CloudBees

The State of Code Abundance 2026:

The Gap Between AI Output and Enterprise Control

AI-powered development has delivered on its promise of generating more code faster, at enterprise scale. This report examines the gap between enterprise confidence in AI-powered development and the operational, financial, and governance realities emerging in its wake.

Table Of Contents

CloudBees

Executive Summary

AI is generating code faster than enterprises can govern it, and many organizations are already paying the price in production failures.

Based on a survey of more than 200 enterprise technology leaders conducted by an independent research agency, 92% of leaders express confidence in the production readiness of AI-generated code, yet 81% report an increase in production issues linked to it.

AI now generates or assists in writing 61% of the average enterprise codebase, and 64% of engineering organizations describe it as "widely adopted" or "fully integrated" into their workflows. The result is “code abundance”: a new condition in which AI generates code faster than enterprises can verify, govern, and attribute what it produces.

52% of organizations report a significant increase in software development output due to AI coding tools.This new era of code abundance introduces an unfamiliar reality, one in which writing code is no longer the most time-consuming part of software delivery. The harder part is governing what AI produces, and proving the value it delivers. While 68% believe AI has clearly delivered business value, organizations can only attribute a third of their AI-related spend to specific business outcomes.

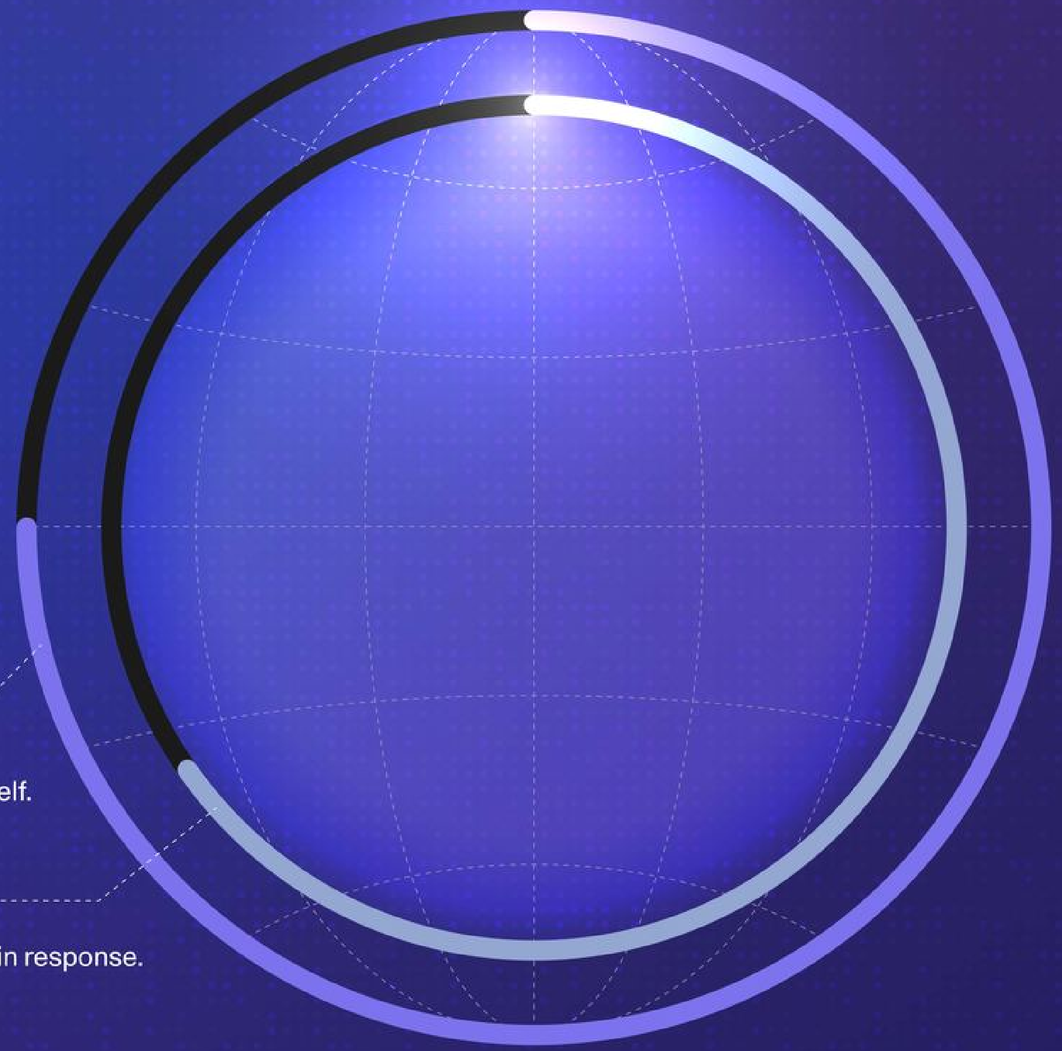
The cost picture is just as stark. AI-related costs are accumulating within existing infrastructure budgets — CI/CD, testing, security scanning — in ways that don't appear as AI spend on the balance sheet. 54% report a significant increase in CI/CD infrastructure spend over the past 12 months, while 53% say testing, security, and deployment costs rose significantly. Only 45% describe their AI spend as highly predictable quarter-to-quarter. This unpredictability is also creating a new form of operational friction: “token anxiety.” As AI usage scales, organizations are struggling to forecast and govern consumption-based costs tied to prompts, agents, and model interactions. The result is growing pressure to balance AI-driven productivity gains against increasingly opaque infrastructure economics.

