

The State of AI-Native Software Delivery **2026**

Why faster code generation exposed
infrastructure as the new bottleneck.
A synthesis of the industry's 2024–
2026 research.

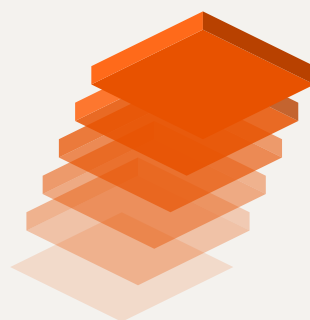


Table of Contents

Key Findings	3
1. How Much Code AI Writes Now	4
2. What Happened to Delivery Speed	5
3. Quality and Stability	6
4. The Review Bottleneck	7
5. The Infrastructure Gap	8
6. Trust and Governance	9
7. What Separates the Leaders	10
Outlook: 2027	11
About Encore	12

About This Report

This report synthesizes more than 25 primary studies, telemetry datasets, and public disclosures published between January 2024 and June 2026: DORA's research program, pull-request telemetry from LinearB and DX, code-quality analyses from GitClear, Veracode, Sonar, and CodeRabbit, surveys from Stack Overflow, Atlassian, Harness, Spacelift, Firefly, Checkmarx, ISACA, and the Cloud Security Alliance, peer-reviewed benchmarks (ITBench, IaC-Eval, SWE-InfraBench, the METR randomized trial), analyst work from Gartner, Bain, BCG, and McKinsey, and first-party disclosures from Google, Microsoft, and GitHub.

No original survey was conducted for this report. Every statistic was verified against its primary source; citations appear as footnotes on the page where each figure is used. Where sources conflict, the conflict is noted rather than averaged away. Self-reported survey results and telemetry are labeled as such throughout, because the two frequently disagree, and the size of that gap shows up as a finding in its own right.

Taken together, the sources describe an industry that has solved code generation and inherited a delivery problem. The chapters follow that arc.

Key Findings

Sources for each finding are cited in the corresponding section.

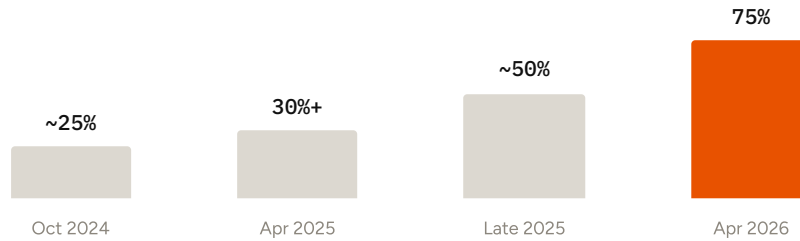
- 1 AI now writes most new code at the frontier.** 75% of new code at Google is AI-generated and approved by engineers, up from a quarter in late 2024. A third of organizations report more than 60% of their code is AI-generated.
- 2 Throughput recovered while stability kept falling.** In 2024, AI adoption correlated with worse delivery throughput and stability. By 2025 throughput turned positive, while instability persisted for a second year.
- 3 The bottleneck moved to review.** AI-generated pull requests wait 4.6x longer for first review and are accepted at 32.7%, versus 84.4% for human-written PRs.
- 4 Staging is leaking.** 43% of AI-generated code that passes QA and staging still requires manual debugging in production, and verifying one AI-suggested fix takes an average of three redeploy cycles.
- 5 Infrastructure is the lagging layer.** 93% of organizations have had an AI-caused infrastructure incident, and 67% say development is moving ahead of infrastructure in AI adoption. Frontier models score up to 97% on application-code benchmarks; the best IT-operations score is 56.2%, and most models remain below 50%.
- 6 Trust is falling as usage rises.** 84% of developers use or plan to use AI tools, while trust in AI accuracy fell from 43% to 33% in a year. Governance trails adoption roughly three to one.
- 7 The platform is the multiplier.** High-quality internal platforms amplify AI's effect on organizational performance, and moving from low to moderate delivery automation more than doubles the likelihood of realizing AI velocity gains, from 26% to 57%. What separates AI leaders from everyone else is the delivery system around the models.

