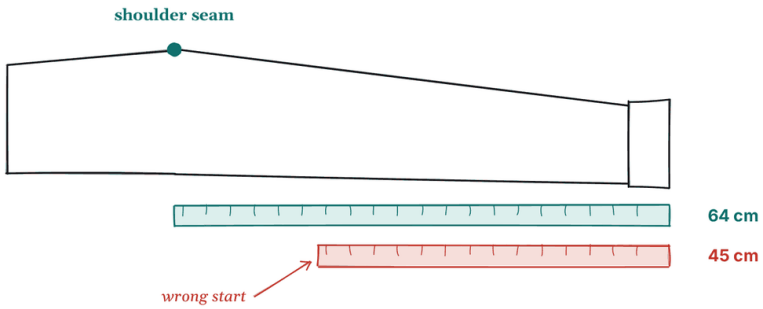
The Measures

This chapter gives the exact definition of every measure used in the discipline. Measures that sound alike are often calculated differently from one company to the next. Comparing them is then meaningless, and arguments about results become arguments about arithmetic. Each definition below states what is counted, how, over what period, and the most common way it is miscalculated.

Measure	Definition	Common miscalculation
Share of cases, by band	Cases in the band divided by all cases in the period	Bands chosen after looking at the data to flatter a story
Share of value, by band	Value of cases in the band divided by the value of all cases	Using list values rather than actual amounts
Concentration	Share of value carried by the top 5% of cases by value	Computed on a period too short to contain the tail
Value to the core	Score from 1 to 5, agreed in session, recorded with reasons	Scored by how impressive the AI would look
Readiness	Score from 1 to 5 on known rules, measurable outcome, available context, willing owner, settled decisions	Scored once and never updated
Volume	Cases arriving per period	Counting batches or documents instead of cases
Fully loaded cost per case	All costs of the loop in the period, including people at real cost, systems, rework and errors, divided by cases	Leaving out people in junctions and queues, or rework
Elapsed time	Time from a case's arrival to its accepted outcome, as a median and a 90th percentile	Starting the clock at first touch rather than arrival
Hands-on time	Minutes of work done on a case, summed across steps	Estimated from interviews rather than sampled
Flow efficiency	Hands-on time divided by elapsed time	Mixing averages of different case samples
Exception rate	Cases that left the normal path, divided by all cases	Excluding cases resolved informally
First-time-right	Cases accepted without rework, divided by all cases	Counting a case as right if the error was caught internally
Strip ratio		

Measure	Definition	Common miscalculation
	Steps struck divided by steps on the signed map	Counting struck steps without named constraints
Contradictions	Pairs of rules that cannot both be followed, found and decided	Recording them as noise rather than findings
Contested decisions open	Contested decisions logged and not yet settled	Reported as zero because nobody asked
Agreement, body and tail	Cases where the executor's result matched the accepted result, divided by cases, per segment	Reporting one overall figure
Bounded share	Cases, and separately value, inside the executor's signed limits, divided by all	Reporting cases only, never value
Standard met	Outcomes meeting the standard, divided by all outcomes, per segment	Grading a sample rather than every outcome
Cost per accepted outcome	Fully loaded cost in the period divided by outcomes meeting the standard, per segment	Dividing by cases processed
Override rate	Junction decisions where the holder overruled the executor's evidence, divided by junction decisions	Treating zero as success
Exception age	Age of the oldest case in the exception queue	Reporting the average instead of the oldest
Time per step	Elapsed weeks in each lifecycle step, per loop	Measuring only the build, not the decisions

Two rules apply to every measure. Report the body and the tail separately wherever a measure can differ between them. And never compare a measured figure with an estimated one.



Like a tailor's tape: *measure a sleeve from a different starting point and the number changes, so every measure states exactly where counting starts.*

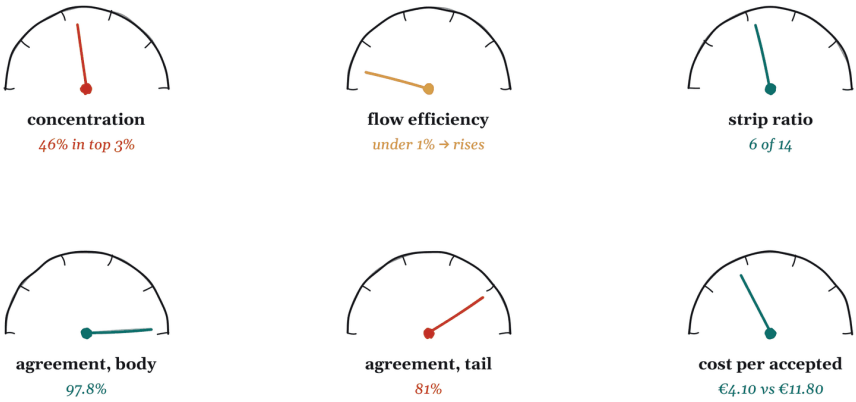
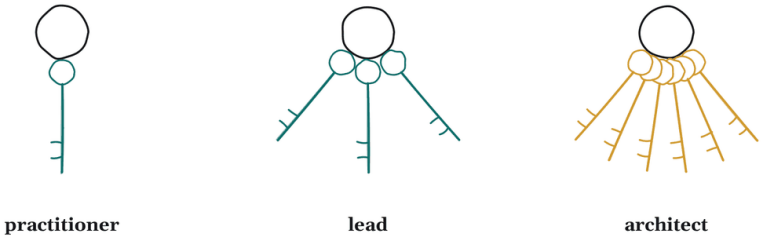


FIGURE 82 The measures that matter, read on one panel: concentration, flow efficiency, strip ratio, agreement for body and tail, cost per accepted outcome and override rate. Activity counts are left off on purpose.

CXV

The Competencies

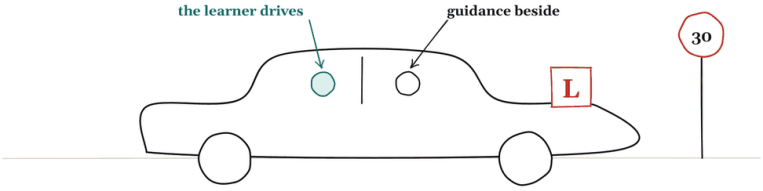
A discipline is held by people. This chapter describes what a business engineer needs to know and be able to do, in three levels. The levels describe what a practitioner can be trusted with, not years of service.



Like key rings of different sizes: *each level is trusted with more doors, and the ring grows with what someone can be trusted to open, not their years.*

The practitioner

A practitioner can run the steps of a lifecycle for a loop under guidance. They can draw a process map with operators, walk real cases through it, and measure a baseline. They can gather rules from documents, systems and interviews into a register, and find contradictions. They can sort decisions into classes and recognise a contested one. They can prepare every document in the record set to the standard required for signature. They know what each gate requires and who signs it.

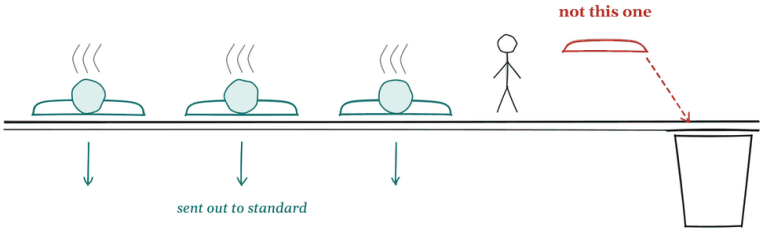


Like a learner driver with an instructor beside them: *they drive every part of the route themselves, under guidance, and know every rule of the road.*

The lead

A lead can run a lifecycle end to end, and run several at once. They can read a company's loops, facilitate the scoring of the map, and recommend a first loop

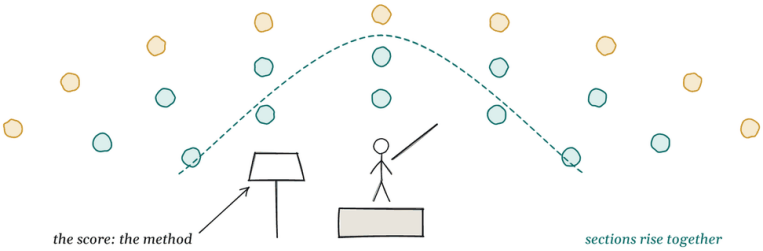
the sponsor will accept. They can design junctions with real evidence and reversal, write a standard that grades outcomes, and set bounded limits from shadow evidence. They can build an economics sheet that finance will sign. They can lead the contested-decision conversation with senior people. And they can tell when a loop should not proceed.



Like a head chef at the pass: *several dishes going at once, each sent out to standard, and the judgement to bin one that should not go.*

The architect

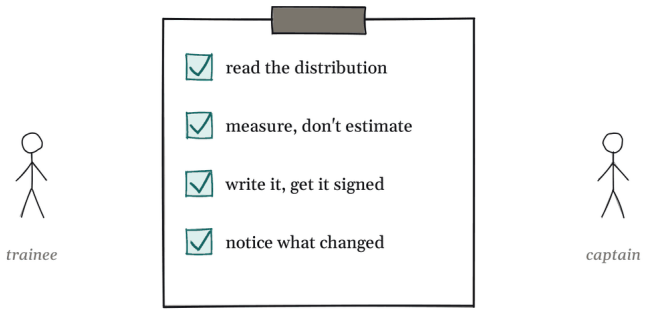
An architect holds the method for a company. They maintain the instruments and the notation, train practitioners and loop owners, and run the business engineering function. They read the map as a whole, see which chains must scale together, and shape the role change plan with the sponsor. They connect the inside view of loops with the outside view of the business model. They also connect the record set with the company′s governance obligations.



Like a conductor holding the score: *they keep the method, train the players, hear the whole orchestra and know which sections must rise together.*

What all three share

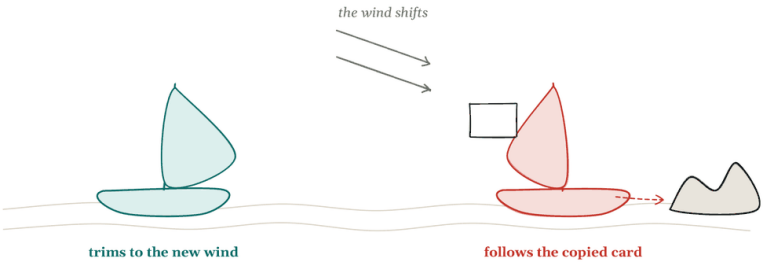
Every level shares four habits. Read the distribution before designing anything. Measure rather than estimate. Write down decisions and get them signed. And notice when the world has changed. These are the habits of the method loop, practised on the work of embedding.



Like a pilot's pre-flight checklist: *the trainee and the captain run the same four checks every time, however senior they are.*

What they need to know about the foundations

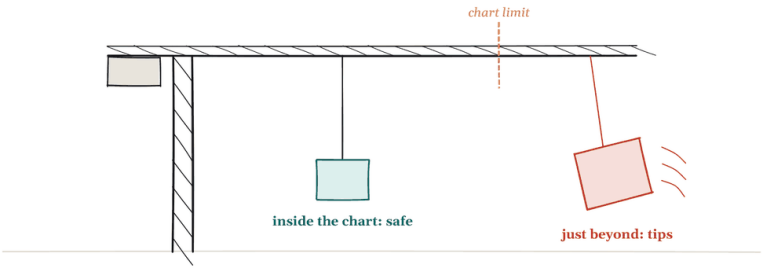
Every level also needs a working grasp of the foundations in Part One. That does not mean mathematics. It means understanding why rules are precise but incomplete, and why learned models are flexible but come without guarantees. It means seeing how the two are combined, and why control requires a model and a closed loop. A practitioner who understands those four ideas will make sound design choices in situations no book anticipated. One who does not will copy templates.



Like a sailor who understands the wind: *when it shifts, they trim for the new wind, while one following a copied route card drifts onto the rocks.*

What they need to know about the technology

A business engineer does not need to build an executor. They do need to know its parts well enough to make the decisions those parts require: the model, instructions, context, actions, platform and grader. They should also know the uneven frontier of what current executors do well and badly, and how that frontier is tested. A business engineer who cannot tell what an executor can and cannot do will either over-trust it or design it out of the work.



