harness

The State of Agent DLC 2026

Agents unsupervised: confidence without control

Executive summary

Organizations are pushing AI agents into production faster than they can govern them.

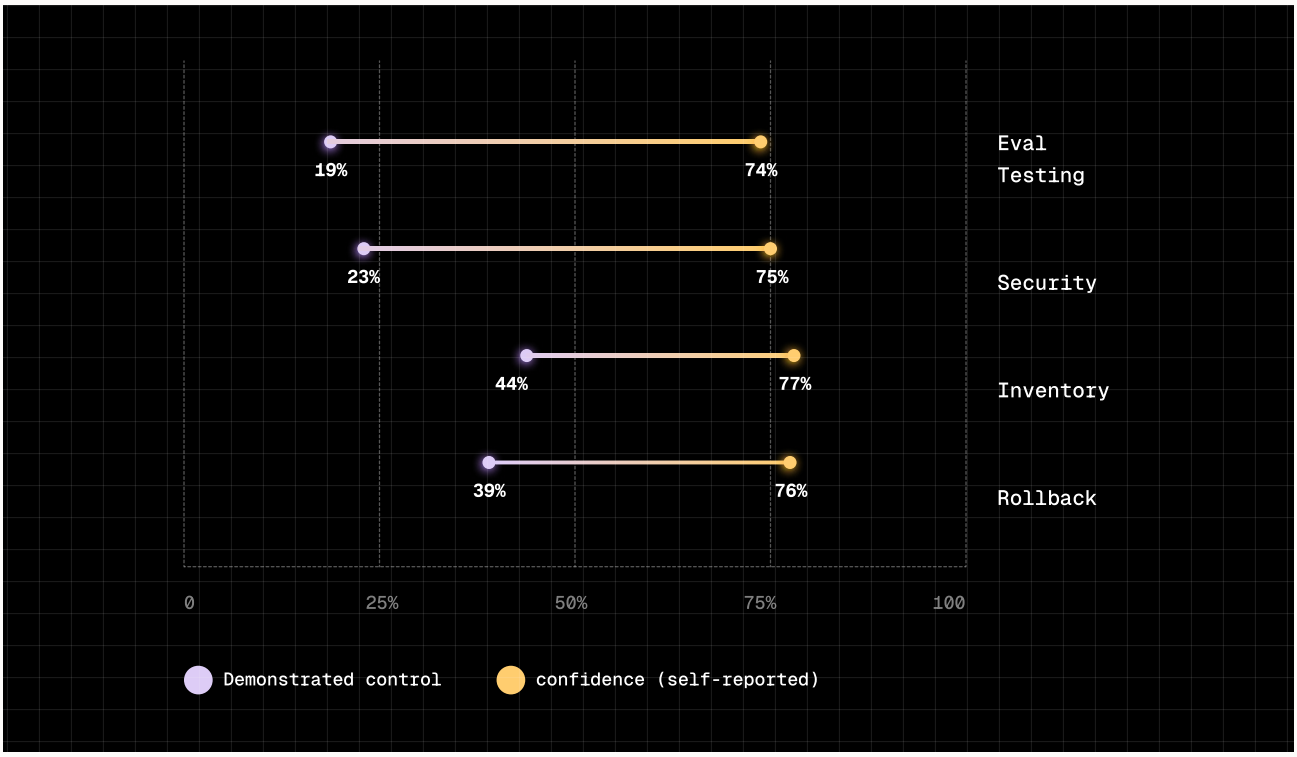
An AI agent isn't like deterministic software: the same agent can behave differently from one run to the next. That means every stage of the software development lifecycle that organizations trust — test, deploy, secure, optimize, release, govern — has to be rebuilt for the agent development lifecycle (Agent DLC). From our survey analysis, we found out that most organizations haven't rebuilt these stages yet. They've just kept deploying more agents.

We wanted to understand the gap between an organization's confidence and the actual controls in place, stage by stage. 700 engineers and engineering leaders from across the US, UK, France, Germany, and India were surveyed to provide information about their organization's Agent DLC, how they build, test, secure, deploy, and govern the AI agents deployed in their environments.

75% of the surveyed organizations have deployed agents in production, while the rest have them deployed in pre-production. We asked these organizations how confident they are across five domains: testing, security, inventory, cost, and rollback, and every answer landed in the mid-70s. That confidence, however, doesn't mean the controls exist to back it up.

The lack of controls may be evidenced by the fact that 87% of organizations experienced an agent-related security event this year, 60% overspent their agent budget, and incidents per 100 agent-related changes increased for most adopters since they started deploying agents.

The good news: organizations can accurately identify their own weak points and 76% already have a budget in place or planned within six months to address them. This report analyzes where the gap comes from and what steps are needed to instill true confidence.



