DRAFT · V0.2 · JUNE 11, 2026 · UPDATED JULY 16, 2026

The Missing Ledger: Decision Intelligence for Organizations through Business-as-Code

From Storytelling to Decision Science

Calvin Pak

Independent · Reference implementation: Monocles.ai

Disclosure: the author created Monocles.ai, a decision-intelligence platform.

ABSTRACT

Organizations run on decisions, yet decisions are the one thing they never record, observe, or evaluate. We argue this absence (the *missing ledger* of organizational decision-making) is why management knowledge is dominated by post-hoc narrative (success stories, the halo effect) rather than evidence, and why corporate "strategy" is so often back-filled and non-repeatable. Building on Simon's view of decision-making as the core of administration, and on the well-established but rarely-operationalized theory of sequential decision-making under uncertainty (Markov decision processes), we present *Decision Intelligence as a discipline* and *Business-as-Code* (BaC) as its execution methodology. BaC instruments the organization so that decisions become observable (an explicit ontology of state), recorded (every action with its context and rationale), and evaluable (decision quality and calibration, separated from outcome luck). We give a three-layer architecture (authority, process, interface) built on a recurring hard/soft split; an account of how autonomous agents serve as a continuous reconciliation layer; a codifiability framework that says *which* decisions can be automated, evaluated, or must remain human; and an evaluation framework that turns A/B testing from interfaces into *decisions*. The aim is a framework that is rigorous and, above all, implementable in any organization.

Keywords: decision intelligence; bounded rationality; Markov decision processes; process mining; organizational ontology; autonomous agents; decision evaluation; calibration; Business-as-Code.

---

1 Introduction

Herbert Simon opened the modern study of organizations with a claim that has never been refuted and never been operationalized: that "decision-making is the heart of administration," and that the vocabulary of management theory must be derived from the logic and psychology of human choice.[1] An organization, in this view, is more than a structure, a culture, or a balance sheet: it is a machine for making and executing decisions under conditions of limited information, limited attention, and limited computational capacity, what Simon called *bounded rationality*.[1]

If decisions are the unit of organizational performance, one would expect organizations to record them with the same rigor they bring to cash. They do not. A mid-sized firm can tell you, to the cent, what it spent on travel last quarter, but it cannot tell you what decisions it made, what alternatives it considered, what it believed at the time, or whether those decisions were any good. The decision (the actual unit of value creation) leaves no trace. This is the *missing ledger*: the organization meticulously books every dollar and never books a single choice.

The consequence is that management knowledge is built on the wrong evidence. We watch the rocket land on the moon and praise the fuel. We celebrate outcomes and back-fill the "strategy" that supposedly produced them, with no empirical record of the decision itself, the alternatives that were rejected, the rationale that was offered, or the information that was actually available at the moment of choice. A company succeeds, and a literature springs up to explain why its culture, its leadership, or its boldness was destined to win, until the same company falters, and the same traits are recast as arrogance and drift. The story is always available after the fact. The data never was.

This paper argues that the missing ledger is the central blind spot of management as a practice, and that it is now *fixable*. The arrival of language-capable software agents, combined with a disciplined way of representing a business in machine-readable form, makes it possible to instrument an organization so that its decisions become observable, recorded, and evaluable: the three properties that turn any human activity from anecdote into science. The discipline that names this goal, **Decision Intelligence**, is not ours to coin; it was established over the past decade as a field of decision *support* and *modeling*, by Pratt's causal decision diagrams,[24] by Kozyrkov's framing of decision intelligence at Google as "turning information into better actions at any scale,"[25] and now by an analyst category of decision-intelligence platforms that model, orchestrate, and audit decisions.[26] Our subject is the adjacent and largely unbuilt half of that discipline: decision *instrumentation and execution*, the recording and evaluation of the decisions an organization actually makes, rather than decision support for humans *before* the choice. The execution methodology we build on for that purpose is **Business-as-Code** (BaC), itself a term and practice that predates this paper (§3) and that we extend rather than originate.

The contribution is a *synthesis plus one genuinely new mechanism*: three ideas wired into a single loop.

- **The decision ledger as an automatic byproduct of executing Business-as-Code.** The closest prior art either proposes *manual* capture of decisions for organizational memory[34] or logs only the decisions made by *autonomous AI* for governance.[35] Neither frames the executable SOP as the *emitter* of the decision record. When execution runs through codified procedures, the ledger is produced as a side effect of doing the work: you get it for free. This is the paper's strongest novel hook.
- **The firm-as-(PO)MDP as a management thesis, evaluated by off-policy evaluation.** We treat the organization itself as a partially observable Markov decision process and the recorded ledger as the log against which its decision policy is assessed off-policy, turning a control-theoretic model of the firm into a measurement program.
- **The decision record as the primary unit of evaluation.** Decision quality, calibration, and off-policy ROI are computed *on the recorded decision* as a first-class, theory-grounded object, not as an audit or governance afterthought bolted onto a platform.

Stated in one sentence: Pratt, Kozyrkov, and the analyst category established Decision Intelligence as decision *support*; Aera and Singh recently extended it to *recording* the decisions made by AI;[33][35] Koenders named the gap of un-recorded human decisions;[34] this paper makes the decision record an *automatic emission of executable Business-as-Code*, and formalizes its evaluation as off-policy assessment of the organization treated as a POMDP, grounding decision *instrumentation*, not decision support, in Simon's bounded rationality.

The goal, stated plainly, is to make organizational decision-making an evidence-based, evaluable, improvable discipline, and to do so in a way that is implementable in a real firm, not merely admirable on the page.

2 The Missing Ledger

The behavioral tradition in organization theory established, decades ago, that the firm is best understood as a coalition of decision-makers operating under bounded rationality. Simon showed that real choosers do not optimize; they *satisfice*, accepting the first option that clears an aspiration threshold.[1] Cyert and March extended this into a behavioral theory of the firm in which organizational decisions emerge from standard operating procedures, attention allocation, and the resolution of conflict among subgoals.[2] March later distilled the lesson: organizations are, above all, machines for processing decisions, and most of what they do is governed by rules and routines rather than fresh calculation.[3] The decision is the organizational atom.

And yet the decision is the one atom the organization never weighs. Consider the contrast with money. Since Pacioli codified double-entry bookkeeping in 1494, every commercial transaction has been recorded twice, reconciled, and rolled into statements that can be audited and compared across time and across firms.[20] Five centuries of accounting discipline mean that the *financial* life of a business is fully instrumented. Its *decisional* life is not instrumented at all. We do not record the context in which a choice was made, the options that were on the table, the rationale that was offered, the information that was available, who decided, or why. There is no ledger of decisions, no reconciliation of decisions, and therefore no statements that could be audited or compared.