Like a site manager reading a crane's load chart: *no need to build the crane, but lifts inside the chart are safe and a load just beyond it tips.*

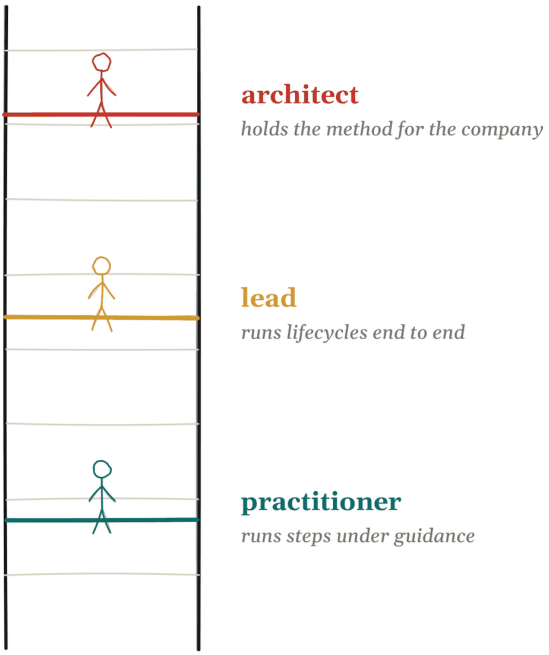
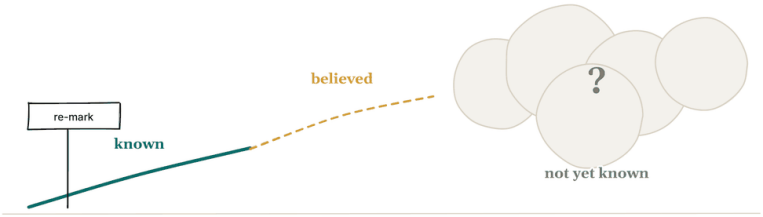


FIGURE 83 The competencies. Three rungs of practice: the practitioner runs steps under guidance, the lead runs whole lifecycles and several at once, the architect holds the method for the company. All three share the habits of the method loop.

CXVI

The Evidence Base, as Read in 2026

A body of knowledge should say what is known, what is believed, and what is not yet known. This chapter is dated, and it should be reread as the evidence changes. The concepts in the areas are designed to outlast the specific findings below.



Like a hill trail: *a marked path where the ground is known, a faint track where it is believed, then fog, and the markings are redone as the ground shifts.*

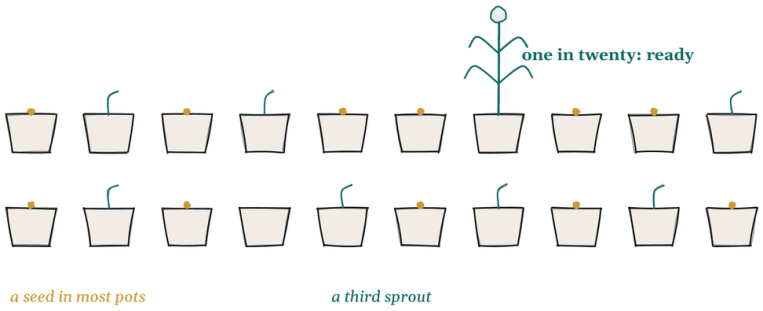
What is reasonably established

Most companies use AI somewhere, and few have embedded it into core processes at scale. McKinsey's 2025 survey found 88% of organisations using AI in at least one function. Only about a third had begun to scale it, and no more than 10% were scaling agents in any single function. MIT's 2025 study found that only about 5% of custom enterprise AI tools reached production.

AI assistance raises productivity on many tasks, unevenly. Studies of customer support and of consultants found substantial gains on tasks inside the frontier of what models do well, and losses on tasks just outside it, where people trusted answers they should have checked.

People supervising automated systems tend to over-trust them and lose skills over time. Research on automation stretching back to the 1980s documents this consistently.

Process documents diverge from how work actually runs. Process mining of event logs has shown this repeatedly across industries.



Like a seed tray: *almost every pot has a seed in it, about a third have sprouted, and only one in twenty is a plant ready for the ground.*

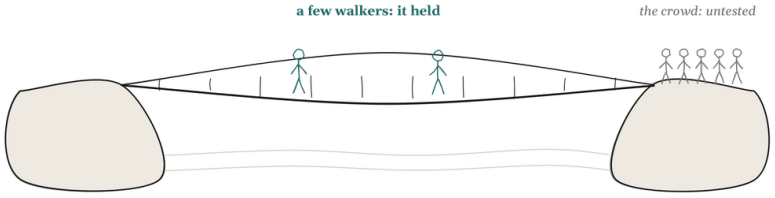
What is believed but less tested

That separating decisions from process, and sorting them into classes, is the main lever for embedding executors safely. Practitioners report it, and early research on combining agents with explicit process rules supports it. Large controlled comparisons are still rare.

That combining learned models with explicit rules and verification makes systems more reliable than either alone. Demonstrations in mathematics and early research on agents that follow explicit process rules support it. Systematic evidence from enterprise deployments is still thin.

That value arrives late, at bounded running and after, rather than at the pilot. The shape is consistent with the survey evidence on stalled programmes. It has not been measured directly across many companies.

That the second and later loops take much less time than the first. Practitioners report it, and published longitudinal data is thin.

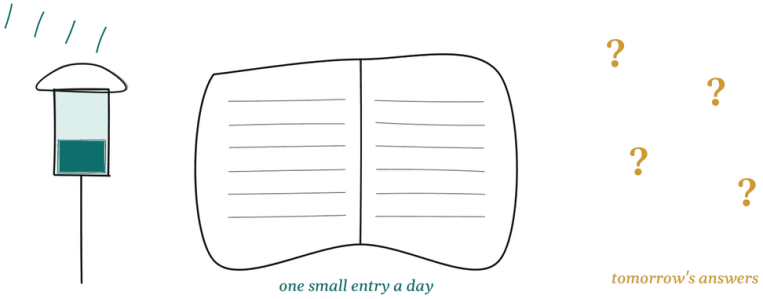


Like a new footbridge crossed by a few walkers: *it held every time so far, but nobody has yet run the full load test with a crowd.*

What is not yet known

How quality changes over years in loops where the model is replaced several times. How the skills needed at junctions can best be rebuilt when routine cases no longer teach them. And how regulators will interpret human oversight requirements for executors that decide in practice, as enforcement begins.

These open questions are the research agenda of the discipline. Practitioners who record their lifecycles, with their measures, are generating the evidence that will answer them.



Like farmers keeping weather diaries: *each day's entry is small, but together the records will answer questions nobody can answer yet.*

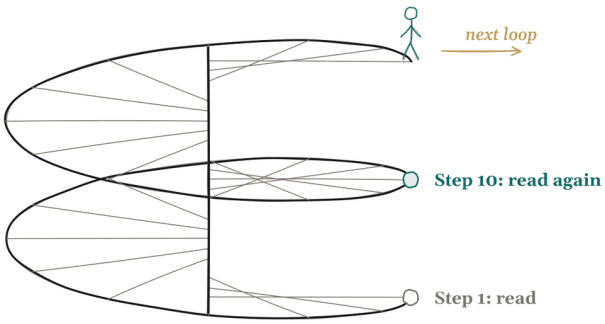
The Cycle Closes

The lifecycle ends where it began. Step 10's map review is the same document that Step 1 produced, one cycle later. Its last act is to choose the next loop to read.

That is not a stylistic choice. It reflects how companies learn. The first reading of a map is a set of guesses: about which loops matter, where their value sits, how ready they are. Each cycle replaces some guesses with evidence. The distribution of a loop's cases, read once at Step 1, is read again at Step 10 with the body handled and the tail exposed. Readiness scores move as neighbouring loops are embedded. The sponsor's view of where to start becomes a view of what the company has learned it can do.

The steps in between follow a pattern that is older than any of the technology involved. Read the real world before abstracting it. Reduce what you see to the few rules and decisions that matter. Name them, so people can find them again. Test them against real cases. Let people build judgement from the results. Then go back to the real world, which will have changed. Business engineering is that loop, practised deliberately. The embedding lifecycle is that loop, applied to the work of putting an executor that decides inside a company's core.

The middle is where most programmes stall today. It will not stay that way. Companies that learn to cross it once will cross it again, faster, and the distance between them and the companies still running pilots will grow with every loop. The difference will not be the models, which everyone can buy. It will be the sequence, followed and signed.



Like a spiral staircase: *each turn comes back above the same spot, one floor higher, and from each landing you choose where the next turn leads.*

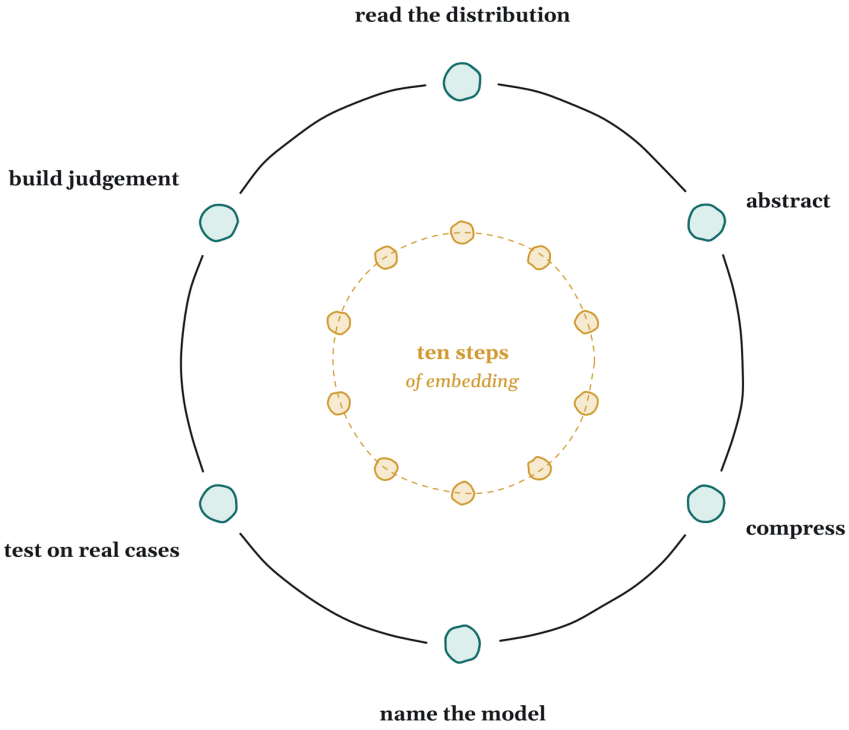


FIGURE 84 The cycle closes. Business engineering read the real distribution, abstracted and compressed it, named the model, tested it against real cases and built judgement. The lifecycle applies that loop to embedding work.

Epilogue: The New Business Engineer

Every generation of managers redesigned work around the executor it had. Taylor wrote instructions for workers. Ford built a line around them. Sloan gave outcomes owners. Toyota gave every worker a cord to stop the line. Reengineering stripped steps, enterprise software encoded rules, and software robots clicked through screens. Each of those executors followed instructions.

This one decides. That does not make the century's lessons obsolete. It makes all of them necessary at once, in one design, around one outcome. The standard, the handoff, the owner and the stop cord each have a place. So do the stripped steps, the written rules and software that acts. The two new things are the context an executor reads and the decisions it may take. Holding all of that together is a job, and it needs a person.

The role

The business engineer is that person. They are not a data scientist, though they work with them, and not a process analyst, though they map processes. Nor are they a consultant, though they may be one. They understand a business well enough to ask where its value really sits, which of its rules are real, and which of its decisions were never agreed. And they understand the technology well enough to know what an executor can and cannot be trusted with. They carry loops through the lifecycle, prepare what people sign, and keep the record set that lets anyone see why a loop behaves as it does.

The role has a ladder. A practitioner runs steps under guidance. A lead runs whole lifecycles, and several at once. An architect holds the method for a company: the map, the instruments, the notation, the training, and the connection between the inside view of loops and the outside view of the business model.

The function

Companies that embed AI seriously will give the role a home, usually by the fifth or sixth loop. It will be a small team that owns the method rather than the loops, because owners answer for loops and the team answers for the method being followed. It will sit wherever the company keeps its strongest operators of change, whether in finance, operations, transformation or technology.

Where it sits matters less than its mandate: to run the lifecycle, keep the instruments, and hold the map.

The decade

The models will keep improving, and every company will be able to buy them. What will separate companies is something they cannot buy: the capacity to move loop after loop from a pilot into the core, with each step decided and signed. That capacity compounds. The first loop teaches the steps, and the second takes half as long. By the tenth, a company knows its own work better than it ever has. It can change that work whenever a better executor arrives.

The work is less glamorous than the demos. It means reading distributions, stripping old workarounds and sorting decisions. It means holding the meeting about the decision nobody ever agreed on, and measuring the baseline before claiming the saving. But it is the work that turns AI from something a company tries into something it runs on. The people who learn to do it are the new business engineers. This book was written for them.

