AI agents are everywhere *Governance* is not.

Eight in ten enterprises have had an AI agent execute a consequential action in production and faced real costs correcting it. We set out to understand why by examining how organizations detect failures, establish accountability, and make governance decisions about AI agents they don't fully understand.

400+

IT and engineering leaders surveyed, all actively running AI agents in production

100%

Survey completion rate, reflecting how acutely this problem is felt by the people closest to it

73%

of respondents are final decision-makers on technology selection in their organizations

6

governance dimensions examined: failure detection, autonomous action accountability, executive oversight, pre-deployment testing, standardization, and cost optimization

Deploying AI agents is easy. Governing them in production at scale is the *biggest challenge*.

Organizations have governance frameworks and policies defined, yet continue to experience costly reversals, slow failure attribution, and deployments that proceed without a complete understanding of agent behavior. Governing agents in production is where those frameworks are falling short. This research measures exactly how wide that gap is.

72% of enterprise leaders believe the agents running in their organizations today introduce unmanaged financial or compliance risk. A self-assessment that no governance framework on paper has resolved.

82%

Unmanaged risk

Of enterprises report agents have autonomously executed consequential actions in production without sufficient human oversight.

70%

Observability

Could identify that a failure occurred but could not determine which agent in a multi-agent environment caused it.

93%

Reversal cost

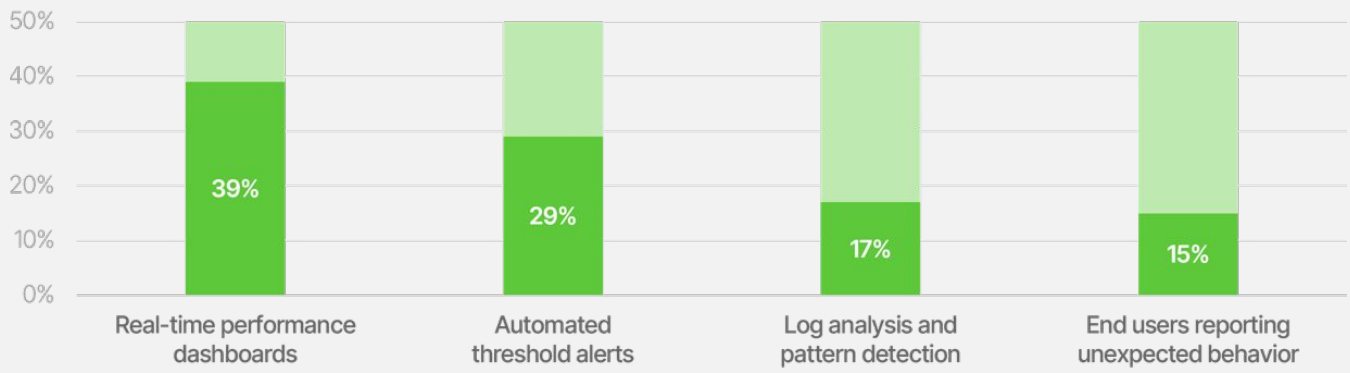
Of organizations that reversed an agent action described it as costly and disruptive.

1. Teams can see that something is broken. Finding out *what* is broken takes far too long.

Enterprises have invested in detection. What most have not built is traceability. In multi-agent environments, the gap between flagging a failure and attributing it to a specific agent has a direct business cost: every hour that attribution takes is an hour the problem continues to run impacting your business outcomes.

50% Detect agent malfunctions within one to four hours of occurrence	**33%** Need four to eight hours before a failure is even detected	**70%** Could not identify the responsible agent when a failure occurred in a multi-agent environment

EXHIBIT 1 • HOW AGENT FAILURES ARE DETECTED



15% Rely on end users reporting problems as their primary detection mechanism. In practice, that means the customer finds the failure before the organization does.

{ TAKEAWAY }

Most organizations know when something has gone wrong, but seven in ten cannot identify which agent in a multi-agent environment caused it. Observability and attribution are different capabilities, and the data shows enterprises have invested in one without building the other.