Enterprise leaders are confident in AI-generated code itself, but it's what happens downstream that's exposing the gaps. Today's governance failures will pose an even higher risk as autonomous agents move from early adoption to mainstream delivery pipelines.

Key Findings

Writing Code Is The Easy Part

70% of engineering leaders say maintaining their test suite is now a bigger burden than writing code itself, and 62% have increased automated testing in response.



70 %

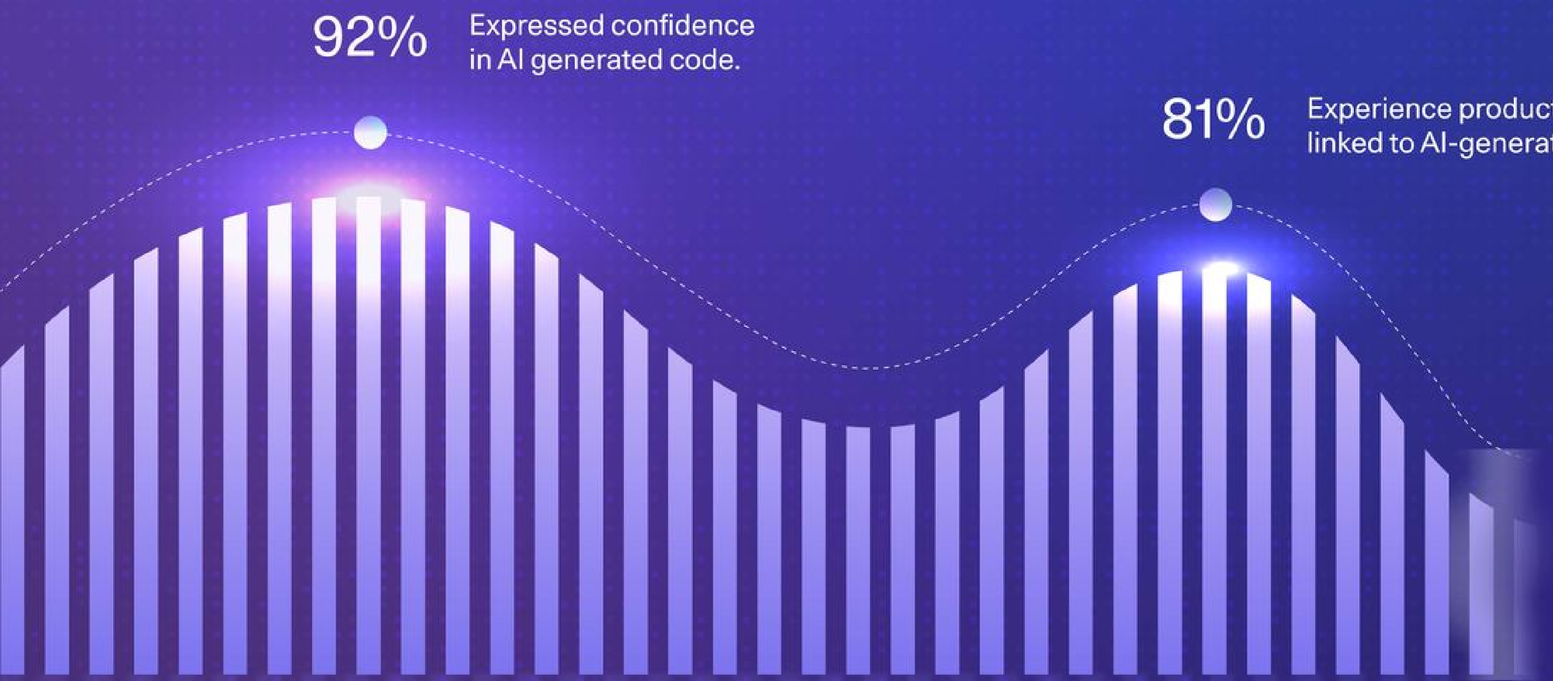
Say maintaining their test suite is a bigger burden than writing code itself.