2.1 The missing-data problem

Call this the *missing-data problem* of management. Because decisions are not recorded, they cannot be studied empirically. Because they cannot be studied empirically, the knowledge that fills the vacuum is necessarily anecdotal and retrospective. This is a failure of *instrumentation* on the part of organizations rather than of rigor on the part of researchers. An astronomer without a telescope is reduced to mythology, and that is precisely the position of anyone who wishes to reason about organizational decisions from the existing record.

2.2 The storytelling and halo fallacies

The void is filled by stories. Rosenzweig's *The Halo Effect* dismantles the business-success literature with surgical care: when a company performs well, observers attribute to it a coherent culture, a visionary leader, and a sound strategy; when the same company falters, the same attributes are re-described in negative terms, all without any independent measurement of the supposed causes.[4] Performance casts a halo over everything else, and the celebrated "great" firms tend, on follow-up, to regress toward the mean. Mauboussin's *The Success Equation* makes the statistical companion point: outcomes in business are a blend of skill and luck, and the more luck contributes, the more strongly results regress, so the firms we lionize are disproportionately the lucky ones.[5]

The professional poker player Annie Duke gives this its sharpest name. In *Thinking in Bets* she describes *resulting*: the cognitive error of judging the quality of a decision by the quality of its outcome.[6] A sound decision can lose and a reckless one can win, because the world is noisy; to evaluate decisions by outcomes alone is to let the noise grade the work. Taleb's *Fooled by Randomness* sharpens the warning: survivorship and randomness conspire to dress up luck as competence, especially in domains with long feedback lags.[7] And Kahneman supplies the underlying machinery: the narrative fallacy that imposes coherent causal stories on random sequences, hindsight bias that makes the past look more predictable than it was, and the overconfidence that makes us certain of stories we have no right to believe.[8]

2.3 The deep point: ROI is measured, the decision is not

Every one of these critiques converges on a single structural fact. Organizations measure *outcomes* (revenue, margin, return on investment) and they measure them well. What they never measure is the *decision*: the quality of the reasoning given what was knowable at the time. Outcome and decision quality are different variables, and the gap between them is exactly the noise that resulting ignores. Without a record of the decision, the only feedback an organization ever receives is the outcome, and the only knowledge it can produce is the story it tells about that outcome. You cannot do science on data you never collected; you can only narrate.

This is the gap Decision Intelligence is meant to fill. The remedy is *instrumentation* rather than better stories or more disciplined intuition: make the decision observable, record it with its context and rationale, and evaluate it on its own terms, apart from the luck of its result. The rest of this paper is concerned with how an organization can actually be built so that this happens by default.

3 Related Work

The ideas in this paper sit at the confluence of three active lines of work: a decade-old field called Decision Intelligence, a 2024–2025 wave of "business-as-code" methodologies, and a small but growing literature that proposes to record decisions directly. We engage each in turn, with particular care for the work that comes closest, so that the precise boundary of our contribution is visible rather than implied.

3.1 Decision Intelligence as a field: support and modeling, not instrumentation

Decision Intelligence is an established discipline. Lorien Pratt developed and popularized it as a discipline at Quantellia, originally "decision engineering" (c. 2008) and rebranded "decision intelligence" (c. 2013), together with *Causal Decision Diagrams*: causal graphs that link decisions to levers, intermediate effects, and outcomes, as a way to help humans reason about complex decisions *before* they are made.[24] Cassie Kozyrkov, as Google's first Chief Decision Scientist, popularized decision intelligence as "the discipline of turning information into better actions at any scale," combining data science with the social and managerial sciences; her emphasis is decision *hygiene*: helping humans decide well.[25] Industry analysts have since defined a category of *Decision Intelligence Platforms*: software to support, automate, and augment the decision-making of humans or machines, with capabilities spanning decision modeling, orchestration of execution flows, and the ability to evaluate, govern, and audit decision outcomes.[26] The analyst definition already includes auditing outcomes; this paper treats the decision record not as one platform feature among many but as a first-class, theory-grounded unit derived from a (PO)MDP model of the firm. The common thread in all of this prior work is that its center of gravity is decision *support and modeling for humans*. Ours is decision *instrumentation and execution*: recording, as data, the decisions the organization in fact makes, and evaluating them after the fact.

3.2 Business-as-Code and its lineage

Business-as-Code has a lineage we build on. We have developed the program elsewhere as an engineering discipline for the firm;[23] here we situate it. The economic thesis (that AI agents take ownership of outcomes formerly delivered by human labor, inverting software-as-a-service into "software that does the work") was named *Service-as-Software* by Foundation Capital.[27] The runtime flavor we lean on most, in which AI agents execute business logic defined as code and workflows are authored in Markdown, was made public as *Business-as-Code* (paired with Services-as-Software) by Drivly.[28] Boston Consulting Group has advanced a sibling operating-model concept, *Enterprise as Code*, for human-machine collaboration in the AI era.[29] The specific mechanism at the heart of our SOPs (compiling a natural-language standard operating procedure into a validated, agent-executable workflow) is itself a named 2025 pattern, appearing as "Agent SOPs" in agent frameworks and as "Agent Operating Procedures" that compile natural-language intent into validated workflows.[31][32] Our use of BaC is as the *vehicle* for instrumentation (the reason the decision ledger can be emitted automatically), not as an invention of the methodology itself.

3.3 Recording decisions: the closest prior art

The work nearest to our title concept records decisions as durable artifacts, and three efforts deserve direct comparison. The closest existing synthesis is Aera Technology's *Agentic Decision Intelligence*: its "Decision Data Model" captures the decisions made, their context, the actions taken, and the outcomes as organizational memory, while its "Control Room" logs reasoning, trade-offs, and provenance and tracks recommendation adoption, human overrides, and outcomes against objectives.[33] This is, to our knowledge, the closest deployed artifact to a decision ledger paired with evaluation. It differs from this paper in kind: Aera is a vertical, vendor-proprietary "cognitive operating system" grounded in ERP data for supply-chain and finance operations: not SOPs authored as source code by the business itself, not grounded in Simon, the POMDP, or off-policy evaluation as a theory, and not a general theory of the organization. Willem Koenders argues, complementarily, that organizations want data-driven decision-making yet capture no data on the decisions themselves, and proposes a centralized decision inventory with generative-AI retrieval;[34] this names the gap precisely but proposes *manual* capture for memory and retrieval, with no execution mechanism, no Business-as-Code, and no decision-quality-versus-outcome evaluation. Raktim Singh uses the very term *Decision Ledger*, capturing decision intent, evidence, controls, ownership and approvals, model and policy and tool versions, and outcomes;[35] but it is framed for the governance and forensic auditability of *autonomous AI decisions*: defensibility, not whole-organization decision science, with no Simon or POMDP grounding and a ledger built for accountability rather than decision-quality evaluation. The structural pattern that predates all of these comes from software engineering: Michael Nygard's *Architecture Decision Records*, lightweight per-decision documents (title, status, context, decision, consequences) whose collection forms a "decision log."[36] The ADR is the closest existing structural template for "record a decision as a unit"; we generalize it from the codebase to the firm and make it an automatic emission of execution rather than a manual note.

