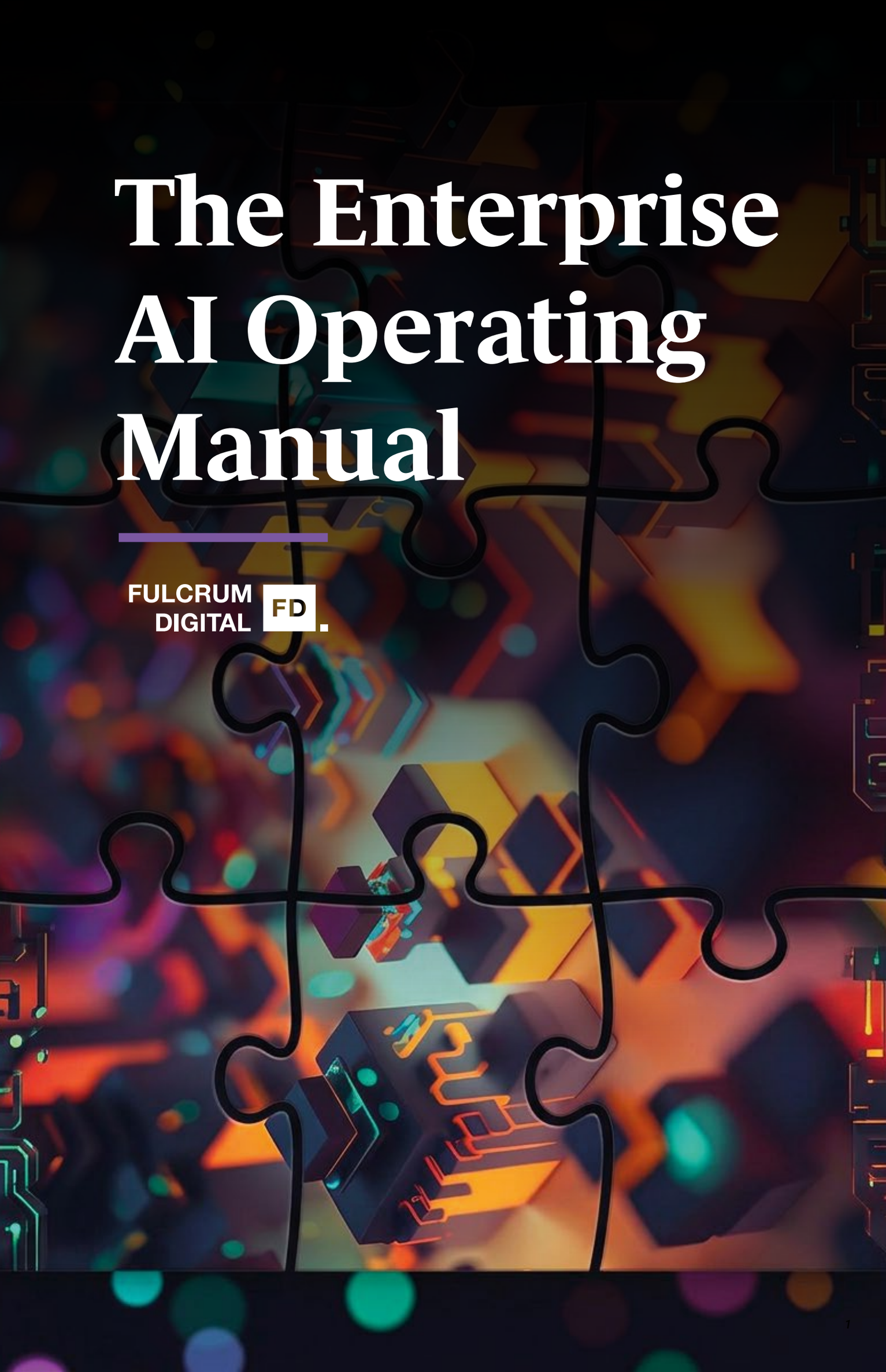
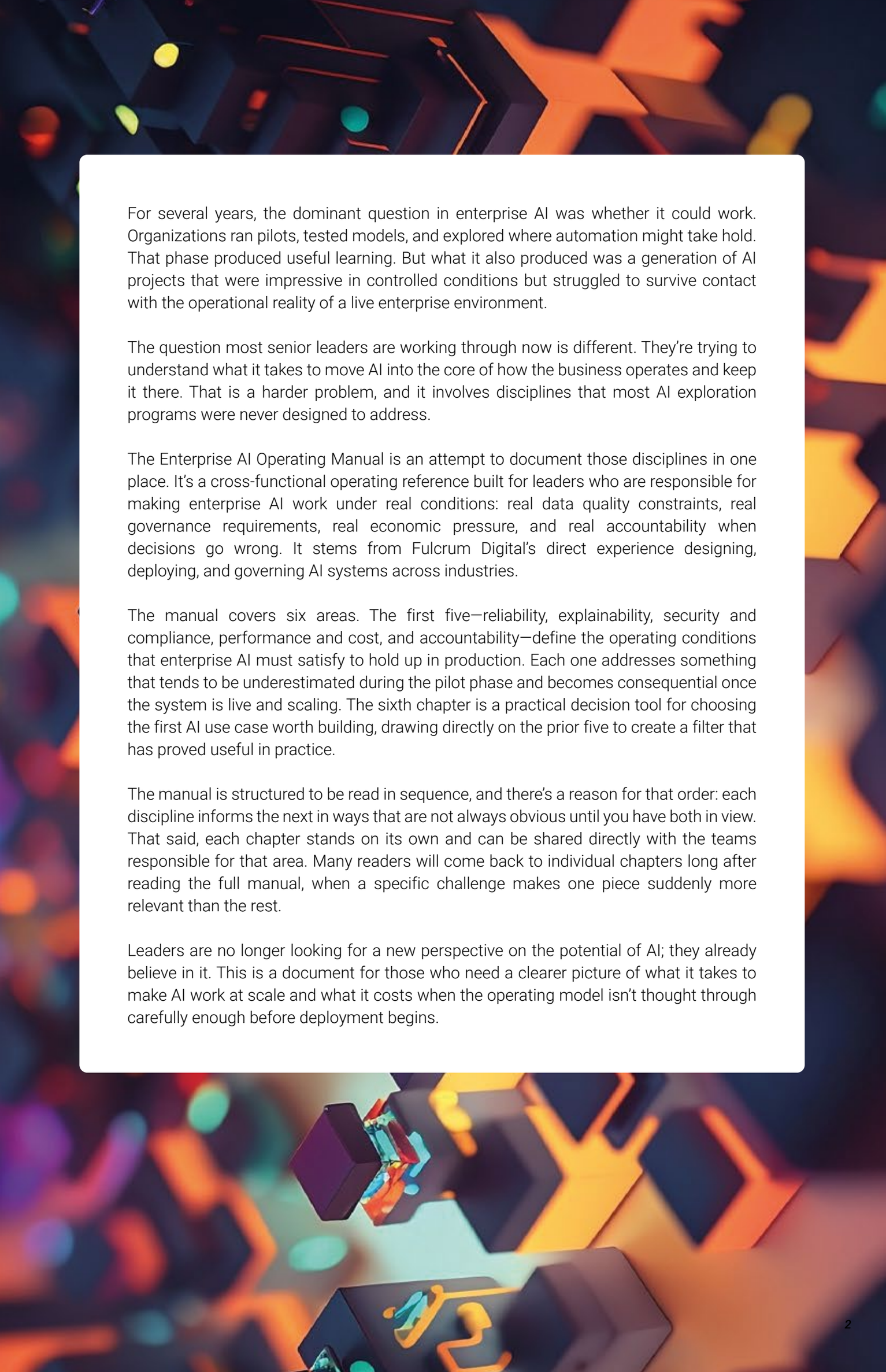
The Enterprise AI Operating Manual

FULCRUM DIGITAL FD.





For several years, the dominant question in enterprise AI was whether it could work. Organizations ran pilots, tested models, and explored where automation might take hold. That phase produced useful learning. But what it also produced was a generation of AI projects that were impressive in controlled conditions but struggled to survive contact with the operational reality of a live enterprise environment.

The question most senior leaders are working through now is different. They're trying to understand what it takes to move AI into the core of how the business operates and keep it there. That is a harder problem, and it involves disciplines that most AI exploration programs were never designed to address.

The Enterprise AI Operating Manual is an attempt to document those disciplines in one place. It's a cross-functional operating reference built for leaders who are responsible for making enterprise AI work under real conditions: real data quality constraints, real governance requirements, real economic pressure, and real accountability when decisions go wrong. It stems from Fulcrum Digital's direct experience designing, deploying, and governing AI systems across industries.

The manual covers six areas. The first five—reliability, explainability, security and compliance, performance and cost, and accountability—define the operating conditions that enterprise AI must satisfy to hold up in production. Each one addresses something that tends to be underestimated during the pilot phase and becomes consequential once the system is live and scaling. The sixth chapter is a practical decision tool for choosing the first AI use case worth building, drawing directly on the prior five to create a filter that has proved useful in practice.

The manual is structured to be read in sequence, and there's a reason for that order: each discipline informs the next in ways that are not always obvious until you have both in view. That said, each chapter stands on its own and can be shared directly with the teams responsible for that area. Many readers will come back to individual chapters long after reading the full manual, when a specific challenge makes one piece suddenly more relevant than the rest.

Leaders are no longer looking for a new perspective on the potential of AI; they already believe in it. This is a document for those who need a clearer picture of what it takes to make AI work at scale and what it costs when the operating model isn't thought through carefully enough before deployment begins.

Index

Chapter 1

Reliability

Continuity across data, time, and change

Quick Read

AI only works in the real world when it works every day, under real load, with real customers, across messy systems and even messier teams. Reliability is the difference between an impressive demo and an AI program your business can actually bet on. It decides whether AI becomes a multiplier or another expensive experiment you can't defend in the next board review.

The truth is simple: **most AI failures aren't "AI failures" at all; they're breakdowns in data plumbing, brittle integrations, human bandwidth, operational discipline, or unclear ownership.** Reliability is the layer that keeps all of that from sinking the investment. It gives executives something rare in this space: predictability, defensibility, and room to scale without fear.

If you run a P&L

- Reliability determines whether AI lifts margin or adds hidden operational debt your teams quietly absorb.
- It's what stops drift, outages, and rework from silently eroding ROI quarter after quarter.

If you're accountable to customers

- Reliability is the line between "AI-enhanced experience" and "escalation waiting to happen."
- Your customers won't remember why the failure happened. Only that it happened on your watch.

If you answer to a board or regulator

- Reliability is what makes your AI decisions defensible: traceable data, governed models, auditable operations, and accountable human oversight.
- It's the only foundation strong enough to support scale, scrutiny, and long-term trust.

Bottom line: Reliability is not a technical feature. It's an enterprise safeguard. Get this right, and every other part of your AI strategy has room to succeed. Get it wrong, and nothing you build will behave the same way tomorrow as it does today.

Definition & Context

In engineering, reliability has always meant one thing: performance without failure, for as long as the system is expected to function.

Classical reliability theory, originally defined in hardware design, treated it as a function of failure rate, time, and operating environment. A component was reliable if it could perform its intended task under specified conditions for a specified period. Over the years, this thinking evolved into software reliability, emphasizing code stability, fault tolerance, and mean time to failure.

In the age of AI, the definition of reliability builds on classical software principles. Kaur and Bahl *(2023)* define it as "the probability of the failure-free operation for a specified period of time in a specified environment," emphasizing three elements: failure, time, and environment. In AI systems, those same variables still define reliability, but each behaves differently as data and context evolve.

Building on that foundation, modern standards such as NIST's AI Risk Management Framework treat reliability as a pillar of trustworthiness, alongside safety, explainability, and robustness. They emphasize that dependable AI isn't judged only by accuracy, but by consistency of performance across time, context, and operating conditions.

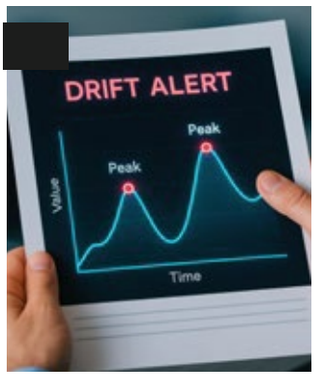
In practice, that means reliability now stretches from mechanical durability to digital persistence. It covers the ability of an AI system to continue performing as designed, even as the data, deployment environment, or business context evolve. It requires architectures that anticipate drift, infrastructure that monitors degradation, and governance that links performance to accountability.

The research community calls this resilience in behavior. For enterprises, it's something simpler: **the confidence that your system will still make the right call when no one is watching.**

But even with clear definitions and maturing standards, reliability breaks down in practice for reasons that have little to do with algorithms and everything to do with how AI is treated inside enterprises.

What People Miss

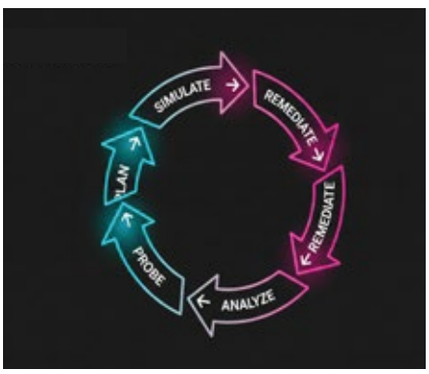
Despite decades of reliability research, AI systems still fail in familiar ways. Much of this stems from treating learning systems like static software rather than dynamic, data-dependent organisms. And the consequences show up early: MIT's 2025 NANDA initiative found that while expectations around AI are high, only a narrow fraction of pilots convert into sustained value. Reliability gaps, not ambition, drive most of that stall. The patterns behind this stall are consistent across research and enterprise deployments.



Accuracy mistaken for reliability

In conventional testing, a model that performs well on benchmark data is considered "ready." But **reliability** isn't about performance once; **it's about performance again and again as the input world shifts.** In production, even well-trained models tend to lose precision over time as input patterns drift and feedback loops reshape their context. Vela, D., Sharp, A., Zhang, R. et al. (2022) call this "AI aging": the temporal degradation of model quality that occurs when retraining lags behind data volatility. **The failure here is less about the algorithm and more about the assumption that short-term accuracy guarantees long-term dependability.**

Reliability treated as static



Traditional reliability engineering **focuses on mean time between failures. AI systems don't fail that way. They drift.** Their behavior changes gradually, often invisibly, until outputs no longer reflect reality. As RAND's work on algorithmic resilience notes, systems can maintain perfect operational uptime while their decision quality degrades under the radar. Operationally flawless but functionally unreliable. **Reliability isn't a status; it's a continuous relationship between model, data, and context.**

Context ignored

Reliability doesn't live inside the model alone. It depends on data governance, process discipline, and human oversight. **A 2023 Cornell study on hallucination in large language models found factual inconsistencies in about 30% of BART-generated summaries and 27% of those from PEGASUS, triggered by small changes in prompts or retrievals.** That's reliability in its most delicate form: context becoming both the enabler and the source of fragility. Add bias, feedback loops, and dependency on user input, and reliability quickly erodes unless contextual integrity is deliberately maintained.

The market is already responding to this volatility. The 2024 Lucidworks Global Generative AI Benchmark Report notes a sharp year-over-year correction in planned AI investment. **Only 63% of organizations expected to increase spending in 2024, down from 93% in 2023.** This reflects a shift toward more evidence-driven expectations of system reliability.

That is why grounded architectures matter.

Techniques like Retrieval-Augmented Generation (RAG) offer a glimpse of what disciplined reliability looks like in practice. Broader analyses from Stanford and Cornell show that large language models still produce factual errors in anywhere from one-third to nearly nine-tenths of domain-specific outputs, figures that have already cooled enterprise confidence in unsupervised generation. RAG mitigates that fragility by grounding generative systems in verifiable, time-aware data rather than static memory. It's an early signal that reliability can be designed, not just maintained, through smarter architectures.

The 10 Reliability Killers Executives Keep Running Into

Data Drift

When inputs shift quietly, your model stays confident while getting progressively wrong.

Overconfident Models

Systems that never surface uncertainty force teams to trust outputs that should have been flagged as low-confidence.

Bottlenecked Human Review

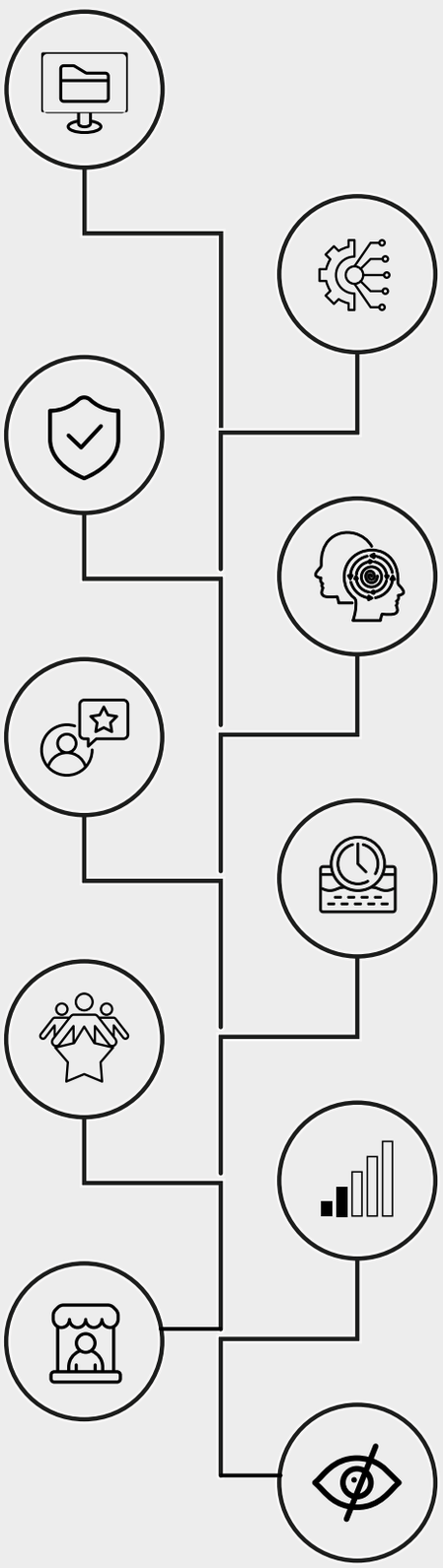
HITL moments exist on paper, but bandwidth, unclear ownership, or missing skills turn them into rubber stamps.

Operational Blind Spots

Teams can't see latency spikes, drift patterns, or error surges until customers feel it first.

Vendor Overpromising

Tools marketed as "autonomous" require babysitting, manual fixes, and constant patching behind the scenes.



Shadow Integrations

Hidden scripts, brittle connectors, and hand-built bridges introduce silent failures no dashboard will warn you about.

Hallucinations Masquerading as Insight

Generated responses that sound plausible but are actually wrong and still flow downstream into decisions.

Model Aging

Performance decays quietly as real-world patterns evolve, leaving your system "accurate" only in last year's context.

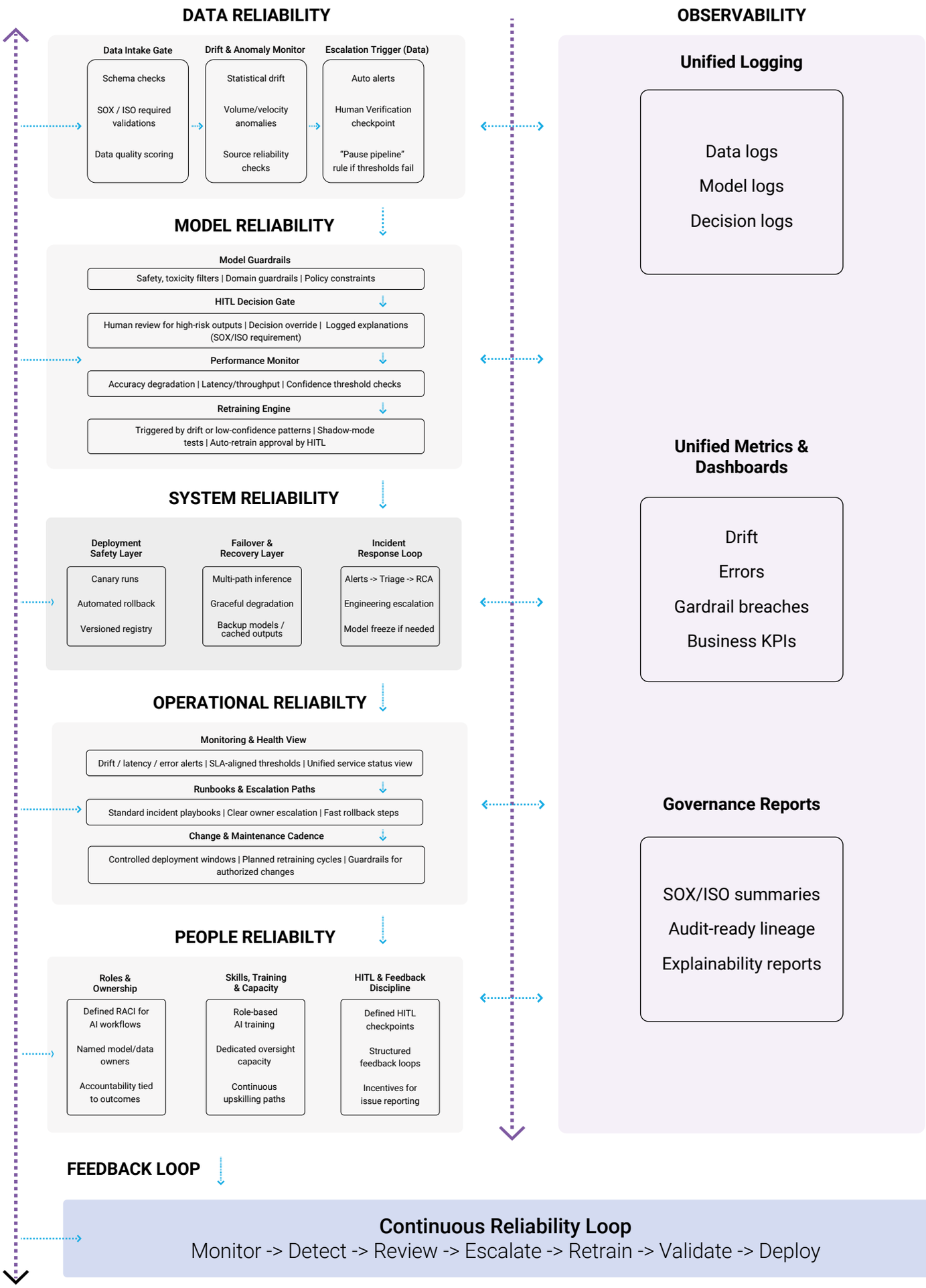
Weak Escalation Paths

Issues route to inboxes, not owners, causing delays, duplicated work, and unresolved risks.

Compliance Ambiguity

Unclear accountability for audits, logs, review steps, and decision traceability creates gaps you only discover during an investigation.

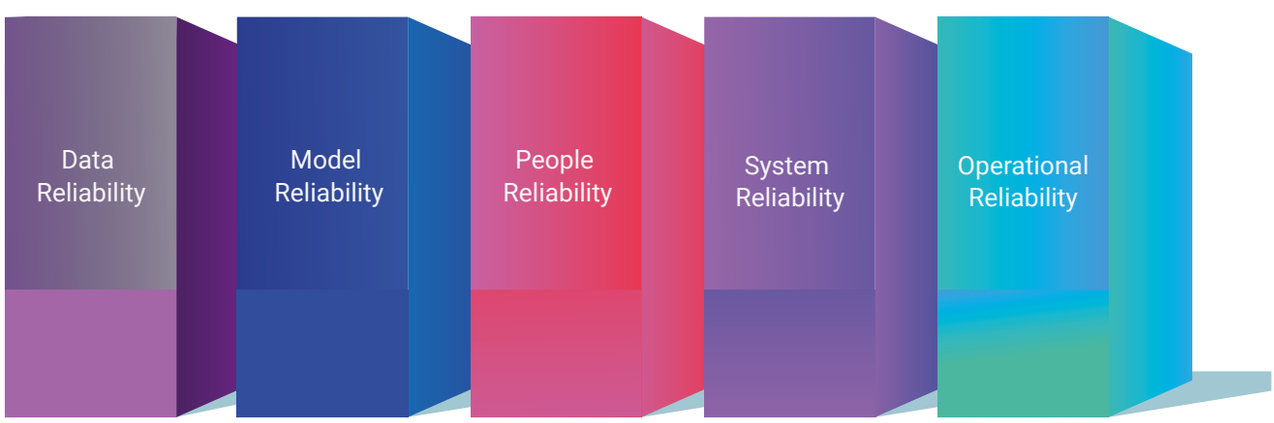
The Fulcrum AI Reliability Layers



How to Do It Right

The antidote to these failure modes is not a single technique but a way of operating; habits engineered into every layer of the system. The organizations that sustain reliability the longest treat it as a design principle, not a repair function. Their success usually rests on four interlocking pillars: data reliability, model reliability, system reliability, operational reliability, and people reliability.