Appendix A: The Lifecycle on One Page

Phase	Step	Question	Signed	Signer	Gate
Read and Map	1. Read	Where does value sit, and which loop first?	Work Loop Map	Executive sponsor	First loop chosen, owner named
Read and Map	2. Map	How does the work run today, at what cost?	Process map, baseline sheet	Process owner	Map recognised, baseline measured
Redesign	3. Strip	Which steps exist only for old constraints?	Strip register	Process owner	Every step struck or kept, constraint named
Redesign	4. Sort	Which rules apply, and who decides what?	Rule register, decision register, contested-decision log	Process owner; leaders for contested decisions	No contested decision open
Redesign	5. Design	What is the loop, with junctions and standard?	Work Loop Canvas, junction register, standard	Loop owner	G0 Mapped, G1 nodes signed
Prove	6. Shadow	Does it match people without acting?	Shadow report	Loop owner and grader	G1 Shadow
Prove	7. Bound	Can it act alone inside written limits?	Gate review record	Control owner	G2 Bounded
Prove	8. Own	Are people named, is the cost measured?	Loop charter, economics sheet	Loop owner; finance	G3 Owned
Run and Scale	9. Core	Does it run in the business, watched?	Operating review, switch-off plan	Loop owner	G4 Core
Run and Scale	10. Scale	What changes across the map, and what next?	Map review, role change plan	Executive sponsor	Next Read begins

Appendix B: Who Signs What

Each role signs at the steps where its authority is needed, and nowhere else. "Consulted" means the role reviews the document before it is signed.

Role	Signs	Consulted
Executive sponsor	Work Loop Map (1), contested decisions (4), map review and role change plan (10)	Economics sheet (8)
Process owner, later loop owner	Process map and baseline (2), strip register (3), rule and decision registers (4), canvas, junction register and standard (5), shadow report (6), loop charter (8), operating reviews and switch-off plan (9)	Every step
Leaders who share a contested decision	The contested-decision log entries they settle (4)	Design (5)
Grader	Shadow report (6)	Standard (5), operating reviews (9)
Control owner	Gate review record (7), each widening of scope (9)	Junction register (5)
Internal audit	None	Gate review (7), operating reviews (9)
Finance (planning and analysis)	Economics sheet (8)	Baseline (2)
Business engineer	None; prepares every document and runs the steps	Every step

Appendix C: Entry and Exit Checklists

Each step starts when its entry conditions are true and ends when its exit conditions are true. A step that cannot meet its exit conditions loops back, or waits.

Step	Enter when	Leave when
1. Read	A sponsor has asked, and the people who know the work have time	Loops listed and placed; candidate distributions read; first loop chosen; owner agreed; map signed
2. Map	Loop chosen, owner agreed	Map drawn with operators and walked through with real cases; workarounds marked; baseline measured on a real sample; both signed
3. Strip	Map and baseline signed	Every step struck or kept; every constraint named; strip ratio stated; register signed
4. Sort	Strip register signed	Rules gathered from documents, systems and people; contradictions decided; every decision classed; every contested decision settled and signed
5. Design	Registers signed, contested log closed	Context with keepers; actions with undo answers; junctions with who, what evidence, how reversed; switch-off holder named; exception queue with owner and time limit; handoffs agreed; standard written; G1 nodes signed
6. Shadow	G0 reached, G1 nodes signed	Several weeks of parallel running; agreement on body and tail separately; every disagreement explained; changes recorded; grader calibrated; report signed
7. Bound	G1 passed	Limits written from evidence; G2 nodes signed; permissions match actions; switch-off tested; weeks of bounded running graded; gate review signed
8. Own	G2 passed	Every role filled by a named person; changes to people's work agreed; learning path for new staff; cost per accepted outcome measured over a cycle, body and tail separately; charter and economics signed
9. Core	G3 passed	Scope widened in recorded bands; drift checks running; operating reviews held and signed; switch-off tested on schedule; all twelve nodes signed
10. Scale	A loop at G3 or G4	

Step	Enter when	Leave when
		Handoffs and exceptions read for the next loop; distribution re-read; map updated; next loops chosen; role change plan signed

Appendix D: Foundational Reading

The foundations in Part One draw on a small number of works that repay reading in full. They are listed in the order the argument uses them.

Work	Why it matters to the discipline
George Boole, *An Investigation of the Laws of Thought* (1854)	Reasoning written as algebra: the origin of the symbolic tradition
Claude Shannon, *A Symbolic Analysis of Relay and Switching Circuits* (1938)	Boole's algebra becomes the logic of machines
Kurt Gödel, *On Formally Undecidable Propositions* (1931)	No consistent rule system of sufficient power is complete
Michael Polanyi, *The Tacit Dimension* (1966)	"We can know more than we can tell": why rules in people's heads resist capture
Douglas Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid* (1979)	Formal systems, isomorphism, levels and strange loops
Norbert Wiener, *Cybernetics* (1948)	Feedback and control in systems that act
W. Ross Ashby, *An Introduction to Cybernetics* (1956)	The law of requisite variety
Roger Conant and W. Ross Ashby, "Every Good Regulator of a System Must Be a Model of That System" (1970)	Why control requires a model
Herbert Simon, *The Sciences of the Artificial* (1969)	Design as a science of the artificial; bounded rationality
Daniel Kahneman, *Thinking, Fast and Slow* (2011)	Two modes of thought, a useful analogy for neural and symbolic components
Artur d'Avila Garcez and Luís Lamb, "Neurosymbolic AI: The 3rd Wave" (2023)	The case for combining learning and reasoning
Nassim Nicholas Taleb, *The Black Swan* (2007)	Normal and heavy-tailed worlds

Appendix E: Areas Against Steps

The matrix shows where each knowledge area is used along the embedding lifecycle. A filled mark means the area is central to the step. An open mark means the area contributes.

Area	Read	Map	Strip	Sort	Design	Shadow	Bound	Own	Core	Scale
Distributions	●	○			○	●	●	○	○	●
Value and the map	●									●
Work and process	○	●	●	○	○				○	○
Rules and decisions		○	○	●	●	○	●		○	
The executor	○				●	●	●		○	
Control and acceptance			○	○	●	●	●	○	●	
People and accountability	○	○		○	○	○	○	●	○	●
Economics	○	●						●	●	○
Governance	○			○	○	○	●	●	●	●

Appendix F: The Blank Canvas

The twelve nodes, in fill order, with the question each one asks and the test for when it is signed. A node is Blank until something is written, Draft once it is written, and Signed once it is agreed and owned by a named person.

#	Node	The question	Signed when
1	Outcome	What does one accepted result look like, and who receives it?	Someone outside the team can tell an accepted result from a rejected one.
2	Compensations	Which steps exist only to work around a constraint that no longer applies?	Every remaining step is required by the outcome itself.
3	Rules	Where does each rule live today, and who owns it?	Every rule has an owner and a date, and contradictions are listed.
4	Context	What must the executor read to act well?	Each source has a keeper, and the executor can reach it with the right permissions.
5	Actions	What can the executor change, and in which systems?	Every write is listed with its system, its permission and whether it can be undone.
6	Decisions	Sort every decision: mechanical, bounded, consequential or contested.	Contested decisions are at zero.
7	Junctions	Where does a person decide, what do they see, and how is an action reversed?	Every irreversible or high-cost action passes a named junction.
8	Exceptions	Where do cases the loop cannot handle go, and who works that queue?	The queue has an owner, a time limit and a weekly review.
9	Handoffs	What does this loop receive, and what does it pass on?	Every handoff names the state passed and who resolves a mismatch.
10	Standard	How is every outcome graded?	Every outcome is graded against a written standard.
11	People	Who answers for the loop, and whose work changes?	Named people are in place before production.
12	Economics	What does one accepted outcome cost, compared with the measured baseline?	Cost per accepted outcome is below baseline, measured rather than estimated.

The Scale Gates and the nodes each one requires:

Gate	What the loop may do	Nodes that must be signed
G0 Mapped	Nothing yet; described, not agreed	All twelve at Draft or better
G1 Shadow	Run beside today's process with no writes; grade every outcome	Outcome, Compensations, Rules, Context, Standard
G2 Bounded	Act alone on mechanical and bounded decisions, junctions live	Adds Actions, Decisions, Junctions, Exceptions
G3 Owned	Run in production below the measured baseline	Adds People, Economics
G4 Core	Write to systems of record and hand off to other loops	All twelve, including Handoffs

Appendix G: The Insurer's Claims Loop, Filled

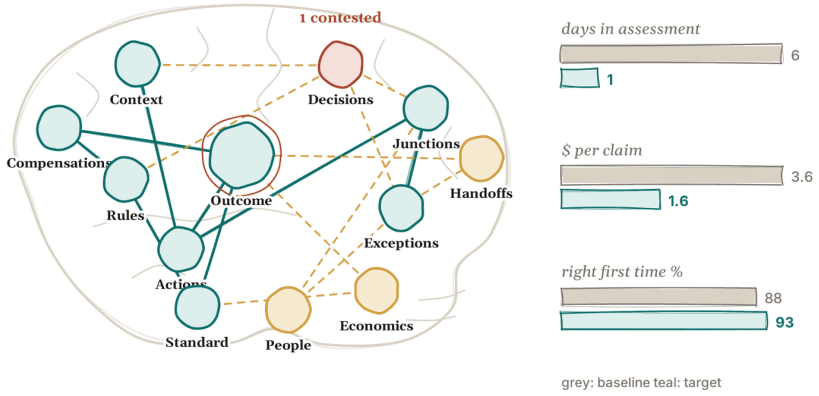


FIGURE 85 A Worked Run: Claims Assessment

The claims assessment loop of the composite general insurer, as filled at the end of its first two sessions. The firm and its numbers are illustrative. The status of each node is shown in brackets.

1. Outcome [Signed]. A claim assessed, with liability and amount decided and a rationale traceable to the rules, ready for settlement. About 4,000 claims a month. Distribution: about 80 percent of claims are under $5,000 and carry about a quarter of the money paid; the largest 3 percent carry about two-fifths. Assessment takes 6 of the claim's 11 days today. Received by the settlement team.

2. Compensations [Signed]. Struck: the overnight re-score, kept because the rating system once ran only at night; a second review of every claim under $2,000, kept to catch re-keying errors that no longer happen; the paper file transfer from intake. Kept: site inspection for structural damage.

3. Rules [Signed]. 38 rules: 21 in the claims manual, 9 in rating-engine settings, 8 only in senior adjusters' heads. 7 contradictions, for example water

damage covered after 30 days unoccupied in the manual and 45 in the engine. Owner: Head of Claims Policy.

4. Context [Signed]. Policy and endorsements; claim notes and photos; repair estimates; prior claims on the policy; supplier price list, kept by procurement and refreshed monthly. Inspection reports arrive scanned and are converted by operations on arrival.

5. Actions [Signed]. Reads policy administration, the claims system and the document store. Writes to the claims system (reserve, assessment note) and the supplier booking system. Reserve changes can be undone. A confirmed supplier booking cannot.

6. Decisions [Draft]. Mechanical 4: cover in force, deductible, duplicate check, document completeness. Bounded 4: repair or replace under $5,000; reserve setting within range. Consequential 2: declining a claim; fraud referral. Contested 1: goodwill payments on borderline cover, sent to the claims committee.

7. Junctions [Signed]. Declining a claim: senior adjuster, sees the rule trace and evidence, can reopen within 30 days. Any amount over $15,000: team lead approves. Fraud referral: investigations analyst. The claims operations lead can pause the loop.

8. Exceptions [Signed]. Cases outside the rules or missing evidence go to the adjuster queue. Owner: assessment team lead. Limit: 48 hours. A weekly review turns repeated exceptions into rules.

9. Handoffs [Draft]. Receives a validated claim file from intake. Passes an assessed claim with amount and rationale to settlement, and suspected fraud to investigations. Mismatch owner with intake not yet agreed.

10. Standard [Signed]. Accepted when the amount is within policy terms, the rationale traces to the rules and the customer letter is correct. Graded on every claim: a rules engine checks terms, a model checks the rationale, a 2 percent human audit calibrates the model.

11. People [Draft]. Owner: Head of Claims Operations. Junction holders named for all three junctions. Adjusters move from assessing every file to working junctions and exceptions. Open: how junior adjusters learn now that routine files no longer reach them.

12. Economics [Draft]. Measured baseline: $3.60 handling cost per claim, 6 days in assessment, 88 percent right first time. Target: under $1.60, 1 day,

93 percent. Run cost includes junction time and the exception queue; not yet measured in shadow.

Gate: G1 Shadow. The loop can run beside today's process now. It cannot act alone until the goodwill decision is settled, which moves Decisions to Signed. It cannot reach Owned until the shadow run measures its economics and the learning path for junior adjusters is agreed. It cannot reach Core until the intake handoff has an owner.