3.4 What this paper adds

This paper's contribution is a synthesis under one theory together with a single new mechanism: **the decision ledger emitted automatically as a byproduct of executing Business-as-Code, with the recorded decision treated as the primary unit of an off-policy evaluation of the organization modeled as a POMDP.** The nearest neighbors stop short of exactly this. Aera logs AI decisions in a proprietary vertical, Koenders proposes manual capture, and Singh records AI decisions for audit[33][34][35]. None makes the executable SOP the *emitter* of the ledger, and none grounds the evaluation in the sequential-decision theory of the firm.

4 A Theory of Organizational Decisions

To instrument decisions we need a model of what a decision is. The model already exists; it has simply never been wired into the running organization. We build it up in two steps, from the single isolated choice to the sequential, stateful stream of choices that a business actually is.

4.1 From decision trees to expected value

The foundational object of decision analysis is the decision tree. A decision-maker facing a single choice under uncertainty enumerates the available actions; for each action, the possible states of the world with their probabilities; and for each combination, a payoff. The recommended action is the one whose probability-weighted payoff (its *expected value*) is highest, adjusted for the decision-maker's attitude toward risk. Raiffa formalized this as a teachable discipline,[9] and Howard built the engineering practice of decision analysis around it, insisting on the separation of a decision's quality from its outcome long before the idea reached a popular audience.[10] The crucial move is that the tree records the *structure of the choice* (alternatives, beliefs, payoffs), which is precisely the data the missing ledger omits.

4.2 From single choices to sequential decisions: the (PO)MDP

A business, of course, does not make one decision; it makes an unending stream of them, each changing the state from which the next is made. The canonical model for sequential decision-making under uncertainty is the *Markov decision process* (MDP), introduced through Bellman's dynamic programming[11] and developed into a mature theory by Puterman.[12] When the decision-maker cannot directly observe the true state (the normal condition of any organization), the model generalizes to the *partially observable* MDP (POMDP), in which the agent maintains a *belief*, a probability distribution over the hidden state, and updates it from noisy signals.[13]

BOX 1 · THE SEQUENTIAL DECISION MODEL

```
A (partially observable) Markov decision process is the tu-
ple (S, A, T, R, Ω, O, γ):

S: states of the world
A: actions available to the decision-maker
T(s′ | s, a): transition dynamics
R(s, a): reward earned
Ω, O: observations and how they are generated
γ: how the future is discounted

A policy π maps the current state (or, under partial ob-
servability, the current belief) to an action. The objec-
tive is to choose π to maximize the expected discounted sum
of rewards. Because the true state is hidden, the agent
maintains a belief b and revises it with each observation
(the belief update).
```

The mapping to a business is exact, and Figure 1 makes it visible. The *state* is the condition of the firm and its world. The *policy* is how the firm decides what to do given that condition. The *action* is the thing it does. The *reward* is the outcome that follows. And the *belief update* is how the firm revises its picture of reality as new signals arrive. Each of these abstract terms has a concrete business counterpart, which we will build in the following sections; Table 1 states the correspondence in advance.

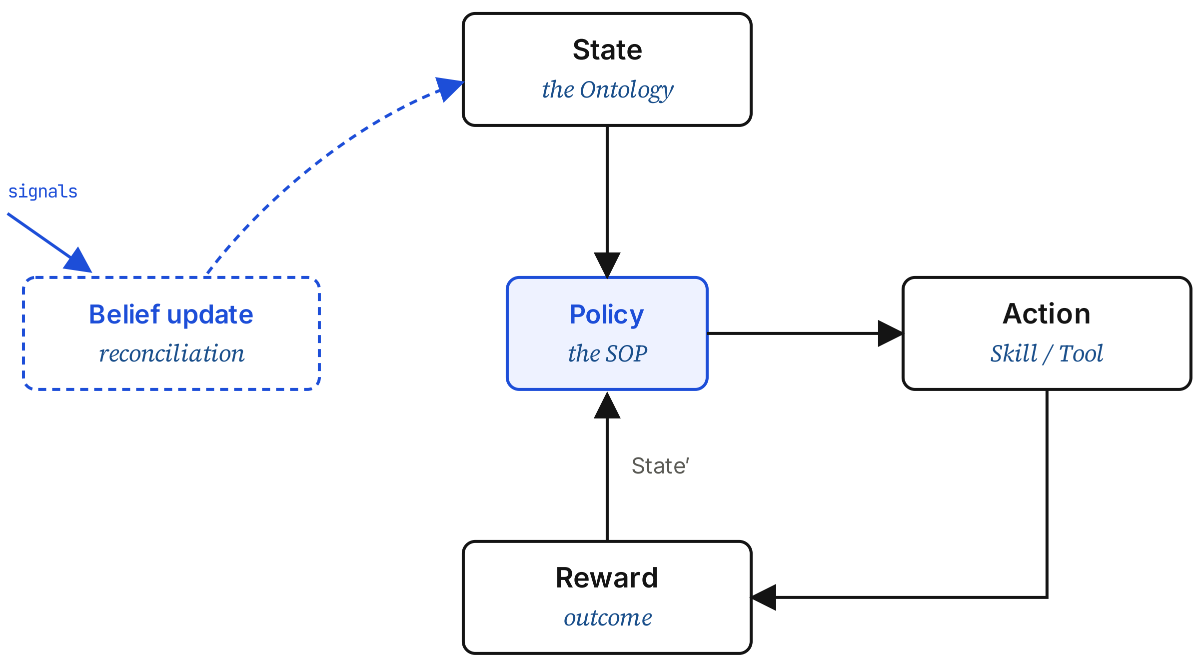

Figure 1: The decision loop, in MDP terms (black) and business terms (blue). The firm reads its *State* (the Ontology), applies a *Policy* (the SOP) to choose an *Action* (a Skill or Tool), and earns a *Reward* (the outcome), which moves the world to State′. A separate *belief-update* path (dashed) takes in observed signals and revises the recorded state, the *reconciliation* loop of §7.

Table 1: The correspondence between the sequential-decision model and Business-as-Code.

Decision-theoretic term	Business-as-Code construct	What it is, concretely
State $s \in S$	the **Ontology**	Entities, their states, and the invariants that constrain them
Policy π	the **SOP**	A codified standard operating procedure: hard rules plus soft strategy
Action $a \in A$	**Skill / Tool**	An atomic capability (Tool) or a composed workflow (Skill)
Belief update	**reconciliation**	Agents diffing world signals against the recorded state
Reward $R(s,a)$	**outcome**	The measured result, logged and attributed to its action

4.1 The insight and the design law

The reframing carries a strong claim. A business has *always* been a partially observable sequential decision process; this is a description, not a metaphor. The reason organizations could never optimize their policy was that the three quantities the mathematics requires were missing, not that the mathematics itself was unavailable (MDPs and POMDPs are standard tools in operations research and artificial intelligence). The *state* was unobserved, scattered across inboxes and people's heads. The *actions* were unrecorded, executed and forgotten. The *rewards* were unattributed, visible only in aggregate at the bottom of a quarterly statement. A theory of control is useless when the system it would control emits no telemetry.