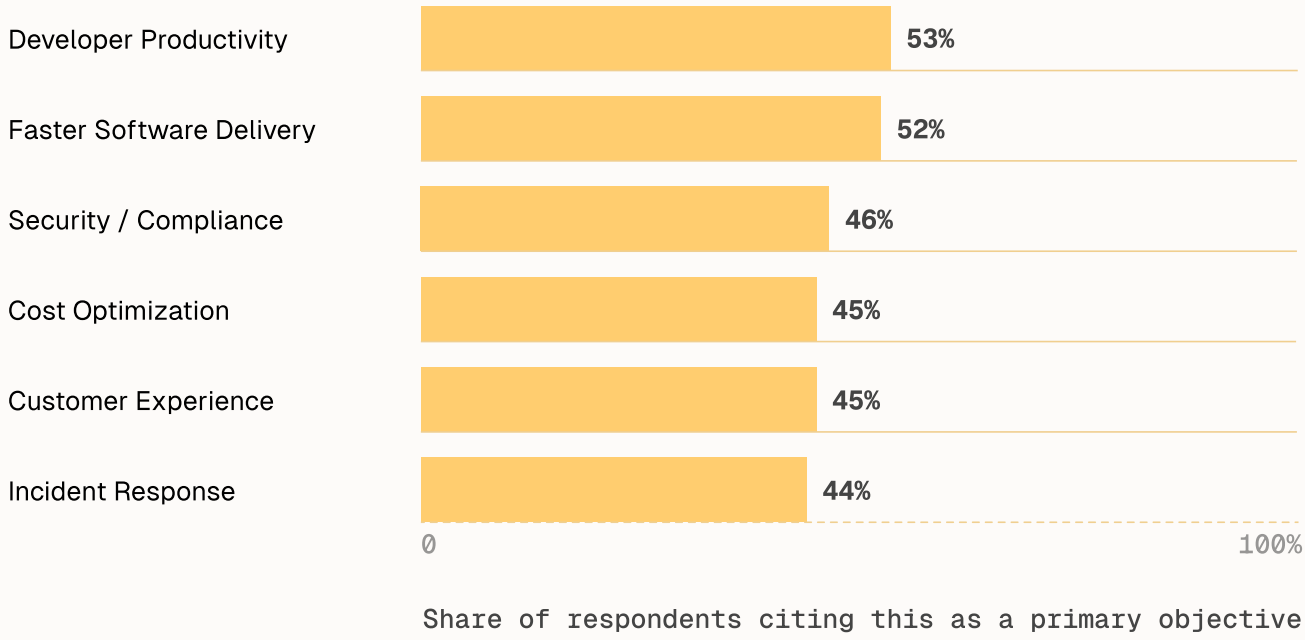
PART 01

Adoption is real, and it's not cautious

Organizations that have started deploying AI agents aren't stopping at pilots. Most (75%) are already running agents in production (33% broadly, across both internal and customer-facing systems). The rest are still pre-production: 14% in a live pilot, 11% at proof of concept.

The primary business objectives for building AI agent initiatives

Developer productivity and faster delivery lead a tight field.



n=700 organizations already running AI agents. Respondents could select multiple objectives.

Organizations use agents to solve the speed problem, not the governance one. No single reason dominates, but the ranking is consistent: teams deployed agents to move faster, and governance was a secondary motive, not the driver.

On average, organizations run just 53% of agent-related changes through a standard build/deploy pipeline before reaching users or production, but that average hides a real spread. More than a third (37%) run less than half their agent-related changes through a standard pipeline; only 7% run nearly all their agent-related changes through a standard pipeline.

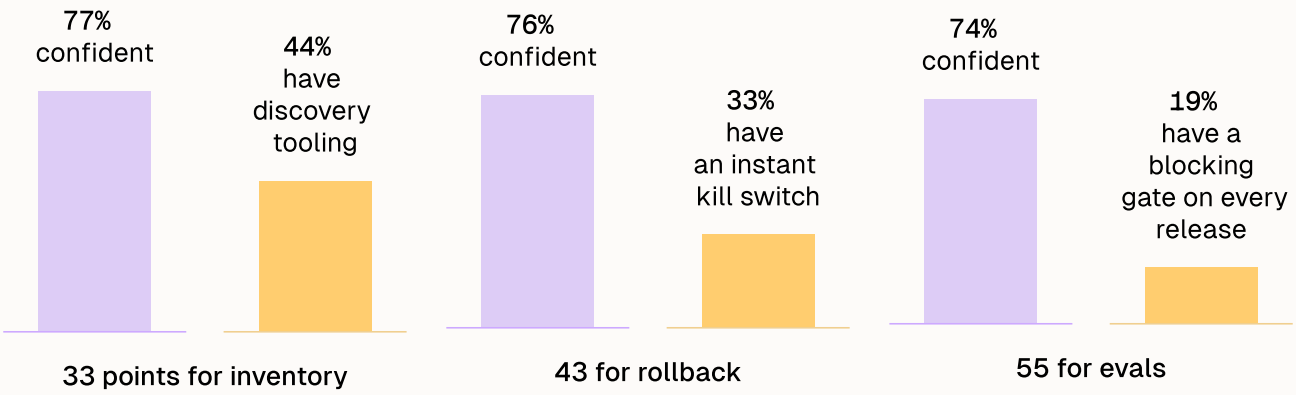
Part 2 is where the gaps start to take shape.

PART 02

The confidence paradox

Confidence lands in the mid-70s in all five domains (74% to 77%) as seen in Part 1. But confidence and control don't relate to each other the same way in every domain. In three (evals, inventory, rollback), there is a straightforward gap.

The size of that gap depends on the domain:



In the other two (security and cost), there is no gap to point at, for different reasons. We found no statistically significant relationship between security confidence and whether an organization experienced a security incident. Organizations are confident they can see spending, but do not have the same confidence that they can control it.

Testing: confidence in evals that aren't universally enforced

74% of organizations are confident that their testing or evaluation would catch a production-impacting failure. Only 19% actually have a gate that automatically blocks every bad release, no exceptions. Among eval-confidence organizations specifically, roughly 80% don't have that gate. In practice, that means a change to an agent that could hurt the production experience can still ship with nothing automated standing in its way; a meaningfully weaker control than the confidence number implies.

The average organization use 4 of the 11 eval methods at once:

Unit / integration tests	Task-success evals	Grounding / hallucination evals
Tool-use correctness evals	Safety / toxicity evals	Data leakage / privacy evals
Permission abuse evals	Regression vs. golden datasets	Latency / performance tests
Token / API cost tests	End-to-end tests	

No single method clears 46% adoption (safety/toxicity checks lead at 45%, with task-success and end-to-end testing not far behind at about 41% each). What this means is that organizations aren't skipping testing; instead they're running a patchwork: four methods at once, on average, with no single one used by a majority.

Security: confidence, but with incidents

87% of all respondents — confident or not — had at least one agent-related security event in the past 12 months. 75% say their agents are secure end-to-end. When we looked specifically at this confidence group, 88% had a security incident this year as well. That's essentially the same rate as the full sample. In other domains, a gap at least tells you something: the control is missing, and adding it should help. For security, the confidence group reported incidents at nearly the same rate as everyone else (88% vs. 87%), so confidence isn't a rough proxy for whether an organization is actually protected. Thus, being confident about agent security didn’t correlate with actually avoiding an incident.