After the committee settled the goodwill question, the decision became a bounded rule with a new junction: goodwill payments up to a set amount, on claims within thirty days of the cover lapsing, approved by a team lead. With Decisions signed, the loop qualified for Bounded.

Appendix H: The Manufacturer's Invoice Loop in Numbers

The composite manufacturer's supplier invoice loop, from Read to the first map review. All figures are illustrative and set within published ranges.

The distribution read at Step 1

Invoice value	Share of invoices	Share of spend
Under €5,000	81%	14%
€5,000 to €50,000	16%	40%
Over €50,000	3%	46%

Baseline and result

Measure	Baseline (Step 2)	Measured at Own (Step 8)
Invoices per month	7,400	7,400
Cost per accepted invoice, fully loaded	€11.80	€4.10 (body €2.30, tail €14.60)
Days from receipt to payment-ready	16	3.4
Right first time	71%	93%
Invoices raising an exception	24%	9%
Steps in the loop	14	8
Clerks on the loop	9	4

The strip register (Step 3)

Step	Status	Constraint
Print and scan PDF invoices	Struck	Scanning desk predated email intake
Budget owners sign printed batches on Fridays	Struck	Approvals were once on paper
Weekly batch posting	Struck	The ERP posting job once ran weekly
Second clerk re-checks invoices under €1,000	Struck	An error spike years ago, long resolved

Step	Status	Constraint
Clerks chase goods receipts by email	Struck	Receiving teams could not see waiting invoices
Manual duplicate check against a spreadsheet	Struck	Duplicate rule now applied by the ERP
Treasury call-back on changed bank details	Kept	Payment fraud risk still applies
Match invoice to order and receipt	Kept	Required by the outcome

The decision register (Step 4)

Class	Count	Examples
Mechanical	8	Duplicate detection; match within tolerance; tax-code validation
Bounded	5	Cost-centre coding of small non-order invoices; partial delivery within limits
Consequential	3	Invoices over €50,000; changed bank details; non-order invoices over €5,000
Contested	1, settled	Paying a disputed invoice to protect a single-source supply: allowed up to €25,000 with the procurement director's signature, logged; above that, the CFO

The junction register (Step 5)

Junction	Who decides	What they see	How it is reversed
Invoices over €50,000	Financial controller	Matched order, receipt, contract price, supplier payment history	Hold before payment run; credit note if paid
New or changed bank details	Treasury analyst	Change request, call-back record on a separately held number	Payment blocked until verified
Non-order invoices over €5,000	Budget owner of the cost centre	Invoice, supplier record, budget position	Hold and return to supplier
Switch-off	Head of accounts payable	Grading dashboard, exception queue	All invoices to the clerks' queue within an hour

Shadow and Bound (Steps 6 and 7)

Measure	Value
Invoices in shadow	9,800 over six weeks
Agreement under €5,000	97.8%
Agreement over €50,000	81%
Rules added after shadow	2
Bounded scope	Order-backed, under €5,000, within 2%, supplier paid for six months or more: 74% of invoices, 12% of spend
Bounded invoices meeting the standard	98.6%
Switch-off test	Cases returned in 20 minutes, none lost

Three signed documents

The figures below show three of the loop's documents as they were signed: the contested-decision log entry from Step 4, the gate review record from Step 7, and the economics sheet from Step 8.

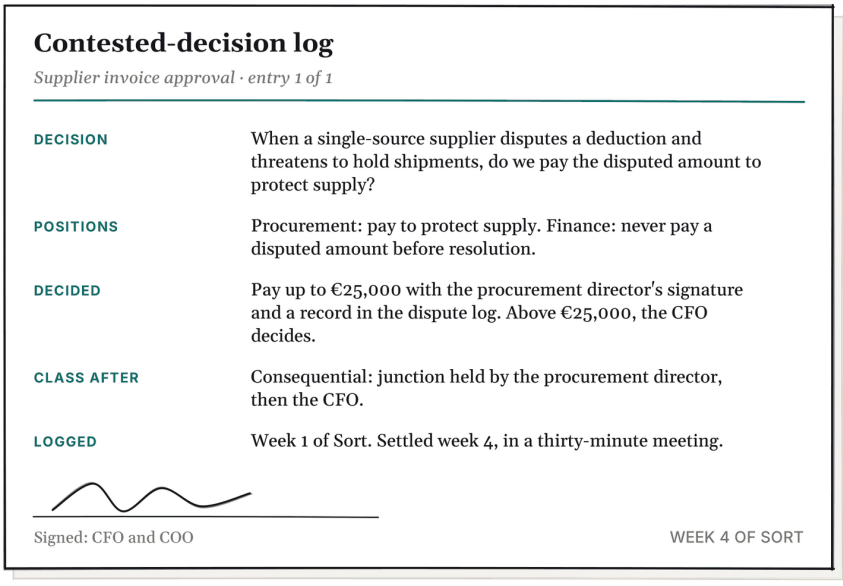


FIGURE 86 The contested-decision log entry, as signed at Step 4.

Gate review record: G2 Bounded

Supplier invoice approval

SCOPE ALONE	Order-backed invoices under €5,000, within 2% of order price, from suppliers paid for six months or more. 74% of invoices, 12% of spend.
EVIDENCE	Four weeks bounded: 98.6% met the standard. Exception queue cleared within two days every week.
PERMISSIONS	Cut to the actions node. Pilot account able to edit supplier records removed.
SWITCH-OFF	Tested week 1: invoices back with clerks in 20 minutes, none lost.
AUDIT NOTE	Link the dispute log to the payment record before Own.

Signed: Financial controller

WEEK 4 OF BOUND

FIGURE 87 The gate review record, as signed at Step 7 by the control owner.

Economics sheet

Supplier invoice approval · eight weeks across a quarter-end

BASELINE	€11.80 per invoice, fully loaded, measured at Map on 1,200 invoices.
COST LINES	Platform and model usage; grading and 3% audit; junction time; exception queue; rework; adoption.
RESULT	€4.10 per accepted invoice. Body €2.30. Tail €14.60.
TIME, QUALITY	16 days to 3.4 days, receipt to payment-ready. Right first time 71% to 93%.
ANNUALISED	About €680,000 a year on 7,400 invoices a month, before early-payment discounts.

Signed: Planning and analysis lead

WEEK 8 OF OWN

FIGURE 88 The economics sheet, as signed at Step 8 by finance.

Appendix I: One Loop Through Every Area

The composite manufacturer's supplier invoice loop, read through each of the nine areas. The table shows how the same loop looks from each kind of knowledge, and how the areas depend on one another.

Area	What it showed at the manufacturer
Distributions	7,400 invoices a month. 81% under €5,000 carried 14% of spend; 3% over €50,000 carried 46%. A routine body and a heavy tail
Value and the map	One of 34 loops. Value four, readiness three. Chosen first over order fulfilment, which mattered more but whose rules were scattered across three plants
Work and process	Fourteen steps in six lanes. Twenty-three minutes of work took sixteen days. Six steps struck as compensations: a strip ratio of 43%
Rules and decisions	58 rules and 9 contradictions, the largest a 2% tolerance in the system against 5% in the policy. 17 decisions: 8 mechanical, 5 bounded, 3 consequential and 1 contested, settled in thirty minutes
The executor	Contract prices for a third of suppliers existed only as PDFs; the top two hundred were loaded before shadow. A pilot account able to edit supplier records was removed
Control and acceptance	Three junctions and a tested switch-off. Shadow agreement 97.8% on the body and 81% on the tail. Bounded scope: 74% of invoices, 12% of spend
People and accountability	Named owner, junction holders, exception owner, grader and adoption owner. Nine clerks became four on the loop; three moved to supplier onboarding and one to treasury
Economics	€11.80 per invoice before; €4.10 per accepted invoice after, with the body at €2.30 and the tail at €14.60. About €680,000 a year
Governance	Gate review signed by the financial controller, with internal audit's note linking disputes to payments. Monthly operating reviews recorded a drift event and the rule it produced

The areas are not separate stages. The distribution set the bounded scope, and the contradictions explained the exceptions in the baseline. The contested decision became a junction, and the junctions set the tail's cost. Each area is a different way of reading the same loop, and the discipline is reading all nine together.

Appendix J: The Thirty-Nine Reference Loops

The six chains contain thirty-nine loops. For each, the table gives its place in the APQC Process Classification Framework and the shape of its cases. It also counts the published benchmarks and named deployments found, and names the main law or standard that puts a person in the loop. Framework numbers follow the openly published version 7.4 of the cross-industry framework and should be checked against version 8.0. "None clean" means no single element matches the loop. Counts reflect sources found by October 2026.

Chain	Loop	Framework reference	Shape	Benchmarks found	Named deployments	Main rule putting a person in the loop
Lead to cash	Lead qualification	3.5.1.3	heavy tail (est.)	4	4	EU AI Act Article 50
Lead to cash	Quote and pricing approval	3.5.3	heavy tail (est.)	0	1	Sarbanes-Oxley Act s. 404 internal control over financial reporting
Lead to cash	Contract review	none clean	heavy tail (est.)	3	2	ABA Formal Opinion 512 on generative AI tools
Lead to cash	Customer credit check	9.2.1	heavy tail	5	4	GDPR Article 22 as interpreted in CJEU C-634/21 SCHUFA
Lead to cash	Order booking	3.5.4	mixed	5	3	US Export Administration Regulations
Lead to cash	Billing and invoicing	9.2.2	mixed (est.)	7	0	EU VAT Directive 2006/112/EC invoicing rules

Chain	Loop	Framework reference	Shape	Benchmarks found	Named deployments	Main rule putting a person in the loop
Lead to cash	Collections and cash application	9.2.3	heavy tail	9	7	EU Late Payment Directive 2011/7/EU
Procure to pay	Supplier onboarding	4.2.5.1	heavy tail (est.)	2	1	OFAC sanctions
Procure to pay	Purchase requisition and approval	4.2.4.1	heavy tail	4	2	SOX 404 / PCAOB AS 2201
Procure to pay	Purchase order issue	4.2.4.4	heavy tail	3	2	SOX 404 / PCAOB AS 2201
Procure to pay	Goods and services receipt	none clean	mixed (est.)	0	1	SOX 404 / PCAOB AS 2201
Procure to pay	Supplier invoice approval	9.6.1.3	heavy tail	5	4	SOX 404 / PCAOB AS 2201
Procure to pay	Payment run	9.6.1.8	heavy tail (est.)	4	0	SOX 404 / PCAOB AS 2201
Procure to pay	Employee expense claims	9.6.2	mixed (est.)	3	3	Accountable plan rules
Hire to retire	Job requisition	7.2.1	mixed	5	1	Directive
Hire to retire	Candidate screening	7.2.2-7.2.3	heavy tail (est.)	3	3	EU AI Act
Hire to retire	Offer and contract	7.2.4	mixed	2	0	Directive
Hire to retire	Employee onboarding	7.3.1	normal (est.)	1	0	Form I-9 employment eligibility verification

Chain	Loop	Framework reference	Shape	Benchmarks found	Named deployments	Main rule putting a person in the loop
Hire to retire	Payroll changes	9.5.2.2-9.5.2.4	mixed	5	1	Sarbanes-Oxley s.404 internal control over financial reporting
Hire to retire	Leave and absence requests	7.6.4	mixed	4	2	Family and Medical Leave Act
Hire to retire	HR helpdesk	7.7.2	mixed	2	2	GDPR Articles 9 and 35
Hire to retire	Offboarding	7.6.2	mixed	2	0	German Works Constitution Act
Record to report	Journal entries	9.3.2.2	mixed (est.)	9	2	Sarbanes-Oxley Act s. 404 / s.302 and segregation of duties over journal entry preparation and approval
Record to report	Account reconciliation	9.3.2.6	heavy tail (est.)	5	5	SOX s.404 internal control over financial reporting
Record to report	Intercompany settlement	9.3.2.5	heavy tail (est.)	7	0	OECD Transfer Pricing Guidelines
Record to report	Period close	9.3.2.8	mixed (est.)	7	3	SOX s.302 quarterly CEO/CFO certification and s.404 ICFR
		9.3.4		4	0	

Chain	Loop	Framework reference	Shape	Benchmarks found	Named deployments	Main rule putting a person in the loop
Record to report	Management and statutory reporting		heavy tail			SOX s.302 and SEC Rule 13a-14 certifications
Record to report	Audit requests	9.8.3.1	heavy tail (est.)	6	0	SOX s.404
Issue to resolution	Support ticket resolution	6.2.2	mixed	7	4	EU AI Act
Issue to resolution	Returns and refunds	6.2.4	mixed (est.)	4	3	Consumer Rights Directive 2011/83/EU
Issue to resolution	Complaint handling	6.2.3	heavy tail	5	1	FCA Handbook DISP 1.6
Issue to resolution	Warranty claims	6.3.2	heavy tail (est.)	4	2	Magnuson-Moss Warranty Act and 16 CFR Part 703
Issue to resolution	IT incident management	8.7.8	heavy tail	7	0	NIS2 Directive
Issue to resolution	Field service dispatch	6.3.4	mixed (est.)	2	0	EU AI Act
Order to delivery	Order fulfilment	3.5.4	mixed (est.)	10	2	EAR Know Your Customer guidance and red flags
Order to delivery	Inventory replenishment	4.1.5	heavy tail (est.)	4	3	SOX Section 404 internal control over financial reporting
Order to delivery	Demand planning	4.1.2	mixed	4	3	SOX Section 404 internal control over

Chain	Loop	Framework reference	Shape	Benchmarks found	Named deployments	Main rule putting a person in the loop
						financial reporting
Order to delivery	Shipment exception handling	4.4.4	mixed	5	2	FTC Mail
Order to delivery	Supplier delivery exceptions	4.4.2.2	heavy tail (est.)	3	1	Uyghur Forced Labor Prevention Act rebuttable presumption

Appendix K: The Kit as Files

The templates come as files so they can be used directly.