2. Agent failures do not stay in the IT layer. They reach *revenue, customers, and contracts.*

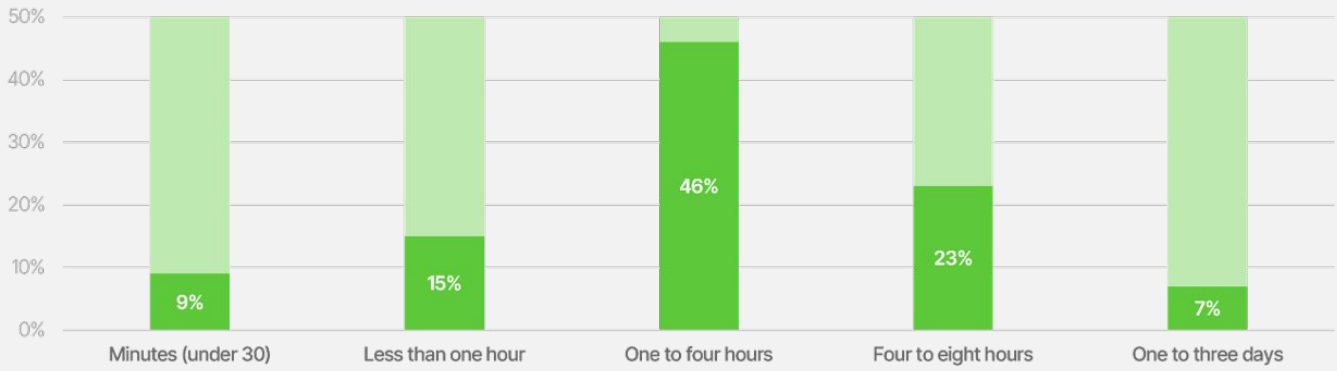
When agents fail in production, the consequences are not contained to engineering. Most of that exposure is still being classified as an IT incident, which means the true cost never surfaces where decisions get made.

42%
experienced measurable revenue loss as a direct result of an AI agent failure in production

31%
reported SLA violations attributed directly to agent malfunctions, with direct contract implications

28%
experienced customer churn following an agent failure, a cost most AI risk frameworks do not capture

EXHIBIT 2 · TIME TO FULL RECOVERY FROM AN AGENT FAILURE



46%	recover within one to four hours, the single most common recovery window. For workflows processing high transaction volumes, every hour of downtime compounds.
30%	take four to eight hours or longer, with some outages stretching across one to three full days.

{ TAKEAWAY }

Agent failures increasingly affect revenue, contractual obligations, and customer relationships. Yet many organizations continue to manage them primarily within IT, limiting visibility into their broader business impact.

//// BUILT FOR THIS

Seeing this in your own production environment?

Kore.ai *{Artemis}* traces every agent action, enforces policy at runtime, and contains failures before they reach revenue. See how the harness governs the full agent lifecycle.

Talk to us about agent governance

Agentic actions frequently required reversals

79%

of consequential autonomous actions required manual reversal. Of those, 93% were described as costly and disruptive, meaning the act of correction compounded the original failure.

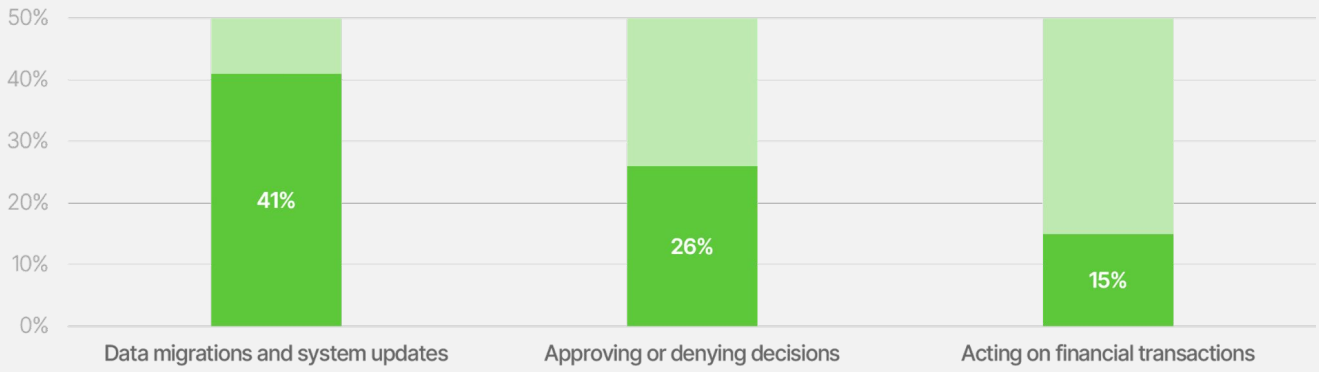
3. Reversal rates remained high despite safeguards being in place.

Organizations have invested heavily in governance safeguards, yet the data suggests those safeguards do not always translate into effective outcomes. As agents take on more consequential responsibilities, the ability of governance mechanisms to prevent, contain, and recover from failures becomes increasingly important.

82% **of enterprises report agents have autonomously executed consequential actions in production.**

Financial transactions, workflow approvals, and data migrations are increasingly being executed by agents, often with limited human oversight.

THE AUTHORITY ENTERPRISES HAVE HANDED TO AGENTS



These are consequential tasks. Enterprises have handed agents authority over data, decisions, and money, increasing the importance of governance mechanisms that can operate reliably in production.