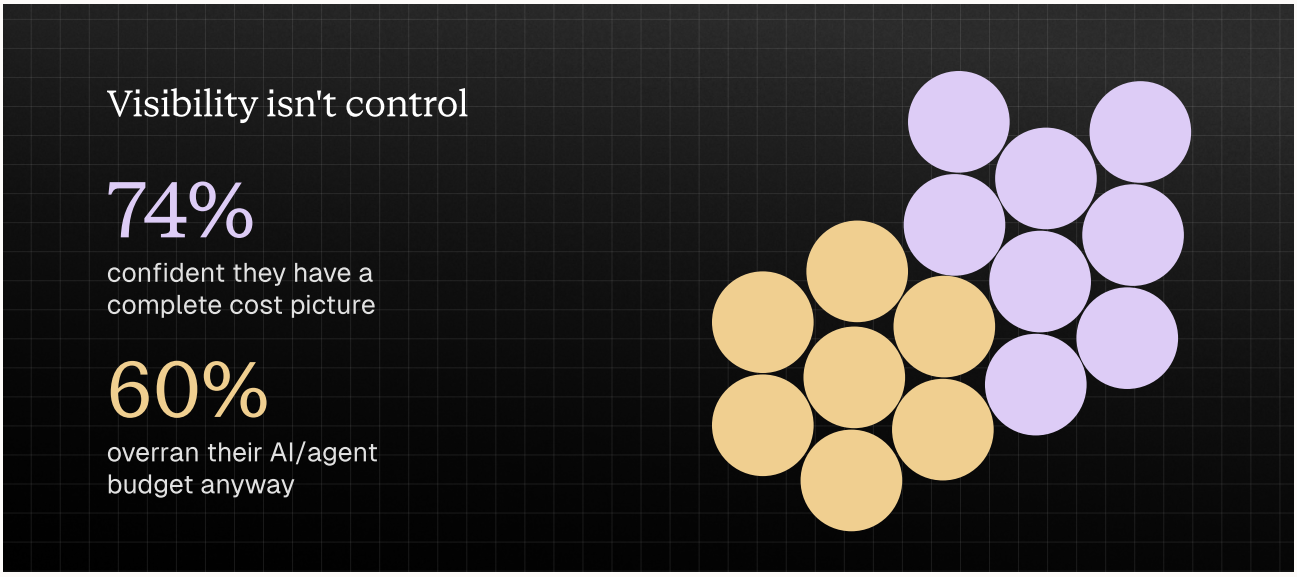
Only 23% have a dedicated AI-agent security layer live in production (29% are currently piloting). Just over half the organizations have moved to purpose-built tooling for Agent DLC in some form. The rest are split between relying on existing general-purpose security tooling, running no dedicated tooling at all, or having limited insight into what tooling is being used, a mix that still leaves real exposure to MCP-specific attack surfaces that general-purpose tools aren’t built to see.

Inventory: confidence in "No Shadow AI," mostly without checking

77% are confident that they have a complete inventory of every agent, MCP server, and LLM running in their production environments and that they have no shadow AI. Only 44% run active discovery or inventory tooling to verify that. 41% of the entire sample — the confident respondents who aren't running any discovery or inventory tooling — are confidently declaring "no shadow AI" with nothing in place to verify it.

Cost: visibility isn't the same claim as control

74% say they have a complete picture of true cost per agent. Separately, 60% overran their agent/AI budget by some amount in the last two quarters. To be clear, these aren't logical contradictions. You can see your spending clearly and still miss a budget target, just as a household can track every dollar and still overspend. What it does mean is that "we have visibility" and "we're in control of the number" are two different claims, and 74% confidence answers the visibility question and 60% answers the (lack of) control question.



Rollback: reversible, but not fast enough

76% believe they could disable or roll back a misbehaving agent in under 15 minutes. 33% have an instant kill switch in production specifically, and 39% have automated rollback specifically. Those aren't the only rollback-related controls on the list: feature flags, environment promotion, manual approval gates, and several others also provide a way to reverse a change, and the vast majority of organizations have at least one of these in place.

The actual gap isn't "two-thirds have no way to roll back anything," because most teams have some way to reverse a bad change. The gap is speed. Two-thirds are missing the fast, purpose-built mechanism that gets you there in minutes. What they have instead is often a manual process, like redeploying code. This manual process might work to reverse a bad change, but it can't meet a 15-minute SLA, the precise recovery benchmark that 76% of organizations expressed high confidence in hitting.

Why control doesn't match confidence

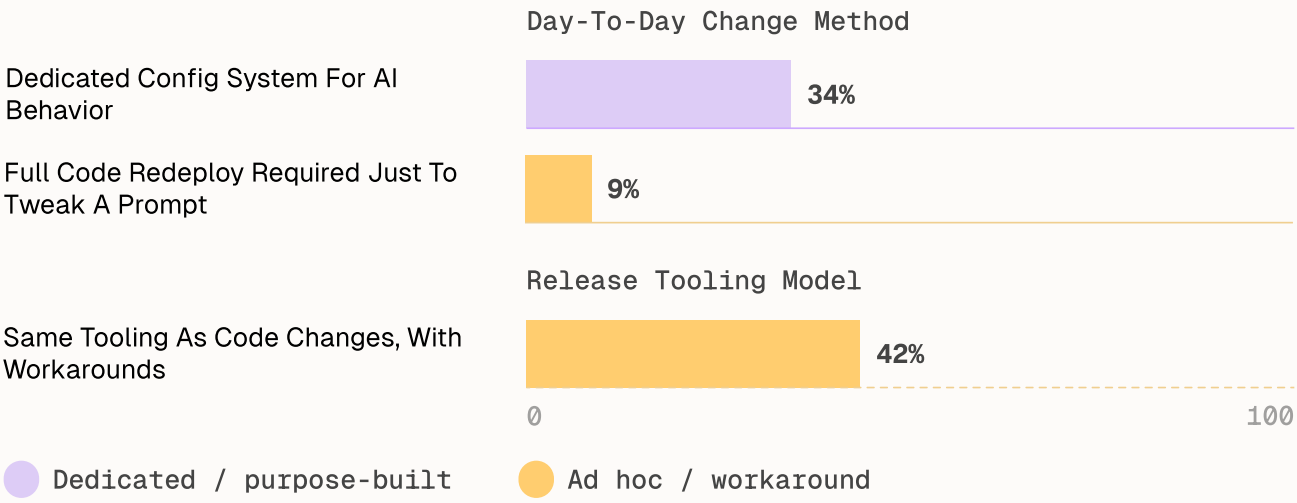
Three domains, the same shape every time: mid-70s confidence and the actual adoption rate of the control sitting well behind it. The likeliest explanation is that confidence tracks whether a control exists (a dashboard, a partial gate, a tool in pilot, or a rollback script someone wrote once), rather than whether that control is sufficient for the failure mode in question. From inside the organization, having a dashboard or a script someone wrote feels like having the necessary controls. When evaluated objectively against specific failure modes, these controls often fall short.