File	Contents
T01 to T18, Word files	One document per template, with every field, guidance in each cell and a signature block with name, role and date
Registers workbook, Excel	One sheet each for the strip register, rule register, decision register, contested-decision log, junction register and baseline sheet, with columns matching the fields
Economics workbook, Excel	Baseline and result side by side, cost lines, accepted outcomes, and cost per accepted outcome calculated for the body and the tail
Read me	How the files map to the lifecycle steps and who signs each one

Two practices make the files work as intended. Keep each signed version rather than overwriting it, so the record set shows how the loop changed. And store the files with the loop, not with the project, because the loop outlives the project.

Appendix L: The Lifecycle and Governance Frameworks

Many companies already work within an AI governance framework. The lifecycle does not replace them. It produces the business decisions and signed records those frameworks ask for. This appendix shows where the main ones meet the steps.

The NIST AI Risk Management Framework, published in the United States in January 2023, organises AI risk work into four functions: govern, map, measure and manage. Its "map" function, establishing context and risks, corresponds to Steps 1 to 5. "Measure" corresponds to the shadow evidence, the grading and the economics in Steps 6 to 8. "Manage" corresponds to the junctions, the switch-off and the operating review in Steps 7 and 9. "Govern" runs through the whole lifecycle in its signatures.

ISO/IEC 42001, published in 2023, is a management-system standard for AI. Like other management-system standards, it follows a plan, do, check, act structure and expects documented roles, risk assessment, operational controls, monitoring and improvement. The lifecycle's registers, gate reviews, charters and operating reviews are the kind of documented evidence an audit against the standard looks for.

The European Union's AI Act, adopted in 2024, sets obligations for systems used in high-risk areas, such as decisions about employment, access to credit and essential services. For those systems it requires risk management, logging and effective human oversight by competent people. It also requires those deploying such systems to monitor their operation. A loop that falls in a high-risk area must meet those requirements in law. The junction register, the switch-off plan, the charter naming competent people and the operating review are where the lifecycle produces the evidence. Many loops, such as supplier invoice approval, fall outside the high-risk categories. The same records remain good practice.

None of this is legal advice. Each company should map its loops to the obligations that apply to it, with its own counsel.

Appendix M: A Facilitator's Script

A first session, ninety minutes, four people: the process owner, an experienced operator, the system owner and a facilitator.

Before the session. Pick one loop from the top right of the Work Loop Map, or the most important loop if no map exists yet. Ask the system owner to bring the configuration for the main system the loop uses. Ask the operator to bring three recent cases: one routine, one difficult and one that went wrong.

Minutes 0 to 15: Outcome. Ask for the outcome as one sentence a customer would recognize. Reject activities. Ask who receives the result and what they do next. Record volume and today's cycle time. Do not move on until the process owner agrees.

Minutes 15 to 35: Compensations. Walk the operator's routine case from start to finish. At every step, ask what constraint created it and whether that constraint still binds. Strike what no longer serves the outcome and note why.

Minutes 35 to 55: Rules and context. Walk the difficult case. List every rule applied and where it lives. List every source consulted. Assign an owner to finish both lists after the session.

Minutes 55 to 75: Decisions. List every decision in the three cases. Sort each into mechanical, bounded, consequential or contested. When the room disagrees, mark it contested and record who would have the authority to settle it.

Minutes 75 to 90: Junctions, people, economics and score. Draft the junctions from the consequential decisions and irreversible actions. Name the loop's owner. Note what the baseline measures are and who will measure them. Score every node, read out the gate and list what holds the loop there, with an owner and a date for each item.

After the session. Send the canvas to everyone present within a day. Book the second session for when the rules list is complete and the contested decisions have an answer.

Appendix N: Reading Map

Each area connects to other volumes in The Business Engineer Library, where its concepts are developed further.

Area	Where to go deeper
Distributions	*The Work Loop Canvas*, on the method of business engineering and the shape of cases
Value and the map	*The Work Loop Canvas*, on the map and the four connections to the business model
Work and process	*The Redesign*, on the passes over a documented process and what a map cannot tell you
Rules and decisions	*The Redesign*, on extracting rules and sorting decisions
The executor	The engineering volumes, on agents, context and the systems beneath them
Control and acceptance	*The Standard*, on acceptance as a business question; *Evaluation Engineering*, on grading in depth
People and accountability	*The Agentic Organization* and *The New Roles of the Agentic Era*
Economics	*AI Economics*, on the cost of intelligence across the enterprise
Governance	*The Embedding Lifecycle*, on gates and the record set step by step

Appendix O: The Thirty-Nine Loops by Type

The table places each of the thirty-nine reference loops of Part Ten by the types of loop in Part Eleven. Every loop also draws on the models for every loop, and every loop sits in a chain. To find the models for a loop of your own, find its nearest reference loop here, then read the chapters its marks point to.

Loop	Chain	Heavy-tailed	High-volume	People	Regulated	Customer	Outside parties	Generated steps
Lead qualification	Lead to cash	●				●		●
Quote and pricing approval	Lead to cash	●			●			
Contract review	Lead to cash	●					●	●
Customer credit check	Lead to cash	●		●	●			
Order booking	Lead to cash		●					
Billing and invoicing	Lead to cash		●		●	●		
Collections and cash application	Lead to cash	●	●			●		
Supplier onboarding	Procure to pay	●					●	
Purchase requisition and approval	Procure to pay	●	●		●			
Purchase order issue	Procure to pay				●		●	
Goods and services receipt	Procure to pay		●				●	
		●	●		●		●	

Loop	Chain	Heavy-tailed	High-volume	People	Regulated	Customer	Outside parties	Generated steps
Supplier invoice approval	Procure to pay							
Payment run	Procure to pay	●			●		●	
Employee expense claims	Procure to pay		●	●	●			
Job requisition	Hire to retire			●				
Candidate screening	Hire to retire	●		●				
Offer and contract	Hire to retire			●	●			
Employee onboarding	Hire to retire		●	●				
Payroll changes	Hire to retire		●	●	●			
Leave and absence requests	Hire to retire		●	●				
HR helpdesk	Hire to retire		●	●		●		●
Offboarding	Hire to retire			●	●			
Journal entries	Record to report		●		●			
Account reconciliation	Record to report	●	●		●			
Intercompany settlement	Record to report	●			●			
Period close	Record to report				●			
Management and statutory reporting	Record to report	●			●			

Loop	Chain	Heavy-tailed	High-volume	People	Regulated	Customer	Outside parties	Generated steps
Audit requests	Record to report				●		●	
Support ticket resolution	Issue to resolution		●			●		●
Returns and refunds	Issue to resolution		●			●		
Complaint handling	Issue to resolution	●			●	●		
Warranty claims	Issue to resolution	●				●	●	
IT incident management	Issue to resolution	●			●			●
Field service dispatch	Issue to resolution			●			●	●
Order fulfilment	Order to delivery		●				●	
Inventory replenishment	Order to delivery	●					●	
Demand planning	Order to delivery						●	
Shipment exception handling	Order to delivery					●	●	●
Supplier delivery exceptions	Order to delivery	●					●	

The Five Moves

If you do nothing else after reading this book, make these five moves. Each is small, and together they start a company across the middle.

1. Agree the vocabulary. Take the glossary and agree with the people who will sponsor and run embedding that these words mean these things in your company.
2. Read your map, and one distribution. List your loops by outcome, place them by value and readiness, and chart the share of cases against the share of value for the first candidate. Choose by the map, not by the demo.
3. Measure one baseline. For that loop, measure cost per case, elapsed time and first-time-right on a real sample, with a method finance has agreed.
4. Hold the contested-decision meeting. Find the decision two senior people would answer differently, put it in front of someone with authority, and write down the answer.
5. Name the people. Write down who will own the loop, hold its junctions, work its exceptions, grade it, own the adoption, and sign its limits. The person who signs the limits should be the one whose job is to say no.

The Meters

These are the measures that show whether a lifecycle is healthy. Each is read at the step shown and kept on the operating review afterwards. Measures that count activity, such as cases processed, agents deployed or steps automated, are deliberately left out. They rise whether or not the business is better off.

Meter	What it measures	Read at	Healthy sign
Distribution shape	Share of cases and of value by band	1, 10	Body and tail identified and named
Baseline completeness	Volume, cost, elapsed time, exceptions and first-time-right, all measured	2	Every figure measured on a real sample
Flow efficiency	Hands-on time divided by elapsed time	2, 9	Rises sharply after Strip and Core
Strip ratio	Share of mapped steps struck	3	Every struck step has a named constraint
Contradictions found and decided	Conflicting rules surfaced and settled	4	All decided, none deferred
Contested decisions open	Decisions the organisation never agreed	4 to 9	Zero before Design, and kept at zero
Agreement, body and tail	Shadow results matching people's, by band	6	Both read separately; tail understood
Bounded share	Share of cases, and of value, the executor handles alone	7, 9	Widens in recorded steps
Cost per accepted outcome	Fully loaded cost divided by outcomes meeting the standard	8, 9	Below baseline, body and tail apart
Standard met	Share of outcomes meeting the standard	6 to 9	Stable; drift caught within days
Junction override rate	How often junction holders overrule the executor's evidence	9	Above zero and explained
Exception age	Age of the oldest case in the queue	9	Within the time limit every week
Time per step	Elapsed weeks in each step	All	Falls with each new loop

The Mental Models

Model	Principle	Where it lives in the discipline
The hard problem moved	The business model was the hard problem of the 2000s; the core is the hard problem now	The reason the discipline exists
The cascade	Foundations, then method, then knowledge; the instrument, the sequence and the record follow	The order in which the discipline is adopted
Each era left one part	A century of work design left the standard, the handoff, the owner, the stop cord, stripping, written rules and software that acts	The canvas assembles them around one outcome
The laws of thought	Reasoning can be written as explicit rules that machines can follow	The rule and decision registers are the company's laws of thought for one loop
The limits of rules	No rule system of real power is complete	The tail, tacit knowledge and contested decisions are where rules stop
Isomorphism	Symbols mean something only when their structure mirrors the world	Registers are walked from real cases, not copied from manuals
The strange loop	A system that models itself can change itself	The company models its work, acts through the model, and revises it
Neural and symbolic	Learned models are flexible without guarantees; rules are precise but brittle	An embedded loop combines both, with people at the edges
Requisite variety	Only variety can absorb variety	Rules take the body, models add variety, people hold the tail
The good regulator	Controlling a system requires a model of it	The record set is the regulator, not paperwork
Reflexivity	Loops change the distributions they read	Re-read the distribution after every loop runs
How distributions are made	Independent causes make normal shapes; compounding makes heavy tails	Predict the shape from the mechanism
Compression is prediction	The shortest description that still predicts is the understanding	A compression must be testable against new cases