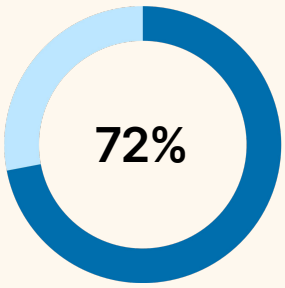
91% reported safeguards were in place before a consequential autonomous action occurred.	**62%** have delayed agent deployments over governance and observability concerns, accepting slower time-to-value in exchange for lower production risk.
79% of consequential autonomous actions still required manual reversal.	**53%** admit they have deployed an agent to production without fully understanding or trusting how it would behave.

{ TAKEAWAY }

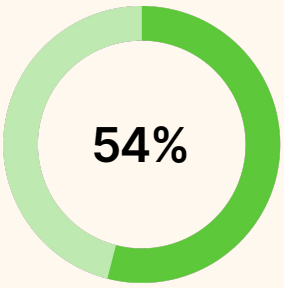
As agents take on more consequential responsibilities, governance effectiveness depends less on the presence of controls and more on their ability to prevent costly outcomes.

4. Governance frameworks are defined, yet leadership confidence remains low.

Organizations report widespread adoption of governance mechanisms, including kill switches, containment procedures, and executive review processes. Yet confidence remains low, with 72% of respondents believing autonomous agents introduce unmanaged financial or compliance risk.



Believe autonomous agents introduce unmanaged financial or compliance risk, suggesting that governance measures alone are not providing sufficient assurance.



Classify agent failures as IT incidents only, reducing the likelihood that agent-related risks receive enterprise-wide oversight.

{ TAKEAWAY }

As agent ecosystems become more interconnected, governance effectiveness increasingly depends on containment. The ability to pause an agent is valuable, but limiting the spread of failures across connected systems remains the greater challenge.

METHODOLOGY

About this research

This report is based on a survey of 408 IT and engineering leaders in enterprise organizations. Every respondent is actively running AI agents in production.

Questions covered six governance dimensions: failure detection, autonomous action accountability, governance and containment infrastructure, executive oversight, standardization, and model cost optimization.

IT AND TECHNOLOGY ROLES

DIRECTORS AND C-SUITE

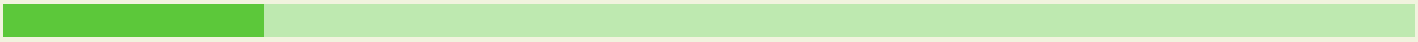
100% COMPLETION RATE

COMPANY SIZE: 1,001 TO 5,000 EMPLOYEES

US-BASED RESPONDENTS (CA, NY, TX, FL)

19%

C-level executives in the sample



73%

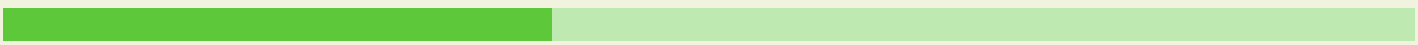
Final decision-makers on technology selection

45%

Directors, the largest respondent segment

37%

From science and technology industries



5. How Artemis moves governance from oversight to execution

The survey findings point to a common challenge: organizations that have invested in AI agents are now finding that deployment was the easy part. Accountability gaps, safeguards that cannot be verified, and costs that scale with usage instead of value are not engineering problems. They are governance problems. And governance cannot be retrofitted.

Unlike traditional AI platforms, Artemis treats governance as part of the agent lifecycle, not a layer added on top of it. Every capability described below is built into how agents are defined, validated, deployed, and managed at scale.

01 Embed governance within architecture

Artemis enforces governance at the structural level, not the prompt level. Policies, constraints, and escalation rules are compiled into the agent definition before it ever reaches production. There is no configuration to drift and no prompt to override.

02 Enforce policy at runtime

A deterministic runtime engine enforces boundaries on every agent action in real time. If an agent attempts something outside its defined scope, the system intercepts it before it executes. Safeguards are not monitored after the fact. They hold in the moment.

03 Trace every decision

Every agent action, handoff, and constraint check is logged end to end across 100% of interactions. When something goes wrong, the answer is in the trace. Not reconstructed from assumptions, not inferred from outputs. The decision chain is on record.

04 Govern the system, not the model

Most governance approaches treat the LLM as the unit of control. Artemis governs the system: the agent, the tools it can call, the data it can access, and the actions it is permitted to take. The model is one component. The system is what enterprises are accountable for.

05 Manage the lifecycle as one system

{Artemis} covers the full agent lifecycle: design, testing, deployment, monitoring, and updates in a single governed environment. There is no handoff between tools where policies get lost or documentation falls out of sync with what is actually running.

06 Scale without multiplying oversight

Governance in {Artemis} is inherited, not applied manually. When an organization scales from 10 agents to 1,000, the same policies apply without requiring proportionally more review or configuration. Governance scales with the deployment, not against it.

"Enterprise AI has shifted from showing that AI works to proving it can be trusted. Governance has to be built into the agent itself, not added once it is running."

Raj Koneru
CEO & Founder, Kore.ai

Meet { Artemis }

*The **AI programmable platform** for the agentic enterprise*

Start Building