62 %

Have increased automated testing in response.

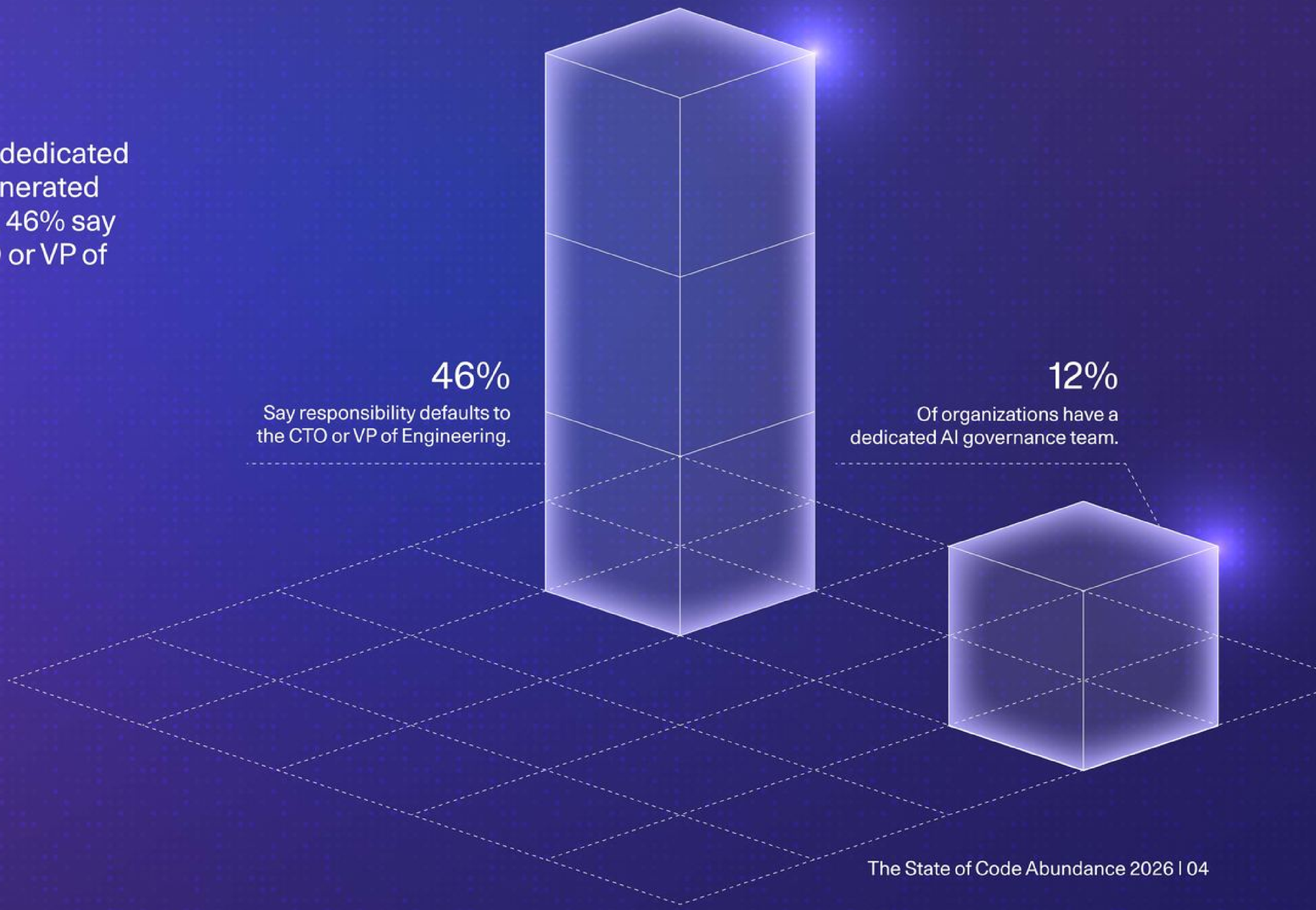
Production Failures Are Common

81% of organizations have seen production issues increase linked to AI-generated code, despite 92% expressing confidence in its production-readiness before it ships.