PART 03

Yesterday's tools for today's agents

Most tooling running today was built to manage predictable software releases, not the nondeterministic behavior of an AI agent, and the retrofit is incomplete in five consistent places: how organizations manage change to an agent's behavior, how they roll the agent out, how they track agent ownership, how completely their tooling covers each governance capability, and whether there's a real gate before an agent reaches production.

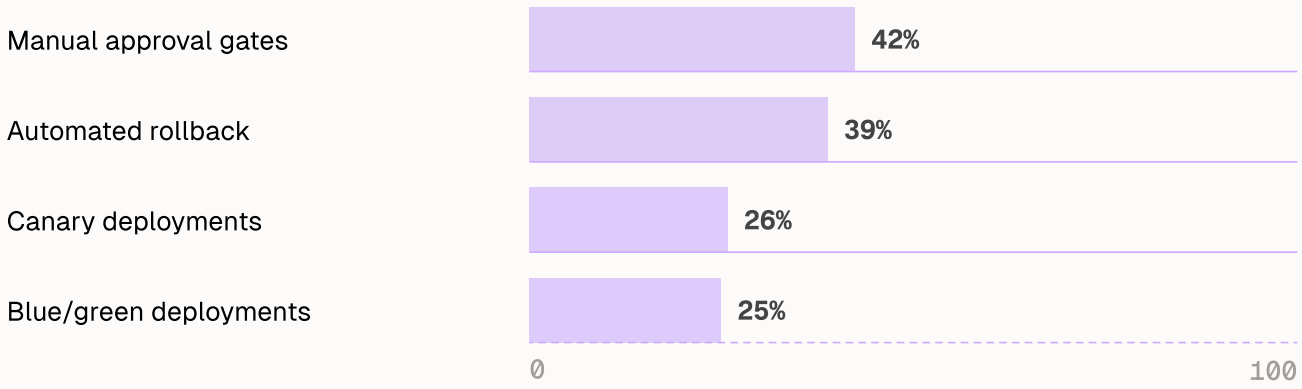
Change management looks different depending on which question you ask



Change management looks different depending on which question you ask. On how teams actually change agent behavior day-to-day: just 34% have a dedicated configuration system for it, and 9% still require a full code redeploy simply to update an agent prompt. Regarding which tooling supports releasing those changes: the largest single group (42%) manages agent changes the same way they'd manage a deterministic software change. For example, a prompt edit gets a pull request, a review, and a merge through the same CI/CD pipeline, even though nothing in that pipeline actually understands what the prompt is. Only 34% have a dedicated configuration system for AI behavior, suggesting less standardization in tooling.

Rollout controls: blunt instruments lead, progressive ones trail

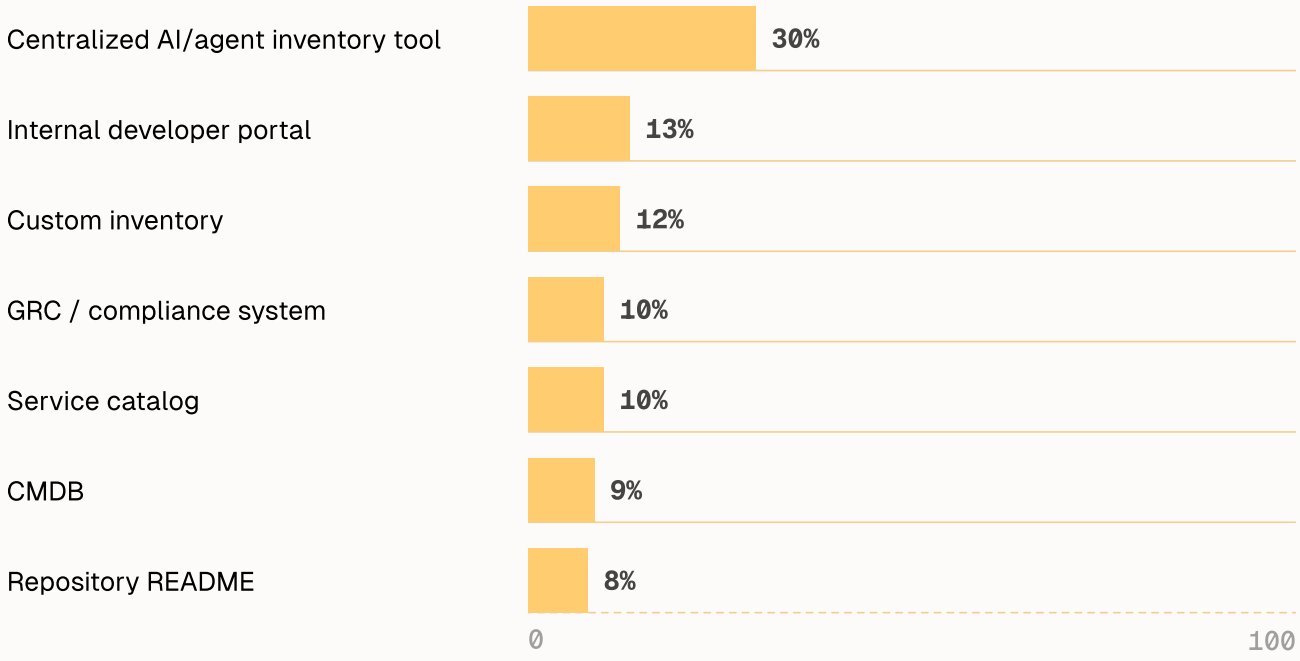
The industry spent years learning to ship progressively and limit blast radius; agent rollouts haven't caught up yet.