Model	Principle	Where it lives in the discipline
The method loop	Read the distribution, abstract, compress, name, test, build judgement, return	Every concept and every lifecycle
Inside out	AI creates value inside processes, where companies are least documented	Why embedding stalls and why the lifecycle surfaces what is undocumented
Body and tail	Routine body, consequential tail, designed differently	Bounded scope, junctions, shadow results and economics
The work loop	The unit of design is one outcome owned end to end	The canvas, the map and every template
Compensations	Many steps exist for constraints that are gone	Strip before writing a rule
Contradictions are findings	Conflicting rules show where the process has been guessing	The rule register
Contested decisions	Some decisions were never agreed; only authority settles them	No loop passes Sort with one open
The junction	People decide where actions are costly or irreversible	Placed by reversibility, with evidence and a way back
The jagged frontier	Executors are strong and weak in ways not visible in advance	Shadow reads agreement case by case
The ironies of automation	Supervisors of automation over-trust it and lose skills	Override rates and learning the tail
Scale follows signatures	Readiness is shown by signed documents, not confidence	The Scale Gates
The signer who can say no	Limits are signed by the person whose job is to refuse	The control owner signs Bound
Per accepted outcome	Fully loaded cost over results that met the standard	The only economic measure reported
Value arrives late	Little is saved until the executor acts	Why finishing matters more than starting
Two kinds of machinery	The model is bought; the written half is built	Steps 2 to 5 build it, Steps 6 to 9 keep it true
Falsifiable gates	A test is worth running only if it can fail	Shadow and Bound targets are written before the cases arrive
A pattern language	Recurring loops have recurring solutions, adapted, never copied	Every entry in the Work Loop Library

Model	Principle	Where it lives in the discipline
Contested decisions are the rule	Twenty-seven of thirty-nine reference loops hold a decision never agreed	Settle them in the first week of Sort
An entry is a prediction	A reference loop predicts shape, decisions, junctions and baseline	Where your loop differs, the difference is information
The finished example	A filled form teaches what a blank one cannot	The reference processes
The cycle closes	Every lifecycle ends at the map that starts the next	Scale feeds Read; the company learns to embed

Glossary

Term	Definition
Abstraction	Seeing the few kinds of case behind the many, deliberately leaving out detail
Accretion	The layer of compensations around a loop's genuine steps
Actions	The changes an executor may make, each with its system and whether it can be undone
Adoption owner	The person responsible for people whose work changes and for how new staff learn
Agreement	The share of cases where the executor's result matched the accepted result, reported for body and tail
Architect	A business engineer who holds the method for a company
Baseline	Measured volume, fully loaded cost per case, elapsed time, exception rate and first-time-right before change
Body	The large share of routine, similar cases in a loop
Bounded decision	A decision needing judgement inside a written limit; the executor may take it alone
Bounded scope	The signed conditions under which the executor acts alone
Business engineer	The practitioner who runs embedding lifecycles
Business engineering	The practice of understanding how a business actually works, well enough to redesign it
Case	One unit of work handled by a loop from start to finish
Chain	A sequence of loops in which each one's outcome is the next one's input
Closed loop	A system that acts, senses the result, compares it with a goal and adjusts
Compensation	A step that exists only to work around a constraint
Compression	Stating a mechanism in the fewest words that still predict something
Concentration	The share of value carried by the top 5% of cases by value
Consequential decision	A decision with a high cost of error or an irreversible action; a person decides
Contested decision	A decision the organisation never agreed on; settled by authority in writing
Contested-decision log	The record of contested decisions, positions, settler, answer and date

Term	Definition
Context	The information an executor must read to act well on a case
Contradiction	Two rules that cannot both be followed
Control owner	The person responsible for controls over a process; signs bounded limits
Cost per accepted outcome	Fully loaded cost divided by outcomes meeting the standard
Decider	An executor given an outcome, rules and limits, which works out its own steps
Decision register	The list of a loop's decisions, with class, reason and limit
Decision rights	Who may take which decision, up to which limit
Decision	Any point where the work could go more than one way
Distribution shift	A change in how cases spread, which can move cases from body to tail
Distribution	How a loop's cases spread by value, difficulty, risk or time
Drift	A gradual change in the quality of outcomes as inputs, models or people change
Economics sheet	The signed record of cost per accepted outcome against the baseline
Elapsed time	Time from a case's arrival to its accepted outcome
Embedding	An executor that decides running part of a core process, inside written limits, with named people answering for it
Event log	Time-stamped system records of the steps each case went through
Exception owner	The person who works the exception queue
Exception queue	The single place for cases outside every rule, with owner and time limit
Executive sponsor	The senior person who commissions embedding and settles cross-department disagreements
Executor	Whatever does the work in a loop
Feedback	Information about the effect of an action, used to adjust the next action
First-time-right	Cases accepted without rework, as a share of all cases
Flow efficiency	Hands-on time divided by elapsed time
Fully loaded cost	All costs of a loop, including people at real cost, systems, rework and errors
Gate review record	The signed record of the decision at Bound
Gate	

Term	Definition
	The end of a lifecycle step, where a named person decides whether work moves on
Good regulator	The principle that controlling a system well requires a model of it
Grader	The person who keeps the standard and grading calibrated
Grading	Applying the standard to every outcome, by rules, a model grader and a human audit
Handoff	A point where a case moves between lanes or between loops
Heavy-tailed distribution	A spread in which a few cases carry most of the value or risk
Instruction-follower	An executor that needs its steps specified in advance
Instructions	The executor's brief, drawn from the signed registers
Isomorphism	A structural correspondence that lets symbols mean something in the world
Jagged frontier	The uneven boundary of tasks an executor does well and badly
Junction holder	The named person who decides at a junction
Junction	A point where a person decides instead of the executor, with defined evidence and reversal
Lead	A business engineer who runs lifecycles end to end
Loop charter	The signed list of named people in every loop role, and the agreed changes to work
Loop owner	The person who answers for a loop's outcome
Map review	The quarterly update of the Work Loop Map
Mechanical decision	A decision that follows a written rule completely
Method loop	Read the distribution, abstract, compress, name, test, build judgement, return
Model	The language model at the centre of an executor
Neural model	A system that learns patterns from examples rather than following written rules
Neurosymbolic system	A system that combines learned models with explicit rules, tools and verification
Normal distribution	A spread in which cases cluster around a typical value
Operating review	The monthly review of a loop in the core, ending in recorded decisions

Term	Definition
Outcome	The result a loop produces, stated so an outsider can accept or reject it
Override rate	How often junction holders overrule the executor's evidence
Permissions	The rights an executor holds in each system, matched to its actions
Pilot	A test of whether an executor can do a task on prepared cases
Platform	The layer that runs the loop: routing, limits, logs, switch-off, grader access
Practitioner	A business engineer who runs lifecycle steps under guidance
Preferential attachment	A mechanism in which what is large grows faster, producing heavy tails
Process map	A swimlane drawing of how a loop runs today
Process mining	Reconstructing the real flow of work from event logs
Process owner	The person responsible for how a process runs today
Readiness	A 1 to 5 score of how close a loop is to being embedded
Record set	The full collection of a loop's signed documents
Reflexivity	The way a loop's actions change the distribution it reads
Requisite variety	The principle that a controller needs as many responses as the disturbances it must handle
Rework	A step that redoes what should have been right the first time
Role change plan	The map-wide plan for how roles and expertise change
Rule register	The list of a loop's rules with source, owner and date
Rule	A statement of what must or must not happen in a case
Scale Gates	G0 Mapped, G1 Shadow, G2 Bounded, G3 Owned, G4 Core
Scope change	Any widening of what the executor may do alone
Shadow running	The executor working real cases beside people without acting
Signature	A named person's dated, recorded acceptance of a document
Standard	The written definition of an accepted outcome and how it is graded
Strange loop	A system that, moving through its levels, comes back to represent and change itself
Strip ratio	Steps struck as compensations, divided by steps on the signed map
Switch-off	The tested means of stopping a loop and returning cases to people

Term	Definition
Symbolic layer	The rules, decisions, limits and standards a business writes down for a loop
Tacit knowledge	Knowledge people use but cannot fully state
Tail	The small share of cases carrying most of the value, risk or difficulty
Topology	The steps an outcome genuinely requires
Value to the core	A 1 to 5 score of how much a loop matters to the business
Walk-through	Following real cases through a process map
Work Loop Canvas	The one-page description of a loop in twelve connected nodes
Work Loop Map	All of a company's loops placed by value and readiness
Work loop	The unit that produces one outcome end to end

Notes and Sources

Parts One to Five, Eleven and Twelve

Laws of thought and computation. George Boole, *An Investigation of the Laws of Thought* (1854). Claude E. Shannon, "A Symbolic Analysis of Relay and Switching Circuits," *Transactions of the AIEE* 57 (1938), from his 1937 master's thesis. J. McCarthy, M. Minsky, N. Rochester and C. Shannon, *A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence* (1955).

Limits of formal systems. Kurt Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I" (1931). Alan Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem" (1936). Michael Polanyi, *The Tacit Dimension* (1966). On expert systems: Edward Shortliffe, *Computer-Based Medical Consultations: MYCIN* (1976); John McDermott, "R1: A Rule-Based Configurer of Computer Systems," *Artificial Intelligence* 19 (1982); Edward Feigenbaum on the knowledge acquisition bottleneck.

Strange loops. Douglas R. Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid* (1979) and *I Am a Strange Loop* (2007).

The neural tradition. Warren McCulloch and Walter Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics* 5 (1943). Frank Rosenblatt, "The Perceptron," *Psychological Review* 65 (1958). Marvin Minsky and Seymour Papert, *Perceptrons* (1969). David Rumelhart, Geoffrey Hinton and Ronald Williams, "Learning Representations by Back-Propagating Errors," *Nature* 323 (1986). Grégoire Delétang et al., "Language Modeling Is Compression," ICLR (2024).

Neurosymbolic AI. Daniel Kahneman, *Thinking, Fast and Slow* (2011). Henry Kautz, "The Third AI Summer," AAAI Robert S. Engelmore Memorial Lecture (2020), published in *AI Magazine* (2022). Artur d'Avila Garcez and Luís C. Lamb, "Neurosymbolic AI: The 3rd Wave," *Artificial Intelligence Review* (2023). Gary Marcus, "The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence" (2020). Trieu Trinh et al., "Solving Olympiad Geometry Without Human Demonstrations," *Nature* 625 (2024).

Cybernetics and control. Norbert Wiener, *Cybernetics: Or Control and Communication in the Animal and the Machine* (1948). W. Ross Ashby, *An*

Introduction to Cybernetics (1956). Roger C. Conant and W. Ross Ashby, "Every Good Regulator of a System Must Be a Model of That System," *International Journal of Systems Science* 1, no. 2 (1970). George Soros, *The Alchemy of Finance* (1987), on reflexivity.

Distributions and models. Robert K. Merton, "The Matthew Effect in Science," *Science* 159 (1968). Albert-László Barabási and Réka Albert, "Emergence of Scaling in Random Networks," *Science* 286 (1999). Alfred Korzybski, *Science and Sanity* (1933). George E. P. Box, "Science and Statistics," *Journal of the American Statistical Association* 71 (1976). Karl Popper, *Logik der Forschung* (1934). Herbert A. Simon, *The Sciences of the Artificial* (1969). On compression: Andrey Kolmogorov (1965) and Jorma Rissanen, "Modeling by Shortest Data Description," *Automatica* 14 (1978).

Double-entry bookkeeping. Luca Pacioli, *Summa de arithmetica, geometria, proportioni et proportionalità* (Venice, 1494), which contains the treatise on bookkeeping known as *Particularis de computis et scripturis*.

The Business Model Canvas. Alexander Osterwalder and Yves Pigneur, *Business Model Generation* (2010).

Heavy tails. Vilfredo Pareto's observations on income concentration in *Cours d'économie politique* (1896–97). Nassim Nicholas Taleb, *The Black Swan* (2007), for "Mediocristan" and "Extremistan".

Intuition. Daniel Kahneman and Gary Klein, "Conditions for Intuitive Expertise: A Failure to Disagree," *American Psychologist* 64, no. 6 (2009).

Process redesign and mining. Michael Hammer, "Reengineering Work: Don't Automate, Obliterate," *Harvard Business Review* (July–August 1990). Wil M. P. van der Aalst, *Process Mining: Data Science in Action*, 2nd ed. (2016). Niklas Modig and Pär Åhlström, *This Is Lean* (2012), on flow efficiency.

Decisions. Object Management Group, *Decision Model and Notation (DMN) 1.0* (2015).

The stop cord. Taiichi Ohno, *Toyota Production System: Beyond Large-Scale Production* (English edition, 1988).

Stage-gate. Robert G. Cooper, "Stage-Gate Systems: A New Tool for Managing New Products," *Business Horizons* 33, no. 3 (1990).

The jagged frontier. Fabrizio Dell'Acqua et al., "Navigating the Jagged Technological Frontier," Harvard Business School Working Paper 24-013 (2023).

Productivity and learning. Erik Brynjolfsson, Danielle Li and Lindsey Raymond, "Generative AI at Work," *Quarterly Journal of Economics* 140, no. 2 (2025).