Accountability Has No Clear Owner

Only 12% of organizations have a dedicated AI governance team. When AI-generated code causes a production failure, 46% say responsibility defaults to the CTO or VP of Engineering.



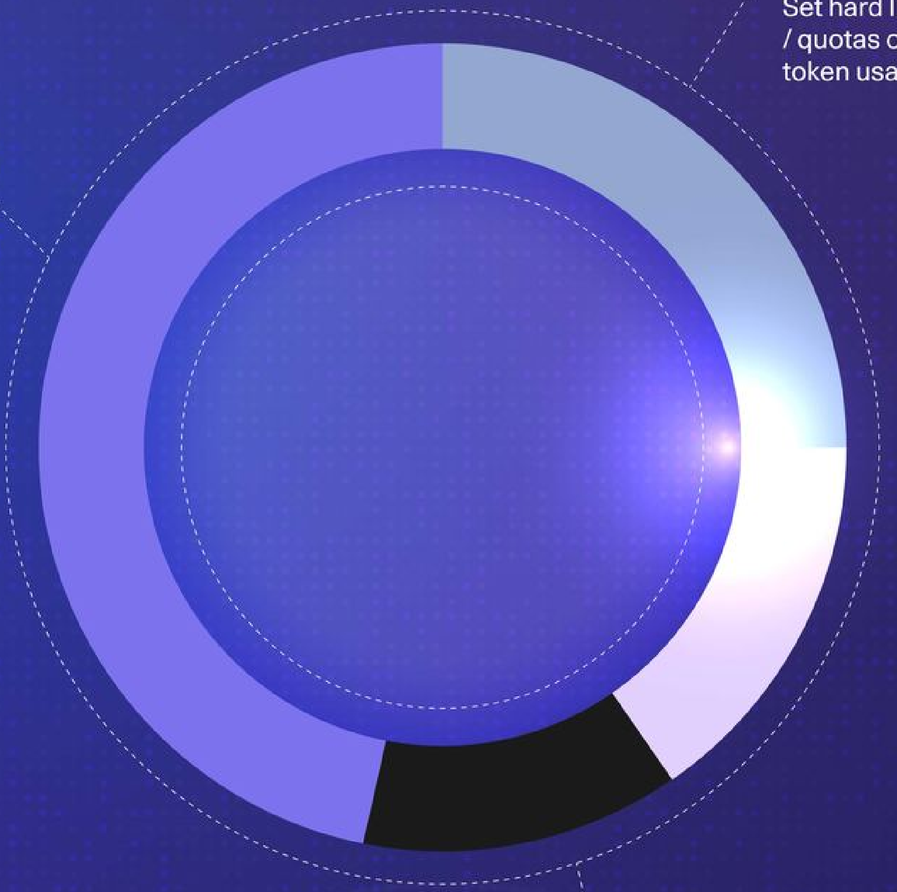
Everything Is Getting More Expensive

54% report a significant increase in CI/CD infrastructure spend in the past 12 months. 53% report a significant increase in testing, security, and deployment costs.

CloudBees

Unpredictable Spend Contributes to Token Anxiety

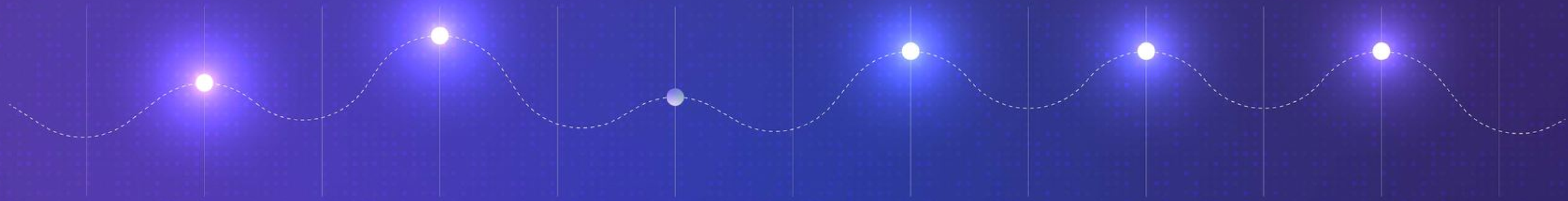
Less than half (45%) of teams describe their AI spend as highly predictable quarter-to-quarter. 27% have set hard limits or quotas on token usage, with just 18% having implemented automated controls to manage that spend.