Each pillar addresses a different point of failure, but together they form a loop: clean data enables stable models; stable models demand resilient systems; resilient systems rely on disciplined human oversight. The following examples show how mature enterprises have made these principles tangible and what similar rigor can achieve elsewhere.



CLEAN DATA ENABLES STABLE MODELS
STABLE MODELS DEMAND RESILIENT SYSTEMS
RESILIENT SYSTEMS RELY ON DISCIPLINED HUMAN OVERSIGHT

1.Data Reliability: Keeping the Inputs Honest

AI reliability begins where data enters the system. When inputs drift or degrade, no amount of model tuning can compensate. In many enterprises, this erosion begins not in the data itself but in the systems that supply it: fragmented sources, inconsistent integrations, and legacy handoffs that introduce silent inconsistencies into the pipeline. Airbnb's Customer Data Infrastructure (CDI) offers a blueprint for large-scale data reliability. Handling over 1.5 petabytes of customer data, Airbnb uses its CDI to unify information across booking, search, and support systems while maintaining quality and consistency. Automated quality scoring, anomaly detection, and continuous monitoring ensure that errors are surfaced and resolved before they affect downstream AI workflows. Together, these practices turn data reliability from a static compliance check into an operational reflex.

Our reliability work begins at the data layer, where pipelines, quality controls, and security checks determine whether downstream models will behave predictably. In production environments, this translates into disciplined preprocessing, classification, and validation flows that keep inputs stable even as data sources evolve.

- Structured ingestion pipelines with schema validation, standardization, and normalization
- Data cleaning, outlier detection, missing value handling, and temporal alignment through AI Ops
- PII masking, tokenization, and classification integrated into upstream data governance
- Unified workflows that convert unstructured sources into structured, AI-ready formats
- Pre-deployment data-flow security checks to catch integrity issues before they reach models

SUMMARY Data Reliability

Reliable AI starts with reliable inputs. If the data feeding the system is incomplete, inconsistent, or distorted by fragmented sources or weak system integrations, nothing downstream can be trusted.

Airbnb's Customer Data Infrastructure shows what "grown-up" data reliability looks like at scale: unified flows across products, continuous quality checks, and monitoring that catches issues before they hit customer-facing decisions.

FULCRUM IN PRACTICE

Structured ingestion pipelines and integrations keep data flowing consistently across systems.

AI Ops handles cleaning, outlier detection, and temporal alignment so models aren't compensating for broken inputs.

PII masking, tokenization, and upstream classification embed governance and security into the data path.

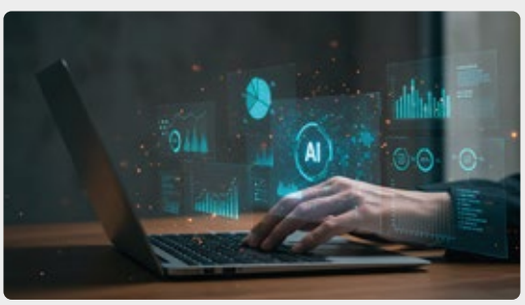
Takeaway: Make data quality an operational responsibility, not an IT clean-up task.

2.Model Reliability: Ensuring the Math Ages Gracefully

Models don't just need accuracy; they need endurance. JPMorgan Chase's AI Threat Modeling Co-Pilot (AITMC) is part of the bank's broader AI-risk-governance strategy, combining generative-AI-driven automation with expert oversight. By integrating threat-modeling tradecraft into its co-pilot, JPMorgan Chase enables engineers to identify potential vulnerabilities earlier and more efficiently across the software-development lifecycle, achieving roughly 20% faster modeling and uncovering additional novel threats per analysis. Reliability here isn't incidental; it's engineered into the model-governance lifeycle itself.

We treat every model as a governed asset: versioned, monitored, stress-tested, and tracked for drift, bias, and performance aging. Reliability here comes from lifecycle discipline: validating behavior before deployment, observing degradation early, and ensuring interventions happen before quality slips.

- Version-controlled model lifecycle with approval workflows, A/B tests, and shadow-mode evaluation
- Continuous drift detection, hallucination monitoring, and accuracy/latency threshold checks
- Bias and fairness audits with protected-attribute handling and explainability tools (LIME/SHAP)
- Prompt and model-level security tests to prevent degradation from injection or misuse
- Retraining triggers tied to drift patterns, confidence drops, and HITL-reviewed exceptions ycle itself.



SUMMARY Model Reliability

A reliable model isn't just accurate on day one. It stays accurate as data, patterns, and operating conditions evolve.

JPMorgan Chase's AI Threat Modeling Co-Pilot shows what disciplined model governance looks like: versioning, automated checks, expert oversight, and faster identification of emerging vulnerabilities.

FULCRUM IN PRACTICE

Versioned, approval-based model lifecycles keep behavior predictable through each release.

Drift, hallucination, and performance monitoring ensure degradation is caught early, not after impact.

Bias audits and explainability checks make model decisions defensible across compliance, audit, and customer-facing contexts.

Takeaway: If the model is learning, aging, and adapting, your governance needs to do the same.

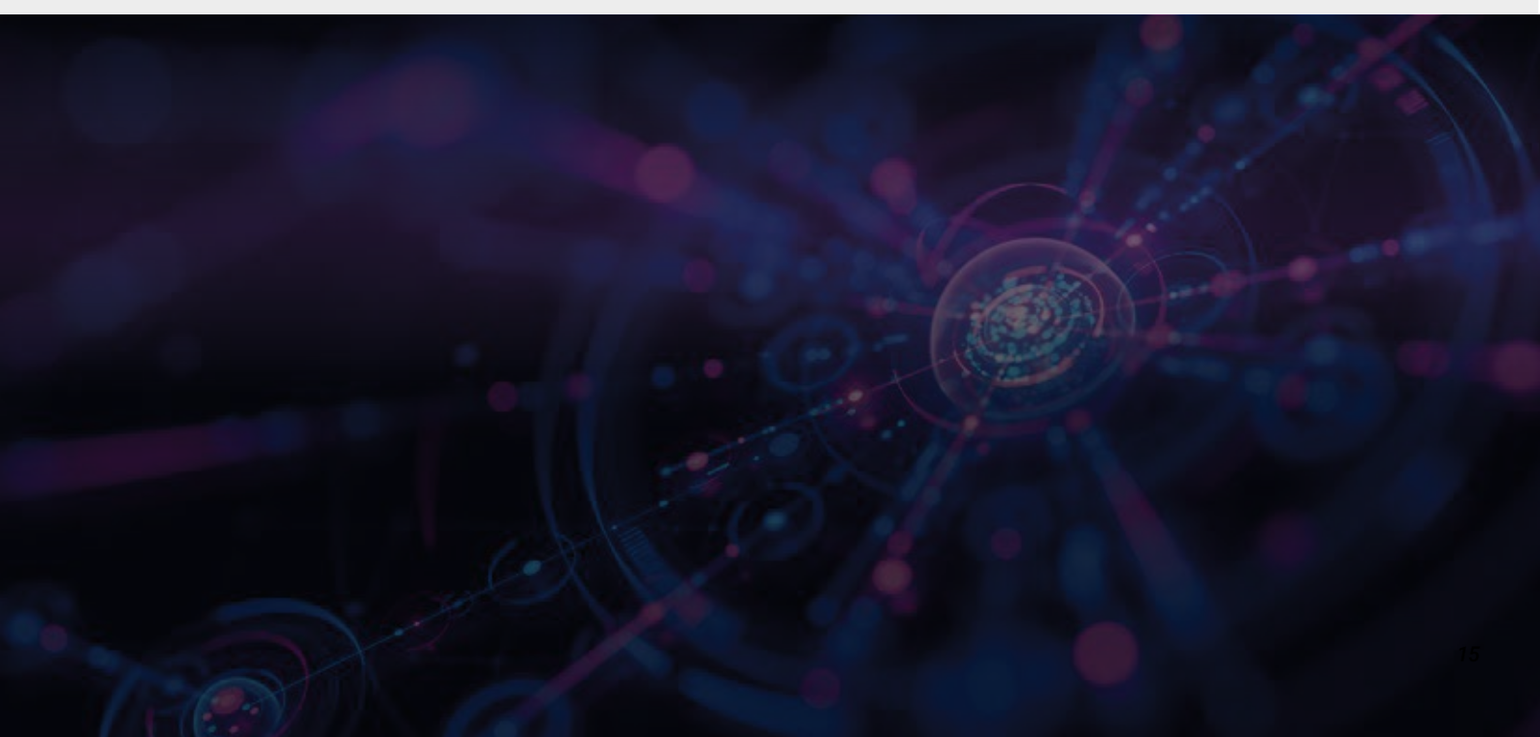
3.System Reliability: Designing for Failure, Not Around It

Even the most robust models fail if the systems around them can't absorb shocks. Netflix's Chaos Monkey, part of its Simian Army testing suite, demonstrates how reliability is tested deliberately, not left to chance. By injecting controlled faults into live microservices environments, Netflix validates its infrastructure's ability to recover before users ever notice. For AI architectures, the analogy is clear: resilience must be built into deployment pipelines and tested under stress, not assumed from uptime metrics.

Much of this resilience depends on the reliability of the integration fabric (APIs, connectors, and service dependencies) because a single brittle link can cause a cascading failure even when the model itself is perfectly healthy.

We build multi-layered resilience into how agents run, recover, and roll forward, ensuring that quality holds even when individual components encounter stress, load, or unexpected conditions.

- Canary deployments, automated rollbacks, and versioned registries for controlled releases
- High-availability configurations with failover paths, graceful degradation, and cached fallback outputs
- Load and resilience testing across latency, throughput, and error-rate thresholds
- Real-time anomaly detection supported by distributed tracing and log aggregation
- Incident response workflows covering alerting -> triage -> root-cause analysis -> resolution



SUMMARY System Reliability

AI systems stay reliable only when the surrounding infrastructure is designed to fail safely and recover fast.

Netflix's Chaos Monkey proves that resilience is engineered, not assumed, by testing failures in production so the system strengthens before users ever feel the impact.

FULCRUM IN PRACTICE

Canary releases, automated rollbacks, and versioned registries ensure controlled, low-risk deployment.

High-availability and graceful-degradation patterns keep AI workflows stable under stress or spikes.

Real-time anomaly detection and structured incident workflows catch issues before they cascade into outages, including failures triggered by brittle connectors or integration points. .

Takeaway: Design your AI systems to expect failure and rehearse recovery before customers ever see a flaw.

4. Operational Reliability: When People and Systems work together

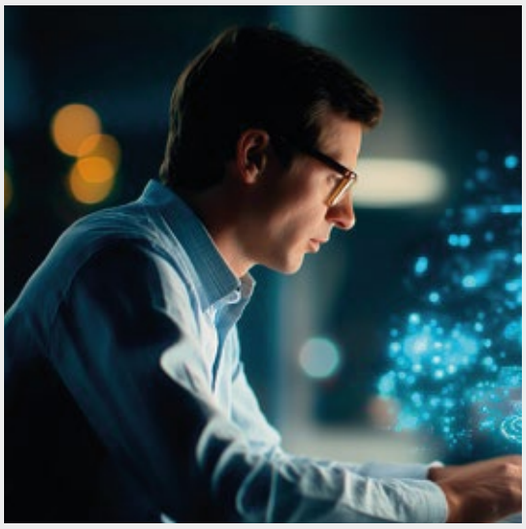
Reliable AI depends on reliable operations. Uber's Michelangelo platform unifies the end-to-end ML lifecycle—managing data, training, deployment, prediction, and monitoring—across teams and use cases. It provides integrated monitoring and visibility so that engineers can track model behavior in production, assess performance, and retrain when needed. By standardizing workflows and tooling, Michelangelo blends automation with human judgment, turning observability into collaboration. Operational reliability, at its best, isn't about control; it's about continuous alignment between people, models, and outcomes.

Reliable operations also require clear human checkpoints. In many enterprise environments, especially those governed by SOX- or ISO-based oversight, certain decisions still need a verified human review through well-defined HITL moments. Not as a workaround, but as an accountability layer that prevents unverified outputs from moving downstream and creates discipline around when humans must intervene and when the process can safely absorb automation.

The second half of operational reliability is what happens when something drifts or degrades. This is where continuous monitoring, alerting thresholds, and defined escalation paths matter. Operational failures often emerge not from the model but from overloaded teams, unclear ownership, or workflow bottlenecks, what many organizations experience as operational reliability debt. A system is only as reliable as its ability to recognize when it is no longer performing as intended and route that signal to the right team before it becomes a disruption. Together, this blend of observability, intervention, and escalation keeps the system honest long after deployment.

This is where systems and people reinforce each other. Our operational layer focuses on monitoring, escalation, HITL checkpoints, and governance so that drift, degradation, or user-impacting issues are surfaced and corrected before they propagate.

- Real-time dashboards tracking latency percentiles, throughput, error rates, and accuracy drift
- Incident playbooks with automated alerting, rollback protocols, and severity-based escalation
- HITL checkpoints for high-risk operations, with override and approval workflows aligned to SOX/ISO
- Structured feedback loops feeding evaluation, regression testing, and model refinement
- Governance controls enforcing audit logs, risk assessments, and compliance-ready documentation



SUMMARY Operational Reliability

AI is only as reliable as the day-to-day operations that monitor it, intervene when needed, and keep it aligned with real-world conditions.

Uber's Michelangelo shows what strong operations look like: unified monitoring, standardized workflows, and teams equipped to retrain, override, or correct the system before issues become business-impacting.

FULCRUM IN PRACTICE

Real-time dashboards expose drift, latency, errors, and quality degradation before customers feel the impact.

Defined HITL and escalation workflows ensure high-risk decisions get human review, not blind automation.

Incident playbooks and governance controls keep operations auditable, traceable, and compliant as systems evolve, even when workloads spike or ownership lines blur.

Takeaway: Stabilize the day-to-day, and the models will follow. Operational reliability is where AI becomes sustainable in the hands of stretched teams and real enterprise constraints.

5.People Reliability: When Capability Keeps the System Steady

AI reliability depends on whether the people around the system can sustain it. McKinsey's 2025 research shows that 41% of employees feel apprehensive about AI, often because they're unsure they can maintain or oversee it. In most enterprises, failures emerge from stretched teams, unclear ownership, and weak review discipline, not from the model itself. Alexi's legal research platform illustrates the opposite. Its AI outputs are consistently reviewed, corrected, and reinforced by trained legal experts, creating a stable loop of human judgment and model refinement. That is the core of people reliability: skilled, supported teams who know when to trust the system, when to intervene, and how to keep it aligned with real-world expectations.

People reliability ensures that skilled teams, clear ownership, and predictable workflows keep AI systems steady long after deployment. Our platforms, particularly FD Ryze Infinity, are designed so that human capability, not just model logic, is a core part of how reliability holds under real-world conditions.

- HITL checkpoints embedded in multi-agent orchestration
- Defined override and approval workflows for sensitive or high-risk actions
- Escalation paths and incident protocols that route issues to the right owners
- Structured feedback loops feeding model refinement and continuous improvement
- Role-based dashboards that distribute workload and centralize monitoring signals .

SUMMARY People Reliability

AI stays reliable only when the people who oversee it have the clarity, capacity, and competence to keep the system steady.

Alexi's legal research platform shows how reliability improves when skilled experts consistently review and refine AI outputs, creating a disciplined human–model feedback loop that strengthens over time.

FULCRUM IN PRACTICE

- HITL checkpoints ensure expert oversight is built into multi-agent orchestration.
- Override and approval workflows give teams the authority to intervene at the right moments.
- Role-based dashboards and structured escalation paths keep responsibility clear and workloads manageable.

Takeaway: Invest in the people running the system. Reliability fails fastest where ownership and capability are weakest.

Up Next: Explainability

Reliable AI doesn't emerge from any single control but from the interaction of all five layers: data, models, systems, operations, and people. When these layers reinforce each other, reliability stops being an aspiration and becomes an operating condition. This foundation is what the next piece of the puzzle builds on, because integration only works when the system underneath can be trusted to behave the same way tomorrow as it does today.

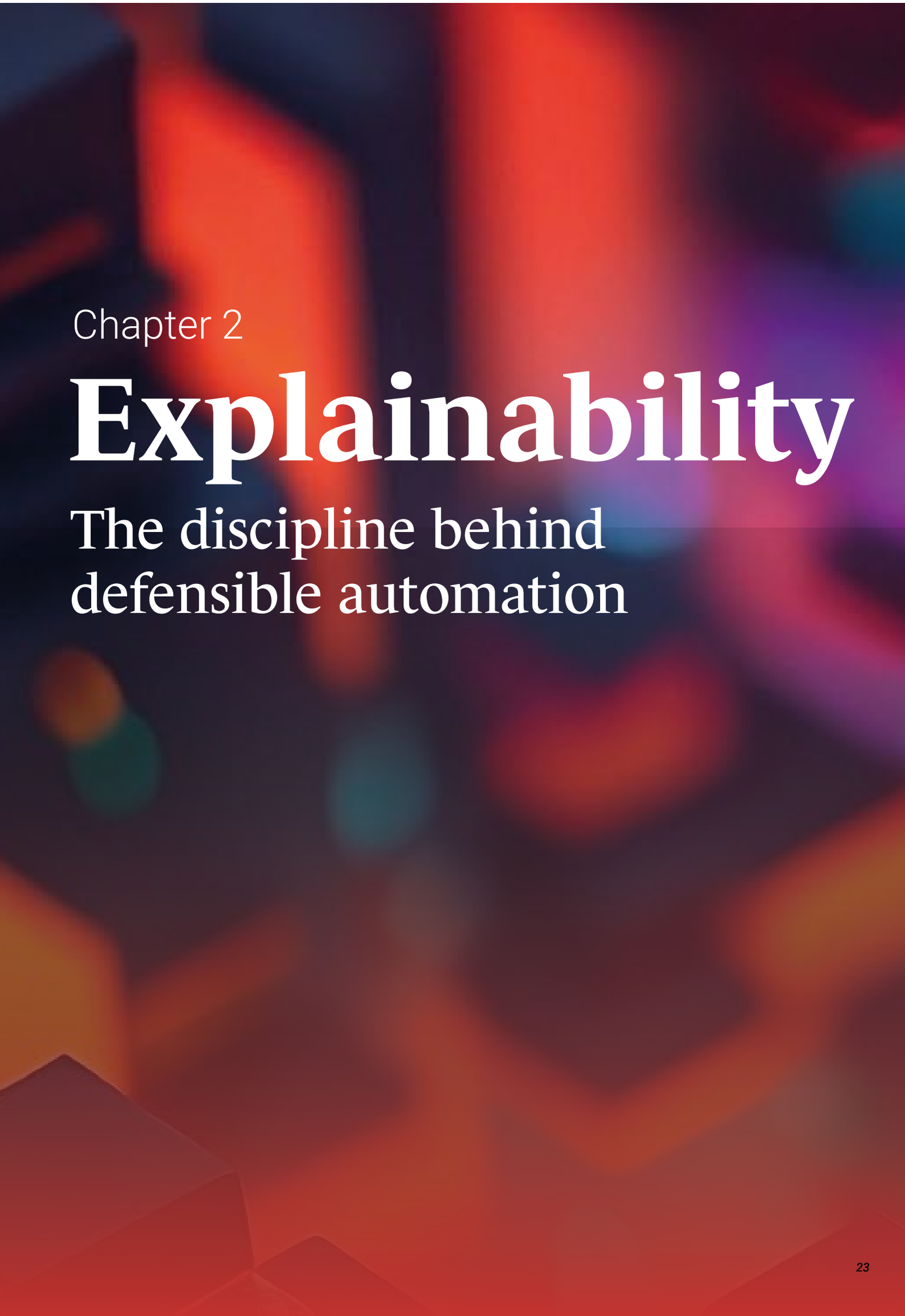
Sources

- The TAILOR Handbook of Trustworthy AI
- Temporal quality degradation in AI models
- The Root Causes of Failure for Artificial Intelligence Projects and How They Can Succeed
- Enhancing AI Reliability: The Power of Context in Large Language Models
- Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models
- What Is Retrieval-Augmented Generation, aka RAG?
- Rethink Enterprise AI: 11 Technology Trends Driving the Next Shift
- Building a Reliable Customer Data Infrastructure: Lessons from Airbnb
- Revolutionizing Threat Modeling with AI: The Threat Modeling Co-Pilot
- Building Threat Modeling Tradecraft into an Artificial Intelligence-based Copilot
- Meet Michelangelo: Uber’s Machine Learning Platform
- 2024 State of Generative AI in Global Business
- Superagency in the workplace: Empowering people to unlock AI’s full potential
- To Get Better Customer Data, Build Feedback Loops into Your Products
- Alexi’s Human-in-the-Loop Process Highlighted in Harvard Business Review Article

Content Lead:
Nishita Pereira - Senior Group Manager, Marketing & Communication

Internal Expert Contributors:

Bhaskar Gandavabi - Senior Vice President, Technology & Innovation
Anubhav Mukherjee - AI Platform and Solutions Innovator (AI Innovation Hub)
Chandrashakher Aryasomayasula - Associate Vice President, AI Engineering (AI Innovation Hub)
Malavika Nair - Associate, Delivery Support (AI Innovation Hub)



Chapter 2

Explainability

The discipline behind defensible automation

Quick Read

Most executives working with AI are tired of the noise around it: sealed systems, confident outputs, and decisions that become difficult to defend once scrutiny begins. What matters now is not whether a model performs well in isolation, but whether its decisions can be understood, reviewed, and governed without exposing the organization to unnecessary risk.

Explainability provides that footing by making the basis of automated decisions visible, stable, and accountable within the systems that use them.

Enterprises are evaluating AI through a different lens today, and it leads to three practical questions:

Why is explainability what makes AI viable at scale?

- It turns AI from a black box into an inspectable component of the workflow, clarifying when decisions can proceed automatically and when human judgment must intervene.
- It exposes reasoning paths early, preventing silent failures from compounding across systems and teams.
- It keeps automation aligned with business rules and constraints as models evolve, rather than allowing logic to drift unnoticed.

Why does this sit with executive leadership?

- When decisions are challenged—by regulators, auditors, customers, or the board—the explanation becomes the artifact under review, not the model itself.
- Without system-level explanations, technically correct outcomes can still be indefensible in practice.
- When teams cannot reliably explain outputs, they slow execution, escalate unnecessarily, and introduce manual workarounds that erode the value of automation.

What is at stake operationally?

- Commercial: Productivity gains diminish when exceptions require repeated human interpretation of opaque decisions.
- Reputational: Trust breaks down quickly when outcomes cannot be justified consistently or clearly.
- Operational: Reviews, rechecks, and escalations multiply when reasoning is fragmented or unstable.

Explainability becomes dependable only when it is treated as both a design requirement and a governance commitment. When built into models, workflows, and oversight structures from the outset, explainability stops being a disclosure exercise and becomes part of how the organization maintains control as AI systems scale.