Rollout controls skew toward broad, manual instruments more than granular, progressive ones: manual approval gates (42%) and automated rollback (39%) — both all-or-nothing controls — lead adoption, while progressive techniques like canary (26%) and blue/green (25%) deployments trail well behind. The industry spent years teaching itself to ship progressively and limit blast radius, but isn't applying the same practice to Agent DLC yet.

Ownership has no standard system of record

How organizations record agent ownership, risk level, and compliance requirements:

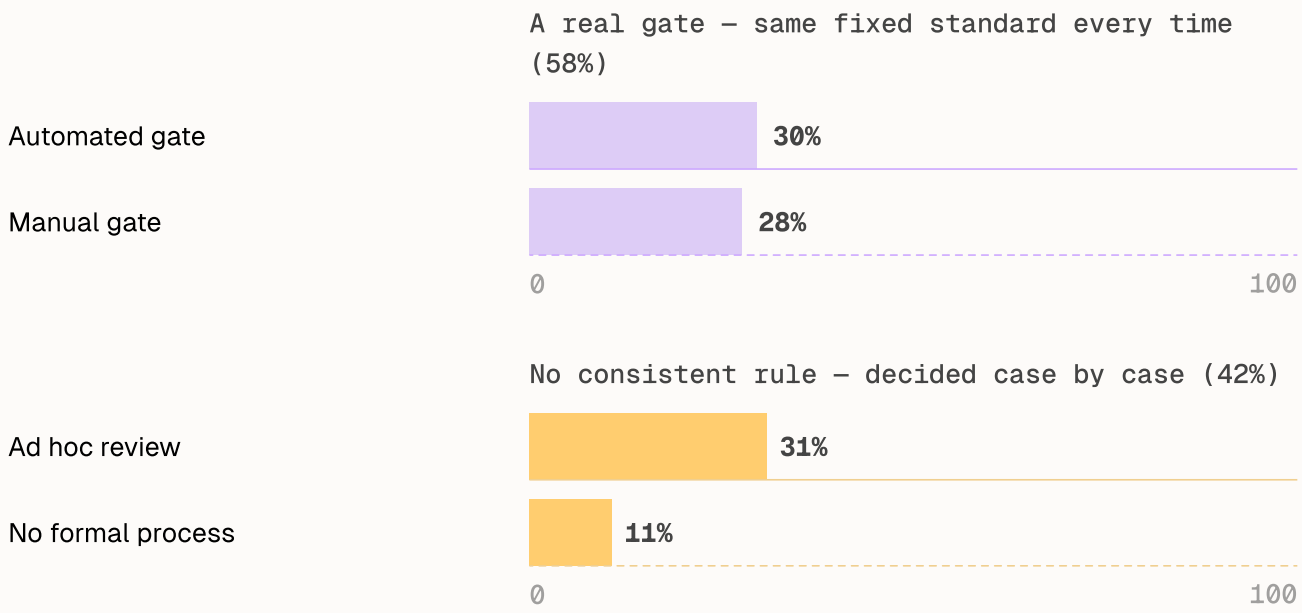


Ownership lacks a standard system of record. A centralized AI/agent inventory tool leads at just 30%, followed by an internal developer portal (13%), and a custom inventory (12%). The rest is split across GRC/compliance systems (10%), service catalogs (10%), CMDBs (9%), and 8% still track ownership in a repository README.

The survey looked at the 10 capabilities: versioning, permissions, evals, cost telemetry, rollback, inventory, and the rest. For each one, more than 80% of organizations have at least partial tooling support. So, capability by capability, the market looks well covered.

Now look at organizations instead of capabilities. For each organization, count how many of the 10 capabilities it actually supports. Here, 25% of organizations are missing real support for 3 or more of them.Same data, two different counts: one counts organizations per capability, the other counts capabilities per organization. The first count looks healthy. The second doesn't.

The production-promotion gate: more than 4 in 10 decide case by case



The **production-promotion** gate is arguably the most important finding in this section. Among organizations that promote agent changes to production, 58% check every change against a fixed, repeatable standard (30% through an automated scorecard, 28% through a manual one, where a human applies the same checklist every time). The other 42% have no such standard: 31% rely on ad hoc review, where the criteria shift from one reviewer, or one change, to the next, and 11% have no formal process at all. More than 4 in 10 organizations that promote agent updates to production are deciding whether to trust each change on the fly, with no consistent rule behind the decision either way.

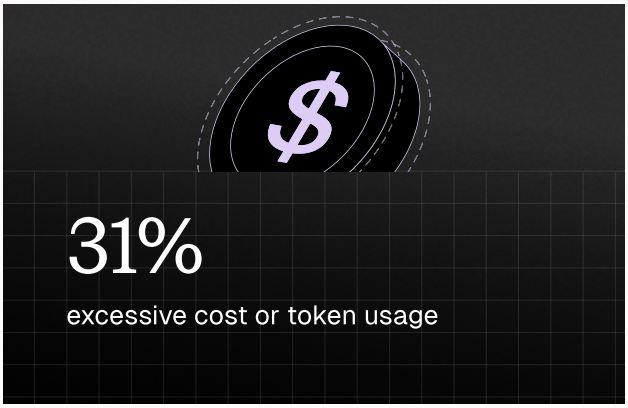
Put together: 62% of organizations are managing agent releases with tooling built for humans and/or have no consistent standard for deciding when a change is safe to promote to production. This is where the incidents in Part 4 are most likely coming from.