CloudBees

ROI Is Often Unclear

68% of organizations believe AI has clearly delivered business value, yet they can attribute only 31% of their AI-related spend to specific business outcomes. 36% track AI spend without measuring ROI or don't measure it at all.



The Code Abundance Readiness Evaluation (CARE) Score

To understand how enterprises assess their own readiness for the demands of code abundance, CloudBees developed the Code Abundance Readiness Evaluation (CARE) score.

This proprietary framework measures how enterprise leaders perceive their own readiness across six dimensions critical to governing AI-generated code at scale: cost visibility, budget predictability, productivity measurement, governance maturity, cross-pipeline visibility, and token governance.

To calculate the CARE Index score, each response across the six dimensions is converted to a 0–100 scale, with the most positive response scoring 100 and the least scoring 0. Scores are then averaged across all six dimensions to produce a composite score out of 100.

Across all six dimensions, the majority of enterprise technology leaders rate their organizations as capable and in control of AI-generated code and its associated costs.

Cost visibility: 54% describe their ability to track and attribute AI costs as very clear.

Budget predictability: 89% describe their AI spend as predictable or very predictable quarter-to-quarter.

Productivity measurement: 51% are very confident that they can accurately measure AI productivity gains and ROI.

Governance maturity: 86% have fully or mostly implemented formal policies and guardrails for AI tool usage, yet only 56% say those processes are always enforced.

Pipeline visibility: 86% report full or high visibility into AI-generated code across the delivery pipeline.

Token governance: 86% have fully or mostly implemented specific limits and controls on AI token usage.

Based on these six dimensions, enterprises score 83.6 out of 100 on the CARE Index—reflecting high self-reported confidence in their readiness to manage AI-generated code at scale. That confidence is, at times, at odds with what organizations are actually experiencing.

Enterprise Leaders Are “All In” on AI

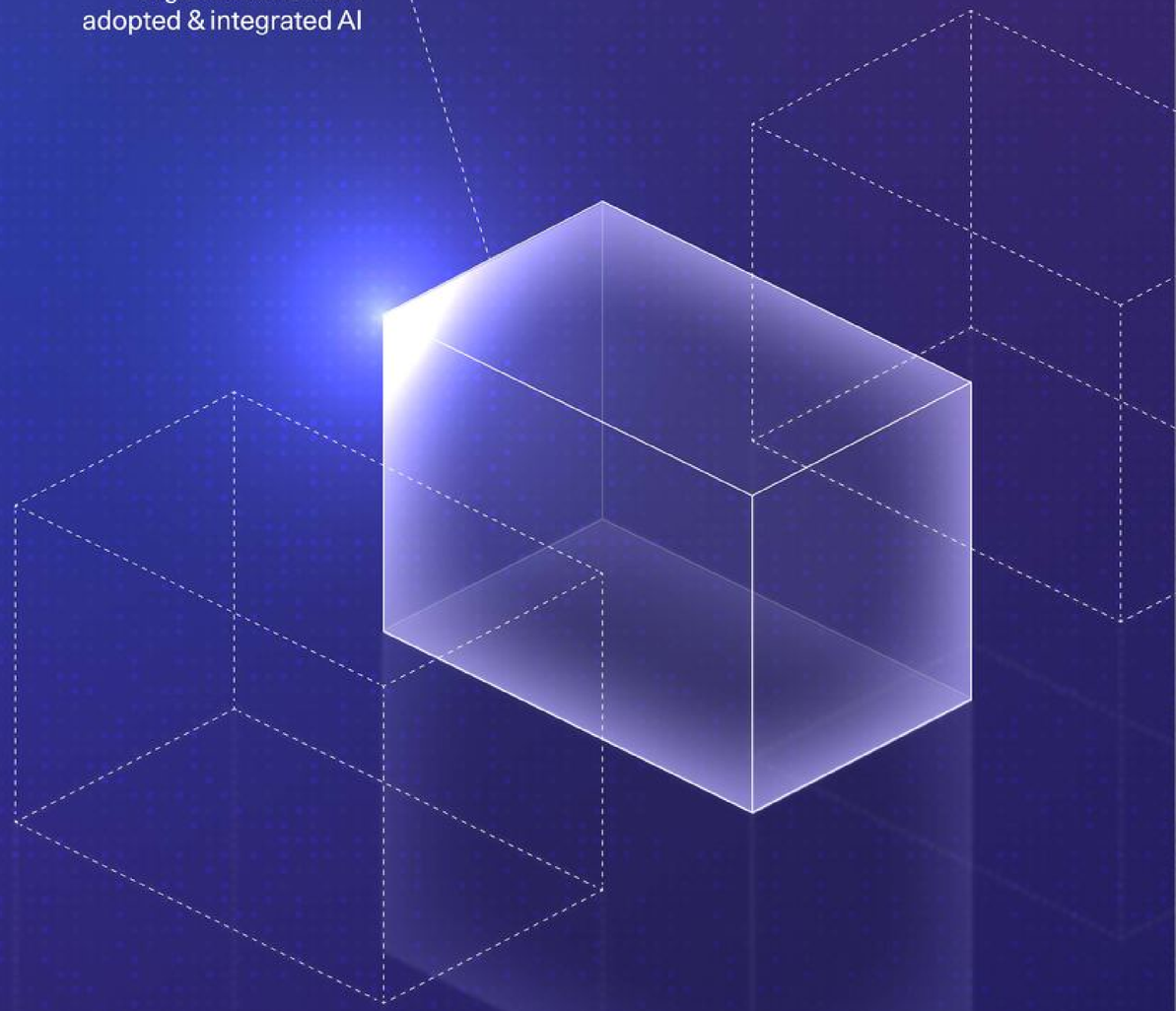
AI-driven code generation has moved from experimentation to enterprise standard. And by most measures, they feel ready for it. Nearly two-thirds (64%) of engineering organizations now describe it as widely adopted or fully integrated, and 52% report a significant increase in software development output due to AI coding tools.