From this follows the design law that organizes the rest of the paper: **to improve decisions, you must observe the state, record the action, and attribute the reward.** We should be candid that the gap here is practical rather than theoretical. The decision-process view of the firm is decades old; the contribution of Business-as-Code is to make it *operational*: to supply the telemetry that the theory has always presupposed and the organization has never produced.

5 Business-as-Code: the Execution Methodology

Business-as-Code is the proposal that an organization be represented in machine-readable form with the same completeness that software systems are.[23] The analogy to a full software stack is deliberate. A running application has a database (its state), a backend (its logic), and a frontend (its interface). A running business has exactly the same three concerns, and BaC names them so they can be built.

5.1 The Ontology as the unifying state

At the base is the **Ontology**: an explicit, machine-readable model of the business's entities, the states those entities can be in, the transitions between states, and the invariants that must always hold. A customer, an invoice, a project, a hire, a contract: each is an entity with a defined lifecycle. The Ontology is the database of the business, the canonical record of *what is true right now*. It is, in MDP terms, the state space made explicit.

The idea of a declarative model of the company has prior art that we adopt here as lineage. Rothmann's *Company-as-Code* proposes a declarative domain-specific language (in the style of infrastructure-as-code) for roles, policies, and compliance, together with graph relationships, yielding a versioned, queryable, testable digital representation of the firm.[30] Earlier still, analysts introduced the *Digital Twin of an Organization*: a dynamic software model of an organization.[37] We distinguish our Ontology from both. Those are largely *static structural* models: descriptions of how the company is shaped. Our Ontology is *live and stateful*: it is the substrate that SOPs execute against, kept true by continuous reconciliation (§7), and it is the very thing whose transitions *emit the decision ledger*. The model is the running state of the firm rather than a portrait of it.

BOX 2 · ACID AS A NORMATIVE DESIGN TARGET

Database systems guarantee that transactions are *atomic*, *consistent*, *isolated*, and *durable* (ACID). Human-run firms satisfy none of these: two departments hold contradictory views of the same customer (inconsistency), half-finished reorganizations leave the org in an invalid intermediate state (non-atomicity), and decisions vanish (non-durability).

We do not claim firms *are* ACID. We claim ACID is a *normative target* that a BaC runtime can impose: state transitions are committed atomically against invariants, the recorded state stays internally consistent, and every committed change is durable and auditable. The point of writing the business in code is to make these guarantees enforceable rather than aspirational.

5.2 Tools, Skills, and SOPs

On top of the Ontology sits a compositional hierarchy of capability. At the bottom are **Tools**: atomic capabilities, each doing one thing: send an email, post a ledger entry, query a balance, transition an entity's state. Tools compose into **Skills**: reusable workflows that chain several tools to accomplish a recurring task, such as onboarding a customer or closing a support ticket. Skills, together with their bindings to the Ontology and their governance, compose into **SOPs**: standard operating procedures that encode whole business processes and policies. An SOP is a skill graph plus the Ontology entities it reads and writes plus the rules that govern it. In the language of Table 1, the SOP *is* the policy π.

5.3 The two-layer SOP

The defining feature of a BaC standard operating procedure (and the property that distinguishes it from a generation of failed attempts to codify business processes) is that it has two layers.

- A **hard layer** of deterministic rules, constraints, and gates that are *enforced* by the runtime. A refund above a threshold requires a second approver; an entity cannot move to `shipped` without a tracking number. These are not suggestions; the runtime refuses transitions that violate them.
- A **soft layer** of strategy, objectives, and decision criteria that an LLM-based agent *reasons over*. Which of three eligible ven-

dors to choose; how to phrase a delicate customer reply; whether an ambiguous case warrants escalation. These are matters of judgment, expressed in natural language and exercised by the agent within the hard layer's constraints.

This two-layer structure is what lets an SOP encode *uncertainty and strategy*, and it is exactly what prior process-automation technologies could not do. Business process management (BPM) and the Decision Model and Notation standard (DMN) represent decisions as deterministic decision tables: given these inputs, return that output.[15] Such tables are excellent for the fully codifiable case and useless for everything else, because they have no way to represent a judgment call. A business process is a decision table wrapped around a core of genuine judgment, not a decision table with missing rows. The two-layer SOP keeps the table where a table belongs (the hard layer) and delegates the residue of judgment to an agent that can actually reason about it.

6 The Three-Layer Architecture

The hard/soft split is not confined to the SOP. It recurs at three layers of the architecture, and that recurrence is the structural spine of the framework. Each layer governs a different aspect of organizational action (who may act, how they act, and how they interact), and each layer draws the same line between what is enforced and what is left to judgment.

Figure 2: One principle (the *hard/soft* split) expressed at three layers. **Authority** bounds what an agent may do (the Agent Proxy: scope, cap, time-to-live, revoker). **Process** is the codified SOP (hard rules plus soft strategy). **Interface** meets humans through structured Forms and Menus (hard) and free-form Chat (soft). The enforced/judgment line (dashed) recurs at every band.

6.1 Authority: the Agent Proxy

Before an agent acts, the organization must decide what it is permitted to do. This is the authority layer, and it is the domain of the *Agent Proxy* pattern developed in a companion paper.[18] Rather than granting an agent the ambient authority of the human it acts for (the classic recipe for the "confused deputy"), the Agent Proxy issues a narrowly scoped, capability-bearing credential: a *scope* of permitted actions, a *cap* on magnitude (a spending limit, say), a time-to-live, and a *revoker* that can withdraw the authority at any moment. The hard side enforces these bounds absolutely; the soft side is the agent's latitude to negotiate, transact, and exercise intent *within* them. This is the layer that makes agentic decisions, negotiations, and transactions safe to delegate at all.

6.2 Process: the codified SOP

The middle layer is the two-layer SOP of §5: a hard skeleton of rules and gates that the runtime enforces, wrapped around a soft core of strategy and criteria that the agent reasons over. It is the policy π of the decision process, now expressed in a form that is simultaneously executable and auditable.

6.3 Interface: chat, form, and menu

The top layer is how humans engage the system, and here the hard/soft split appears as a choice of interface idiom. **Chat** is the soft interface: it traffics in *verbs*, accepting free-form intent in natural language, ideal when the human knows what they want but not the precise shape of the operation. **Form** is a hard interface: it traffics in *nouns*, structured fields rendered in-flight to force process compliance: if an SOP requires three pieces of information before a state can change, the form will not submit without them. **Menu** is the read interface: *views* and dashboards onto the Ontology. The same duality that bounds authority and codifies process now shapes the human surface: free where judgment is wanted, enforced where compliance is required.