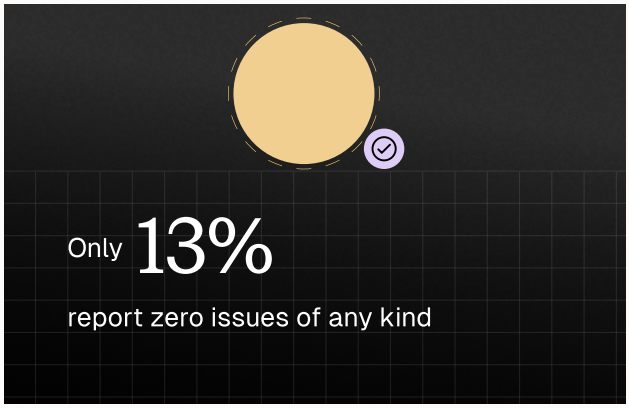
PART 04

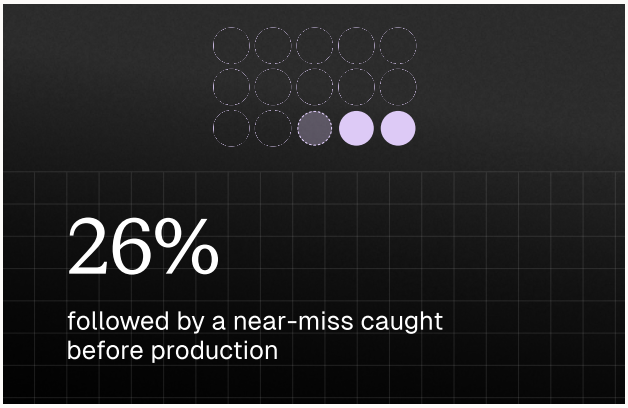
The incidents are already here

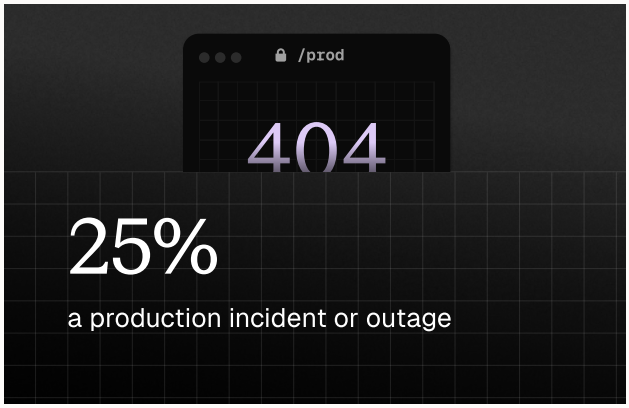
58% of organizations report an increase in production incidents per 100 changes since deploying AI agents into their environment. Only 25% report a decrease.

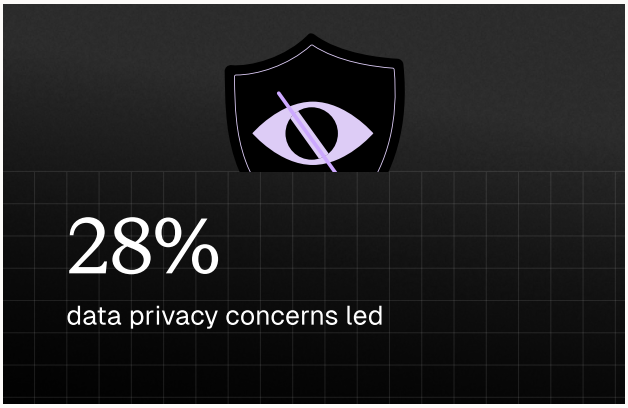
Agent-related issues in the past 12 months cluster tightly together

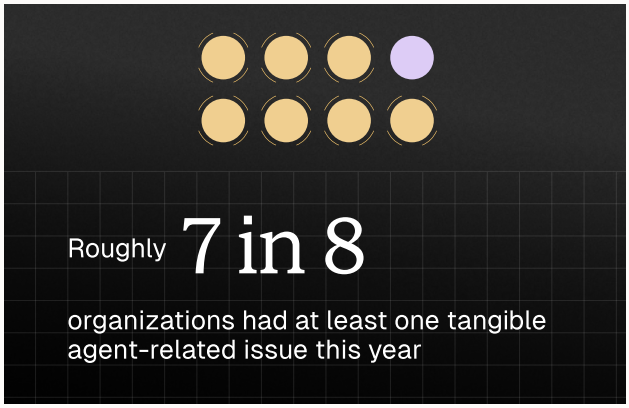






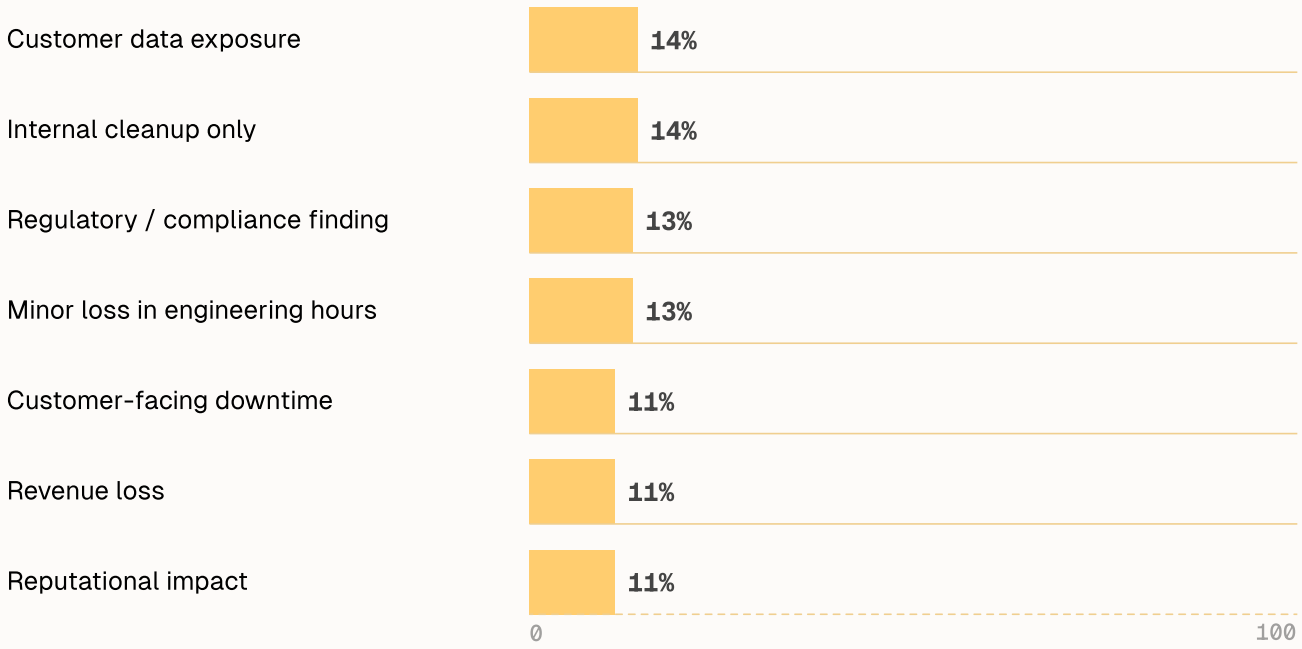






The most severe consequence of an agent incident is broad and evenly spread

All seven disclosed outcomes fall within three points of each other — no clear leader.