56% of enterprise technology leaders believe current industry benchmarks definitively capture the true productivity impact of AI coding tools, and 51% are very confident that their organization can accurately measure the productivity gains attributed to AI-generated code. AI-generated code.

As AI takes on more of the writing, the pressure in the delivery process is shifting downstream. When asked where the primary bottleneck in their delivery process sits today, more leaders point to what happens after code is written—reviewing it, testing it, and deploying it (57%)—than to writing it (35%).

64%

Of engineering organizations have adopted & integrated AI



The Gaps Behind the Green Lights

Despite 92% expressing high confidence int he production-readiness of AI-generated code, 81% of organizations have experienced an increase in production issues they attribute directly to it.

The same pattern holds on the value side. 68% of organizations believe AI has clearly translated into measurable business outcomes, yet they can attribute only 31% of their AI-related spend to specific outcomes. 54% measure productivity primarily by time saved, a self-reported metric that reflects activity, not business impact.

Without attribution, these perceived benefits are little more than hopeful assumptions.

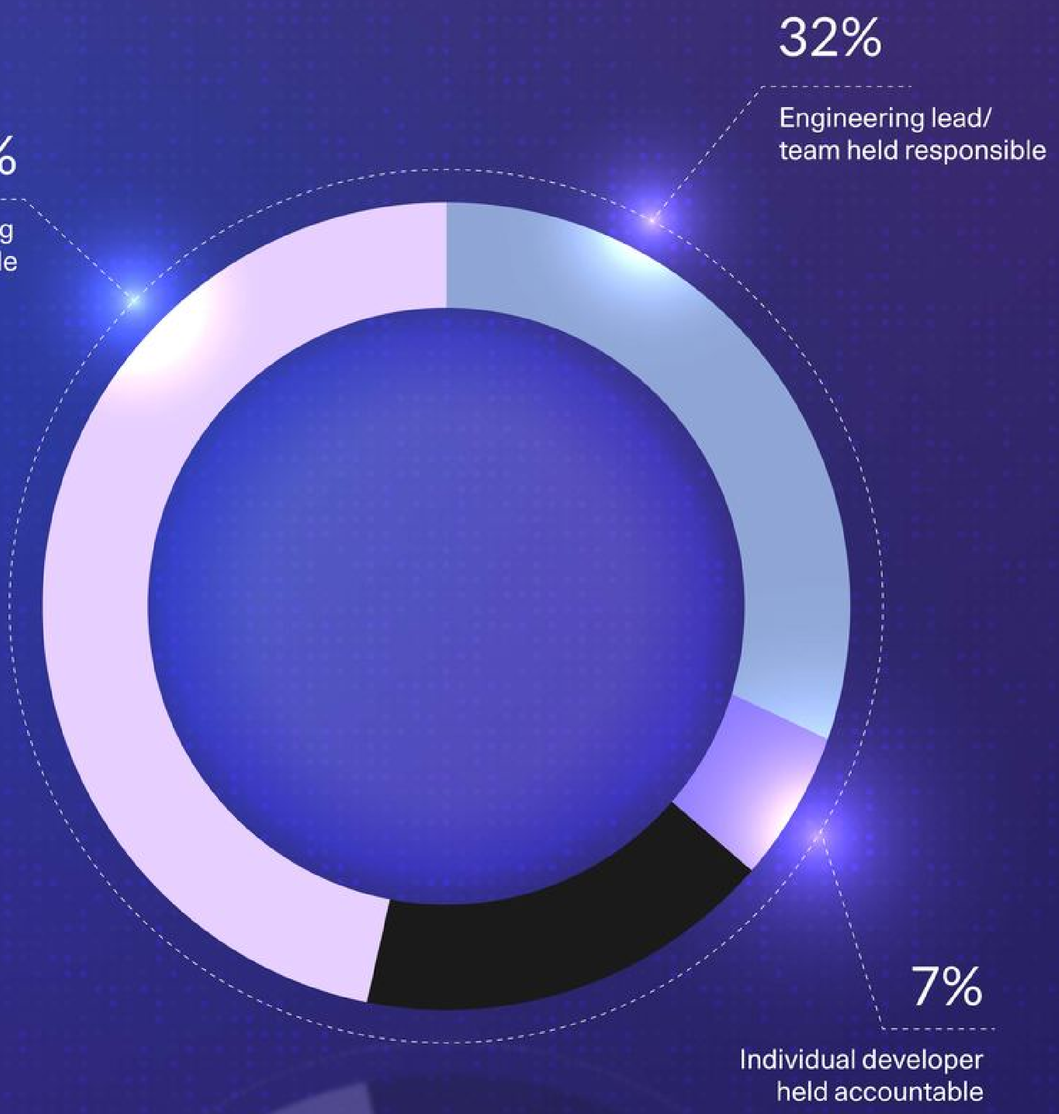
When Something Breaks, Nobody Owns It

When AI-generated code causes a production failure, the question of accountability does not have a clean answer. Only 12% of organizations have a dedicated AI governance team in place to absorb that responsibility. For most, the data reveals a system that defaults upward:

- **46% say responsibility falls to the CTO or VP of Engineering**
- **32% say it lies with the engineering lead or team that used the tool**
- **7% hold the individual developer who shipped the pull request accountable**