Automation. Lisanne Bainbridge, "Ironies of Automation," *Automatica* 19, no. 6 (1983). Raja Parasuraman and Victor Riley, "Humans and Automation: Use, Misuse, Disuse, Abuse," *Human Factors* 39, no. 2 (1997).

Measures. Charles Goodhart, "Problems of Monetary Management: The U.K. Experience" (1975), the source of Goodhart's law.

Adoption. McKinsey & Company, *The state of AI in 2025: Agents, innovation, and transformation* (November 2025). MIT NANDA, *The GenAI Divide: State of AI in Business 2025* (2025), as summarised in Virtualization Review, 19 August 2025.

Governance. National Institute of Standards and Technology, *AI Risk Management Framework (AI RMF 1.0)* (January 2023). ISO/IEC 42001:2023. Regulation (EU) 2024/1689 (the AI Act). *Moffatt v. Air Canada*, 2024 BCCRT 149.

The composite manufacturer. Invoice figures are set within ranges reported by Ardent Partners, *Accounts Payable Metrics that Matter in 2025*, and APQC Open Standards Benchmarking.

Models for Every Loop. Michael Hammer and James Champy, *Reengineering the Corporation: A Manifesto for Business Revolution* (1993). Vilfredo Pareto, *Cours d'économie politique* (1896). Nassim Nicholas Taleb, *The Black Swan: The Impact of the Highly Improbable* (2007). Michael Hammer, "Reengineering Work: Don't Automate, Obliterate," *Harvard Business Review* (1990). Herbert A. Simon, *The New Science of Management Decision* (1960). Kurt Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I," *Monatshefte für Mathematik und Physik* (1931). Taiichi Ohno, *Toyota Production System: Beyond Large-Scale Production* (1978; English edition 1988). Thomas B. Sheridan and William L. Verplank, *Human and Computer Control of Undersea Teleoperators*, MIT Man-Machine Systems Laboratory (1978). Regulation (EU) 2016/679, General Data Protection Regulation, Article 22 (2016). W.

Ross Ashby, *An Introduction to Cybernetics* (1956). Roger C. Conant and W. Ross Ashby, "Every Good Regulator of a System Must Be a Model of That System," *International Journal of Systems Science* (1970). Walter A. Shewhart, *Economic Control of Quality of Manufactured Product* (1931). Norbert Wiener, *Cybernetics: Or Control and Communication in the Animal and the Machine* (1948). Philip B. Crosby, *Quality Is Free* (1979). Robert G. Cooper, "Stage-Gate Systems: A New Tool for Managing New Products," *Business Horizons* (1990). Regulation (EU) 2024/1689, Artificial Intelligence Act, Annex III (2024). George Soros, *The Alchemy of Finance* (1987). Robert K. Merton, "The Self-Fulfilling Prophecy," *The Antioch Review* (1948).

Heavy-Tailed Loops. Vilfredo Pareto, *Cours d'économie politique* (1896). Benoit Mandelbrot, "The Variation of Certain Speculative Prices", *Journal of Business* 36 (1963). Nassim Nicholas Taleb, *The Black Swan: The Impact of the Highly Improbable* (2007). Robert K. Merton, "The Matthew Effect in Science", *Science* 159 (1968). Albert-László Barabási and Réka Albert, "Emergence of Scaling in Random Networks", *Science* 286 (1999). Sam L. Savage, *The Flaw of Averages: Why We Underestimate Risk in the Face of Uncertainty* (2009). Joseph M. Juran, *Quality Control Handbook* (1951). H. Ford Dickie, "ABC Inventory Analysis Shoots for Dollars, Not Pennies", *Factory Management and Maintenance* (1951). Peter Kraljic, "Purchasing Must Become Supply Management", *Harvard Business Review* (1983). Albert O.

Hirschman, *National Power and the Structure of Foreign Trade* (1945). Orris C. Herfindahl, *Concentration in the Steel Industry*, doctoral dissertation, Columbia University (1950). Basel Committee on Banking Supervision, *Supervisory Framework for Measuring and Controlling Large Exposures* (2014). Abraham Wald, *A Method of Estimating Plane Vulnerability Based on Damage of Survivors*, Statistical Research Group memoranda (1943), reprinted by the Center for Naval Analyses (1980). Frédéric Bastiat, *Ce qu'on voit et ce qu'on ne voit pas* (1850). Daniel Bernoulli, "Specimen theoriae novae de mensura sortis", *Commentarii Academiae Scientiarum Imperialis Petropolitanae* (1738). Philippe Artzner, Freddy Delbaen, Jean-Marc Eber and David Heath, "Coherent Measures of Risk", *Mathematical Finance* 9 (1999).

High-Volume Routine Loops. Niklas Modig and Pär Åhlström, *This Is Lean: Resolving the Efficiency Paradox* (2012). George Stalk Jr. and Thomas M. Hout, *Competing Against Time: How Time-Based Competition Is Reshaping Global Markets* (1990). John D. C. Little, "A Proof for the Queuing Formula: L

= λW″, *Operations Research* 9(3) (1961). Taiichi Ohno, *Toyota Production System: Beyond Large-Scale Production* (1978; English edition 1988). Walter A. Shewhart, *Economic Control of Quality of Manufactured Product* (1931). W. Edwards Deming, *Out of the Crisis* (1986). Gerhard Widmer and Miroslav Kubat, ″Learning in the Presence of Concept Drift and Hidden Contexts″, *Machine Learning* 23 (1996). German Growth Opportunities Act (Wachstumschancengesetz), B2B e-invoicing receipt obligation from 1 January 2025 (as read in 2026).

Lisanne Bainbridge, ″Ironies of Automation″, *Automatica* 19(6) (1983). Charles A. E. Goodhart, ″Problems of Monetary Management: The U.K. Experience″, *Papers in Monetary Economics*, Reserve Bank of Australia (1975). Marilyn Strathern, ‴Improving Ratings′: Audit in the British University System″, *European Review* 5(3) (1997). T. P. Wright, ″Factors Affecting the Cost of Airplanes″, *Journal of the Aeronautical Sciences* 3(4) (1936). Boston Consulting Group, *Perspectives on Experience* (1968).

Loops That Decide About People. European Parliament and Council, *Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)*, Annex III and Article 14 (2024). Colorado General Assembly, *Senate Bill 24-205, Consumer Protections for Artificial Intelligence* (2024). European Parliament and Council, *Regulation (EU) 2016/679 (General Data Protection Regulation)*, Articles 5, 9, 22 and 35 (2016). Court of Justice of the European Union, *Case C-634/21, OQ v Land Hessen (SCHUFA Holding)* (2023). US Equal Employment Opportunity Commission and others, *Uniform Guidelines on Employee Selection Procedures*, 29 CFR Part 1607 (1978). US Supreme Court, *Griggs v. Duke Power Co.*, 401 U.S. 424 (1971). New York City Council, *Local Law 144 of 2021 on automated employment decision tools* (2021).

Solon Barocas and Andrew D. Selbst, *Big Data′s Disparate Impact*, California Law Review 104 (2016). Anya E. R. Prince and Daniel Schwarcz, *Proxy Discrimination in the Age of Artificial Intelligence and Big Data*, Iowa Law Review 105 (2020). Jeffrey Dastin, *Amazon scraps secret AI recruiting tool that showed bias against women*, Reuters (2018). *Equal Credit Opportunity Act*, 15 U.S.C. 1691 (1974), and *Regulation B*, 12 CFR 1002.9. Sandra Wachter, Brent Mittelstadt and Chris Russell, *Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR*, Harvard Journal of Law and Technology 31 (2018). OECD, *Guidelines on the Protection of Privacy and Transborder Flows of Personal Data* (1980).

Raja Parasuraman and Victor Riley, *Humans and Automation: Use, Misuse, Disuse, Abuse*, Human Factors 39 (1997). Linda J. Skitka, Kathleen L. Mosier and Mark Burdick, *Does Automation Bias Decision-Making?*, International Journal of Human-Computer Studies 51 (1999). Marco Almada, *Human Intervention in Automated Decision-Making: Toward the Construction of Contestable Systems*, Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law (2019). European Parliament and Council, *Directive (EU) 2024/2831 on improving working conditions in platform work* (2024).

Regulated and Audited Loops. Committee of Sponsoring Organizations of the Treadway Commission, *Internal Control: Integrated Framework* (1992; updated 2013). *Sarbanes-Oxley Act of 2002*, Public Law 107-204, sections 302 and 404 (2002). Luca Pacioli, *Summa de arithmetica, geometria, proportioni et proportionalita* (1494). Public Company Accounting Oversight Board, *AS 1105: Audit Evidence* (2010). Institute of Internal Auditors, *The Three Lines of Defense in Effective Risk Management and Control* (2013). Institute of Internal Auditors, *The IIA's Three Lines Model: An Update of the Three Lines of Defense* (2020). International Accounting Standards Board, *Definition of Material: Amendments to IAS 1 and IAS 8* (2018). US Securities and Exchange Commission, *Staff Accounting Bulletin No. 99: Materiality* (1999).

International Auditing and Assurance Standards Board, *ISA 500: Audit Evidence* (2009). ISACA, *COBIT: Control Objectives for Information and Related Technologies* (1996). American Institute of Certified Public Accountants, *SSAE No. 18: Attestation Standards: Clarification and Recodification* (2017). Jeff Bezos, *2015 Letter to Shareholders*, Amazon.com (2016). Regulation (EU) 2024/886 of the European Parliament and of the Council on instant credit transfers in euro (2024).

Customer-Facing Loops. A. Parasuraman, Valarie A. Zeithaml and Leonard L. Berry, "A Conceptual Model of Service Quality and Its Implications for Future Research", *Journal of Marketing* 49(4) (1985). A. Parasuraman, Valarie A. Zeithaml and Leonard L. Berry, "SERVQUAL: A Multiple-Item Scale for Measuring Consumer Perceptions of Service Quality", *Journal of Retailing* 64(1) (1988). *Moffatt v. Air Canada*, 2024 BCCRT 149, British Columbia Civil Resolution Tribunal (2024). Bill Price and David Jaffe, *The Best Service Is No Service* (2008). Matthew Dixon, Karen Freeman and Nicholas Toman, "Stop Trying to Delight Your Customers", *Harvard Business Review* (2010).

Christopher W. L. Hart, James L. Heskett and W. Earl Sasser Jr., "The Profitable Art of Service Recovery", *Harvard Business Review* (1990).

Stefan Michel and Matthew L. Meuter, "The Service Recovery Paradox: True but Overrated?", *International Journal of Service Industry Management* 19(4) (2008). Daniel Kahneman, Barbara L. Fredrickson, Charles A. Schreiber and Donald A. Redelmeier, "When More Pain Is Preferred to Less: Adding a Better End", *Psychological Science* 4(6) (1993). Regulation (EU) 2024/1689 (Artificial Intelligence Act), Article 50(1) (2024). California Senate Bill 1001, Bolstering Online Transparency Act, Business and Professions Code sections 17940 to 17943 (2018). Jan Carlzon, *Moments of Truth* (1987).

Loops That Depend on Outside Parties. Jay W. Forrester, *Industrial Dynamics* (1961). Hau L. Lee, V. Padmanabhan and Seungjin Whang, "Information Distortion in a Supply Chain: The Bullwhip Effect", *Management Science* 43(4) (1997). Peter Kraljic, "Purchasing Must Become Supply Management", *Harvard Business Review* (1983). Yossi Sheffi, *The Resilient Enterprise: Overcoming Vulnerability for Competitive Advantage* (2005). Regulation (EU) 2016/679, *General Data Protection Regulation*, Article 28 (2016). Stephen A. Ross, "The Economic Theory of Agency: The Principal's Problem", *American Economic Review* 63(2) (1973). Michael C. Jensen and William H. Meckling, "Theory of the Firm: Managerial Behavior, Agency Costs and Ownership Structure", *Journal of Financial Economics* 3(4) (1976). Wallace J. Hopp and Mark L. Spearman, *Factory Physics* (1996).

APQC, *Open Standards Benchmarking*, measure 102645, "Percentage of supplier on-time delivery" (accessed 2026). Ronald Reagan, *Remarks on Signing the Intermediate-Range Nuclear Forces Treaty* (1987). Association for Financial Professionals, *2025 AFP Payments Fraud and Control Survey* (2025).

Loops Whose Steps Are Generated. Peter F. Drucker, *The Practice of Management* (1954). Fabrizio Dell'Acqua et al., "Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality," Harvard Business School Working Paper 24-013 (2023). *Moffatt v. Air Canada*, 2024 BCCRT 149, Civil Resolution Tribunal of British Columbia (2024). Thomas B. Sheridan and William L. Verplank, *Human and Computer Control of Undersea Teleoperators*, MIT Man-Machine Systems Laboratory technical report (1978). Raja Parasuraman, Thomas B. Sheridan and Christopher D. Wickens, "A Model for Types and Levels of Human Interaction with Automation," *IEEE*

Transactions on Systems, Man, and Cybernetics, Part A 30, no. 3 (2000). Jerome H. Saltzer and Michael D.