7 The Live Organization: Ontology and Reconciliation

An Ontology that is written once and left to rot is worse than no Ontology at all, because it lends false confidence to a stale picture. The hardest problem in any enterprise-modeling effort is not building the model but keeping it true. We call this the *ontology-liveness problem*, and the central operational claim of this paper is that autonomous agents are the answer to it.

7.1 Agents as a continuous reconciliation layer

The accounting analogy is precise and worth taking literally. The Ontology is the *ledger* of business state. The world (the stream of emails, data feeds, meeting transcripts, and external events) is the *bank statement*: the independent record against which the ledger must be reconciled. In bookkeeping, reconciliation is the disciplined practice of comparing the two and resolving every discrepancy. Business-as-Code makes reconciliation continuous and assigns it to agents.

The agent is therefore not merely an actor that executes skills; it is a *bookkeeper*. Its reconciliation loop, shown in Figure 3, runs as follows. It senses a signal from the world. It diffs that signal against the ledger: does the recorded state still match reality? It proposes a state transition to close any gap. And then it branches on stakes: low-stakes transitions are *auto-committed*, while high-stakes transitions are routed to a human for confirmation. Either way, the transition is logged. This is, in the strict sense, *double-entry business state*: every change to the ledger is backed by an observed signal and recorded with its provenance.

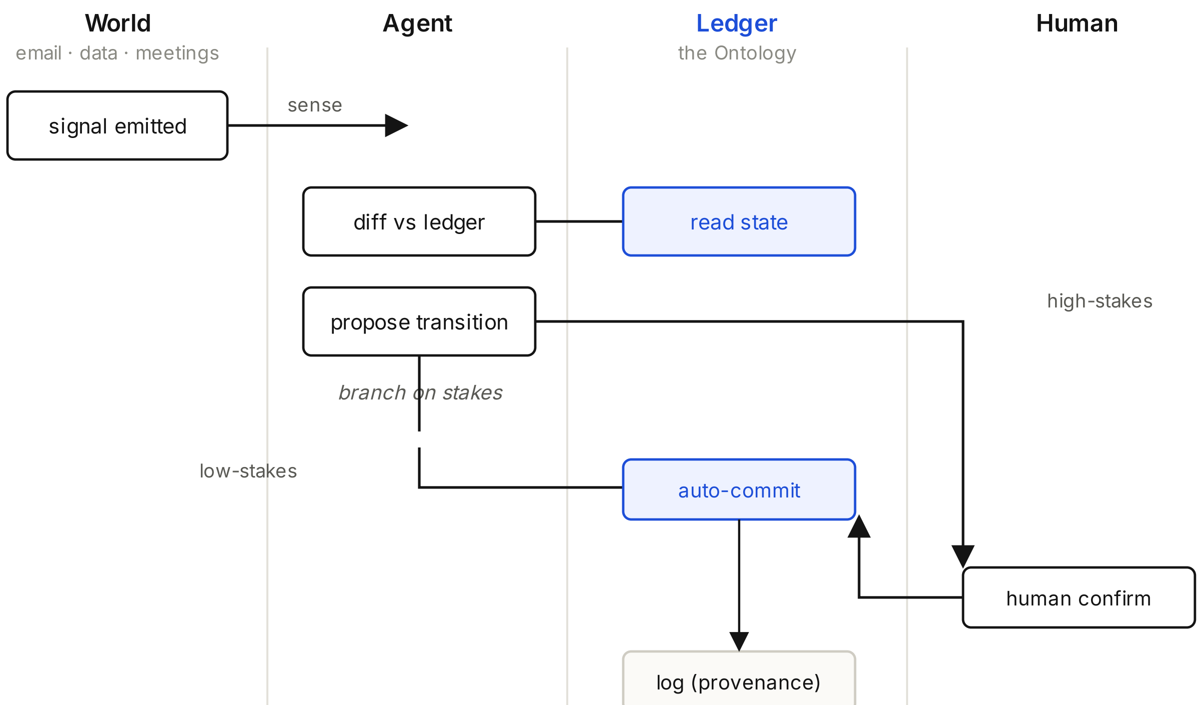

Figure 3: The reconciliation loop as a swimlane across *World*, *Agent*, *Ledger*, and *Human*. The agent senses a world signal, diffs it against the ledger, proposes a state transition, then branches on stakes: low-stakes transitions auto-commit, high-stakes transitions are routed to a human for confirmation. Every committed transition is logged with its provenance.

7.1 Off-SOP decisions as uncommitted changes

No organization runs entirely on codified procedure. People improvise, route around the official process, and invent ad-hoc responses to novel situations, the *shadow processes* that every real firm runs alongside its official ones. In a BaC organization these off-SOP actions are treated as *uncommitted changes* rather than forbidden. The principle is: log everything, even the unscripted. Over time, the accumulated log of real behavior can be mined: van der Aalst's discipline of *process mining* reconstructs the processes an organization actually executes from its event logs, including the shadow ones management never sanctioned.[14] Recurring off-SOP patterns are then candidates for *promotion* into first-class SOPs. The organization learns its own procedures from the trace of its own behavior rather than from a designer's idealization of it.

7.2 Recording rationale, not just action

One further discipline distinguishes a decision ledger from a mere audit log. It is not enough to record *what* was done; the ledger must capture the *context* available at the time, the *rationale* offered for the choice, and, where they matter, the *human motivations* in play: the incentives, the power dynamics, the known biases. These are the very fields the decision tree of §4 demands and the missing ledger of §2 omits. They are also the fields that make decision *science* possible: without recorded rationale you can audit compliance but you cannot evaluate judgment, and judgment is the thing we set out to improve.

Here lies the mechanism that separates this account from its closest neighbors. The decision ledger is a *byproduct of execution* rather than a system the organization must remember to maintain. Because every action runs through a codified SOP (an executable policy that reads the Ontology, branches on the soft layer's judgment, and commits a transition), the SOP *is itself the* *emitter* of the record. The alternatives, the context, the rationale, and the outcome are written as the procedure runs, not transcribed afterward. This is what the prior art does not do: where one line of work proposes that humans manually file decisions into an inventory,[34] and another logs only the decisions taken by autonomous AI for audit,[35] Business-as-Code yields the ledger of *all* codified organizational decisions for free, as the exhaust of running the business through code. You do not collect the data; executing the SOP collects it for you.

8 The Codifiability Boundary

Nothing in the preceding sections claims that all decisions can or should be automated. A central part of the framework is knowing where automation stops. Not every decision is codifiable, and the framework must say *which* are, on what axes, and what to do at each tier.