Definition & Context

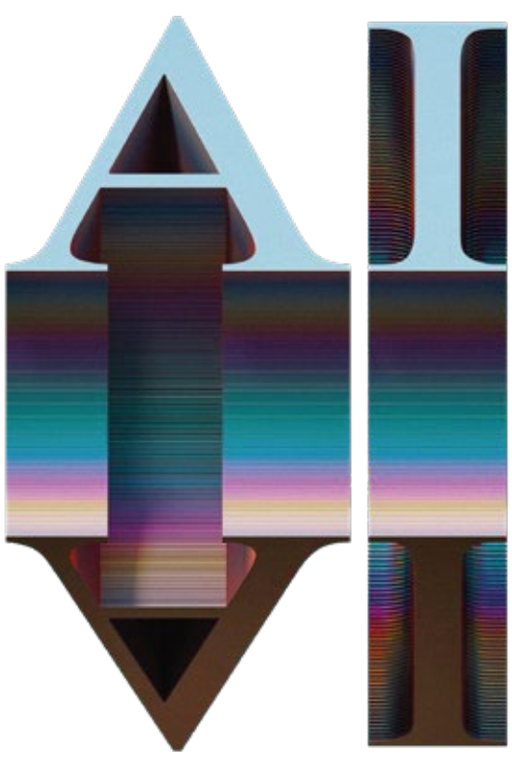
Explainability addresses a structural gap in modern AI systems: the distance between what a model computes and what a human understands. As learning systems grow more complex, the internal logic behind their outputs becomes less accessible, even to those responsible for evaluating or governing their behavior. Most of these systems operate as opaque black boxes, producing outputs without exposing the reasoning that led to them; an opacity that undermines an essential condition for enterprise use: the ability to understand why a system acted the way it did, whether its reasoning is sound, and where its boundaries lie.

In operational settings, this gap is not abstract. It surfaces when decisions cannot be defended during audits, when accountability for outcomes becomes ambiguous, and when organizations are forced to rely on post-hoc justification rather than verifiable system rationale.

Across technical and regulatory communities, explainability is treated as a fundamental property of trustworthy AI. The National Institute of Standards and Technology (NIST) describes it as the capacity of a system to provide evidence or reasons for its outputs and processes in a way that is meaningful, accurate, and aligned with its knowledge limits. The Defense Advanced Research Projects Agency (DARPA) frames the requirement more directly: AI systems should be capable of explaining their rationale to human users, characterizing their strengths and weaknesses, and conveying some understanding of how they will behave in the future. Both perspectives converge on the same operational requirement: systems must reveal enough of their internal logic for humans to judge whether an output is appropriate and when it should be trusted.

Explainability is distinct from interpretability. Interpretability concerns the meaning a human assigns to a model's decision, while explainability concerns the mechanism that produced it. The difference is consequential. Human interpretation is flexible and context-dependent; it draws on background knowledge and intuition. Algorithmic explanations, by contrast, are rigid reflections of the model's structure and training data.

NIST's report on Psychological Foundations of Explainability and Interpretability in Artificial Intelligence illustrates this with a rental-application scenario. A machine-learning model trained on historical rental data may classify an applicant as high risk because, in its training set, short or nonexistent rental histories correlated with higher rates of payment problems. A human assessor may reach a similar outcome, but the reasoning differs. Humans often justify the decision through contextual logic, such as linking missing rental history to insufficient evidence of reliability. Both paths lead to the same result, but through different mechanisms. When those mechanisms diverge, users either misunderstand the system or place misplaced confidence in its outputs.



In organizational settings, this distinction has practical consequences. Interpretations formed by individuals cannot be governed, audited, or consistently reproduced. However, explanations generated by systems can. When decisions are defended through human reasoning rather than system-level rationale, accountability fragments across teams, reviews become narrative exercises, and risk ownership becomes unclear. Explainability shifts responsibility from individual judgment to organizational control by ensuring that the basis for a decision exists as a durable system artifact rather than an informal explanation after the fact.

This same divergence is also where bias enters: **people routinely apply inference shortcuts, while models reproduce statistical patterns, including those shaped by historical inequities.**

In high-stakes environments, this same gap between computation and comprehension becomes a system-level risk. DARPA's Explainable Artificial Intelligence (XAI) Program highlights this challenge explicitly in domains where users must make rapid, consequential decisions that rely on AI-generated recommendations. Without explanations that illuminate why a system acted and under what conditions it might fail, operators cannot calibrate trust, identify errors, or understand when human intervention is required.

The implications extend directly into enterprise decision systems. Explanations support the formation of accurate mental models, improve decision quality, and establish the conditions under which an AI system can be used appropriately. NIST's Artificial Intelligence Risk Management Framework (AI RMF 1.0) makes this explicit: effective explanations enhance user understanding, task performance, trust, and the ability to correct errors.

This expands explainability beyond model introspection. It becomes an operational requirement that ties model behavior to human reasoning, governance expectations, and domain constraints. A system may produce accurate predictions, but without explanations that reveal rationale, limitations, and decision pathways, accuracy alone cannot guarantee responsible or reliable use. In practice, explainability ensures that AI behaves as part of a larger decision system, one where humans remain accountable for choices that depend on the model's output.

This is the context in which explainability matters: not as a transparency exercise, but as a mechanism that links mathematical behavior to human judgment in a way that preserves trust, enables oversight, and keeps the system aligned with real-world conditions.

What People Miss

Although explainability is widely acknowledged as essential, several aspects of it remain loosely handled in enterprise environments. The gaps are not primarily technical failures so much as assumptions about what explanations are for and what they can realistically support. These oversights shape how systems are evaluated, how decisions are made around them, and how risk is interpreted once an AI model begins influencing real operations.

Explanations Treated as Inherently Clarifying

A frequent gap is the assumption that the presence of an explanation improves understanding. Google's People + AI Research (PAIR) Guidebook shows that the effect is far less predictable. Explanations influence how people read and judge a model's output, but the alignment between an explanation and a user's reasoning is uneven. Polished or confident - sounding explanations can prompt overreliance, while sparse or incomplete explanations can create unwarranted doubt even when the model's logic is sound. The issue is not the availability of explanations but the way they shape confidence. When explanation quality is treated as interchangeable with explanation presence, teams gain a narrative without gaining the clarity needed to make reliable decisions, escalate risk appropriately, or know when intervention is required.

Explainability Framed as a Convenience Rather Than a Control

Explainability is often positioned as a way to make model outputs easier to interpret, but its role extends far beyond usability. In enterprise systems, explainability functions as a control mechanism: a means of enforcing accountability, enabling oversight, and constraining risk. Regulations such as the GDPR, which requires meaningful information about the logic involved in automated decisions, and the EU AI Act, which sets transparency obligations for high-risk systems, treat explanation quality as part of the oversight surface rather than as optional disclosure. Explanations identify which factors influenced a decision and whether those factors are appropriate, defensible, and aligned with policy or law. Like financial or compliance controls, they establish a verifiable basis for review, challenge, and escalation. When explainability is reduced to a usability feature, organizations forfeit this control layer, leaving decisions difficult to audit and accountability difficult to assign in practice.

Black-Box Models Assumed to Be Explainable After the Fact

Another gap appears when explainability is treated as something that can be added once a model is built. Post-hoc techniques attempt to approximate how a black-box system arrived at its output, but they operate outside the model's internal structure. DARPA's XAI work draws a clear distinction between explanations produced by design and those reconstructed after the fact, noting that approximations can diverge from the model's actual reasoning. Interpretable systems such as Microsoft's Explainable Boosting Machine illustrate the alternative, exposing feature effects directly rather than inferring them after execution. This distinction matters because reconstructed explanations rarely hold up under scrutiny. Post-hoc narratives may appear coherent, but they are difficult to reproduce, validate, or defend in audits, regulatory reviews, or legal challenges. When explainability is not designed into the system itself, organizations risk relying on explanations that sound reasonable but cannot be reliably tied back to how a decision was actually made.

These gaps point to a broader pattern: **explainability rarely fails on definition, but on how it is integrated into the system's design and operation.** When explanations are treated as controls rather than artifacts, they must be engineered, governed, and maintained with the same discipline as other critical assurance mechanisms. The next step is to consider what it takes to build that structure deliberately; where explainability is engineered into the model, the workflow, and the governance fabric from the outset.

When AI Decisions are Challenged, This is What Gets Examined

In reviews, audits, and escalations, the explanation becomes the artifact under scrutiny.

What teams often prepare

- Model accuracy metrics
- Performance benchmarks
- Prompt logic or instructions
- Feature importance or confidence scores

What gets examined under scrutiny

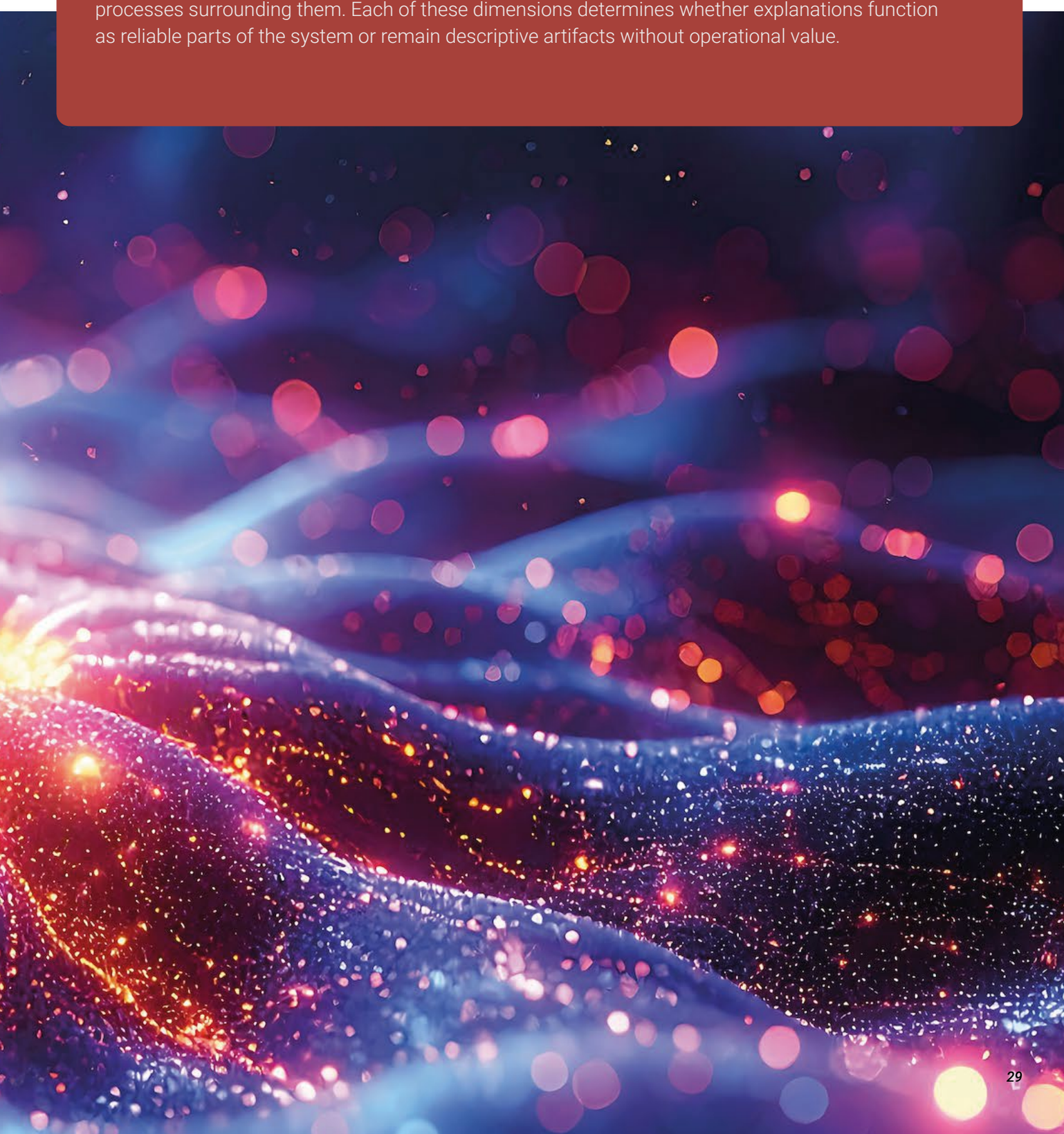
- Information sources used at decision time
- Reasoning consistency across versions and time
- Decision context inside the workflow
- Human intervention and override records

Performance explains outcomes. Systems explain accountability.

How to Do It Right

Enterprise teams recognize the need for explainability, but few treat it as a design requirement. Surveys continue to show explainability rising as a top adoption risk while remaining one of the least mitigated in practice. In McKinsey's recent studies, 40% of organizations cited explainability as a key risk in adopting generative AI, and it ranks among the two most commonly reported risks overall. Yet it is still one of the least actively mitigated.

The challenge is not awareness but the absence of a structured approach for building explanations that hold up under real use. Effective explainability depends on choices made at the model level, the stability of explanation methods, how explanations enter decision pathways, and the governance processes surrounding them. Each of these dimensions determines whether explanations function as reliable parts of the system or remain descriptive artifacts without operational value.

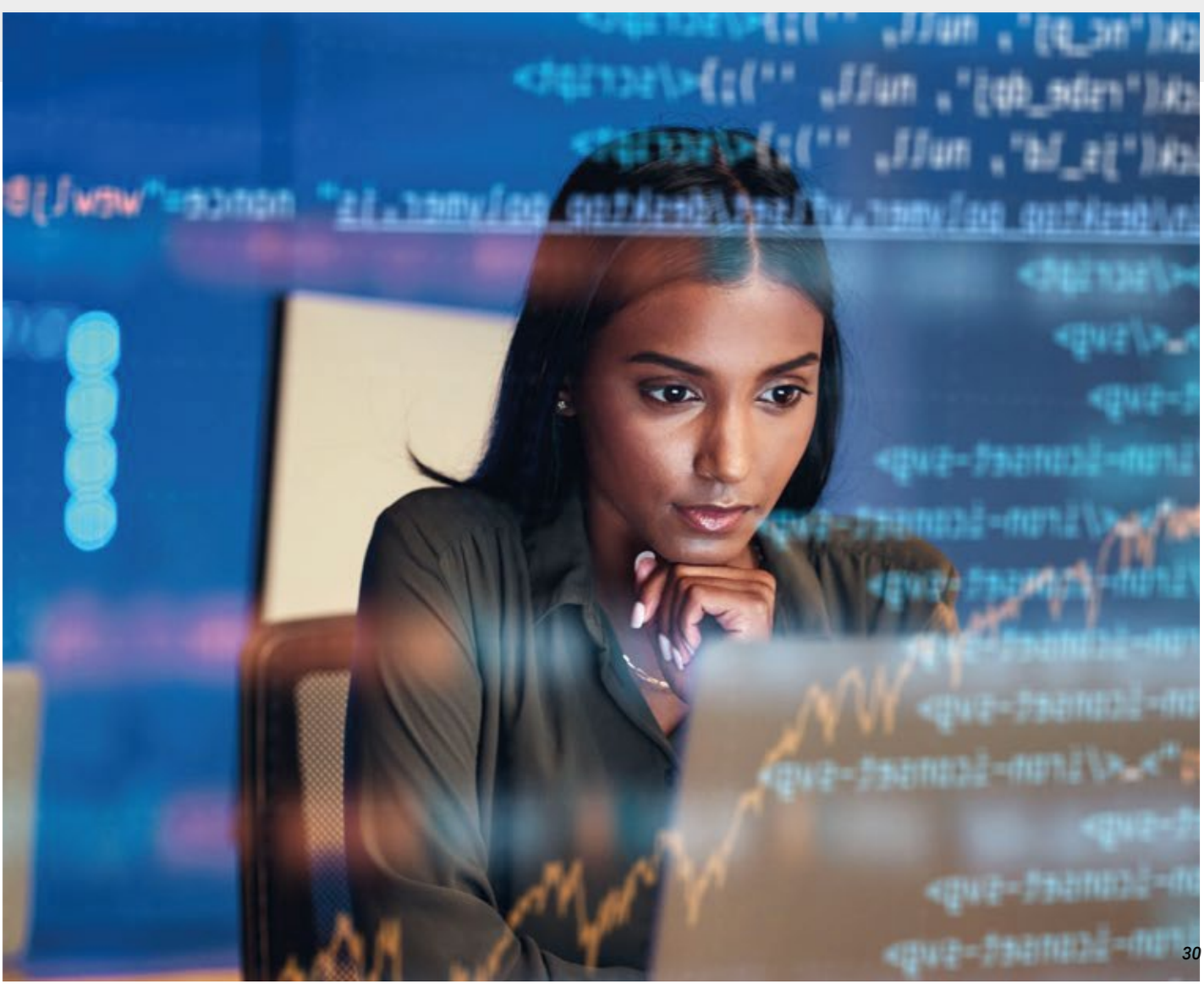


1. Design for Transparency at the Model Level

Effective explainability starts with choosing models whose behavior can be examined rather than inferred. Microsoft's Explainable Boosting Machine is an example of how this works in practice. Built on GAM and GA2M algorithms, EBM separates feature effects so they can be viewed independently, revealing which signals genuinely shape the prediction. In contrast to black-box models that rely on complex interactions or seemingly random feature combinations, EBMs make the rationale visible by design. For enterprise systems, this architectural transparency reduces the gap between model computation and the reasoning teams must defend, enabling explanations that remain stable as the system evolves.

Explainability at the model level depends on designing transparency directly into how systems operate under enterprise conditions. Rather than relying on post-hoc interpretation, our agentic AI platform FD Ryze Infinity is designed so data, reasoning, and control layers remain explicitly separated and inspectable as part of normal operation.

- Explicit separation between data retrieval layers and model generation logic
- Structured data pipelines that surface source provenance, context, and scope of use
- Policy and constraint layers applied upstream to shape outputs deterministically



SUMMARY Model-Level Transparency

Explainability starts with models your teams can actually read, not reverse-engineer under pressure.

Microsoft's Explainable Boosting Machine (EBM) is a good proof point: it uses glass-box, GAM/GA2M-based modeling so feature effects are visible by design, giving stakeholders a clear view of why a prediction was made, not just what it is.

FULCRUM IN PRACTICE

- Explicit separation between data retrieval layers and model generation logic.
- Structured data pipelines that surface source provenance, context, and scope of use.
- Policy and constraint layers applied upstream to shape outputs deterministically.

Takeaway: Choose architectures your teams can defend in the boardroom, not just optimize on a benchmark.

2. Stabilize Explanation Quality in Real Use

Explanations must remain consistent across the lifecycle of a model. When explanation methods change as the system evolves, teams lose the ability to compare decisions, audit behavior, or understand whether a shift reflects the data or the explanation tool. Google's PAIR initiative shows how even small inconsistencies in explanation format or detail can miscalibrate user confidence. Treating explanation quality as a component with its own stability requirements—tested, versioned, and monitored—keeps explanations aligned with the model's true behavior. This stability allows explanations to function as system signals rather than intermittent commentary.

We treat explanation quality as part of the system lifecycle, not as a static artifact attached to a model. As models and workflows evolve, explanation behavior is managed with the same controls used for core system components to preserve comparability, auditability, and trust.

- Explanation logic versioned alongside models, prompts, and retrieval workflows.
- Explanation behavior reviewed whenever models or workflows change
- Monitoring for shifts in sources, reasoning paths, or confidence signals that indicate explanation drift

SUMMARY Explanation Stability

Executives need explanations that stay consistent as the model evolves. When rationale changes from one version to the next, confidence erodes and comparisons lose meaning.

Google's PAIR research highlights that even small inconsistencies in explanation format or detail can distort user confidence and decision accuracy. Stability needs to function as a control surface, not a cosmetic layer.

FULCRUM IN PRACTICE

- Explanation logic versioned alongside models, prompts, and retrieval workflows
- Explanation behavior reviewed whenever models or workflows change
- Monitoring for shifts in sources, reasoning paths, or confidence signals that indicate explanation drift

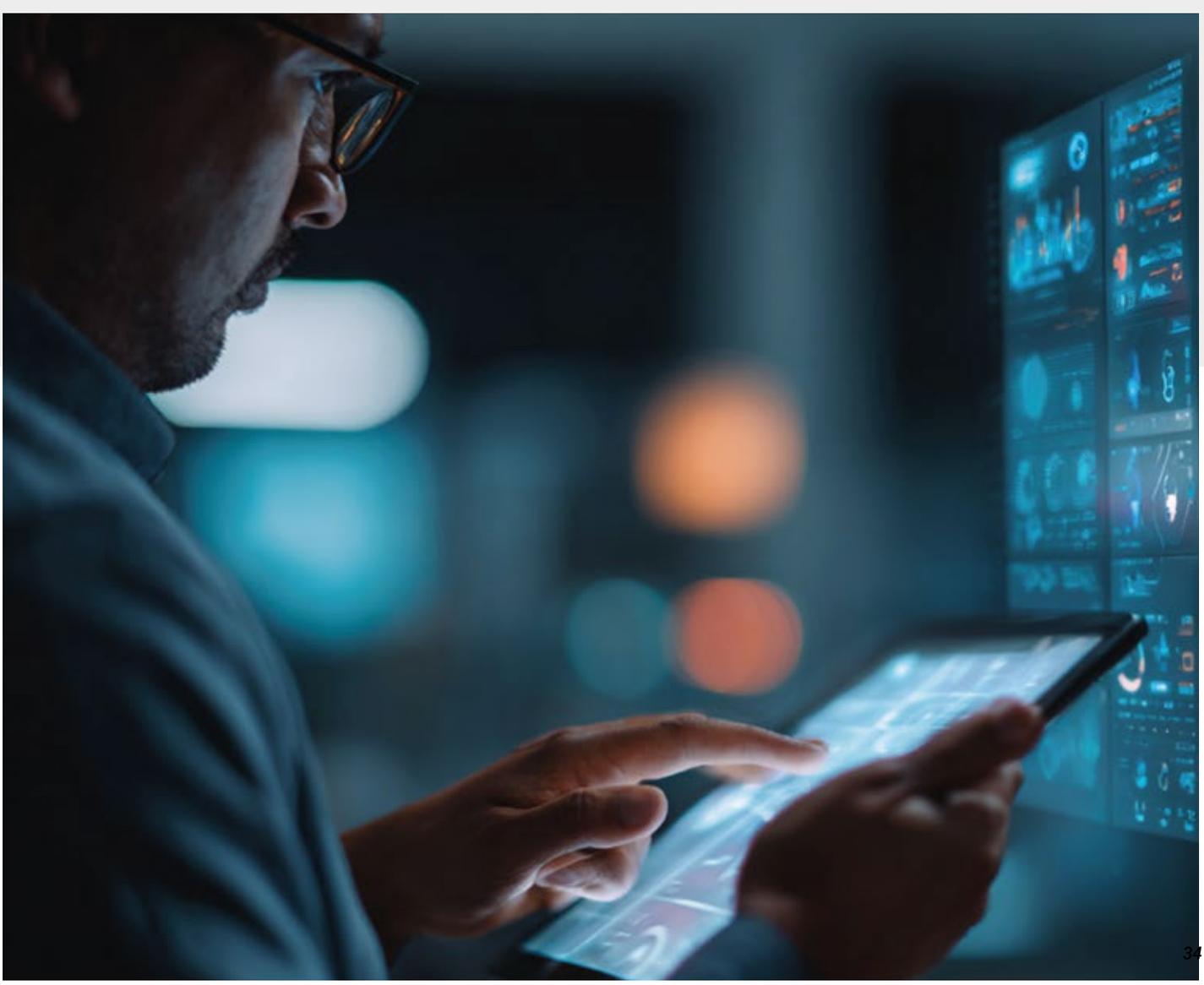
Takeaway: If your explanations shift unpredictably, your governance will too. Lock the logic before you scale the system.

3. Integrate Explainability into Decision Pathways

Effective explainability requires more than generating a rationale. It must sit in the same pathway where the decision is taken. Watson OpenScale demonstrates how this can operate in practice. Its explainability layer surfaces the features driving a prediction at the moment a reviewer encounters it and pairs them with counterfactual tests that show how the outcome would shift under different inputs. This positions explanations as part of the workflow rather than an afterthought. Decisions are accepted, escalated, or overridden with a clear understanding of the model's reasoning and its boundaries. This is the practical requirement of doing explainability well: placing the rationale where it actively shapes the action that follows.

Explainability is only effective when it appears at the moment a decision is being shaped, not after it has already been taken. We integrate explanations directly into decision pathways so reasoning, risk context, and source signals are visible where routing, escalation, or override decisions occur.

- Explanations surfaced within agent and workflow steps where decisions are made
- Routing and escalation logic informed by visible reasoning and source context
- Human validation gates applied for higher-risk actions, supported by clear explanatory signals



SUMMARY Explainability In Decision Pathways

An explanation no one sees at the moment of action is just decoration. Reliability only shows up when the rationale sits inside the workflow that uses it.

IBM's Watson OpenScale demonstrates this well. Explanations appear at the decision point, paired with counterfactuals that show how outcomes would change under different inputs. It's explainability as an operational instrument and not just an accessory.