When looking at incidents, the consequences are broad and evenly distributed. The most common "most severe consequence" is customer data exposure (14%). This is narrowly ahead of internal cleanup only (14%), meaning no external harm at all. Close behind: a regulatory/compliance finding (13%) and a minor loss in engineering hours (13%). The rest — customer-facing downtime, revenue loss, reputational impact — sit just behind that, each around 11%, with no clear leader among them.

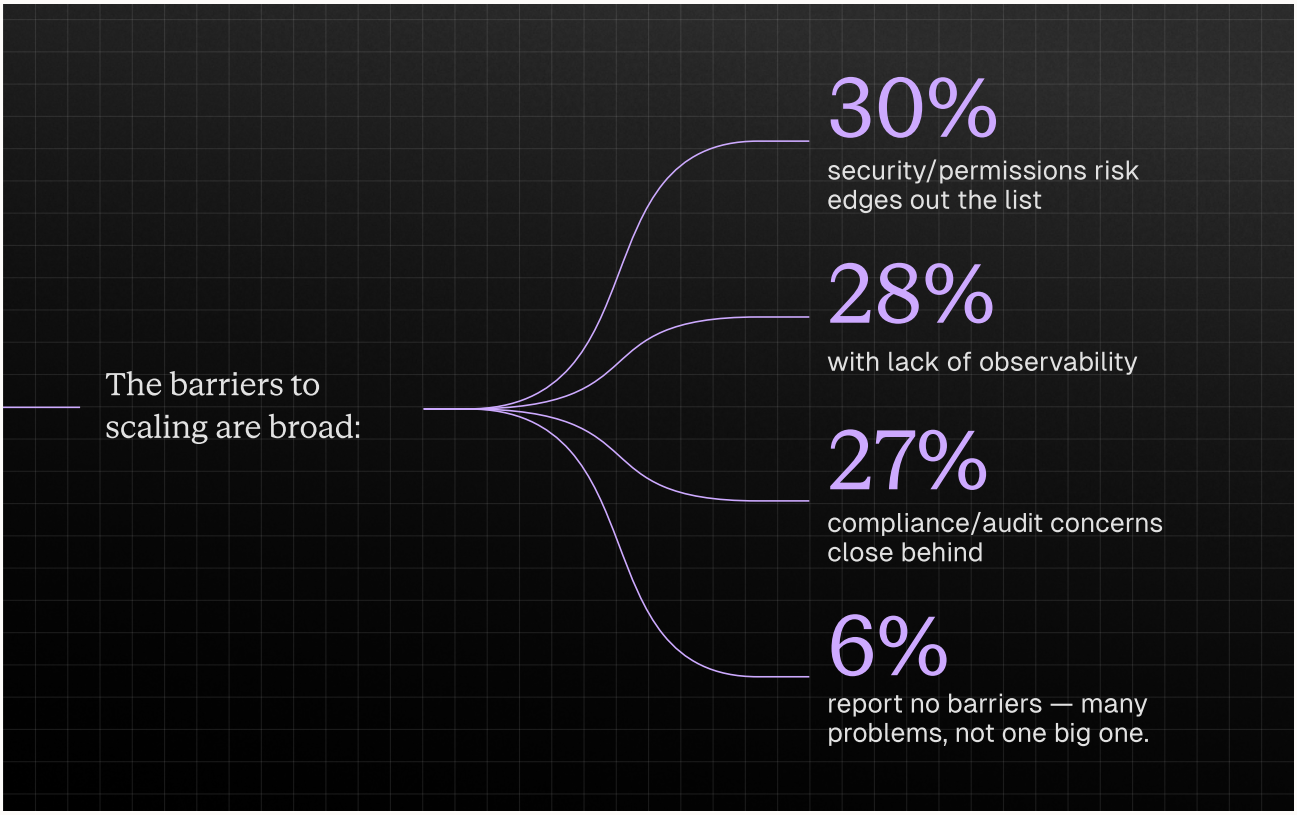
This data isn't an argument that AI agents are net negative; it's that the version of Agent DLC most enterprises have deployed today (partial evals, workaround config, and an inconsistent promotion gate for over 4 in 10 adopters) is currently generating more incidents for most organizations.

We also checked whether any single security control tracked with fewer incidents. None did. Organizations with a given control and those without it reported incident rates that differed by only a few points.