8.1 The axes of codifiability

Five properties of a decision determine how far it can be pushed into code, illustrated along the spectrum in Figure 4.

- **Observability.** Can the inputs to the decision be captured as data? A decision that turns on signals no instrument records cannot be codified.
- **Objective specifiability.** Can we write down what a good outcome is? Where the objective is contested or ineffable, there is no target to optimize toward.
- **Mapping stability.** Does the same situation call for the same response over time, or does the right answer drift as the environment shifts?
- **Frequency.** Does the decision recur often enough to accumulate the data that codification and evaluation require?
- **Tacitness.** Polanyi's dictum that "we know more than we can tell" names the limit case: knowledge that resists articulation cannot be written down as a rule.[16] Nonaka's account of knowledge creation describes how tacit knowledge is, at best, partially and effortfully externalized.[17]

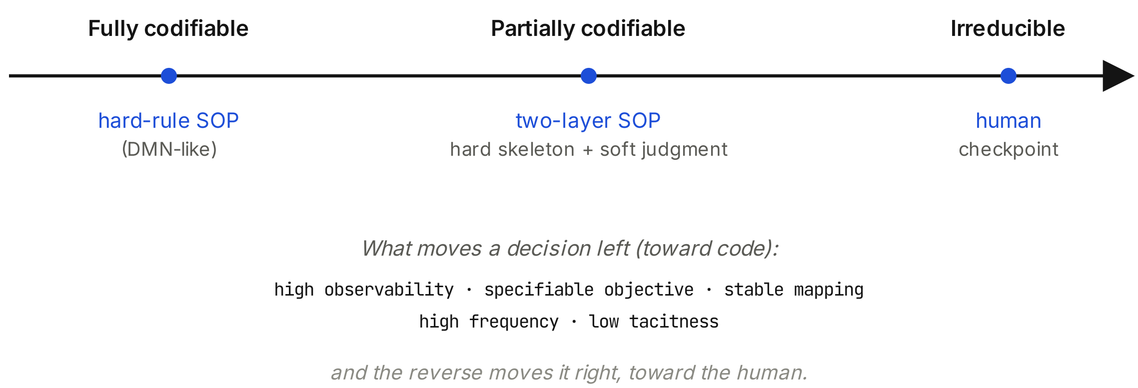

Figure 4: The codifiability spectrum. A decision moves *left* (toward a hard-rule SOP) as it becomes more observable, more objectively specifiable, more stable, more frequent, and less tacit; it moves *right* (toward irreducible human judgment) as those properties weaken. The partially codifiable middle is the home of the two-layer SOP.

8.1 The three tiers

The spectrum resolves into three operating tiers. A **fully codifiable** decision (observable inputs, a clear objective, a stable mapping, high frequency, no tacit content) belongs in a hard-rule SOP, the DMN-style decision table, executed deterministically with no agent in the loop. A **partially codifiable** decision belongs in the two-layer SOP: a hard skeleton handles the structured part while the soft layer hands the genuine judgment to an agent. An **irreducible** decision (tacit, contested, or rare) belongs at a human checkpoint, where the system's job is to surface the right context and record the human's choice and rationale, not to make it.

8.2 The normative principle: minimize the judgment surface

The framework prescribes a direction of travel. Just as secure systems design seeks to minimize the *trusted computing base* (the body of code that must be trusted absolutely), a BaC organization should **minimize the judgment surface**: push as much of every decision as possible into verifiable hard rules, and keep the soft residue that depends on unverifiable agent judgment as small as it can be. Every part of a decision that can be made a checkable rule is a part that can be audited, tested, and trusted. The soft layer is where the risk lives, so the discipline is to keep it lean.

9 Evaluating Decisions

We arrive at the payoff of the whole construction. Once decisions are observed, recorded with context and rationale, and tied to their outcomes, they can be *evaluated*, and evaluation is the core of decision intelligence. The first move is to refuse the conflation that the missing ledger forced on us, and to separate two quantities that the outcome alone runs together.

9.1 Decision quality versus outcome

Decision quality asks: given the information available at the time, was the reasoning sound: were the right alternatives considered, the relevant beliefs stated, the objective served? **Outcome** asks: what actually happened? Outcome is noisy and luck-dominated, and grading decisions by outcome alone is the *resulting* error of §2.[6] This separation is the spine of an established evaluation literature that long predates our framework: Spetzler and the Strategic Decisions Group formalized it for management as *Decision Quality*, with its six elements of a high-quality decision evaluated independently of how the outcome happened to break;[38] Pfeffer and Sutton wrapped the same instinct into evidence-based management;[40] and Roger Martin's framing of the organization as a factory that produces decisions is a direct forebear of treating the decision as the unit of output.[41] A recorded decision ledger lets us evaluate quality directly, on the process, rather than inferring it backwards from a result the world may have corrupted. The novel wiring here is its substrate rather than the distinction itself: we compute these evaluations on the *auto-emitted* decision record as the primary unit of analysis, rather than treating evaluation as an audit add-on layered onto a system that was not built to produce the record in the first place.

BOX 3 · FOUR COMPONENTS OF DECISION EVALUATION

1. Constraint adherence: did the decision respect the hard layer's rules and the authority bounds? *Verifiable* mechanically from the log.

2. Decision quality: was the soft reasoning sound given what was knowable? *Reviewable* by a human or a critic agent against the recorded rationale.

3. Calibration: over many decisions, did stated confidences match realized frequencies? When the agent said "70% likely," did it happen about 70% of the time? *Statistical*, over volume.

4. Outcome / ROI: the realized return, attributed to the decision. *Statistical*, meaningful only over many comparable decisions.

Calibration deserves emphasis, because it is the one component that recovers signal from a single agent making many decisions, none of which is individually conclusive. An agent that is well-calibrated (whose stated 70% confidences come true about 70% of the time across hundreds of decisions) can be trusted to weigh its own uncertainty, regardless of how any individual call turned out. The scoring discipline here is well established: Tetlock's work on superforecasting and the Good Judgment Project established that calibration and Brier scores, computed over many recorded forecasts, separate genuine judgment from confident noise.[39] Calibration is decision quality made measurable in aggregate, and the decision ledger is precisely the record of recorded forecasts that the Brier score needs.

9.2 A/B testing graduates from interfaces to decisions

Online firms long ago learned to randomize *interface* variants and measure the difference: the practice Thomke documents as a culture of disciplined business experimentation.[19] The missing ledger is what kept that practice trapped at the surface. Once decisions are recorded as structured policies, the same machinery applies one level deeper: to *decisions themselves*. We can run randomized *policy* experiments (route a fraction of cases through a revised SOP and compare), turning A/B testing from a tool for buttons into a tool for judgment.