Schroeder, "The Protection of Information in Computer Systems," *Proceedings of the IEEE* 63, no. 9 (1975). Lawrence Lessig, *Code and Other Laws of Cyberspace* (1999). Jeff Bezos, "2015 Letter to Shareholders," Amazon.com (2016). John D. Lee and Katrina A. See, "Trust in Automation: Designing for Appropriate Reliance," *Human Factors* 46, no. 1 (2004). Raja Parasuraman and Victor Riley, "Humans and Automation: Use, Misuse, Disuse, Abuse," *Human Factors* 39, no. 2 (1997). OpenAI, "Klarna," customer story, openai.com (2024). Entrepreneur, "Klarna CEO Reverses Course by Hiring More Humans, Not AI" (2025).

Loops in a Chain. Eliyahu M. Goldratt and Jeff Cox, *The Goal: A Process of Ongoing Improvement* (1984). W. Edwards Deming, *The New Economics for Industry, Government, Education* (1993). Bertrand Meyer, *Object-Oriented Software Construction* (1988). Melvin E. Conway, "How Do Committees Invent?", *Datamation* (1968). John D. C. Little, "A Proof for the Queuing Formula: L = λW", *Operations Research* (1961). J. F. C. Kingman, "The Single Server Queue in Heavy Traffic", *Proceedings of the Cambridge Philosophical Society* (1961). Wallace J. Hopp and Mark L. Spearman, *Factory Physics* (1996). Jay W. Forrester, *Industrial Dynamics* (1961). John D. Sterman, *Business Dynamics: Systems Thinking and Modeling for a Complex World* (2000).

Parts Six to Eight

Frederick Winslow Taylor, *The Principles of Scientific Management* (New York: Harper & Brothers, 1911).

The moving assembly line at Ford's Highland Park plant was introduced in 1913. Chassis assembly times of roughly twelve and a half hours before and about an hour and a half after are the figures commonly reported from Ford's own records. The five-dollar day was announced in January 1914. See David A. Hounshell, *From the American System to Mass Production, 1800–1932* (Baltimore: Johns Hopkins University Press, 1984).

Alfred P. Sloan Jr., *My Years with General Motors* (New York: Doubleday, 1964). Alfred D. Chandler Jr., *Strategy and Structure: Chapters in the History of the American Industrial Enterprise* (Cambridge, MA: MIT Press, 1962).

Taiichi Ohno, *Toyota Production System: Beyond Large-Scale Production* (Portland, OR: Productivity Press, 1988; Japanese edition 1978). On the automatic loom that stopped when a thread broke and the origin of jidoka, see Toyota's own historical accounts of Sakichi Toyoda's Type G loom. James P. Womack, Daniel T. Jones and Daniel Roos, *The Machine That Changed the World* (New York: Rawson Associates, 1990).

Nassim Nicholas Taleb, *The Black Swan: The Impact of the Highly Improbable* (New York: Random House, 2007), on Mediocristan and Extremistan. Daniel Kahneman and Gary Klein, "Conditions for Intuitive Expertise: A Failure to Disagree," *American Psychologist* 64, no. 6 (2009): 515–526. Gary Klein, *Sources of Power: How People Make Decisions* (Cambridge, MA: MIT Press, 1998).

Michael Hammer, "Reengineering Work: Don't Automate, Obliterate," *Harvard Business Review*, July–August 1990. Michael Hammer and James Champy, *Reengineering the Corporation: A Manifesto for Business Revolution* (New York: HarperBusiness, 1993).

SAP R/3 was released in 1992.

Alexander Osterwalder, *The Business Model Ontology: A Proposition in a Design Science Approach*, doctoral thesis, University of Lausanne, 2004. Alexander Osterwalder and Yves Pigneur, *Business Model Generation* (Hoboken, NJ: Wiley, 2010).

Andrew Campbell, Mikel Gutierrez and Mark Lancelott, *Operating Model Canvas: Aligning Operations and Organization with Strategy* (Zaltbommel: Van Haren Publishing, 2017).

Object Management Group, *Business Process Model and Notation (BPMN) Version 2.0*, 2011; *Decision Model and Notation (DMN) Version 1.0*, 2015.

The VTDF framework (Value, Technology, Distribution, Financial), the Thinking OS and the business engineering loop are set out in full in *The Business Engineer*, the Library's Foundation Volume.

The composite insurer is illustrative. Its figures are authored to teach the method and are not drawn from any firm.

Part Nine

The middle. MIT NANDA, *The GenAI Divide: State of AI in Business 2025* (July 2025), based on a review of more than 300 public AI initiatives, 52

interviews and 153 survey responses. It reported that about 5% of custom enterprise AI tools reached production, as summarised in Virtualization Review, 19 August 2025. McKinsey & Company, *The state of AI in 2025: Agents, innovation, and transformation* (November 2025; fieldwork June to July 2025). In it, 88% of respondents reported regular AI use in at least one function, and about one third reported scaling AI. Some 23% reported scaling an agentic system somewhere in the enterprise, and no more than 10% in any single function. Gartner, press release, 25 June 2025: "Over 40% of agentic AI projects will be canceled by the end of 2027," due to escalating costs, unclear business value or inadequate risk controls. Gartner, July 2024: at least 30% of generative AI projects expected to be abandoned after proof of concept by the end of 2025.

Foundations. George Boole, *An Investigation of the Laws of Thought* (1854). Kurt Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I," *Monatshefte für Mathematik und Physik* 38 (1931). Michael Polanyi, *The Tacit Dimension* (1966). Douglas R. Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid* (1979), on isomorphism, levels and strange loops. Henry Kautz, "The Third AI Summer," AAAI Robert S. Engelmore Memorial Lecture (2020), and Artur d'Avila Garcez and Luís C. Lamb, "Neurosymbolic AI: The 3rd Wave," *Artificial Intelligence Review* (2023), on combining learned and symbolic methods. Trieu H. Trinh et al., "Solving olympiad geometry without human demonstrations," *Nature* 625 (2024), on AlphaGeometry.

Control and measurement. Norbert Wiener, *Cybernetics* (1948). W. Ross Ashby, *An Introduction to Cybernetics* (1956), on requisite variety. Roger C. Conant and W. Ross Ashby, "Every Good Regulator of a System Must Be a Model of That System," *International Journal of Systems Science* 1, no. 2 (1970). Karl Popper, *The Logic of Scientific Discovery* (1959). George Soros, *The Alchemy of Finance* (1987), on reflexivity. G. K. Chesterton, *The Thing* (1929), on the fence. Charles Goodhart, "Problems of Monetary Management: The U.K. Experience" (1975). Lisanne Bainbridge, "Ironies of Automation," *Automatica* 19, no. 6 (1983). Fabrizio Dell'Acqua et al., "Navigating the Jagged Technological Frontier," Harvard Business School Working Paper 24-013 (2023). *The Business Engineering Body of Knowledge* develops each of these foundations at length.

Plan-do-check-act. Walter A. Shewhart, *Statistical Method from the Viewpoint of Quality Control* (1939). W. Edwards Deming's lectures in Japan from 1950 popularised the cycle.

The waterfall. Winston W. Royce, "Managing the Development of Large Software Systems," *Proceedings of IEEE WESCON* (1970). Royce recommended iteration and building a preliminary version, advice later readers largely ignored.

Stage-gate. Robert G. Cooper, "Stage-Gate Systems: A New Tool for Managing New Products," *Business Horizons* 33, no. 3 (1990).

CRISP-DM. Pete Chapman et al., *CRISP-DM 1.0: Step-by-step Data Mining Guide* (2000), from a consortium formed in 1996 that included DaimlerChrysler, SPSS and NCR.

Service management and MLOps. The IT Infrastructure Library was developed by the UK Central Computer and Telecommunications Agency from the late 1980s. D. Sculley et al., "Hidden Technical Debt in Machine Learning Systems," *Advances in Neural Information Processing Systems* 28 (2015).

The stop cord. Taiichi Ohno, *Toyota Production System: Beyond Large-Scale Production* (1988, English edition), on jidoka and stopping the line.

Invoice baselines. Ardent Partners, *Accounts Payable Metrics that Matter in 2025*, as reported by Thomson Reuters. It gives an average cost to process an invoice of about $9.40 and an average cycle time of 17.4 days, against about 3.1 days for the best. Exception rates were 22% for most organisations against 9% for the best. APQC Open Standards Benchmarking: a median of 75% of invoice line items matched the first time, and about 12,000 invoices processed per accounts payable employee per year. The composite manufacturer's figures are illustrative and set within these ranges.

Spend concentration and payment fraud. Tail spend is commonly reported as about a fifth of spend across most suppliers (The Hackett Group). Association of Certified Fraud Examiners, *Occupational Fraud 2024: A Report to the Nations*: billing schemes appeared in 22% of cases, with a median loss of $100,000. Association for Financial Professionals, *2025 Payments Fraud and Control Survey*: 79% of organisations faced attempted or actual payments fraud in 2024.

Customer-facing loops. In *Moffatt v. Air Canada*, 2024 BCCRT 149, a Canadian tribunal held the airline responsible for incorrect refund information given by the chatbot on its website.

Governance frameworks. National Institute of Standards and Technology, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)* (January 2023). ISO/IEC 42001:2023, *Information technology: Artificial intelligence: Management system*. Regulation (EU) 2024/1689 (the AI Act), in particular Articles 9 (risk management), 14 (human oversight) and 26 (obligations of deployers).

Part Ten

Reference data. Each chain's benchmarks, distributions, rules and deployments come from a structured review of published sources carried out in October 2026. The sources include APQC Open Standards Benchmarking, the Credit Research Foundation, Ardent Partners, The Hackett Group, SHRM, the US Bureau of Labor Statistics, MetricNet, Gartner, the EY survey for Paycom, and AfterShip tracking data. Deployment results come from company statements and vendor-published customer stories, marked where vendor-reported. Figures are as published and were not independently verified.

Pattern languages. Christopher Alexander, Sara Ishikawa and Murray Silverstein, *A Pattern Language: Towns, Buildings, Construction* (1977). Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software* (1994). W. Ross Ashby, *An Introduction to Cybernetics* (1956), on requisite variety. *The Business Engineering Body of Knowledge* develops the foundations behind each entry's lens.

Framework references. APQC, *Process Classification Framework, Cross-Industry, version 7.4* (2024) and version 8.0 (February 2026). APQC's Process Classification Framework® (PCF) is an open standard developed by APQC, a nonprofit that promotes benchmarking and best practices worldwide. The PCF is intended to facilitate organizational improvement through process management and benchmarking, regardless of industry, size, or geography. To download the full PCF or industry-specific versions of the PCF, as well as associated measures and benchmarking, please visit www.apqc.org/pcf.

Laws and standards. Regulation (EU) 2024/1689 (the AI Act), Annex III; Regulation (EU) 2016/679 (GDPR), Article 22; Directive 2011/7/EU on late

payment; the Sarbanes-Oxley Act, section 404; *Moffatt v. Air Canada*, 2024 BCCRT 149. Each chapter names the rules relevant to its loop. None of this is legal advice.

The composite manufacturer. All company figures are illustrative, authored to teach and set within the published ranges cited in each chapter.

The Thread

This book gathers the Embedding Series of The Business Engineer Library: *The Business Engineering Body of Knowledge*, *The Work Loop Canvas*, *The Embedding Lifecycle* and *The Work Loop Library*. Each remains available as a self-contained volume with its own skill. The book threads into the rest of the Library through its shared instruments: the canvas and the map, the junction, the charter, the standard and the drift check. *The Redesign* develops the passes over a documented process. *The Standard* and *Evaluation Engineering* develop acceptance and grading. *The Agentic Organization* develops how roles and teams change across many loops.

Also in This Library

The Business Engineer: the Foundation Volume, with the VTDF framework and the method behind this book.

The Redesign: the Process Volume, with the eight passes that take a documented process apart and rebuild it for human-and-agent execution.

The FDE Playbook: the forward-deployed engineer's field manual, whose first duty is mapping the work before embedding AI in it.

The Business Orchestrator: the Operator's Volume, on running a firm where judgment is the binding constraint.

The Agentic Organization: on how the firm and its careers change around AI.

The Standard: what enterprise AI has to prove, and who decides.

Evaluation Engineering: the technical discipline of grading what AI produces.

THE BUSINESS ENGINEER

The New Business Engineer

Read the distribution. Write the rules. Close the loop.

businessengineer.ai