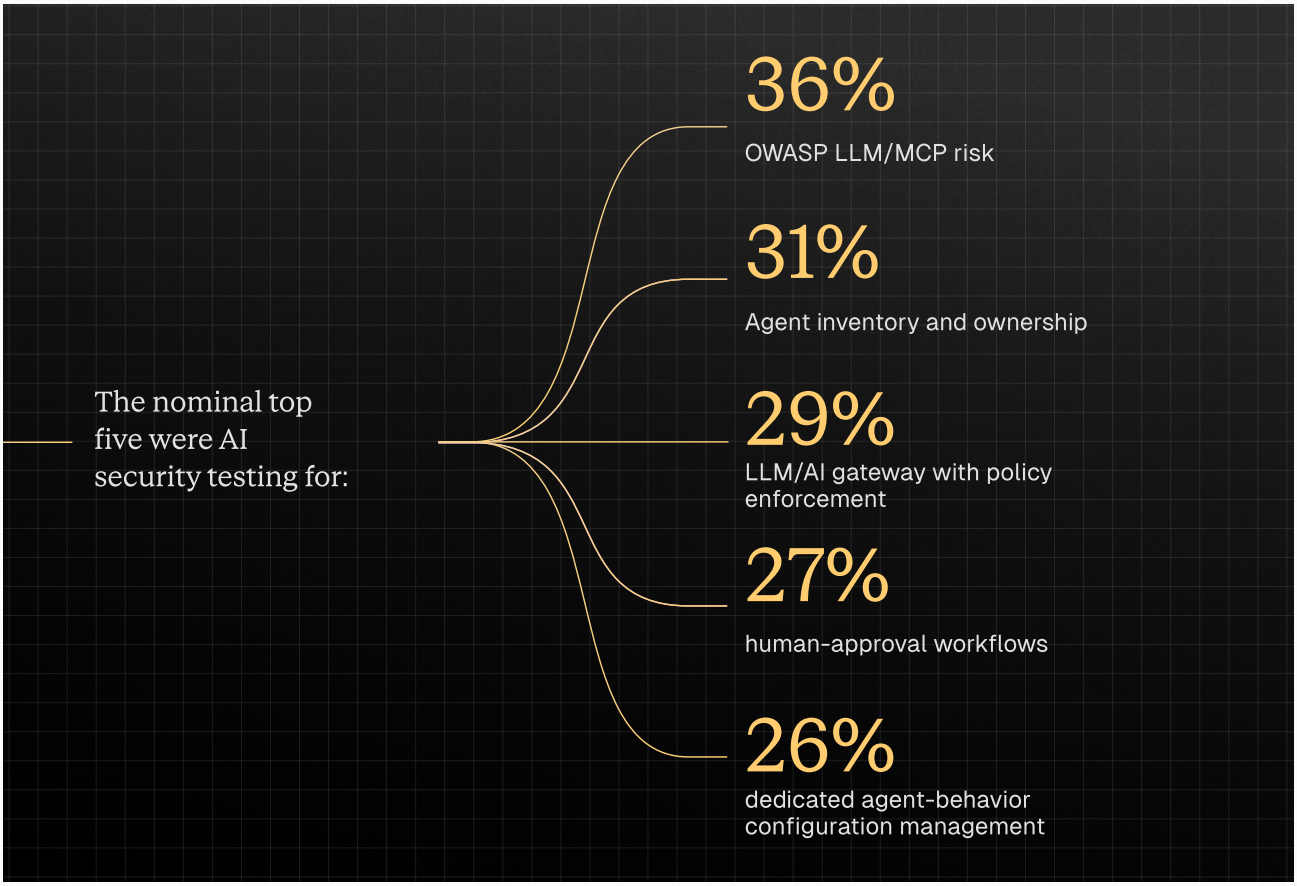
PART 05

What comes next: barriers, priorities, and the investment window

When asked which barriers are delaying organizations from scaling agents into production, the responses were broad, rather than concentrated.



We asked what capabilities matter most for scaling agents safely, out of fifteen options. The responses clustered tightly — every option landed between roughly 22% and 36%, which, at this sample size, is close to a statistical tie across the board.



These are five things clearly on people's minds, not a ranked priority list. It closely mirrors the barriers list, suggesting organizations know their weak points.

Current tooling is fragmented: the leading approach, AI-platform-native tooling, covers just 24% of the market. Only 9% have an integrated, purpose-built control plane across the lifecycle today. Another 13% describe their approach as either "mostly manual" or a set of disconnected point tools, including 5% who are working entirely by hand.

84% see high or moderate value in a single control plane spanning test, deploy, secure, optimize, release, govern. That's a striking number. Teams are closing or planning to close this gap by investing in Agent DLC tooling. 45% are already investing or plan to within 3 months, and 76% expect to invest within 6 months. Only 1% have no plans at all.

Bottom line: confidence and the actual presence of a control are two different things, and most organizations are reporting on the former without checking the latter. Across most domains in this report, the message is the same: don't assume that confidence means a control is in place. Check explicitly. The organizations best positioned aren't the ones with fewer incidents by default. They're the ones who don’t treat confidence as a stand-in for control, and verify, domain by domain, what controls are actually working.

Methodology: how to read this report

This report is based on a survey of 700 technology professionals at large enterprises in the United States, United Kingdom, France, Germany, and India, conducted by Sapio Research in July, 2026 on behalf of Harness.

Who was surveyed:

Respondents were screened for organizations with 1,000 or more employees, 100 or more developers, and annual revenue above $100 million, working in software engineering, IT operations or infrastructure, or technology leadership. Critically, participation required that the organization had already deployed AI agents in production, in pilot, or at least in a live proof of concept. Organizations still exploring, researching, or not yet started were not surveyed.

What that means for the figures:

Throughout this report, "organizations" refers to this population. Percentages describe organizations that have already committed to running AI agents, not enterprises generally. Adoption figures in particular should be read as reflecting how far committed adopters have progressed, not how widespread adoption is.

How to read the numbers:

Where respondents could select more than one answer, percentages show how many chose each option, not how many have the capability at all. A capability selected by 33% of respondents may still be present in nearly all of them alongside others. Figures are self-reported.

How we define Agent DLC:

Harness defines the Agent Development Lifecycle (Agent DLC) as the full lifecycle for building, testing, securing, deploying, operating, and governing an AI agent — distinct from the software delivery lifecycle the agent may participate in. It exists because agent behavior varies from run to run, and standard testing and release checks weren't designed for that kind of variability.

About the authors

Trevor Stuart

Trevor is the Senior Vice President and General Manager at Harness, where he leads the company's Growth and Monetization strategy. He is responsible for how customers discover, purchase, and scale their use of the Harness platform. Trevor brings a wealth of experience across operations, product management, and startup investing, and has a proven track record of building and scaling high performing teams in both startup and large enterprise environments.

Before joining Harness, Trevor was the President and Co-Founder of Split Software, which was acquired by Harness in 2024. At Split, Trevor played a pivotal role in driving the company's growth and product innovation. Prior to founding Split, he was responsible for leading product simplification initiatives at RelateIQ, which was acquired by Salesforce.

Keith Mann

Keith Mann is Field CTO and Head of Research for Harness. He is responsible for capturing the wealth of information about DevSecOps that our customers provide us, analyzing it thoroughly and objectively, and publishing it as trustworthy research for the benefit of everyone.

Prior to joining Harness, Keith spent ten years as Senior Director, Analyst at Gartner, where his work included leading the DevSecOps Magic Quadrant. He has published over 40 research papers, spoken at dozens of industry events, and advised thousands of clients. He has also been a software engineer, architect, and agile transformation leader in the course of his 35-year career.