And we need not always run the experiment live. **Off-policy evaluation** (OPE) is the established technical name (in reinforcement learning, recommender systems, and online advertising) for exactly this problem: estimating how a *new* decision policy would perform from the logged data of decisions made under the *old* one.[21] It reweights the historical record to simulate the counterfactual without ever exposing a customer to the untested policy. We invoke OPE by name deliberately, because it is the term a machine-learning-literate reader will expect for "evaluate decisions from the recorded ledger," and because it makes the claim precise: the decision ledger is the off-policy log, and the organization's SOP is the policy being evaluated. The same causal apparatus that lets us reason about interventions from observational data[22] lets us ask, of the decision ledger, "what if we had decided differently?"

9.3 The coupling: evaluation method follows the codifiability tier

There is a deep and useful symmetry between §8 and §9. The very axes that make a decision *codifiable* (above all observability and frequency) are the axes that make it *statistically measurable*. A decision that recurs thousands of times with observable inputs can be both encoded as a soft-layer policy and evaluated by off-policy ROI estimates with real confidence intervals. A decision that happens twice a decade can be neither. Evaluation method is therefore *gated by the codifiability tier*: frequent, observable decisions get statistical, off-policy ROI evaluation; rare, irreducible decisions get qualitative decision-quality review against their recorded rationale. The framework tells you not only how to make a decision but how you will be able to grade it.

BOX 4 · SOFT-LAYER ROI

For a recurring decision handled by the soft layer, define the return on the agent's judgment as:

$$\text{soft-layer ROI} = V(\pi_{\text{agent}}) - V(\pi_{\text{baseline}}) - C_{\text{reasoning}}$$

where $V(\pi)$ is the expected value realized under policy π (estimated live or off-policy), π_{baseline} is the prior rule or human default, and $C_{\text{reasoning}}$ is the cost of the agent's deliberation. The soft layer earns its place only when this quantity is reliably positive: the decision-intelligence test for whether judgment should be delegated to an agent at all.

10 Toward Self-Improving Organizations

Once decisions are recorded and evaluable, a loop closes that has never closed in management before. Agents can read the ledger of past decisions and their evaluations, learn which policies earned positive soft-layer ROI, and iterate: revising SOPs, tightening calibration, promoting successful off-SOP patterns into first-class procedures. This is the agentic self-improvement loop: sense, decide, act, evaluate, revise. An organization built this way improves its policy continuously, rather than merely executing it, learning from its own recorded experience: the path toward genuinely agentic, more autonomous organizations.

We state this with deliberate restraint. The loop is bounded by everything that came before it. The soft layer's judgment remains unverifiable in the strong sense; the irreducible-human tier does not shrink to zero; and a self-improving system optimizing a mis-specified objective improves in the wrong direction faster. Self-improvement is a consequence of the framework, not a promise that exceeds it.

11 Extensibility to Personal Decision Intelligence

The logic of the missing ledger is not peculiar to firms. An individual, too, makes a stream of consequential decisions (about money, career, health, relationships) and records almost none of them: not the alternatives, not the rationale, not the beliefs held at the time. The same remedy applies at the scale of one person. A personal ledger that observes the decision, records its context and reasoning, and later evaluates quality and calibration apart from outcome luck turns a life of unexamined choices into a dataset one can actually learn from. Decision Intelligence is, at bottom, the discipline of not throwing away the record of one's own choices, and that discipline scales down to the individual as cleanly as it scales up to the enterprise.

12 Limitations

The framework rests on several load-bearing assumptions, and candor requires naming where each could give way.

- **Ontology liveness.** The entire edifice presumes the recorded state can be kept true. Agent reconciliation (§7) is our proposed answer, but it is an empirical bet, not a proof; an Ontology that silently drifts from reality poisons every decision that reads it.
- **The limits of codifiability.** The most consequential organizational decisions (the rare, strategic, tacit ones) sit firmly in the irreducible tier. The framework instruments and records them but cannot automate or statistically evaluate them, and these are often the decisions that matter most.
- **Unverifiable soft judgment.** The soft layer depends on LLM reasoning, which is neither reliable nor verifiable in the way a hard rule is. Minimizing the judgment surface contains this risk; it does not eliminate it.
- **Statistics need volume.** Off-policy ROI, calibration, and randomized policy tests all require many comparable decisions. High-stakes, low-frequency decisions resist statistical evaluation precisely where rigor would be most valuable. For the genuinely single-shot decision the framework shifts mode rather than failing: it grades decision *quality* rather than outcome, reasons in Bayesian belief rather than long-run frequency, borrows base rates from a reference class, and scores the decision-maker's calibration across many heterogeneous one-shots even when no single one repeats.
- **A changing world (non-stationarity).** Off-policy evaluation assumes the environment that produced the ledger still holds: that past and present decisions face the same dynamics. A shift in the macroeconomy, in tariffs or trade policy, in interest rates, or in the geopolitical backdrop can change the dynamics underneath a business almost overnight (as many have experienced firsthand in recent years), alongside the slower drift of markets and competitors. A policy can then look good or bad in the record for reasons that have nothing to do with its quality and everything to do with a shifted regime. Recording the environmental context as part of the state (so drift becomes movement *through* the state space rather than a change *of* it), weighting recent decisions more heavily, and detecting regime breaks all help, but cleanly separating "the policy got worse" from "the world changed" is the fundamental confound of all longitudinal decision data. It can be bounded, not eliminated.
- **Goodhart's law.** Once decision quality is measured, the measure becomes a target, and people and agents will optimize the metric rather than the underlying judgment. Instrumenting decisions changes how decisions are made (sometimes for the worse), and the design must anticipate the distortion it induces.

13 Conclusion

Decisions are the missing ledger of the organization. We book every dollar and not a single choice, and so the knowledge we have about management is a knowledge of stories (outcomes celebrated, strategies back-filled, luck mistaken for skill) rather than a knowledge of evidence. The remedy is instrumentation rather than better storytelling: make the decision observable through an explicit ontology of state, record it with its context and rationale, and evaluate its quality and calibration apart from the luck of its result. The theory that decisions can be optimized this way is old; what is new is the ability to supply the telemetry it always needed.

Business-as-Code is that methodology. It renders the organization as a running system: an Ontology of state, a hierarchy of tools and skills, two-layer SOPs that encode both rule and judgment, agents that reconcile the ledger against the world, and an evaluation discipline that grades decisions on their merits. The same hard/soft split organizes authority, process, and interface; the same codifiability axes that decide what can be automated decide what can be measured. None of it requires the organization to be more than it is, only to stop discarding the record of what it does. The aim throughout has been a framework that is rigorous and, above all, *implementable*: a way to turn management from storytelling into decision science, in a firm one could actually build.

The organization has always been a sequential decision system that could see its own state only in part; what it lacked was a *record* of decisions, not a theory of them. The mechanism this paper supplies is that record, produced as a byproduct of execution: the decision ledger as an *automatic emission of executable Business-as-Code*, with the recorded decision as the primary unit of an off-policy evaluation of the organization modeled as a POMDP. You do not build a ledger and hope it gets filled; you run the business through code, and the ledger is what the running leaves behind.