When there are no humans behind the code, ownership of what breaks becomes harder to assign.

The governance frameworks meant to fill that gap aren't consistently doing so either. 93% of organizations say they have a formal process for reviewing and releasing AI-generated code into production, but only 56% say those processes are always enforced.



More Code, More Cost, Less Control

A decade ago, cloud computing blindsided enterprise finance teams with unpredictable costs and no governance playbook. Organizations adopted fast, figured out the economics later, and scrambled when the bills arrived. The same pattern is playing out today with AI across every layer of the delivery stack.

Sixty-three percent of organizations say they have a fully implemented strategy for scaling CI/CD infrastructure to match AI code volume. Yet 54% report a significant increase in CI/CD infrastructure spend in the past 12 months, and 53% saw a significant increase in testing, security, and deployment costs. Not a single organization reported a decrease in any category.

The majority of enterprises surveyed are already spending between $1M and $10M annually on CI/CD infrastructure, before accounting for the cost increases driven by code abundance.

Rising costs are easier to control when you can anticipate and plan for them.

However, just 45% describe their AI spend as highly predictable quarter-to-quarter. Only 27% have set hard limits or quotas on token usage, and just 18% have implemented automated controls to manage that spend.

The barriers to aligning cost governance with AI adoption are as much about the market as they are about internal execution. 31% of enterprises cite constantly shifting AI tools and platforms as their biggest obstacle. Another 25% say the industry simply has no established benchmarks or frameworks for AI cost management to work from yet.