FULCRUM IN PRACTICE

- Explanations surfaced directly within decision workflows where routing, escalation, or override occurs
- Reasoning and source context made visible at points of human review and intervention
- Human validation gates applied for higher-risk actions, supported by clear explanatory signals

Takeaway: Explanations should shape decisions. If they sit outside the workflow, they aren't protecting the system.

4. Anchor Explainability in Governance

Explainability becomes durable when it is treated as a governed requirement rather than an attribute of a specific model. IBM approached this by formalizing transparency through FactSheets and an ethics-by-design process that required models to advance with their rationale and limitations clearly recorded. This approach shows how to operationalize explainability: embed it in review cycles, assign ownership, and make documentation part of the system's lifecycle. Governance ensures explanations remain accurate as models shift, preventing silent divergence between what the system does and what the organization believes it does.

We anchor explainability in governance by defining how explanatory artifacts are owned, reviewed, and relied upon in audits, regulatory inquiries, and incident analysis.

- Explanatory records maintained as authoritative system artifacts for audit and review purposes
- Clear ownership assigned for explanation quality, with accountability embedded in governance forums
- Periodic governance reviews to verify alignment between model behavior, recorded rationale, and organizational understanding

SUMMARY Governance-Anchored Explainability

Explainability only endures when it's treated as a governed asset and not a feature. Organizations need to operationalize transparency rather than rely on memory or tribal knowledge.

IBM got this right years ago with a disciplined approach that treated transparency as a compliance artifact. Its FactSheets framework required every model to carry its logic, limits, data lineage, and risks wherever it was deployed. The lesson? If the rationale isn't recorded, it isn't real. And if it isn't maintained, it can't be trusted.

FULCRUM IN PRACTICE

- Explanatory records maintained as authoritative system artifacts for audit and review
- Clear ownership assigned for explanation quality, embedded in governance and review forums
- Periodic governance reviews to verify alignment between model behavior, recorded rationale, and organizational understanding

Takeaway: If you want explainability to hold up in a board meeting, assign ownership and make transparency a gated part of the lifecycle and not a post-launch chore.

Integrated explainability creates a consistent view of how a system arrives at its outputs. The durability of that view depends on the protections around it: how the system manages access, safeguards its logic, and meets the controls imposed by policy and regulation.

As models take on more consequential workloads, these protections become the boundary between a defensible decision and a liability the business must absorb. Those constraints form the next piece of the puzzle.

Up Next: Security & Compliance

As AI systems expand across data, teams, and third-party services, control weakens at the seams. The Security & Compliance chapter focuses on keeping authority intact after deployment, through clear ownership, enforceable boundaries, continuous monitoring, and controls that survive real operational change.

It addresses how organizations stay secure, compliant, and defensible when AI is no longer contained by static policies or periodic reviews.

Sources

- Explainable AI in practice
- Four Principles of Explainable Artificial Intelligence
- Artificial Intelligence Risk Management Framework (AI RMF 1.0)
- The People + AI Guidebook
- Fewer Than 1% of Explainable AI Papers Validate Explainability with Humans
- Le problème du videur : la crédibilité des explications de l'IA en question
- On the Relation of Trust and Explainability: Why to Engineer for Trustworthiness
- Entry into force of the European AI Regulation: the first questions and answers from the CNIL
- The state of AI in early 2024: Gen AI adoption spikes and starts to generate value | McKinsey
- The state of AI in 2025: Agents, innovation, and transformation | McKinsey
- Glass Box ML Model: Microsoft's InterpretML to explain a Telco customer churn model
- Global explanation stability in Watson OpenScale explainability metrics
- How IBM makes AI based on trust, fairness and explainability
- Responsible Use of Technology: The IBM Case Study

Content Lead:
Nishita Pereira - Senior Group Manager, Marketing & Communication

Internal Expert Contributors:

Bhaskar Gandavabi - Senior Vice President, Technology & Innovation
Anubhav Mukherjee - AI Platform and Solutions Innovator (AI Innovation Hub)
Chandrashakher Aryasomayasula - Associate Vice President, AI Engineering (AI Innovation Hub)
Malavika Nair - Associate, Delivery Support (AI Innovation Hub)

Chapter 3

Security & Compliance

Navigating AI risks with foresight and precision

Quick Read

AI brings speed and scale, but it also exposes parts of the business that usually stay out of sight. Security and compliance shape how much of that exposure you can control once AI enters real workflows. They ensure the organization can respond to AI challenges before they escalate into security breaches, legal risks, or operational chaos.

What leaders should understand about AI security and compliance:

- They determine whether scrutiny stops at the system level or escalates to key leadership: CISO, General Counsel, or Compliance officers.
- They draw the line on AI autonomy, defining when responsibility shifts from technology to business leadership (CIO/COO).
- They set the pace for AI scale, ensuring the CFO isn't left managing risks the business didn't prepare for.

What signals leaders should be watching across the enterprise:

- Teams using AI outputs without clarity on ownership, approval, or supervision of decision-making processes.
- Systems expanding across departments without clear governance over data access or user permissions.
- Decisions advancing through workflows without visibility into how models arrived at them or without audit trails.

What actions create stability when the systems keep changing:

- Treat governance as a working practice. Assign owners, set review cycles, and keep oversight continuous.
- Create explicit boundaries by defining who can use AI systems, what data they interact with, and how exceptions or deviations are handled.
- Test the system under stress and fix weaknesses before they reach customers or partners.

AI becomes easier to manage once leaders treat control as an ongoing, integral part of operations rather than a one-time milestone. Get that foundation right, and performance and cost can be leveraged to drive growth, instead of becoming unexpected risks.

Definition & Context

Security and compliance in AI are often discussed as legal or technical obligations, but in practice they function as enterprise risk controls that determine whether AI systems are permitted to operate at all. They define whether an organization can demonstrate due care over how AI systems use data, make decisions, and operate under scrutiny. As AI moves into regulated, customer-facing, and revenue-critical workflows, security and compliance become less about preventing every failure and more about establishing enforceable boundaries, ownership, and evidence of control when those systems are challenged.

What distinguishes AI from earlier digital systems is not just the sensitivity of the data it touches, but the opacity, adaptability, and scale at which it operates. Models can infer sensitive attributes, replicate training data, be manipulated through inputs, or behave unpredictably across contexts. Traditional security controls, designed for static software and deterministic rules, do not automatically extend to these behaviors. As a result, organizations can meet baseline security standards and still fail to control AI-specific risk. This gap is visible in enforcement actions and breaches where the issue was not unauthorized access alone, but unclear accountability for model behavior, insufficient controls over how systems adapt, or the inability to demonstrate compliance after the fact (e.g., GDPR enforcement actions against Clearview AI for biometric processing, data leakage into third-party LLMs, or biased automated decisions).

In this context, security and compliance operate as the enforcement layer of the AI lifecycle, defining what the system is allowed to do, who is responsible for it, and under what conditions it may continue operating.

Explainability makes system behavior legible; security and compliance determine whether that behavior is controlled, constrained, auditable, and legally defensible. Without explainability, organizations cannot articulate how decisions are made.

Without security and compliance, they cannot control who can access, influence, or override those decisions, nor demonstrate adherence to regulatory expectations once AI outputs are questioned.

Why Traditional Security Controls Break in AI Systems

Earlier Digital Systems Were Secured As…	AI Systems Operate Very Differently
Deterministic software with fixed logic paths	Decisions emerge from models, data, policies, and orchestration layers
Clearly bounded applications with known interfaces	Outputs depend on retrieved enterprise data, not model memory alone
Data accessed explicitly, not inferred	Behavior adapts as prompts, sources, or workflows change
Behavior reviewed at release, not during operation	Reasoning paths shift without visible code changes
Decisions executed by systems, owned by teams	Decisions are distributed across agents, tools, and humans
Controls that worked here:	**Controls that are required here:**
Static access controls	Continuous monitoring of reasoning and sources
Periodic security reviews	Versioned prompts, retrieval logic, and policies
Role-based permissions	Governed explanation artifacts, not just outputs
Point-in-time compliance checks	Ownership defined at decision points, not system edges

Why this matters for leadership

When controls assume stability but systems are built to adapt, accountability shifts upward. Under scrutiny, leadership is expected to demonstrate ongoing control, explain decisions made in production, and show that safeguards traveled with the system as it evolved.

Modern AI security and compliance therefore extend beyond perimeter defense or policy documentation. They encompass data governance, identity and access control, model evaluation, audit trails, incident response, and continuous risk assessment across development, deployment, and operation. Frameworks such as NIST's AI Risk Management Framework and ISO/IEC 42001 reflect this shift by emphasizing lifecycle governance, role clarity, and evidence-based controls rather than one-time certifications.

The urgency of this shift is reflected in enterprise behavior. A growing majority of organizations now express concern about AI's regulatory impact, are actively developing AI compliance policies, and plan to pursue audits or certifications within the next 12–24 months. This momentum is driven less by innovation ambition and more by exposure. Regulatory penalties, reputational damage, and operational disruption have become credible outcomes of unmanaged AI risk.

Seen through this lens, security and compliance are not adjacent to AI strategy; they are preconditions for safe deployment and sustained scale. They determine whether AI systems can be trusted with sensitive data, allowed to automate consequential decisions, and defended under legal, regulatory, and public scrutiny. As AI systems become more autonomous and embedded, the absence of enforceable security and compliance controls does not slow adoption; on the contrary, it concentrates risk at the organizational level.

What People Miss

Although security and compliance are widely cited as foundational to responsible AI, they are often treated as background requirements rather than active system properties. Most organizations are aware of relevant regulations, standards, and frameworks. What they underestimate is how quickly those safeguards weaken once AI systems move from policy design into live, evolving operation, where models, data, users, and workflows change faster than formal controls can be updated, and where responsibility for risk shifts from documentation to execution.

This gap shows up clearly in outcomes. **IBM's Cost of a Data Breach Report 2025** notes that organizations take an average of 280 days to identify and contain a breach, a delay that reflects how risk accumulates after systems are already in use. The most common breakdowns don't stem from missing controls but from how those controls are scoped, enforced, and sustained across the AI lifecycle, often leaving organizations unable to demonstrate effective oversight once harm has already occurred.

Treating compliance as proof of safety instead of a starting line

Many organizations assume that meeting regulatory requirements or aligning to a recognized framework means their AI systems are secure. In reality, compliance largely reflects a moment in time. Once models are deployed, they begin interacting with new data, new prompts, new users, and new operational pressures that weren't fully anticipated during review. Audits and policies struggle to keep pace with that change.

Research from compliance benchmarking studies consistently shows a gap between documented controls and how those controls are enforced day to day, especially after deployment. The result is a false sense of safety: systems appear compliant on paper while real exposure accumulates quietly in production environments, where incidents are harder to trace, responsibility is diffused, and post-hoc justification replaces verifiable control.

Underestimating how fast security breaks when AI crosses boundaries

AI systems rarely operate within a single, well-defined perimeter. They span cloud platforms, third-party models, internal tools, APIs, and human workflows, often without clear ownership at the seams. What organizations miss is how quickly security responsibility fragments across these boundaries, creating zones where no single team has enforceable authority over use, access, or impact.

Samsung's 2023 ChatGPT incident illustrates this clearly: sensitive source code and internal documents were exposed not through a system breach, but through normal employee use of a public AI tool that sat outside formal controls. The LLaMA model leak followed a similar pattern at a different layer, where custody and distribution rules failed once the model left its original environment. In both cases, risk emerged not from malicious intent, but from unclear boundaries around who could use what, where, and under whose authority. This left organizations unable to assert control once information crossed those lines.

Focusing on technical risk while liability accumulates elsewhere

Many AI security discussions stay framed as engineering concerns: prompt injection, data leakage, model extraction. What gets missed is where the consequences actually land. When Clearview AI built its facial recognition system by scraping billions of images without consent, the technical capability worked exactly as intended. The failure surfaced elsewhere, through regulators, courts, and market access. Multi-country fines, mandatory data deletion orders, and outright bans followed, effectively shutting Clearview out of entire regions.

The risk was never confined to the model. It manifested as legal exposure, reputational damage, and loss of operating permission, outcomes that were foreseeable long before enforcement but treated as secondary to technical performance.

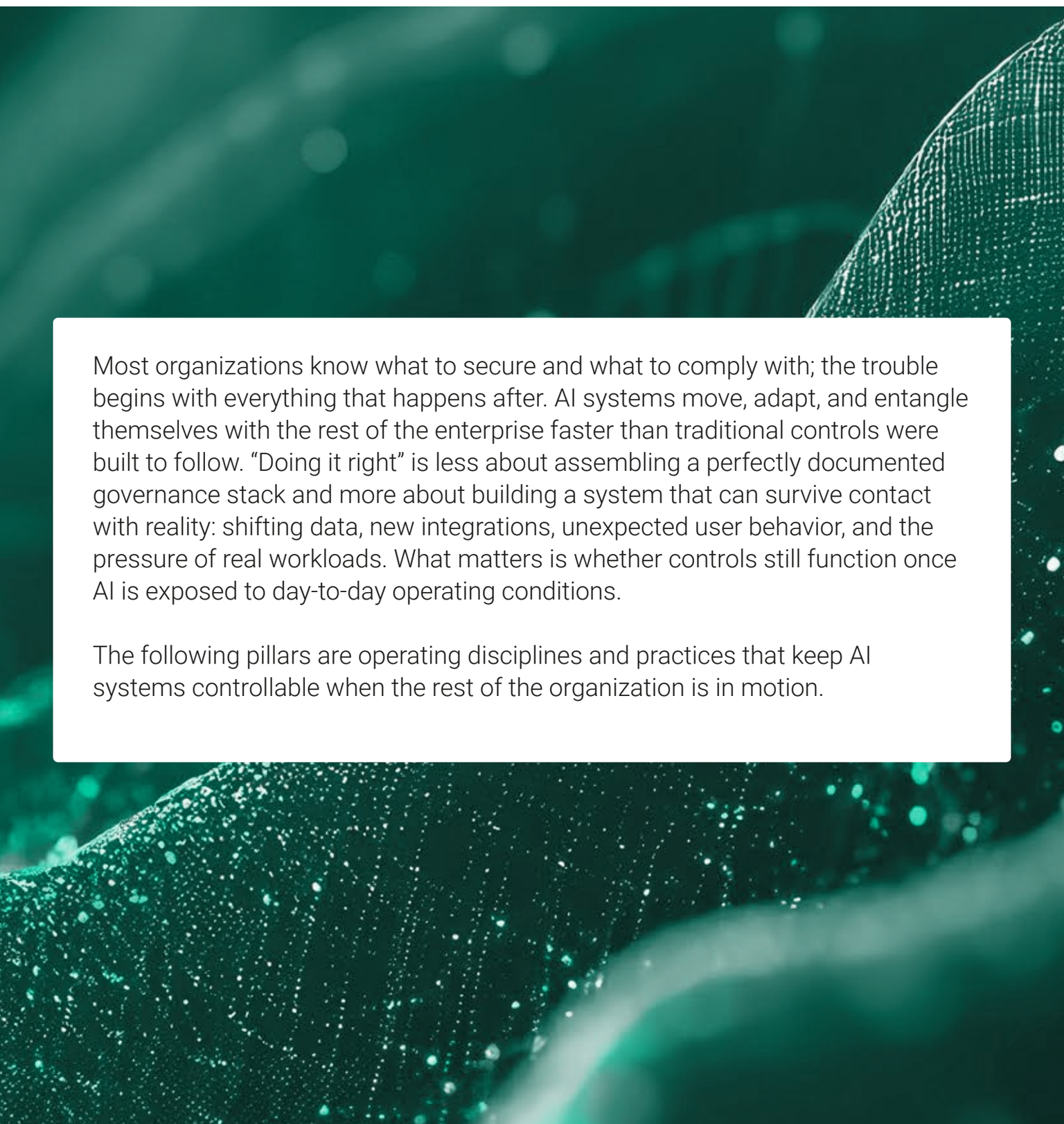
Confusing governance artifacts with governance in action

In many organizations, AI governance exists on paper: policies, model reviews, ethics statements, and compliance attestations. What's missed is that governance only works when it actively shapes how systems run. The SafeRent case illustrates this gap clearly. The tenant-screening model operated inside a regulated housing context, influencing accept-or-reject decisions tied to protected classes. Governance technically existed, but there were no enforceable checkpoints to halt the system when risks surfaced, no escalation path to intervene, and no mechanism to correct behavior before harm accumulated.

The result was not just bias exposure, but prolonged operation of a risky system until legal action forced change, revealing governance that documented intent but lacked the power to interrupt execution.

These misses rarely stem from carelessness. They arise from treating AI security and compliance as static obligations rather than systems that must function under real operating conditions. Once models are live, risk no longer sits in documentation. Closing that gap requires shifting from control-as-evidence to control-as-infrastructure, which is the first step to doing it right.

How to Do It Right



Most organizations know what to secure and what to comply with; the trouble begins with everything that happens after. AI systems move, adapt, and entangle themselves with the rest of the enterprise faster than traditional controls were built to follow. “Doing it right” is less about assembling a perfectly documented governance stack and more about building a system that can survive contact with reality: shifting data, new integrations, unexpected user behavior, and the pressure of real workloads. What matters is whether controls still function once AI is exposed to day-to-day operating conditions.

The following pillars are operating disciplines and practices that keep AI systems controllable when the rest of the organization is in motion.

1. Govern AI Like an Operating System, Not a Documentation Exercise

Most organizations treat AI governance as a checklist to complete, not a system to run. Policies get written, approvals get logged, and compliance feels "done" the moment a model goes live. That assumption breaks the moment the system starts interacting with real data, real users, and real downstream decisions. Governance that only exists on paper cannot control how a system behaves once it is in motion.

The gap shows up everywhere. According to A-LIGN's 2025 Compliance Benchmark Report, 63% of organizations report they have no governance policies to oversee or detect shadow AI, and an even larger share, 87%, have no defined processes to mitigate AI-specific risks. The issue is not awareness, but enforceability: controls exist, but they are not wired into how systems actually operate.

Organizations need to treat governance like a live operating environment. That means assigning durable ownership, defining escalation paths, and ensuring review mechanisms persist beyond initial approval. Risk owners must be named; review cycles must be scheduled; audit and monitoring responsibilities must be embedded into the architecture rather than delegated to periodic reviews.

This shift is already visible in how leading institutions structure their AI oversight. BNP Paribas' modernization of its risk assessment function illustrates the change. Instead of relying on episodic checks, they rebuilt their workflow around continuous evaluation: models that adapt as new data arrives, alerts for anomalies that surface in real time, and oversight that follows the system rather than waiting for quarterly reporting. Governance moved from episodic verification to continuous supervision.

Industry standards are moving in the same direction. ISO 42001 explicitly requires organizations to maintain ongoing risk controls, role clarity, and operational transparency across the AI lifecycle. Its first real-world milestone—Synthesia receiving ISO 42001 certification through A-LIGN—demonstrates what mature AI governance looks like in practice: a management system capable of sustaining control as models, use cases, and integrations evolve. Google's Secure AI Framework reinforces the same principle by shifting focus from perimeter checks to continuous protection surfaces. The EU AI Act formalizes this expectation by making post-deployment monitoring a core obligation rather than a best practice.

The lesson here is simple. **AI systems don't remain governable unless governance itself remains active.** Controls that sit in documentation lose authority once systems change, and AI systems change continuously.

To govern effectively, organizations must build for motion rather than stability: roles that persist, checks that trigger automatically, and oversight that travels with the system as it evolves.

We operationalize AI governance at the system level, where models, data, agents, and human decisions intersect. Controls are applied through orchestration, monitoring, and enforced decision boundaries rather than policy artifacts. This allows governance to travel with the system as it scales and changes.

- Ownership assigned to agents, workflows, and business outcomes, not individual models or tools\
- Decision rights split explicitly between autonomous actions and mandatory human escalation
- Audit logs linking outputs to prompts, models, agents, and source data

SUMMARY

Govern AI Like An Operating System

Executives need governance that continues to function after deployment, not controls that expire at launch. Effective AI governance operates as an active layer of the architecture, with clear accountability, enforced decision boundaries, and visibility that persists as systems adapt and scale.

BNP Paribas shows what this looks like in practice. Their risk assessment function moved from episodic reviews to continuous oversight, pairing real-time signals with teams responsible for acting on them. Governance became embedded in day-to-day operations rather than reserved for audit cycles.

FULCRUM IN PRACTICE

- Ownership follows decisions through the system
- Autonomy bounded by mandatory human escalation
- Decisions logged with audit-ready evidence

Takeaway: Build governance into the system runtime: ownership, escalation, and evidence generation must execute automatically as the system operates.

2. Make AI Boundaries Explicit, Owned, and Enforced

AI systems don't always fail because of the unpredictable behavior of a model. They also fail because no one can say with certainty who owns what at the seams, between data systems, tools, teams, and third-party services. These seams are where security and compliance decisions are enforced in practice, and unless those boundaries are defined, staffed, monitored, and enforced, even a well-designed control system weakens the moment AI touches a real workflow.

A clear illustration comes from Scotiabank's compliance operations, where risk monitoring spans employee communications, financial transactions, and internal systems. What makes the program effective is the clarity of ownership. Every interaction surface has an accountable team, mapped permissions, and conditions under which escalations are triggered. The controls work because the boundaries are visible and actively managed and responsibility doesn't dissolve at handoffs.

Global standards echo the same expectation. ISO 42001 calls for explicit ownership across the AI lifecycle: who governs a model, who monitors it, who interprets its behavior, and who steps in when conditions change. Microsoft's enterprise governance guidance takes the same position, assigning accountable owners for AI outcomes and not just for the systems that surround them. NIST reinforces this structurally through its Governance function, which centers AI risk management on role clarity and responsibility tracing rather than security configurations alone.

Together, these approaches reflect a shift enterprises need to make: boundaries are not an architectural detail but the enforcement layer. When organizations take the time to make them explicit, assign owners, and integrate them into everyday workflows, AI becomes easier to control, easier to audit, and far less likely to create surprises that spill into legal, operational, or customer-facing problems. Without that discipline, even the strongest safeguards degrade into hopeful intentions rather than reliable protection.

In enterprise AI, risk accumulates at the seams: between systems, handoffs, and third-party integrations. We enforce boundaries through centralized orchestration and identity controls, applied through the FD Ryze platform, where agent interactions, data access, and action execution are governed. This keeps accountability intact as models are updated, vendors change, and workflows scale.

- Access control enforced through granular RBAC, IAM, and tenant isolation
- Permission scopes and consent requirements applied at workflow handoffs and integrations
- Configuration, prompts, and agent logic governed as versioned artifacts with audit trails and rollback

SUMMARY

Make AI Boundaries Explicit, Owned, And Enforced

It's not the complexity of an AI system that results in executives losing control. More often, it's because the boundaries between teams, tools, and data flows are unclear. When everyone touches the system but no one owns the seams, security and compliance become guesswork.

Scotiabank's compliance operations are a strong proof point because their monitoring framework works in a way where every interaction surface — communications, transactions, internal systems — has defined owners, mapped permissions, and clear escalation paths. The boundaries are visible, staffed, and enforced.

FULCRUM IN PRACTICE

- Ownership is enforced at workflow boundaries
- Authority transitions are explicitly defined to ensure automation never outruns human accountability
- Boundary breaches surface immediately, with clear escalation paths before risk compounds

Takeaway: Force boundary clarity. For every AI-enabled workflow, require named ownership, defined authority transitions, and an explicit exception path before scale.

3. Harden the Model Before the World Does

For organizations to get AI security right, they need to all start from the same premise that pressure is inevitable. It will come from users, partners, APIs, automated pipelines, and even well-intentioned employees. In enterprise environments, that pressure translates directly into security exposure, audit findings, and loss of control if the system's behavior has not been exercised in advance. Hardening isn't about waiting for that pressure but about shaping how the system behaves when it arrives.

A useful pattern comes from India's cybersecurity ecosystem. CERT-In, SEBI, and the RBI treat adversarial testing as an operating rhythm rather than a final checkbox. Continuous VAPT, red-team drills, and simulated attack exercises are the normal way a system proves it is ready for production. The focus is on demonstrable readiness under conditions regulators and auditors expect systems to withstand. The underlying principle here is not to trust a model that hasn't already survived the conditions it will face once deployed.