References

1. Simon, H. A. *Administrative Behavior: A Study of Decision-Making Processes in Administrative Organizations.* Macmillan, 1947 (4th ed., Free Press,

1997).

2. Cyert, R. M., and March, J. G. *A Behavioral Theory of the Firm.* Prentice-Hall, 1963.
3. March, J. G. *A Primer on Decision Making: How Decisions Happen.* Free Press, 1994.
4. Rosenzweig, P. *The Halo Effect: ... and the Eight Other Business Delusions That Deceive Managers.* Free Press, 2007.
5. Mauboussin, M. J. *The Success Equation: Untangling Skill and Luck in Business, Sports, and Investing.* Harvard Business Review Press, 2012.
6. Duke, A. *Thinking in Bets: Making Smarter Decisions When You Don't Have All the Facts.* Portfolio, 2018.
7. Taleb, N. N. *Fooled by Randomness: The Hidden Role of Chance in Life and in the Markets.* Random House, 2001.
8. Kahneman, D. *Thinking, Fast and Slow.* Farrar, Straus and Giroux, 2011.
9. Raiffa, H. *Decision Analysis: Introductory Lectures on Choices under Uncertainty.* Addison-Wesley, 1968.
10. Howard, R. A. "Decision Analysis: Practice and Promise." *Management Science* 34(6), 1988, pp. 679–695.
11. Bellman, R. *Dynamic Programming*. Princeton University Press, 1957.
12. Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, 1994.
13. Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. "Planning and Acting in Partially Observable Stochastic Domains." *Artificial Intelligence* 101(1–2), 1998, pp. 99–134.
14. van der Aalst, W. M. P. *Process Mining: Data Science in Action.* 2nd ed., Springer, 2016.
15. Object Management Group. *Decision Model and Notation (DMN), Version 1.3.* OMG, 2019.
16. Polanyi, M. *The Tacit Dimension.* University of Chicago Press, 1966.
17. Nonaka, I. "A Dynamic Theory of Organizational Knowledge Creation." *Organization Science* 5(1), 1994, pp. 14–37.
18. Pak, C. "Taming the Confused Deputy: The Agent Proxy Pattern for Bounded Delegated Authority." Working paper, 2026.
19. Thomke, S. H. *Experimentation Works: The Surprising Power of Business Experiments.* Harvard Business Review Press, 2020.
20. Pacioli, L. *Summa de Arithmetica, Geometria, Proportioni et Proportionalita.* Venice, 1494. (First printed treatment of double-entry bookkeeping.)
21. Dudík, M., Langford, J., and Li, L. "Doubly Robust Policy Evaluation and Learning." *Proc. ICML*, 2011.
22. Pearl, J. *Causality: Models, Reasoning, and Inference.* 2nd ed., Cambridge University Press, 2009.
23. Pak, C. "Business Engineering: Business-as-Code." *reboot.mba*, 2026.
24. Pratt, L. *Link: How Decision Intelligence Connects Data, Actions, and Outcomes for a Better World.* Emerald Publishing, 2019. (Pratt developed and popularized Decision Intelligence as a discipline at Quantellia (originally "decision engineering," c. 2008; rebranded "decision intelligence," c. 2013) and developed Causal Decision Diagrams.)
25. Kozyrkov, C. Writings on decision intelligence as Google's first Chief Decision Scientist, defining it as the discipline of turning information into better actions at any scale. Google adopted the term *decision intelligence* circa 2018.
26. Gartner. *Magic Quadrant for Decision Intelligence Platforms.* Gartner, Inc., 2026 (inaugural). Defines DIPs as software to support, automate, and augment decision-making of humans or machines, including decision modeling, orchestration of execution flows, and evaluation, governance, and auditing of decision outcomes.
27. Chen, J., and Gupta, J. "Service-as-Software: The $4.6 Trillion Opportunity." Foundation Capital, 2024. (AI agents take ownership of outcomes formerly delivered by human labor.)
28. Clevenger, N., and Drivly / Platform.do. "Business-as-Code" and "Services-as-Software." Public materials, 2025. (AI agents executing business logic defined as code; workflows and SOPs authored in Markdown.)
29. Boston Consulting Group. "Enterprise as Code." BCG, December 2025. (An operating model for the AI era; defining operations as code for human-machine collaboration.)
30. Rothmann, D. (42futures). "Company as Code." February 2025. (A declarative, infrastructure-as-code–style DSL for roles, policies, and compliance with graph relationships; a versioned, queryable, testable digital representation of the company.)
31. Amazon Web Services. "Agent SOPs" in the Strands Agents framework. AWS, November 2025. (Natural-language / Markdown standard operating procedures that agents run.)
32. Decagon. "Agent Operating Procedures (AOPs)." 2025. (Natural-language SOP intent compiled into validated agent workflows.)
33. Aera Technology. "Agentic Decision Intelligence," the Decision Data Model, and the Control Room. Aera Decision Cloud, 2022–2024. (Captures decisions, context, actions, and outcomes as organizational memory; logs reasoning, trade-offs, and provenance and tracks recommendation adoption, human overrides, and outcomes against objectives.)
34. Koenders, W. "The Missing Link in Data-Driven Organizations: Decision Traceability." Medium, April 2025. (Argues organizations capture no data on their own decisions; proposes a centralized decision inventory with generative-AI retrieval.)
35. Singh, R. "The Decision Ledger: How AI Becomes Defensible, Auditable, and Enterprise-Ready." January 2026. (Captures decision intent, evidence, controls, ownership and approvals, model/policy/tool versions, and outcomes for governance and forensic auditability of autonomous AI decisions.)
36. Nygard, M. "Documenting Architecture Decisions." 2011. (Architecture Decision Records: a lightweight per-decision record of title, status, context, decision, and consequences; the collection forms a decision log.)
37. Gartner. "Digital Twin of an Organization (DTO)." Gartner, Inc., introduced 2017. (A dynamic software model of an organization.)
38. Spetzler, C., Winter, H., and Meyer, J. *Decision Quality: Value Creation from Better Business Decisions.* Wiley, 2016. (Strategic Decisions Group; the six elements of decision quality, evaluated independently of outcome.)
39. Tetlock, P. E., and Gardner, D. *Superforecasting: The Art and Science of Prediction.* Crown, 2015. (The Good Judgment Project; calibration and Brier scores as the scoring discipline for recorded forecasts.)
40. Pfeffer, J., and Sutton, R. I. *Hard Facts, Dangerous Half-Truths, and Total Nonsense: Profiting from Evidence-Based Management.* Harvard Business School Press, 2006.
41. Martin, R. L. "Rethinking the Decision Factory." *Harvard Business Review*, October 2013. (The organization as a factory that produces decisions.)