Unpredictable conditions like these are what lead to what the industry has started calling "token anxiety"—the creeping uncertainty that comes with AI spend scaling faster than you can understand, attribute, or control it.

Organizations are being asked to govern a landscape that has not yet stabilized.

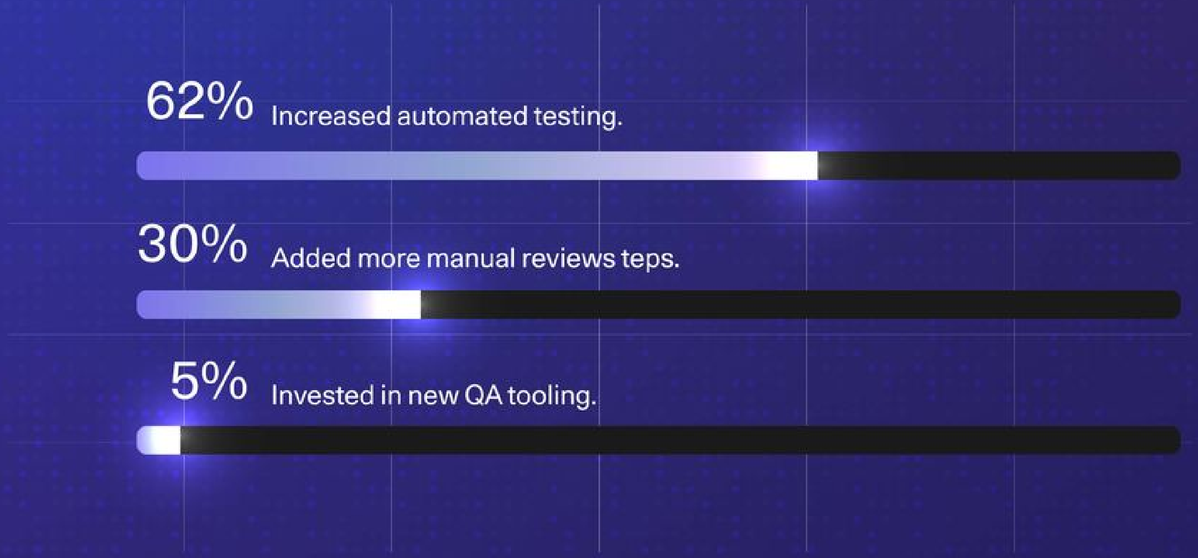
Testing Can't Keep Up

AI coding tools have made writing code faster, but the process of verifying it hasn't kept pace. The result is a widening gap between how much code organizations are shipping and how confidently they can stand behind it.

70% of engineering leaders now say maintaining their test suite is a bigger burden than writing code itself. Ninety-seven percent of organizations report having changed their testing approach in response to higher AI code volume.

- **62% increased automated testing**
- **30% added more manual review steps**
- **5% invested in new QA tooling**

As code volume continues to grow, running more tests will not be enough. Organizations will also need to know which tests are worth running for each change, and which are not.



The Essential Foundation for AI in the SDLC

The data shows that organizations have processes in place to manage AI-driven velocity. They have governance frameworks for reviewing and releasing AI-generated code. They have infrastructure plans for scaling to meet AI code volume. And they are confident that those strategies are working.

So why does the data reveal a clear disconnect between confidence and execution?

One key factor runs through all three: a lack of visibility. A strategy is only as effective as the visibility that supports it, and each tool in your delivery stack only sees its own piece of the picture. AI needs the complete picture to act safely and usefully in your delivery pipeline.

Closing that gap requires a layer above your tools; one that assembles pipeline data, policy definitions, delivery history, and system state into a single, unified view.

CloudBees Unify is just that: a control plane that sits above execution, connecting CI engines, project management systems, and security scanners without replacing any of them. Unify gives teams a single, authoritative view of what is ready to ship, what it costs, and who approved it.

When that unified view is in place, the control plane becomes a context plane for AI: the foundation that makes agents in your delivery pipeline trustworthy. It is also the infrastructure enterprises will need as agentic AI moves from the edges of the delivery pipeline to its center.

Intelligent testing completes that picture, using AI to prioritize the tests most likely to catch relevant failures, so the delivery record reflects what's actually ready, not just what passed. If your organization is navigating the demands of code abundance, CloudBees is built for you.

Book a demo of CloudBees Unify and CloudBees Smart Tests.