That same principle shows up in AI-specific tooling. Microsoft's PyRIT and the AI Red Teaming Agent are engineered to push models the way an adversary would—prompt manipulation, endpoint probing, query variation, pattern bypass attempts—with the goal being controlled discovery. These tools expose the seams that will inevitably be stressed when users, plug-ins, or external agents start interacting with the system. Hardening means you learn those weaknesses before anyone else does and convert unknown exposure into known behavior that can be documented, mitigated, and governed before deployment.

Model-extraction research sharpens the point. An unprotected model can be replicated to 87.3% of its performance through nothing more than repeated queries. With disruption-based defenses, that drops to 35.7%. That delta is the difference between losing your intellectual property and defending it. It's also the clearest evidence that hardening materially changes what attackers can do and how safely the system can operate.

To do this well, enterprises need an operating discipline. Assume pressure. Simulate pressure. Instrument the system so you can see how it behaves under pressure. Hardened AI doesn't make it invulnerable, but it does make it predictable. And predictability is what makes a system secure, governable, and ready for real environments.

Before AI systems are deployed, they need to be exposed to the same pressures they will face in production. This means stress-testing models, agents, and workflows under conditions they will experience once live. Rather than waiting for failure, we proactively simulate adversarial behavior to ensure systems can handle real-world demands from day one.

- Adversarial testing using prompt injection, jailbreak attempts, output manipulation checks, and data-extraction simulations
- Progressive exposure through sandbox, POC, and staged environments, with canary and blue-green deployments to safely evaluate performance under real-world pressure
- Prompts, configurations, and agent logic treated as deployable assets, versioned and tested under adversarial conditions to reinforce system-level security

SUMMARY

Harden The Model Before The World Does

AI doesn't fail quietly; it fails under pressure. And pressure shows up the moment a model is exposed to real users, real data, and real incentives. Executives who want dependable systems need a hardening discipline that tests the model before the environment does.

India's cybersecurity bodies offer a practical template. CERT-In, SEBI, and the RBI treat adversarial testing as routine engineering work, not an afterthought. The goal is to discover the system's weak points under controlled conditions so those weaknesses don't surface in production. The same principle carries through AI-specific tooling like Microsoft's PyRIT, which stress-tests models through prompt manipulation, endpoint probing, and pattern-bypass attempts long before anything reaches a customer or partner.

FULCRUM IN PRACTICE

- Adversarial testing before exposure through prompt injection, jailbreaks, and extraction
- Progressive exposure under controlled conditions
- Treating prompts and agent logic as testable, versioned attack surfaces

Takeaway: Run adversarial and stress testing as part of your build process so your teams, and not users, regulators, or attackers, uncover the system's weak points first.

Security and compliance make an AI system trustworthy, but they don't make it worthwhile. Once the guardrails are in place, the question shifts from "Is it safe?" to "Is it pulling its weight?" Performance and cost tend to expose whatever was left unresolved upstream; models that drift because governance is thin, infrastructure that over-scales because boundaries weren't set, or pipelines that break because the system wasn't hardened early. The next piece of the puzzle picks up there: how to design and run AI systems that stay fast, efficient, and economically defensible as the scale and complexity of the enterprise grow.

Up Next: Performance & Cost

As AI systems scale, performance decisions begin to carry economic consequences. This chapter looks at how latency, accuracy, reliability, and operating constraints can shape cost, value, and long-term viability once systems are live and in use.

Sources

- Dutch regulator fines Clearview AI $33.6 million for GDPR violations
- Clearview AI Fined €30.5m by Dutch Watchdog Over Illegal Data Collection
- Overdue Data Protection Fine for Clearview AI Facial Recognition Software Is Leading to Big Penalties
- Dutch DPA Fines Clearview AI €30.5 Million for Illegal Data Collection
- Fines for GDPR violations in AI systems and how to avoid them
- Samsung Bans ChatGPT Among Employees After Sensitive Code Leak
- Samsung 'bans' employees from using OpenAI's ChatGPT
- Safeguard the Future of AI: The Core Functions of the NIST AI RMF
- How To Align with the NIST AI RMF: Step-by-Step Playbook
- 2025 Global Compliance Trends
- Our journey to becoming the world's first ISO 42001-compliant AI video company
- Cost of a data breach 2025 | IBM
- Facebook's Powerful Large Language Model Leaks Online
- Leaked: Meta's AI Language Model with Unprecedented Power - Bytefeed - News Powered by AI
- AI landlord screening tool will stop scoring low-income tenants after discrimination suit | The Verge
- Final_ AI Security- Case Study
- Google's Secure AI Framework (SAIF) - Google Safety Center
- Secure AI Framework (SAIF) | Google Cloud
- ISO 42001 Risk Controls for AI in High-Assurance Systems
- Final_ AI Security- Case Study
- NIST AI RMF 1.0
- AI in Banking [20 Case Studies] [2025] - DigitalDefynd
- Governance and security for AI agents across the organization - Cloud Adoption Framework | Microsoft Learn
- D-DAE: Defense-Penetrating Model Extraction Attacks
- Run AI Red Teaming Agent Locally (Azure AI Evaluation SDK) - Microsoft Foundry | Microsoft Learn
- Responsible AI and the Evolution of AI Security | Microsoft Community Hub

Content Lead:
Nishita Pereira - Senior Group Manager, Marketing & Communication

Internal Expert Contributors:
Vaibhav Tare – Vice President & Chief Information Security Officer
Bhaskar Gandavabi - Senior Vice President, Technology & Innovation
Anubhav Mukherjee - AI Platform and Solutions Innovator (AI Innovation Hub)
Chandrashakher Aryasomayasula - Associate Vice President, AI Engineering (AI Innovation Hub)
Malavika Nair - Associate, Delivery Support (AI Innovation Hub)

Chapter 4

Performance & Cost

The operating economics of AI

Quick Read

Most AI conversations begin with capability and end with cost. The problem is that by the time cost becomes visible, the system is already embedded in workflows, tied to customer experience, and politically difficult to unwind.

So what truly makes AI survivable in the real world? It isn't superior accuracy alone, but control over how performance translates into economic behavior. That means knowing which use cases justify premium responsiveness, where human oversight should decline over time, how infrastructure is utilized under real demand, and how cost per outcome moves as volume increases.

If you lead an enterprise AI program, focus on:

- Clear limits on performance tiers and usage before broad rollout
- Defined economic thresholds that trigger review or adjustment
- Named ownership for post-launch operating controls
- Visibility into cost per outcome, not just model metrics
- A plan to reduce human review and token intensity as reliability improves

Ask your team to show you:

- Where cost is concentrated today
- How economics change under heavier demand
- Which controls are reviewed regularly and by whom
- What happens to margin if usage doubles
- Evidence that efficiency improves as the system matures

If these answers are clear, the foundation is sound. If they are vague, the work is not finished.

Definition & Context

In classical systems engineering, performance has never meant "how impressive the output looks." It is defined through measurable operating characteristics such as:

A system performs well if it processes work at the required speed, within acceptable error bounds, without degrading as demand increases.These definitions carry directly into AI systems. ISO/IEC guidance frames AI performance in terms of measurable characteristics such as accuracy, precision, recall, and robustness under specified conditions. IEEE 2937:2022 similarly anchors model quality in controlled evaluation environments against defined benchmarks and validation datasets. In both cases, performance is assessed under known inputs and measurable constraints.

That boundary matters.

Standards define behavior under specified conditions. But they don't define what it costs to sustain that behavior once systems are embedded in production environments with unpredictable load, evolving data, and organizational dependencies. Similarly, evaluation frameworks measure correctness. But they don't account for concurrency pressure, infrastructure elasticity, or the economic consequences of guaranteeing low latency at scale.

In enterprise operations, performance becomes a commitment; a guarantee of response time, uptime, consistency, and resilience under real usage. Through put must hold during peak demand. Latency must remain acceptable under parallel calls. Error rates must not \spike as models encounter edge cases. Stability under load becomes as important as benchmark accuracy.

In the same way, cost extends beyond infrastructure line items.

Compute and storage are visible components. Operating AI systems introduces layered costs: monitoring, evaluation cycles, retraining, human review pipelines, rework when outputs fail, compliance documentation, audit preparation, and incident response. Research on AI ROI consistently shows that implementation costs extend well beyond initial deployment, and that infrastructure and integration often exceed early estimates. Surveys from Deloitte and IBM reflect accelerated investment alongside uneven returns. Google's enterprise framing distinguishes short-term productivity gains from long-term transformation, which requires sustained operating investment.

This is where total cost of ownership (TCO) becomes relevant. Performance in production carries cumulative operating costs that rarely appear in initial ROI calculations but compound over time.

The structural issue is that performance and cost are coupled at the architectural level.

Lower latency increases parallel compute demand. Higher reliability requires redundancy and expanded monitoring. Continuous evaluation introduces runtime overhead. Drift detection expands infrastructure footprint. Human-in-the-loop safeguards add review cycles and staffing cost. Each performance commitment embeds an economic obligation into the system.

ROI volatility often reflects this misalignment. Hard ROI, i.e. direct cost savings or revenue lift, is measurable. Soft ROI, i.e. productivity gains or strategic positioning, is harder to attribute. When performance expectations are set without pricing their operating implications, instability appears later because the performance envelope was never economically bounded.

Netflix's long-term investment in performance discipline illustrates the systems dimension of this problem. In consolidating multiple recommendation models into a unified multi-task architecture, the company reduced duplicated pipelines and rising operational complexity. Rather than pursuing incremental model gains in isolation, it addressed technical debt, unified inference infrastructure, and operated under explicit budget targets for model serving. Stability improved not through heavier compute, but through architectural discipline.

The tension is structural. Performance ambition compounds cost exposure as systems mature. For many, AI systems don't collapse at launch but rather, accumulate economic pressure until the operating model becomes unsustainable.

That is the problem this chapter examines.

What People Miss

Technical capability can be measured: benchmarks can be validated, latency can be tracked, and uptime can be enforced. What is harder to measure, though, is how that capability behaves once it is embedded inside a live enterprise with regulatory pressure, customer expectations, legacy systems, and competing incentives. As AI systems move from controlled environments into real operating contexts, their economic profile changes. The gap between what a system can technically achieve and what an organization can sustainably support often widens over time.

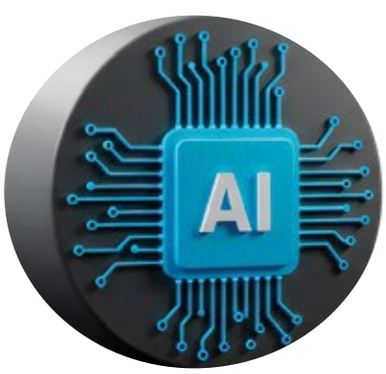
Operating Cost is Treated as a Linear Variable

Enterprise AI budgets are often built on the assumption that operating cost scales proportionally with usage. In practice, cost behaves less like a straight line and more like a compound curve. As systems mature, new obligations surface: integration layers deepen, evaluation cycles expand, compliance reviews intensify, monitoring becomes continuous, and concurrency demands reshape infrastructure. These are not incremental additions but structural expansions. The model may perform as expected, yet the surrounding ecosystem required to sustain it grows in scope and expense, often outpacing the original business case.

Volkswagen Group's Cariad software division illustrates this dynamic in an AI-enabled production environment. Cariad was tasked with building and integrating advanced, AI-driven vehicle software across multiple brands. As codebases grew and system integration deepened, operating losses accumulated despite rising revenue from vehicles running the software. Delays tied to software instability exposed the strain of maintaining performance across a complex, AI-dependent architecture. The financial pressure did not stem from a single failed feature but from the sustained cost of supporting and validating AI-enabled systems at automotive scale, where integration depth, testing, validation, and system coherence proved far more expensive than originally priced.

Performance Can Constrain Value Even When Costs are Controlled

An AI system can operate within budget and meet its technical targets yet still narrow the business's room to maneuver. When performance fluctuates under real-world scrutiny, regulatory oversight intensifies, claims are restricted, and deployment boundaries tighten. The economic impact in such cases does not show up as runaway spending but as constrained operating permission. Revenue assumptions tied to autonomy, scale, or automation becomedifficult to defend when reliability under live conditions remains uneven, even if the underlying infrastructure remains financially stable.



Tesla's Autopilot and Full Self-Driving systems provide a clear instance of this tension. The underlying AI stack—computer vision models, sensor fusion systems, and neural network inference pipelines—continued to operate and evolve. However, documented crashes and regulatory investigations led to recalls, revised claims, and constraints on how the systems could be described and deployed. Infrastructure spending did not publicly spiral, yet real-world performance variability shaped what could be promised, sold, and expanded. In this case, AI performance under scrutiny directly influenced operating permission and value realization, independent of cost overruns.

High Technical Performance Does Not Ensure Economic Viability

Enterprise AI systems are often evaluated on adoption, engagement, and technical functionality. If usage grows and the system performs reliably within defined parameters, the assumption is that value will follow. What gets missed is whether each interaction actually carries economic weight. An AI system can be accurate, responsive, and widely deployed while generating interactions that are too low in value to sustain its infrastructure, staffing, and long-term development costs. Performance may be stable. Costs may be controlled. Adoption may be high. Yet the economic density per interaction remains thin.

Amazon's Alexa ecosystem makes this visible in a large-scale AI deployment. Voice recognition worked. Intent classification functioned. Device adoption surpassed one hundred million units, and interaction volume increased year over year. However, most usage was clustered around low-value tasks such as timers, weather queries, and basic music commands. Maintaining the cloud-based natural language processing stack and hardware ecosystem required significant ongoing investment, while monetization per interaction remained limited. The AI system functioned at scale, but the economic return per use case remained thin relative to its operational footprint.

Scaling Ambition Outruns Validated Operational Fit

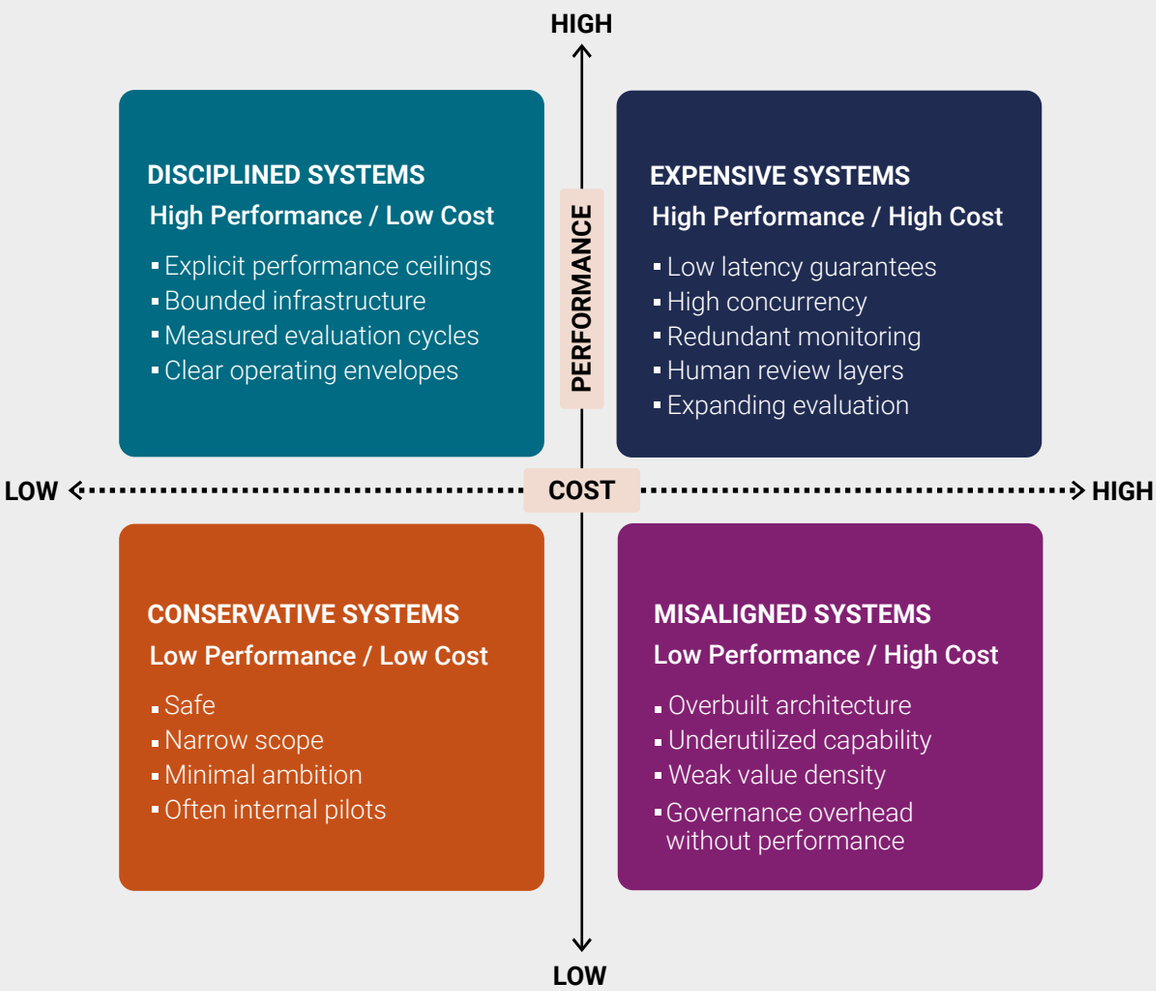
Enterprise AI programs often assume that once a model performs well in controlled evaluation settings, scaling becomes a matter of rollout and change management. The technical milestone is treated as proof of readiness. What gets underestimated is the difference between demonstrable capability and operational fit. Performance in lab conditions does not automatically translate into performance inside complex, high-stakes workflows where decisions are contextual, multidisciplinary, and constrained by institutional norms. When scaling proceeds before that fit is validated, integration layers deepen, stakeholder exposure widens, and the cost of adjustment multiplies. What began as a promising system becomes an expensive coordination problem.

IBM's Watson for Oncology highlights this gap in a clinical AI context. The system generated treatment recommendations using trained models and curated data, supported by significant investment. However, hospitals reported difficulty aligning those recommendations with real-world oncology practice shaped by patient variability, local protocols, and physician judgment. Integration into clinical workflows required substantial effort, and confidence in outputs varied across institutions. The issue was not a lack of technical sophistication, but the gap between modeled performance and validated, repeatable utility in live medical environments. As deployment expanded, integration and oversight costs increased while adoption lagged.

More often than not, AI systems alter the economics of decision-making in ways that are not always visible at the point of approval. What appears stable in dashboards can still be misaligned in incentives, monetization logic, or operating permissions.

Addressing these realities requires more than tighter budgeting or better benchmarks. Enterprises need a deliberate alignment between what the system is designed to optimize, how its outputs translate into business value, and what the organization is structurally prepared to sustain. Without that alignment, even technically sound systems create friction over time.

AI System Performance Vs Cost Matrix



Understanding Performance vs Cost Matrix

X-Axis: Cost Exposure
This represents total operating exposure: compute, monitoring, evaluation cycles, compliance overhead, human review, incident recovery. The full TCO surface.

Y-Axis: Performance Commitment
This reflects guarantees: latency promises, concurrency levels, uptime expectations, redundancy, evaluation depth.

- **Top Left: Disciplined or Economically Stable Systems**
  High performance commitment + Low cost exposure
  This is the ideal state. These systems don't promise more than they can sustain. Performance is strong but economically engineered. This quadrant represents architectural maturity.

- **Top Right: Expensive or Economically Fragile Systems**
  High performance commitment + High cost exposure
  This is where ambition runs hot. These systems may be technically impressive but every performance promise multiplies economic obligation. They become structurally expensive to maintain.

- **Bottom Left: Conservative or Controlled Systems**
  Low performance commitment + Low cost exposure
  These AI systems are safe, predictable, and narrow. Nothing breaks, but nothing scales meaningfully either. This quadrant is economically safe but strategically constrained.

- **Bottom Right: Misaligned Systems**
  Low performance commitment + High cost exposure
  These systems are a waste. High cost without meaningful performance guarantees.
  This is often the result of experimentation that never translated into operating discipline.

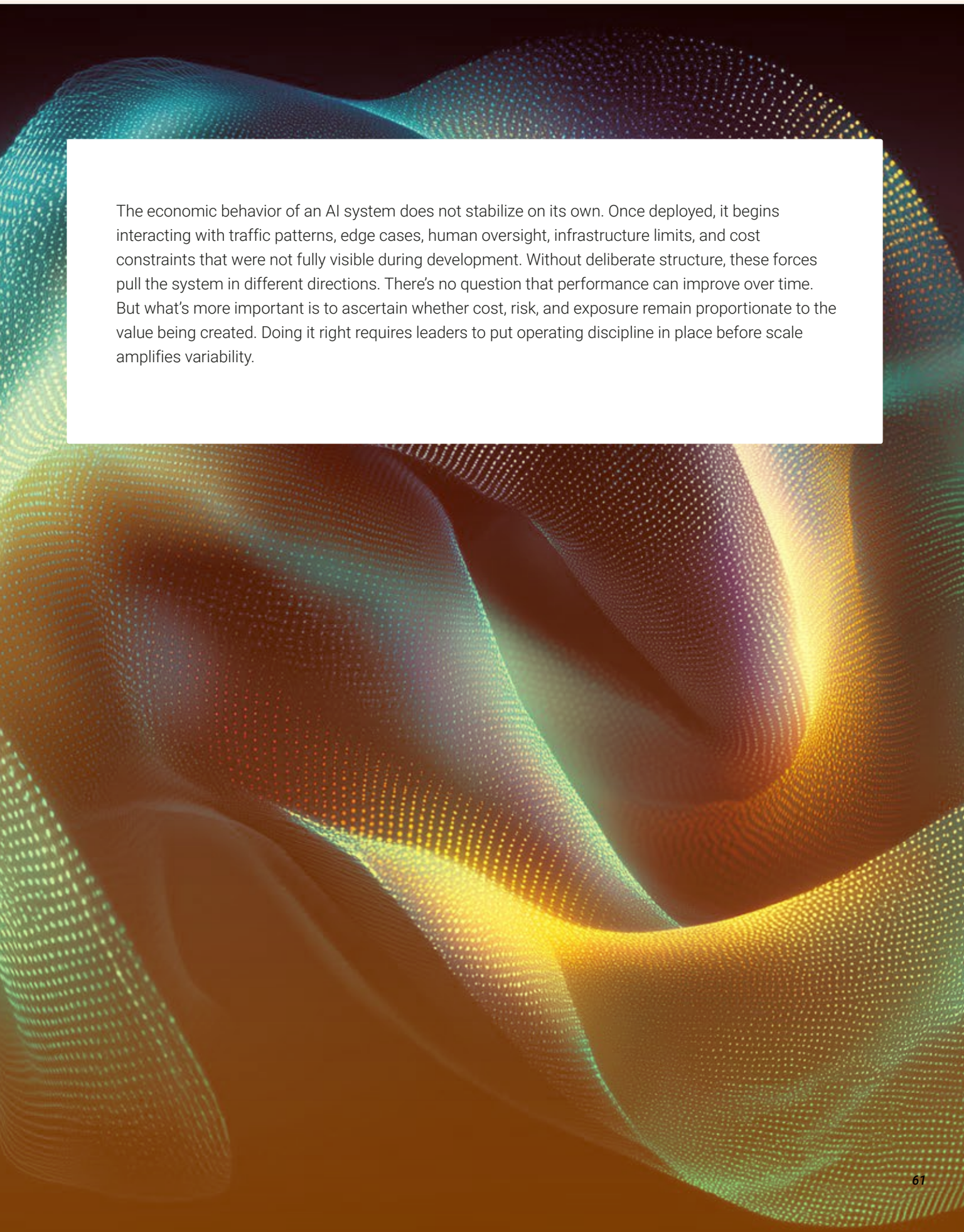
Bottomline: Performance commitments must be economically bounded and cost exposure must correspond to value density.

Performance and cost are independent dimensions but they still influence each other. Instability can emerge from either direction:

- **Over-ambition (top right)**
- **Underutilization (bottom right)**

AI systems risk becoming economically unstable if commitments are mispriced or misaligned.

How to Do it Right



The economic behavior of an AI system does not stabilize on its own. Once deployed, it begins interacting with traffic patterns, edge cases, human oversight, infrastructure limits, and cost constraints that were not fully visible during development. Without deliberate structure, these forces pull the system in different directions. There's no question that performance can improve over time. But what's more important is to ascertain whether cost, risk, and exposure remain proportionate to the value being created. Doing it right requires leaders to put operating discipline in place before scale amplifies variability.

1. Define the Value Equation Up Front

Enterprise AI systems should be designed with explicit performance limits tied to economic logic from the outset. Rather than assuming every user or workflow requires peak responsiveness, organizations need to define where high-speed execution is mission-critical and where slower, lower-cost processing is acceptable. Capacity, responsiveness, and access should be structured intentionally, with clear ceilings and monetization or value alignment behind them. When performance is segmented and bounded early, cost exposure remains visible and controllable. Economic discipline becomes part of system architecture rather than a corrective measure after growth.

Cursor, an AI-powered code editor developed by Anysphere, embeds these constraints directly into its product model. The platform differentiates between fast and slower requests, caps high-performance usage, and allows additional capacity to be purchased when needed. Users can set spending limits and automatically shift to lower-priority processing once thresholds are reached. Performance is not unlimited, nor is it hidden. It is tiered, metered, and priced in alignment with usage. By placing economic boundaries at the product level, Cursor avoids the trap of offering uniform peak performance that would otherwise create unbounded compute exposure at scale.

We treat cost and performance control as architectural decisions rather than operational afterthoughts. Within the FD Ryze platform, incoming prompts are evaluated before compute is committed. The LLM serving layer classifies each request based on query complexity and routes it to the most appropriate model. This allows the platform to match workload intensity with the right compute resource, preventing high-cost models from being used when simpler responses will suffice. Several platform mechanisms reinforce this control layer:

- Complexity-based routing directs prompts to the appropriate model tier.
- Token and budget controls manage usage at user, team, and organizational levels.
- Human validation layers route uncertain outputs for review before additional compute is triggered.

SUMMARY

Value-Bound AI Architecture

The mistake most enterprises make is assuming that peak AI system performance should be universally available across every user, workflow, and use case. But if capability is unconstrained, cost exposure can scale invisibly. Before deployment, leaders should be able to answer a simple question: where is premium performance justified and where is it not?

Cursor's product model demonstrates a clean approach. Performance is tiered, usage is metered, and high-speed compute is bounded by explicit limits and spend controls. This way, users can see and control their exposure. The economic logic is embedded into the product itself, not retrofitted after scale.

FULCRUM IN PRACTICE

- Complexity-based routing matches each prompt to the right model tier.
- Usage controls govern tokens, budgets, and keys across teams and users.
- Validation gates route uncertain outputs for review before further compute or action.

Takeaway: Require every AI initiative to define its performance tiers, spend ceilings, and routing logic before approving scale.

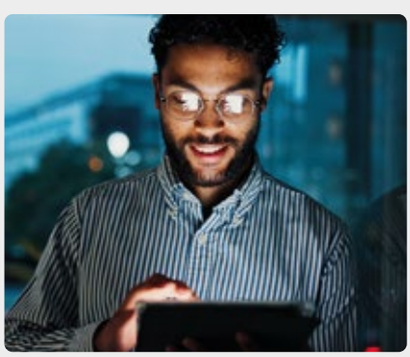
2. Tie Model Performance to Financial Signals

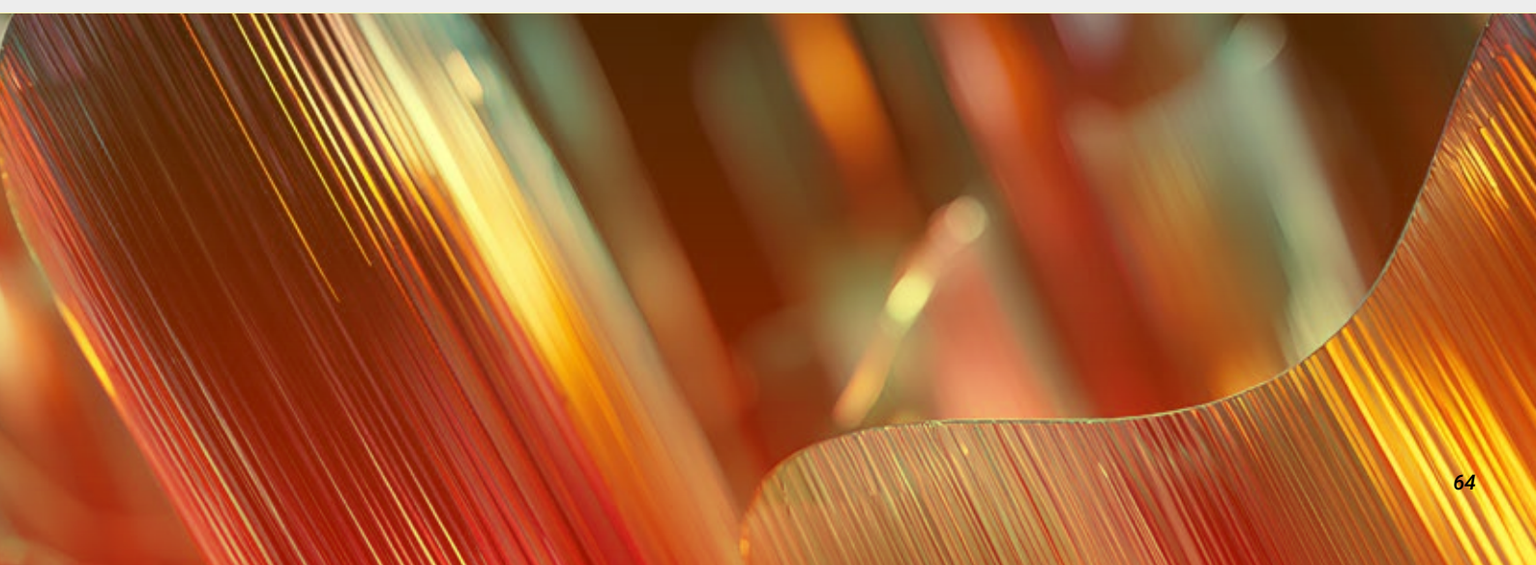
Once an AI system is live, accuracy metrics are not enough. It is important that performance be tracked in the same currency the business uses to make decisions. That means tying model outputs to margin, loss exposure, customer impact, and operating thresholds and not just precision, recall, or latency dashboards. Continuous instrumentation requires defining the economic breakpoints upfront (what level of error is tolerable, what level of drift is costly, what threshold changes revenue exposure), monitoring them in real time, and adjusting both model and policy as conditions evolve. Without this layer, AI systems can remain technically "high performing" while silently eroding value.

Stripe's machine learning-driven fraud detection system, Radar, uses AI models to score every transaction in real time for fraud risk. Radar does not evaluate model quality in isolation; instead, it links fraud detection thresholds to merchant margins, chargeback costs, and false-decline impact. Performance is monitored through precision-recall curves and score distributions, but operational decisions are made at defined economic operating points. Models are retrained as fraud patterns drift, dashboards track block rates and merchant-level effects before release, and thresholds can be tuned to balance fraud loss against revenue friction. In this setup, model performance is inseparable from financial consequence, exactly the discipline enterprise AI systems require to prevent economic drift after deployment.

We monitor model activity alongside the economic signals that determine whether the system is delivering value in production. The platform tracks token consumption, usage behavior, and validation activity through monitoring and reporting layers, giving teams visibility into how AI performance translates into compute cost and operational overhead. This allows performance decisions to be read in business terms, not only in model terms, and makes it easier to spot when quality gains are being offset by rising usage or review effort.

- Token consumption tracking links model activity to real compute cost.
- Operational analytics and reporting show how usage patterns affect system economics.
- Validation activity monitoring helps teams see when review overhead begins to outweigh model gains.





SUMMARY

Financially Accountable Performance

An AI system can hit its precision targets and still increase loss exposure, friction, or operational drag. Once deployed, performance must be measured in the same units the business manages: revenue, cost, risk, and customer impact. Leaders should be able to answer what level of model error changes their economics and how quickly would they know?

In the case of Stripe's Radar system, fraud thresholds are not set in isolation but are tied to merchant margins, chargeback costs, and false-decline impact. Model updates are evaluated not just for technical lift but for their financial effect before release.

FULCRUM IN PRACTICE

- Cost signals sit beside model signals in live system reporting.
- Usage patterns are visible before they turn into economic drag.
- Review effort is tracked so oversight costs do not grow unnoticed.

Takeaway: Define economic breakpoints for every production model and monitor them continuously.

3. Actively Manage Economic Controls in Production

Once an AI system goes live, its economics begin to move. A threshold set conservatively during launch may double human review costs six months later. A prompt refinement intended to improve output quality may instead increase token burn. A spike in low-confidence cases may overload operations without triggering any infrastructure alert. Enterprise AI systems do not remain economically stable without intervention. They require deliberate, recurring adjustment of the levers that determine cost exposure and performance boundaries.

Doing this well means treating those levers as operational controls. Review confidence thresholds against actual resolution cost and risk tolerance. Monitor human-in-loop ratios with an explicit reduction plan tied to measurable accuracy gains. Audit prompt structures for token efficiency as rigorously as output quality. Analyze exception queues not just for volume but for economic weight. Enterprises need to tighten or relax guardrails based on real operating conditions. Eventually, this becomes a cadence: review, adjust, measure, repeat. When those adjustments are routine, cost curves flatten instead of creeping upward.

Economic controls continue after deployment through live monitoring and adjustment of how AI systems behave in production. We track token usage, cost exposure, and validation activity as systems operate, while adaptive agents learn from outcomes and refine behavior over time. This allows teams to observe where review effort is rising, where system confidence is improving, and where operational overhead can be reduced without weakening oversight.

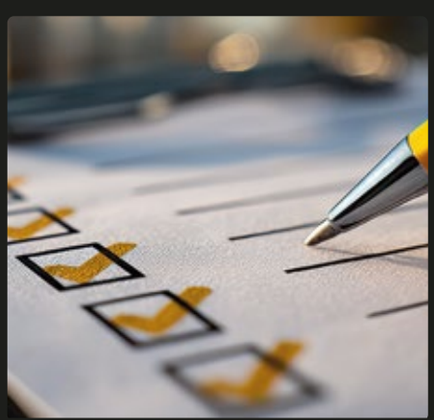
- Live usage tracking gives teams visibility into token consumption and cost exposure in production.
- Human validation layers help review effort shift as confidence and reliability improve.
- Adaptive agents learn from outcomes, reducing avoidable intervention over time.

SUMMARY

Production-Level Economic Discipline

When an AI system enters production, its economics do not stay fixed. Automation rates change, confidence distributions shift, human review expands or contracts, and prompt modifications alter token consumption. Together, these movements determine whether the system becomes more efficient or more expensive over time.

Production discipline means someone is responsible for actively managing those levers. Thresholds are reviewed against real operating cost. Human oversight is reduced deliberately as reliability improves. Prompts are refined for efficiency and exception volumes are evaluated for financial weight. If managed deliberately, these variables can reduce cost per outcome over time.



FULCRUM IN PRACTICE

- Production monitoring tracks token usage, cost exposure, and system behavior in real time.
- Review levels evolve as system confidence improves and human validation becomes more targeted.
- Adaptive agents learn from outcomes, reducing avoidable intervention over time.

Takeaway: Establish a recurring review cycle to monitor AI economics in production and adjust thresholds, review levels, and prompts as system behavior evolves.

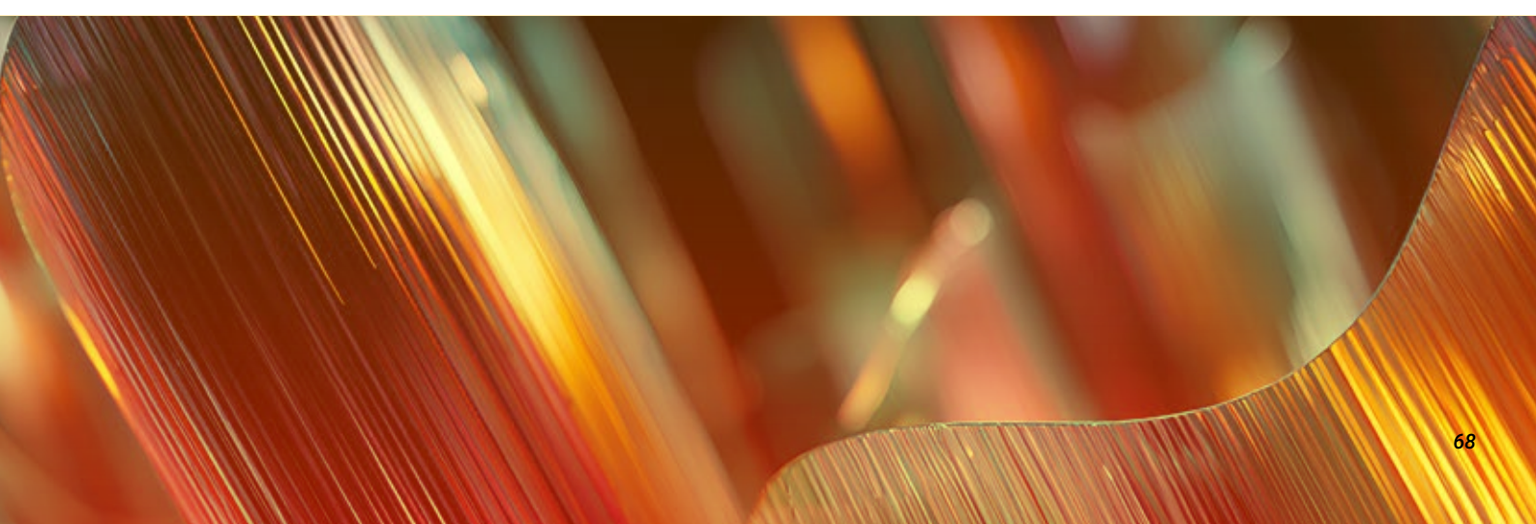
4. Control the Cost Structure Beneath the Model

Once AI systems move into production, cost leakage tends to accumulate across layers: idle GPUs, inefficient batching, unnecessary data transfer, under-optimized memory usage, and poorly governed resource allocation. None of these failures appear dramatic on dashboards. Yet over time, they compound into material economic drag. Enterprises must treat infrastructure not as a static environment but as an actively governed system. Compute must be allocated dynamically, inference stacks must be continuously optimized, network traffic must be controlled, data movement must be minimized, and monitoring must operate at the architectural layer, not just the model layer. Structural governance is what prevents technically sound AI systems from becoming economically inefficient ones.

NVIDIA's Run:ai platform shows what disciplined compute governance looks like in practice. Instead of assigning full GPUs to workloads by default, it enables fractional allocation and priority-based scheduling, increasing utilization and reducing idle capacity waste. At the inference layer, OpenAI's public 80% price reduction for its o3 model—driven by serving stack optimization rather than model downgrade—demonstrates a similar principle: improving infrastructure efficiency to lower unit costs without sacrificing capability. The same structural discipline applies across network configuration, storage design, and data locality, where preventing unnecessary movement and congestion protects long-term AI economics.

We govern the infrastructure beneath the model as a connected operating layer. The platform separates data retrieval, search, and model serving so that requests move through the stack with less unnecessary movement and better control over performance. System health, performance trends, and resource behavior are continuously observed through monitoring and analytics, allowing infrastructure inefficiencies to be surfaced and addressed before they turn into persistent cost drag.

- Layered service architecture separates storage, enterprise search, and model serving.
- Monitoring dashboards and analytics track system health, performance, and utilization.
- Identity and access controls govern how services and resources interact across the stack.



SUMMARY

Infrastructure Cost Control

Model capability may be the visible part of an AI system, but infrastructure determines its long-term economics. Idle GPUs, inefficient batching, unnecessary data transfer, and poorly governed resource allocation rarely trigger alarms. Yet over time, they compound into material cost drag. Leaders need to understand where infrastructure inefficiency is eroding margin.

NVIDIA's Run: AI platform shows what compute governance looks like in practice: fractional GPU allocation and priority scheduling to prevent idle capacity waste. At the serving layer, OpenAI's 80% price reduction for o3, driven by inference stack optimization rather than model downgrade, illustrates the same principle. Cost improvement came from improving infrastructure efficiency and not lowering capability.

FULCRUM IN PRACTICE

- Layered infrastructure keeps data retrieval, search, and model serving from becoming one inefficient stack.
- System-level monitoring surfaces infrastructure waste before it turns into recurring cost drag.
- Governed service access keeps resources controlled across the platform layer.

Takeaway: Every production AI system must track infrastructure efficiency, utilization, and data movement alongside model cost.

AI SYSTEMS ECONOMIC LEAKAGE MAP

MODEL LAYER	INFERENCE LAYER	HUMAN OVERSIGHT	DATA & MOVEMENT	GOVERNANCE
Oversized Model	Idle GPUs	Review Creep	Cross-Region Traffic	No Cost Owner
Context Creep	Poor Batching	Static Thresholds	Duplicate Storage	No Attribution
Prompt Bloat	No Quantization	Queue Backlog	Long Retrieval Chains	No ROI Window
No Batching	Cache Misses	No Reduction Plan	No Data Pruning	No Utilization Target
Low Confidence Gates	Reactive Scaling	Guardrail Drift	Retrieval Blindness	No Theresgold Owner

Enterprise AI becomes sustainable only when its financial behavior is as visible and governable as its technical performance. Leaders should be able to explain where money is being spent, why it is being spent there, and how those economics will evolve under heavier demand. If those mechanics are not understood at today's scale, they will not correct themselves at ten times the volume. It is naive to assume that increased adoption will dilute structural weaknesses; it only amplifies them. For that reason, economic stability must be demonstrated before demand accelerates.

Up Next: Accountability

As AI systems begin influencing real operational decisions, questions about performance and cost give way to a more fundamental concern: who inside the organization is responsible for what those systems do. This chapter examines how enterprises assign ownership, maintain traceability for AI-driven outcomes, and preserve clear decision authority.

Sources

- Artificial Intelligence Risk Management Framework (AI RMF 1.0)
- IEEE Standard for Performance Benchmarking for Artificial Intelligence Server Systems
- The CEO's Guide to Generative AI: Cost of compute | IBM
- AI infrastructure compute strategy | Deloitte Insights
- Beyond Benchmarks: The Economics of AI Inference
- The ROI of AI 2025 | Google
- AI ROI: The paradox of rising investment and elusive returns
- Netflix Research
- VW Group's Cariad Software Division Had A Bad Year. Again
- The Biggest AI Fails of 2025: Lessons from Billions in Losses
- Tesla Autopilot and Full Self-Driving AI Under Fire After Fatal Crash and Public Criticism - OECD.AI
- (Mis-)use of standard Autopilot and Full Self-Driving (FSD) Beta: Results from interviews with users of Tesla's FSD Beta - PMC
- Exclusive | Amazon, in Broad Cost-Cutting Review, Weighs Changes at Alexa and Other Unprofitable Units - WSJ
- Amazon Puts Alexa Under Microscope in Cost-Cutting Review
- What Ever Happened to IBM's Watson? - The New York Times
- How IBM Watson Overpromised and Underdelivered on AI Health Care - IEEE Spectrum
- IBM Watson: From healthcare canary to a failed prodigy
- Report: Anysphere Business Breakdown & Founding Story | Contrary Research
- Stripe: Radar Technical Guide
- O3 is 80% cheaper and introducing o3-pro - Announcements - OpenAI Developer Community
- Maximizing GPU Utilization using NVIDIA Run:ai in Amazon EKS | Containers

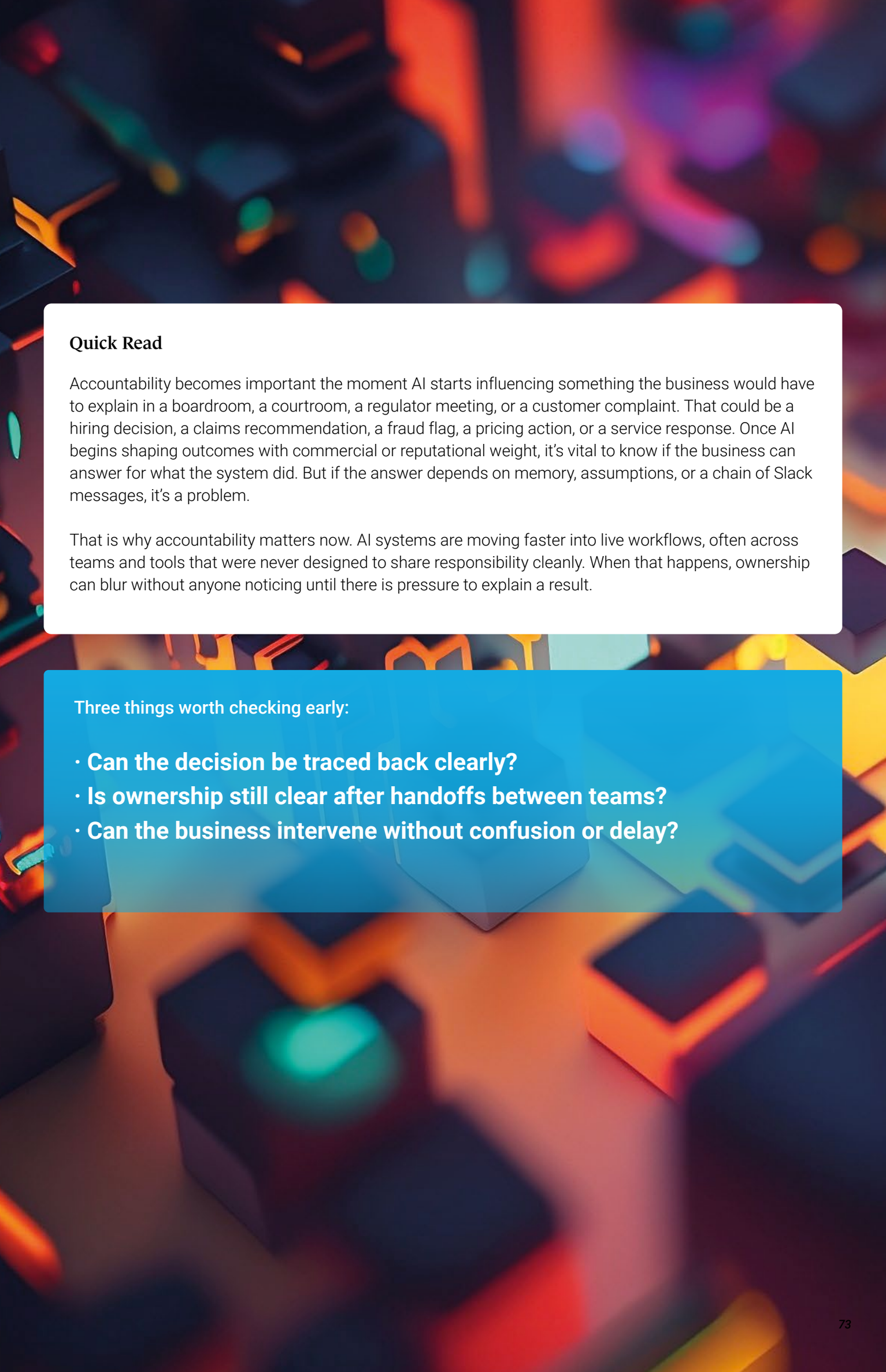
Content Lead:
Nishita Pereira - Senior Group Manager, Marketing & Communication

Internal Expert Contributors:
Bhaskar Gandavabi - Senior Vice President, Technology & Innovation
Chandrashakher Aryasomayasula - Associate Vice President, AI Engineering (AI Innovation Hub)
Shalin Mayank - Senior AI Platform and Solutions Innovator (AI Innovation Hub)
Anubhav Mukherjee - AI Platform and Solutions Innovator (AI Innovation Hub)
Malavika Nair - Associate, Delivery Support (AI Innovation Hub)

Chapter 5

Accountability

Owning outcomes in automated systems



Quick Read

Accountability becomes important the moment AI starts influencing something the business would have to explain in a boardroom, a courtroom, a regulator meeting, or a customer complaint. That could be a hiring decision, a claims recommendation, a fraud flag, a pricing action, or a service response. Once AI begins shaping outcomes with commercial or reputational weight, it's vital to know if the business can answer for what the system did. But if the answer depends on memory, assumptions, or a chain of Slack messages, it's a problem.

That is why accountability matters now. AI systems are moving faster into live workflows, often across teams and tools that were never designed to share responsibility cleanly. When that happens, ownership can blur without anyone noticing until there is pressure to explain a result.

Three things worth checking early:

- **Can the decision be traced back clearly?**
- **Is ownership still clear after handoffs between teams?**
- **Can the business intervene without confusion or delay?**

Definition & Context

Once AI systems move deeper into enterprise operations and after questions like how the system works, how secure it is, or how efficiently it runs have been answered, the next big concern is who inside the organization is responsible for what the system does. Accountability exists to answer that question. It defines who owns the outcomes produced by AI systems and who must answer when those outcomes affect customers, employees, financial performance, or regulatory obligations.

This question becomes unavoidable as AI begins to influence real operational decisions. Systems that recommend hiring candidates, adjust prices in real time, influence underwriting outcomes, or shape customer interactions are no longer confined to analytical support. They participate in decisions that carry legal, financial, and reputational consequences. In these situations, organizations must maintain a clear line connecting those outcomes to human authority within the enterprise. Research on AI governance refers to the difficulty of maintaining that link as the "responsibility gap," where the effects of automated decisions are visible but responsibility becomes harder to trace across complex systems and actors.

Across policy and governance frameworks, accountability is consistently described as responsibility for the impacts produced by AI systems. The OECD's AI governance principles state that organizations and individuals involved in the lifecycle of an AI system remain responsible for its outcomes and impacts, including the obligation to establish governance structures, maintain documentation, and address harms when they occur. Similarly, the NIST AI Risk Management Framework links accountability to clearly assigned ownership, documented responsibilities, and the ability to investigate system outcomes when decisions need to be reviewed. From a governance perspective, the U.S. National Telecommunications and Information Administration describes accountability as a chain connecting how AI systems are built, how they are evaluated, and how organizations respond to their outcomes.

Taken together, these frameworks describe accountability as an operational capability rather than a single policy. Enterprises must know who is responsible for system behavior, maintain the information needed to review decisions, and retain the authority to intervene when outcomes require correction.

This becomes more complex in AI environments because responsibility spreads across many participants. Developers build models, vendors provide tools and platforms, internal teams integrate systems into workflows, and business units rely on those systems during daily operations. Without clearly defined ownership across this chain, responsibility can become diffuse even though the business consequences remain real. This diffusion of responsibility is one of the defining risks of modern AI systems.

Accountability becomes visible in the everyday decisions that organizations must be able to defend. In financial services, accountability may involve identifying who stands behind automated underwriting or credit scoring decisions that influence customer eligibility. In retail and e-commerce environments, accountability can determine who is responsible for pricing systems or recommendation engines that shape how customers are treated. In hiring or talent systems, accountability determines who answers when algorithmic screening tools influence employment decisions. In each case, AI systems participate in the operational machinery of the organization, and accountability ensures that responsibility for those outcomes remains anchored to identifiable roles inside the enterprise.

For that reason, accountability functions as the point where the broader system of enterprise AI governance ultimately converges. Reliable systems ensure decisions behave consistently, while explainable systems allow organizations to understand how those decisions were produced. Security and compliance controls protect the system and its data, and performance and cost management determine whether the system operates sustainably. In the end, it is accountability that ensures the organization can stand behind the decisions those systems influence.

When accountability is clearly defined, enterprises maintain control as AI systems expand across teams and workflows. When it is unclear, even well-designed systems can produce outcomes that no one inside the organization is prepared to explain, justify, or correct.

Distinguishing Accountability from Related AI Governance Terms

EXPLAINABILITY

Explainability refers to the ability to understand how an AI system produced a particular outcome. Techniques such as model interpretation tools or decision explanations allow teams to examine the factors that influenced a result. Explainability helps organizations understand system behavior, but understanding a decision does not by itself determine who is responsible for it.

TRANSPARENCY

Transparency describes the degree to which information about an AI system is visible or disclosed. This can include documentation about training data, model design, system limitations, or operational processes. Transparency supports oversight and trust, yet visibility alone does not assign responsibility for the system's decisions.

AUDITABILITY

Auditability refers to the ability for independent reviewers to examine how an AI system operates. Systems that maintain logs, documentation, and traceable decision records allow auditors to reconstruct what happened and evaluate whether governance controls were followed. Auditability provides evidence for oversight, but it does not define who ultimately answers for the outcome.

RESPONSIBILITY

Responsibility describes the tasks and roles assigned to individuals or teams throughout the lifecycle of an AI system. Engineers may be responsible for model development, product teams for deployment decisions, and operations teams for monitoring system performance. Responsibility distributes work across the organization but does not necessarily determine who bears the consequences of the system's actions.

ACCOUNTABILITY

Accountability exists when the organization clearly identifies who must answer for the outcomes produced by an AI system and who has the authority to intervene when problems occur. It connects the system's actions to decision authority inside the enterprise. While explainability, transparency, and auditability help organizations understand and review system behavior, accountability ensures that responsibility for the system's impact remains anchored to identifiable actors.

LIABILITY

Liability concerns the legal consequences that may arise from the actions of an AI system. Courts or regulators may determine liability when automated decisions lead to harm, discrimination, or financial damage. Liability often depends on how accountability is structured within the organization, since unclear governance can complicate the assignment of legal responsibility.

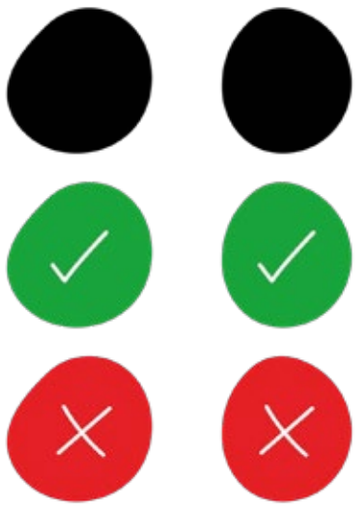
What People Miss

On paper, accountability can appear straightforward in most AI governance frameworks if roles are defined, oversight is assigned, and organizations implement processes intended to ensure that automated systems operate responsibly.

Yet when real incidents occur, accountability often becomes harder to trace than expected. Research shows that even organizations following accepted governance practices encounter blind spots that only become visible once systems operate in real environments.

Treating AI decisions as separate from the organization that deploys them

Many enterprises assume that AI outputs exist in a category separate from ordinary business decisions. Systems may be described as assistants, tools, or automated services, creating the impression that responsibility sits somewhere outside the organization itself. That distinction rarely survives scrutiny once the system's outputs begin affecting customers, employees, or the public.



In 2022, Air Canada's website chatbot provided incorrect information about bereavement fares to a customer. Upon being sued, the airline argued that it should not be liable for the chatbot's statements. But in 2024, the tribunal rejected the claim, noting that the chatbot was simply part of the company's website and that the airline remained responsible for the information it delivered. In the 2025-2026 federal class-action lawsuit Mobley v. Workday, a U.S. federal court allowed claims to proceed against an AI hiring platform under the theory that the vendor's system could function as an employer's agent when screening applicants.

These cases reflect a broader shift: when AI systems perform organizational functions, their outputs are increasingly treated as decisions made by the organization itself.

Assuming that AI responsibility is centralized

Even when enterprises attempt to assign responsibility for AI systems, they tend to assume that AI belongs to a single team. Unlike traditional software systems, which often have clear ownership structures, AI systems operate across multiple layers of an organization. Data teams manage training data, engineers build models, product teams deploy the systems, and business units rely on the outputs for operational decisions.

This fragmentation becomes most visible when something goes wrong. A minor crash involving a Zoox autonomous robotaxi in 2025 illustrates how responsibility can move quickly across layers of an organization. The investigation examined software design, operational controls, corporate oversight, and regulatory compliance. What appeared to be a single system failure ultimately involved decisions made across engineering, management, and governance structures. AI systems frequently sit at the intersection of technology, business process, and decision-making, making accountability harder to isolate than traditional software systems.

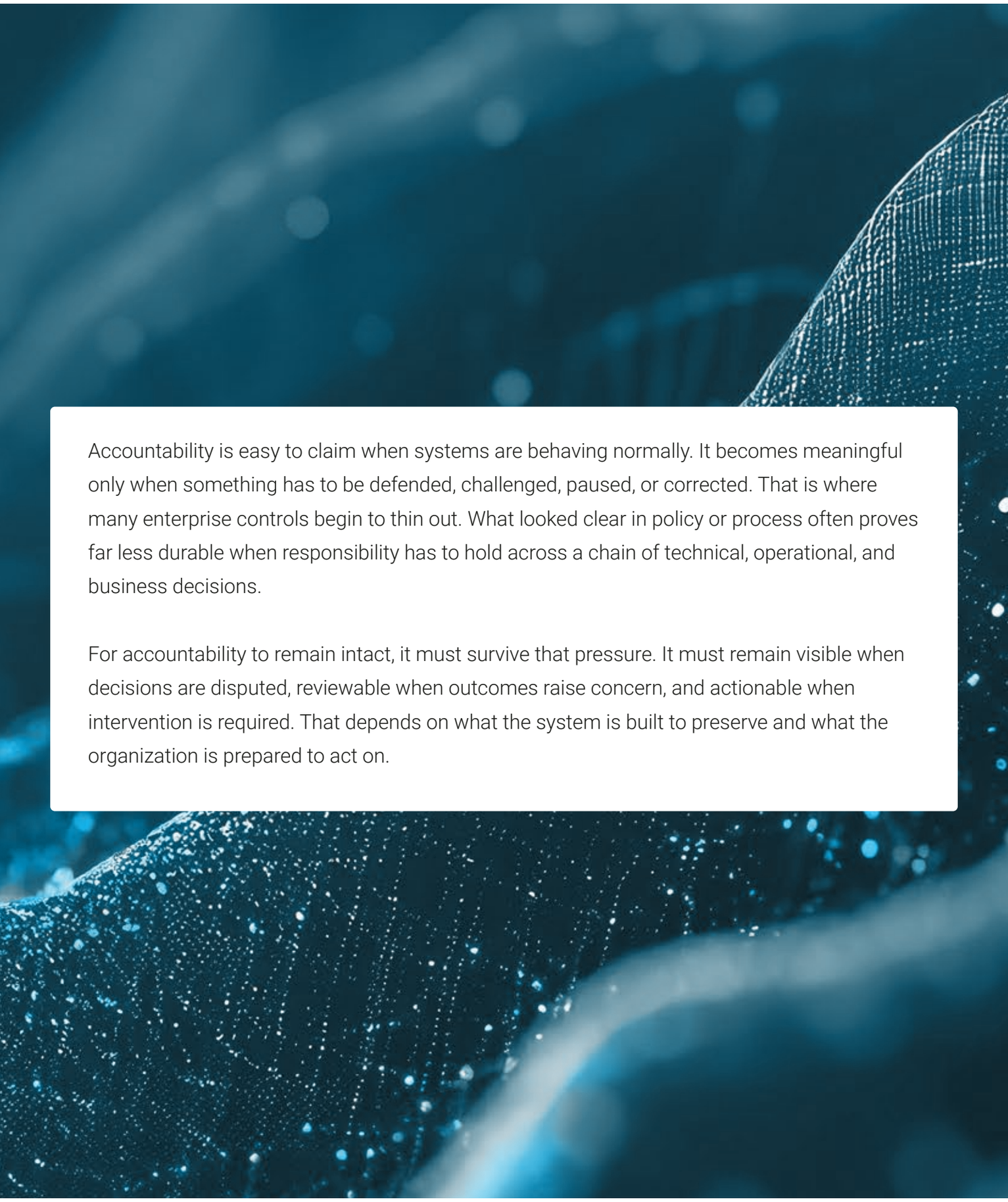
Relying only on human oversight to keep AI accountable

Many regulatory frameworks assume that assigning a human reviewer to AI-driven decisions preserves accountability. Human oversight is widely treated as a safeguard against automated errors or bias, which means that a person monitoring the system can intervene, override incorrect outputs, or halt the system when necessary.

But empirical research suggests that this protection is often weaker than expected. Studies of algorithmic decision-making consistently show that people tend to defer to automated recommendations, a pattern known as automation bias. In high-volume environments, oversight can degrade even further as reviewers face large numbers of alerts or decisions each day. Research on algorithm governance warns that nominal oversight can turn into rubber-stamping, where human reviewers approve automated outputs without meaningful scrutiny. In those situations, oversight mechanisms may legitimize flawed systems rather than provide the protection they were intended to offer.

These blind spots reveal a common theme: accountability cannot be assumed simply because roles, policies, or oversight structures exist. When AI systems begin operating at scale, responsibility must be deliberately engineered into the way the system is designed, monitored, and governed.

How to Do It Right



Accountability is easy to claim when systems are behaving normally. It becomes meaningful only when something has to be defended, challenged, paused, or corrected. That is where many enterprise controls begin to thin out. What looked clear in policy or process often proves far less durable when responsibility has to hold across a chain of technical, operational, and business decisions.

For accountability to remain intact, it must survive that pressure. It must remain visible when decisions are disputed, reviewable when outcomes raise concern, and actionable when intervention is required. That depends on what the system is built to preserve and what the organization is prepared to act on.

1. Make accountability traceable through system evidence

Accountability becomes practical only when organizations can reconstruct how an AI-driven outcome occurred. When a decision affects a customer, employee, or financial transaction, the enterprise must be able to determine what data informed the system, which model produced the result, and how the system behaved at that moment in time. This depends on an operational evidence layer that captures how the system functions in practice.

Treat traceability as part of system architecture to make accountability truly measurable. Decision logs, model version records, monitoring data, and other operational traces allow teams to examine system behavior after the fact and investigate outcomes when questions arise. Governance frameworks such as the NIST AI Risk Management Framework emphasize maintaining this traceability because organizations must be able to review system behavior and respond when decisions require explanation or correction. Accountability only functions when the evidence needed to examine system actions already exists.

Accountability Evidence Checklist

Can your organization reconstruct how an AI decision was made?

The following elements ensure that organizations can examine, investigate, and correct outcomes when necessary.

1. End-to-End Logging
System activity is recorded across the full decision pipeline, capturing inputs, outputs, and key processing steps.

2. Decision Records
Individual AI outputs can be linked to timestamps, users, transactions, or operational events.

3. Model Version Tracking
The exact model version active at the time of a decision can be identified and reconstructed.

4. Operational Monitoring History
Performance metrics, alerts, and system behavior are continuously tracked and retained.

5. Audit Logs
Independent reviewers can access traceable records showing how decisions were generated and handled.

6. Post-Incident Analysis Capability
When anomalies or complaints occur, teams can reconstruct system activity and conduct a structured investigation.

SUMMARY

MAKE ACCOUNTABILITY RECONSTRUCTABLE

When accountability is tested, memory is never enough. Enterprises need evidence that allows them to reconstruct how an AI-driven outcome occurred, what conditions shaped it, and whether the system behaved as expected at that point in time. If that evidence does not already exist inside the operating environment, responsibility becomes difficult to examine and even harder to defend.

FULCRUM IN PRACTICE

Record activity across the full decision pipeline.

Maintain evidence that supports audit and review.

Treat post-incident reconstruction as a standing capability.

2. Anchor accountability across the AI lifecycle

In enterprise environments, AI decisions rarely originate from a single team or function. Data pipelines are prepared by one group, models are developed by another, systems are integrated into workflows by product or engineering teams, and business units rely on the outputs during daily operations. When accountability is treated as the responsibility of a single owner, these distributed roles can leave gaps in oversight once the system begins operating in production.

Enterprises need to structure accountability checkpoints at each stage of the AI lifecycle. Data sourcing, model development, evaluation, deployment, and operational monitoring all carry clearly defined ownership and review responsibilities. Emphasize documenting these roles so organizations understand who is responsible for system behavior at every stage. When accountability is mapped across the lifecycle in this way, responsibility remains continuous even as AI systems evolve, scale, and interact with multiple parts of the enterprise.

Lifecycle Accountability Map

Where accountability should exist across the AI system lifecycle

A lifecycle map ensures that accountability is not limited to the team that built the model. It follows the system from the moment data enters development through ongoing operation and incident response.

LIFECYCLE STAGE	ACCOUNTABILITY CHECKPOINT
Data Sourcing & Preparation	Data ownership is defined and datasets are documented, including provenance, quality checks, and permitted uses.
Model Development	Model design choices, training data, and evaluation criteria are documented and approved by the responsible development team.
Model Evaluation & Testing	Performance thresholds, validation results, and risk assessments are reviewed before the system progresses toward deployment.
Deployment Approval	A designated authority approves the system for operational use, confirming that required governance and safeguards are in place.
Operational Monitoring	Teams continuously track model performance, system behavior, and anomalies during live operation.
Incident Response & Review	When unexpected outcomes occur, responsibility for investigation, corrective action, and communication is clearly assigned.

SUMMARY

CONTINUITY OF OWNERSHIP

In most enterprise AI environments, responsibility is spread across more teams than leaders initially assume. Data, model design, testing, deployment, and operational oversight often sit with different functions, which makes accountability vulnerable at every transition point. What matters most is ensuring ownership remains continuous from development through live operation.

FULCRUM IN PRACTICE

- **Attach accountability to each lifecycle transition.**
- **Preserve review and approval before operational release.**
- **Ensure incident response has named owners from the outset.**

3. Design escalation and intervention mechanisms

Organizations must be able to intervene when an AI system produces outcomes that require correction. As AI systems operate at scale, decisions may occur continuously and often without direct human involvement. In these environments, accountability depends on clearly defined mechanisms that detect abnormal behavior and trigger review or intervention before problems escalate.

Establish escalation protocols that define when systems should be reviewed, who has authority to intervene, and what actions can be taken when risks appear. Triggers may include performance anomalies, customer complaints, regulatory concerns, or unexpected system outputs. Teams responsible for system operations, product oversight, and risk management must know when to step in and how to respond. When escalation pathways are clearly defined and regularly tested, enterprises retain the ability to pause systems, investigate outcomes, and correct course while the system continues to operate in production.

AI Escalation and Intervention Framework

When AI behavior requires review, who steps in, and what happens next

Intervention Triggers

Conditions that signal the need for review or escalation.

- Unexpected or anomalous system outputs
- Model performance degradation or drift
- Customer complaints or disputed decisions
- Regulatory, compliance, or legal concerns
- Operational alerts from monitoring systems

Escalation Authority

Defined roles responsible for reviewing and acting on system behavior.

- System operations or platform team reviews technical alerts
- Product or business owners assess operational impact
- Risk, compliance, or legal teams review regulatory exposure
- Executive or governance oversight for high-impact incidents

Intervention Actions

Operational responses available once an issue is identified.

- Pause or limit system operation
- Revert to a previous model version or rule-based fallback
- Launch structured incident investigation
- Communicate findings and corrective actions to stakeholders

SUMMARY

THE INTERVENTION THRESHOLD

Accountability is tested most visibly when a system has to be challenged, paused, or corrected. In those moments, enterprises need more than technical monitoring. They need clear thresholds for review, defined authority to intervene, and operating paths that allow issues to be contained before they spread across customers, workflows, or regulatory obligations.

FULCRUM IN PRACTICE

Set clear trigger points for intervention and escalation.

Keep authority to review and act visible across the response chain.

Maintain operational paths for pause, fallback, and investigation.

A workable AI system is never the product of a single model, policy, or control. It holds together because the surrounding structure does. Reliability keeps behavior stable. Explainability makes decisions reviewable. Security and compliance protect the system and its boundaries. Performance and cost determine whether it can be sustained. Accountability ensures the organization can stand behind what the system does. When these conditions are designed to function together, AI becomes something the enterprise can operate with clarity rather than tolerate with caution.

That does not mean the hard part is over. It means the organization can finally make a better first move. Once the system requirements are clear, the next question is what the enterprise should commit to first, and what is strong enough to survive everything this section has now made visible.

Up next: The First Move

The final chapter turns from system design to decision discipline. It focuses on how enterprises choose the first AI initiative worth building, and why the strongest starting point is rarely the broadest or most ambitious one.

Sources

- The Conflict Between Explainable and Accountable Decision-Making Algorithms
- OECD AI Principles: Guardrails to Responsible AI Adoption
- AI Accountability: Operationalising the OECD AI Principle 1.5
- Accountability and transparency (NIST AI RMF)
- Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy
- AI Accountability Policy Report | National Telecommunications and Information Administration
- Liability Rules and Standards | National Telecommunications and Information Administration
- The Conflict Between Explainable and Accountable Decision-Making Algorithms
- What Air Canada Lost In 'Remarkable' Lying AI Chatbot Case
- AI Conversations & Chatbot Accountability Under Scrutiny: The Case of the (Too) Helpful Chatbot | BD&P
- Mobley v. Workday: Court Holds AI Service Providers Could Be Directly Liable for Employment Discrimination Under "Agent" Theory | Seyfarth Shaw LLP
- Courts Expand AI Vendor Accountability While Contracts Shift Risk
- Who's Accountable When AI Fails? - Knowledge at Wharton
- The Ownership Gap Creating Hidden Risk in Enterprise AI - Provar
- The flaws of policies requiring human oversight of government algorithms - ScienceDirect
- Human Oversight Doesn't Work: Why Most AI Compliance Systems Fail at the Point of Review
- The impact of AI errors in a human-in-the-loop process - PMC
- ISO 42001 & NIST AI RMF: practical steps for mastering responsible AI governance in 2026

Content Lead:
Nishita Pereira - Senior Group Manager, Marketing & Communication

Internal Expert Contributors:
Bhaskar Gandavabi - Senior Vice President, Technology & Innovation
Chandrashakher Aryasomayasula - Associate Vice President, AI Engineering (AI Innovation Hub)
Shalin Mayank - Senior AI Platform and Solutions Innovator (AI Innovation Hub)
Anubhav Mukherjee - AI Platform and Solutions Innovator (AI Innovation Hub)
Malavika Nair - Associate, Delivery Support (AI Innovation Hub)

Chapter 6

The First Move

How enterprises decide which AI use case to build first and why getting that decision right changes everything downstream.

The five preceding chapters in this manual focused on the disciplines that determine whether an AI system can hold up in real operation: reliability, explainability, security and compliance, performance and cost, and accountability.

This final chapter steps into the one decision that shapes them all: where to begin.

Choosing the first use case happens early in a program, but choosing it well requires a clear view of what production-grade AI truly demands. Once those demands are understood, selection becomes less speculative and more disciplined.

When a use case is chosen with usable data, real ownership, and an accuracy bar grounded in what the system can realistically deliver, everything that follows has a foundation. When that decision is rushed, later engineering and governance work end up compensating for a weak starting point.

What gets bypassed here usually returns later as something harder and more expensive to correct.

The Cost of Doing Too Many Things

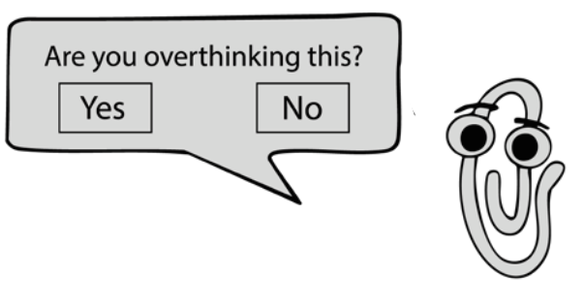
In 2024, enterprises collectively spent over $120 billion on AI initiatives and around 80% of those projects never reached production. The failures were rarely technical; platforms were available; models performed adequately in controlled settings. What failed was selection. Organizations committed resources to use cases before honestly assessing whether those use cases were buildable with what they had at the time.

The costs show up elsewhere too. A study of software development teams found that despite the speed AI tools promise, developer productivity dropped by 19% on average, largely because engineers were spending significant time reviewing and correcting low-quality output rather than building. And when it comes to agents specifically, Carnegie Mellon research found that AI agents fail to complete end-to-end office tasks around 70% of the time because they lack the contextual understanding to navigate how real organizations operate.

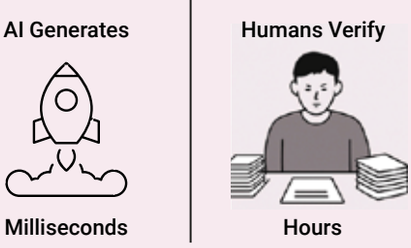
Why Planning Sessions Overproduce

AI planning sessions tend to produce more than they resolve. Use cases accumulate across a whiteboard, everyone leaves with a sense of momentum, and the output is a list that no one is quite sure how to act on. Some items depend on data that is not ready. Others address problems that no single person owns. A few are genuinely worth building, but they end up buried under enough noise that the list itself becomes the obstacle.

The reason this happens is that generating options is easier than making a call. A planning session that ends with forty use cases has produced activity, but it has postponed the harder decision: which one of these is truly ready to be built now, and which ones need to wait.



These aren't arguments against AI but arguments for choosing the right use cases carefully, rather than pursuing volume and hoping the good ones surface.



The speed of generation and the speed of verification are not the same. Every idea added to the list carries a human cost to assess, validate, and build.

- **$120B spent on AI in 2024**. 80% of projects did not reach production.
- **19% reduction in developer productivity** attributed to reviewing AI-generated output.
- **70% failure rate for AI agents** completing real end-to-end office tasks (Carnegie Mellon).

Why Some Use Cases Shouldn't Be on the List

A planning process that only 'adds' is incomplete. The more useful discipline is **removal: taking what's on the list and asking which items shouldn't be there yet.** This requires an understanding of why certain use cases fail, and it comes down to something more specific than readiness.

AI systems are fundamentally probabilistic. They produce outputs based on patterns and likelihoods, which means they will sometimes be wrong. For many tasks, that's acceptable; a content draft that gets edited or a ranked list that a human reviews before acting on. But a significant portion of the use cases that end up on planning roadmaps sit in areas where being wrong carries direct consequences: compliance checks, financial calculations, legal decisions, patient-facing outputs. These tasks require deterministic accuracy. Applying a system that produces probabilistic outputs to a workflow that demands certainty isn't a calibration problem. It's a category mismatch, and no amount of refinement fully resolves it.

Understanding this distinction removes a large number of items from any list, not because AI can't eventually be applied there, but because the use case requires a level of verified, consistent accuracy that the system needs to demonstrate before it earns a place in that workflow.

Three Questions Worth Asking Honestly

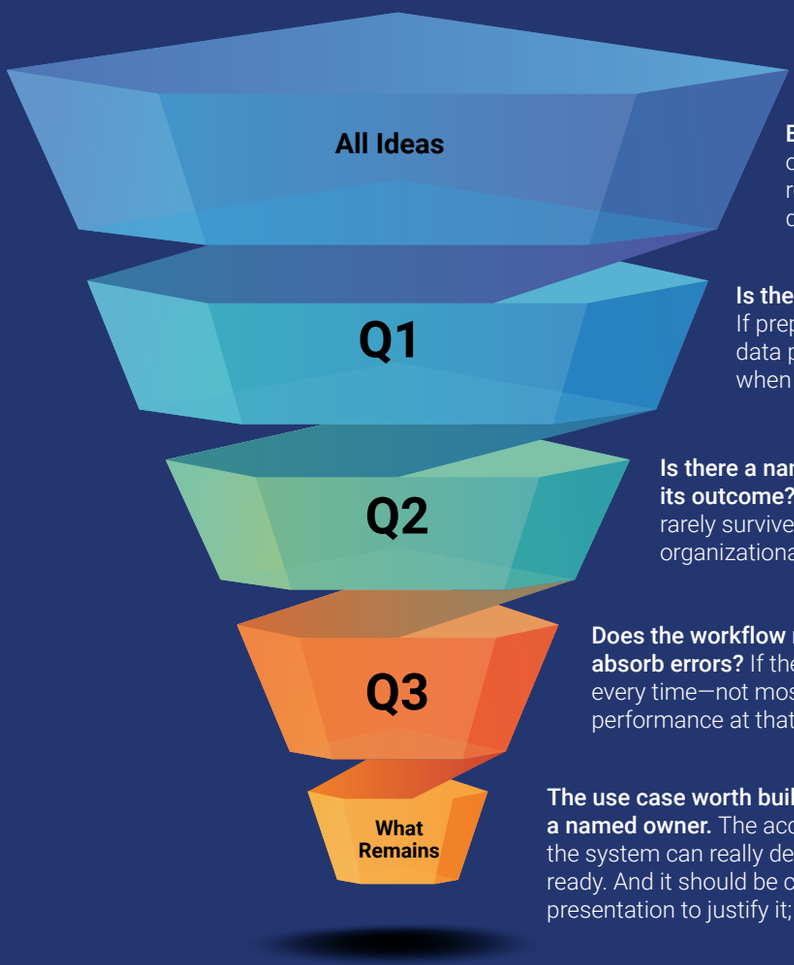
Removing items from a roadmap requires a basis for removal; otherwise, it becomes a political exercise where only the ideas with the loudest advocates are the ones to survive. Three questions, applied consistently across every use case on the list, tend to cut through that.

The first concerns data and its availability in a usable state.
Is it clean, accessible, and reliable enough that the team could begin building with it now?
When the honest answer involves a preparation phase measured in months, the use case belongs on a different list. It's a data project before it becomes an AI project and treating them as the same thing is a sure way of stalling a program.

The second concerns about ownership.
Is there a specific person, with true authority and genuine accountability, who owns the underlying problem?
Broad organizational benefit doesn't satisfy this. When a use case serves everyone in general, it tends to belong to no one in particular. And without a person willing to push implementation through the friction of a real deployment, most initiatives lose momentum before they reach production.

The third is the accuracy question.
Can this workflow absorb the error rate the system will realistically produce, or does it require a level of verified performance the system hasn't demonstrated yet?
If the latter, defer it until that bar has been met.

The Selection Discipline



Everything on the list: workshop outputs, vendor conversations, pilot requests, things that surfaced in last quarter's strategy review.

Is the data available in a usable state today? If preparation work is required first, this is a data project at this stage. Log it and return when the data is ready.

Is there a named person accountable for this problem and its outcome? A use case with no single accountable owner rarely survives the pressures of actual deployment. General organizational benefit is not a substitute.

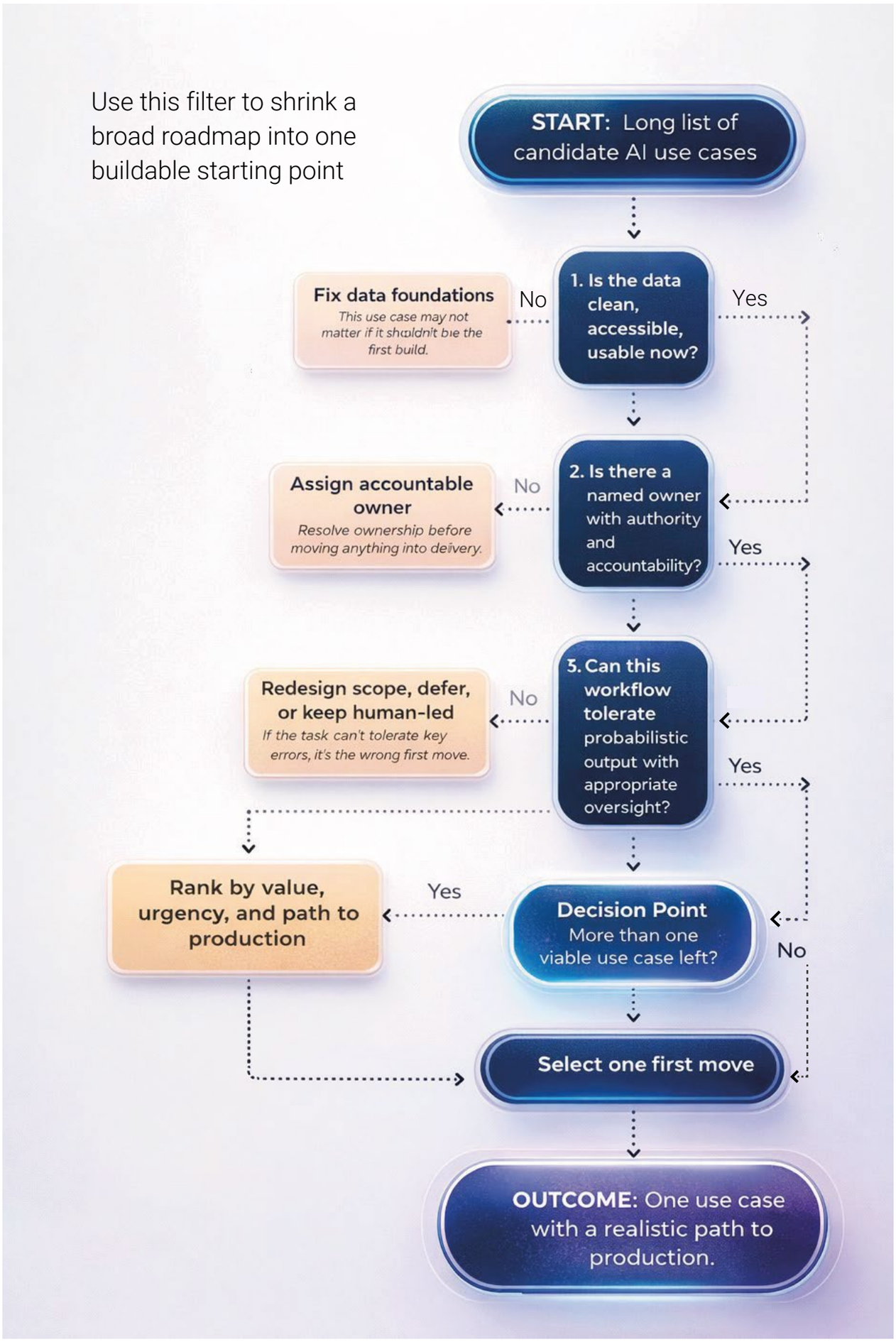
Does the workflow require deterministic accuracy, or can it absorb errors? If the task requires outputs that are correct every time—not most of the time—it needs demonstrated performance at that level before going into production.

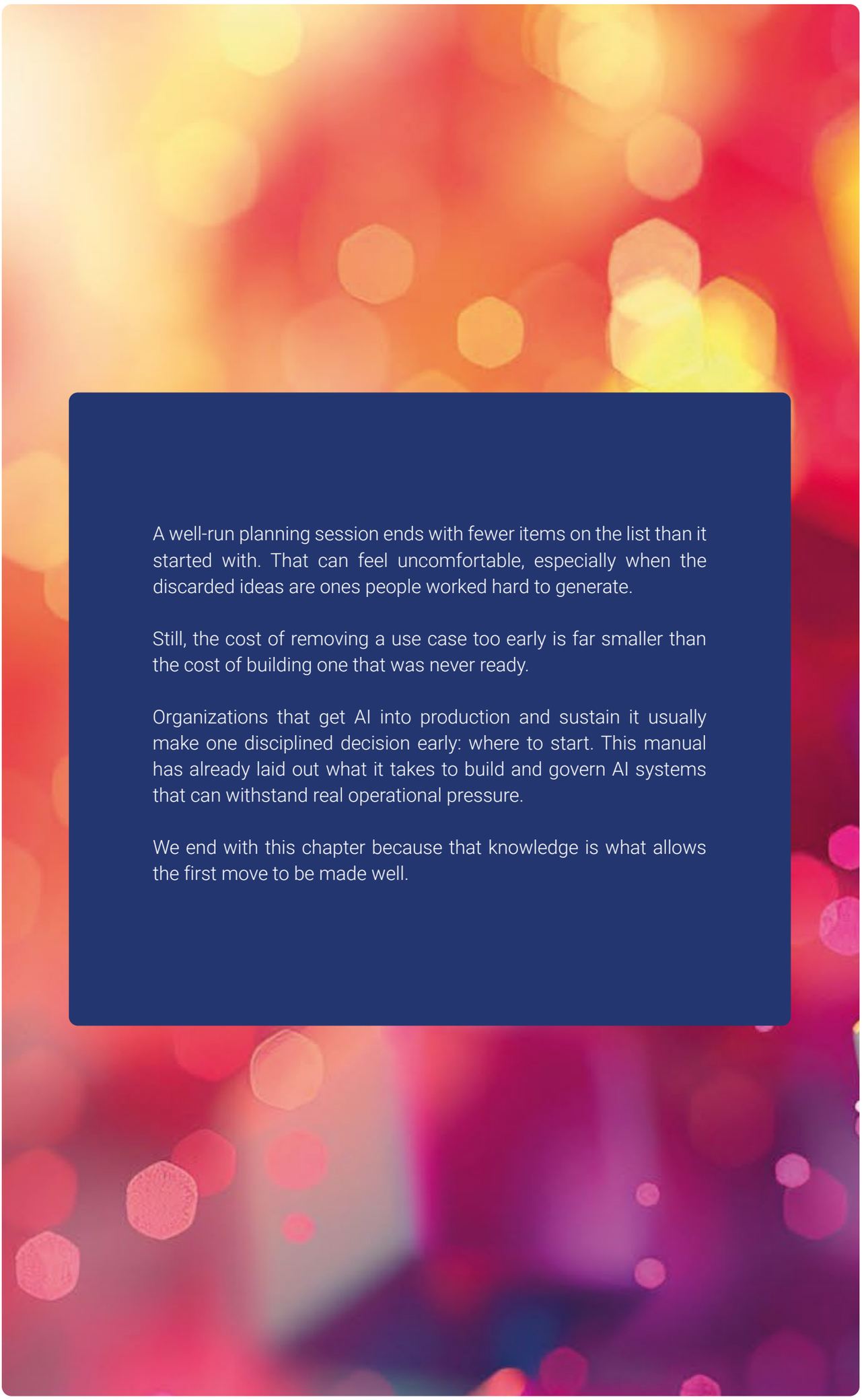
The use case worth building first. The data is available. There is a named owner. The accuracy requirements are matched to what the system can really deliver. When a use case clears all three, it's ready. And it should be clear enough that it doesn't need a presentation to justify it; the case makes itself.

The First-Move Filter

A decision tool for selecting the first AI use case to build

Use this filter to shrink a broad roadmap into one buildable starting point



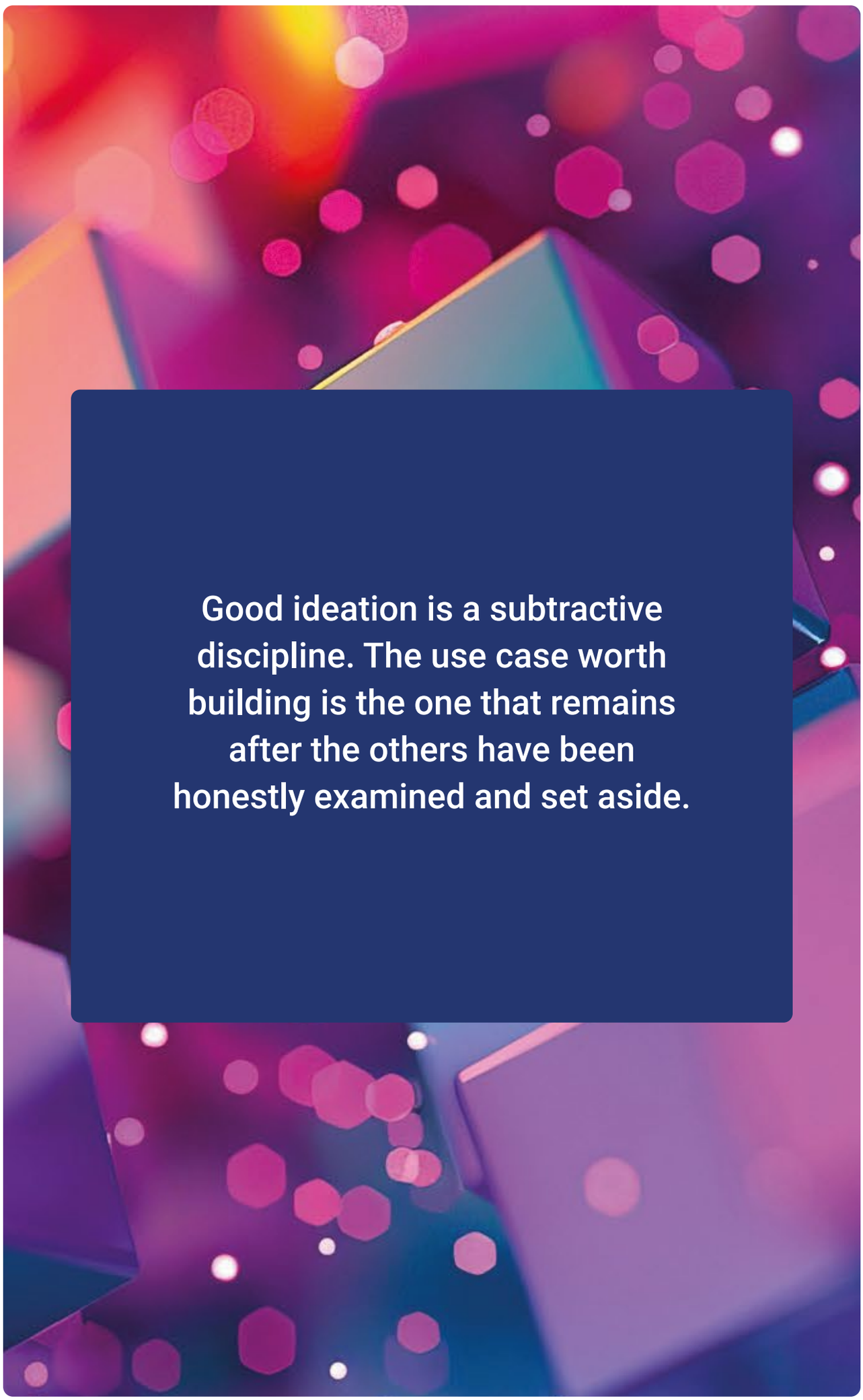


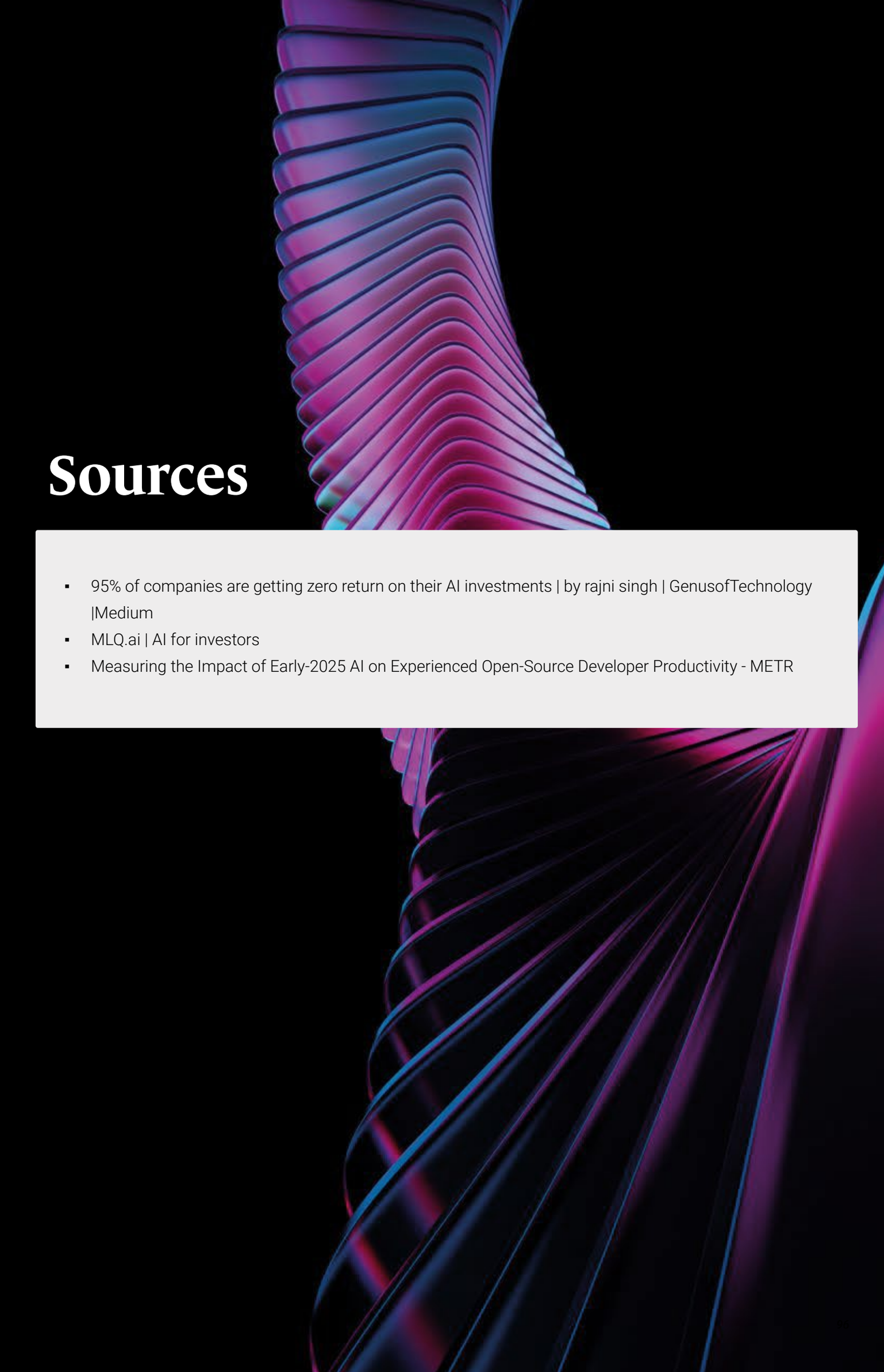
A well-run planning session ends with fewer items on the list than it started with. That can feel uncomfortable, especially when the discarded ideas are ones people worked hard to generate.

Still, the cost of removing a use case too early is far smaller than the cost of building one that was never ready.

Organizations that get AI into production and sustain it usually make one disciplined decision early: where to start. This manual has already laid out what it takes to build and govern AI systems that can withstand real operational pressure.

We end with this chapter because that knowledge is what allows the first move to be made well.


Good ideation is a subtractive discipline. The use case worth building is the one that remains after the others have been honestly examined and set aside.



Sources

- 95% of companies are getting zero return on their AI investments | by rajni singh | GenusofTechnology |Medium
- MLQ.ai | AI for investors
- Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity - METR

One suite.
Infinite ways to make enterprise AI usable

The FD RYZE® suite brings together the layers enterprises need to move from isolated experiments to usable systems. FD RYZE® sets the umbrella vision and operating foundation. FD RYZE® Infinity extends that into autonomous orchestration at enterprise scale. FD RYZE® Nexus applies it in a focused assistant layer that helps teams find, use, and act on the knowledge they already own.

FD RYZE

The broader enterprise AI platform built to connect human judgment and machine capability with greater control, adaptability, and trust. FD RYZE® is designed for organizations moving toward production-grade adoption across complex systems and operations.

RYZE INFINITY

FD RYZE® Infinity is the foundational platform layer within the suite. It is built to deploy and run autonomous, industry-ready agents across complex business operations, from claims and underwriting to payments, reporting, and forecasting, while maintaining governance, compliance, and explainability as the system scales.

FD RYZE Nexus

A purpose-built and deployable AI assistant built on that backbone. FD RYZE® Nexus helps teams surface policies, product knowledge, process guidance, and institutional context from their own documents and systems, with answers returned quickly and tied back to source.

See what the FD RYZE® suite could unlock for your teams.

Book a call with our Fulcrum Digital team.

About Fulcrum Digital

Fulcrum Digital is a global digital solutions company delivering AI, platform engineering, and large-scale system integration. We specialize in building and operating enterprise agentic AI platforms for business, supporting organizations across North America, Europe, Latin America, India, and Australia.

Connect with us
www.fulcrumdigital.com

Headquarters
Fulcrum Digital
New York 259 W 30th St,
Suite 1300 New York, NY 10001

Identifying the right first use case is where the planning ends and the work begins. What comes next—scoping the proof of concept, pressure-testing the assumptions, making governance decisions—is equally exciting and just as difficult. It's where most programs either build something durable or discover they need to go back to the drawing board.

Fulcrum Digital's role in this next stage is one we take seriously: building the proof of concept, designing the architecture, and helping organizations make calls that determine if the business can sustain and scale. We've done this across industries and at different stages of AI maturity, and what we encounter in that range of work is what keeps the thinking interesting and current.

Not every conversation that starts here begins with a fully formed use case. Some organizations come with a problem they haven't yet translated into an AI initiative. Others come mid-build, having hit a wall they didn't anticipate. Some come with a use case that has already been greenlit, but where governance questions were never fully resolved. Some come simply to think out loud with people who have been through it.
What Fulcrum learns from these conversations is part of how this body of work continues to develop. The manual you have just read is a reflection of that accumulated experience. The next version will be shaped by what comes next, including conversations that begin from this page.

If you are at any stage of an AI initiative, whether you're validating an idea or untangling a deployment that didn't go as planned, we want to hear about it.

YOUR NEXT MOVE STARTS HERE.