1 How Much Code AI Writes Now

The clearest public record comes from Google's own disclosures: a quarter of new code in October 2024, well over 30% by April 2025, about half by late 2025, and 75% by April 2026, in each case AI-generated and reviewed or approved by engineers.¹ Microsoft put its share at 20 to 30% in April 2025.²

FIGURE 01 — AI-GENERATED SHARE OF NEW CODE AT GOOGLE

OCT 2024 → APR 2026



Share of new code at Google that is AI-generated, per Google's public statements. Sources: Alphabet earnings calls Q3 2024, Q1 2025, Q4 2025; Google Cloud Next 2026.

The broad market is behind the frontier but moving in the same direction. 90% of developers use AI at work per DORA; 84% use or plan to use AI tools per Stack Overflow's 49,000-respondent survey; 90% of engineering teams use AI coding tools per Jellyfish, up from 61% a year earlier.³ At the intensity end, 34% of organizations already report that more than 60% of their code is AI-generated.⁴ Telemetry is more conservative than the self-reports: across DX's 425 instrumented organizations, 22% of merged code was AI-authored in late 2025, rising past 27% in early 2026, and the gap between measured and self-reported numbers recurs throughout this report.⁵

The adoption shows up in aggregate volume: GitHub recorded nearly one billion commits in 2025, up 25% year over year, with more than 1.1 million public repositories now importing an LLM SDK.⁶ Code generation is a solved and widely deployed capability. The next six sections examine what happens to that code after it is written.

¹ Alphabet Q3 2024 earnings; Q1 2025 earnings call; Q4 2025 earnings call; Google Cloud Next 2026 ("75% of all new code at Google is now AI-generated and approved by engineers").

² Satya Nadella, LlamaCon, Apr 29, 2025 ("maybe 20%, 30% of the code that is inside of our repos today").

³ Google Cloud DORA, 2025; Stack Overflow Developer Survey 2025, n=49,000+; Jellyfish, "2025 State of Engineering Management Report".

⁴ Checkmarx, "Future of Application Security in the Era of AI", Aug 2025, n=1,500+.

⁵ DX, "AI-assisted engineering: Q4 impact report", 2025, 135,000+ developers across 425 organizations; DX Q1 2026 benchmark.

⁶ GitHub, Octoverse 2025.

2 What Happened to Delivery Speed

DORA's two-year arc is the industry's most careful measurement, and it splits the question in half. In 2024, every 25% increase in AI adoption was accompanied by an estimated 1.5% decrease in delivery throughput and a 7.2% decrease in delivery stability. In 2025, throughput turned positive while the instability finding persisted.⁷

FIGURE 02 — TWO YEARS OF DORA MEASUREMENT

2024 VS 2025

DORA FINDING	2024	2025
Delivery throughput	Negative (est. -1.5% per +25% adoption)	Positive, "a key shift from last year"
Delivery stability	Negative (est. -7.2% per +25% adoption)	Still negative, second consecutive year
Framing	Bottleneck downstream of code	"AI is an amplifier" of the surrounding system

Sources: DORA, "Accelerate State of DevOps Report", 2024, and "State of AI-assisted Software Development", 2025.

Self-report and measurement disagree

Self-reported results are strong: 64% of engineering teams report achieving at least a 25% velocity increase from AI.⁸ Measured results are narrower. Bain puts the typical realized gain at 10 to 15%, with the saved time often not redirected; BCG's verdict on engineering organizations is "widespread adoption, shallow impact", with half of CIOs unable to quantify GenAI's effect; and in METR's randomized trial, experienced open-source developers took 19% longer with AI tools while believing they had been 20% faster.⁹ Telemetry lands between the extremes: DX measures an average 3.9 hours saved per developer per week in its 2026 analysis, real but bounded by the fact that coding occupies only about 14% of a developer's day, per Microsoft's time-allocation research.⁵

WHY THE NUMBERS DISAGREE

All of these numbers can be true at once. Individual coding time genuinely shrinks. Organizational delivery barely moves, because writing code is 25 to 35% of idea-to-launch, and the stages after it, review, environments, deployment, absorbed the volume as queue time.⁹ Sections 4 and 5 trace where the gains go.

⁷ Google Cloud DORA, "Accelerate State of DevOps Report", 2024; "State of AI-assisted Software Development", 2025.

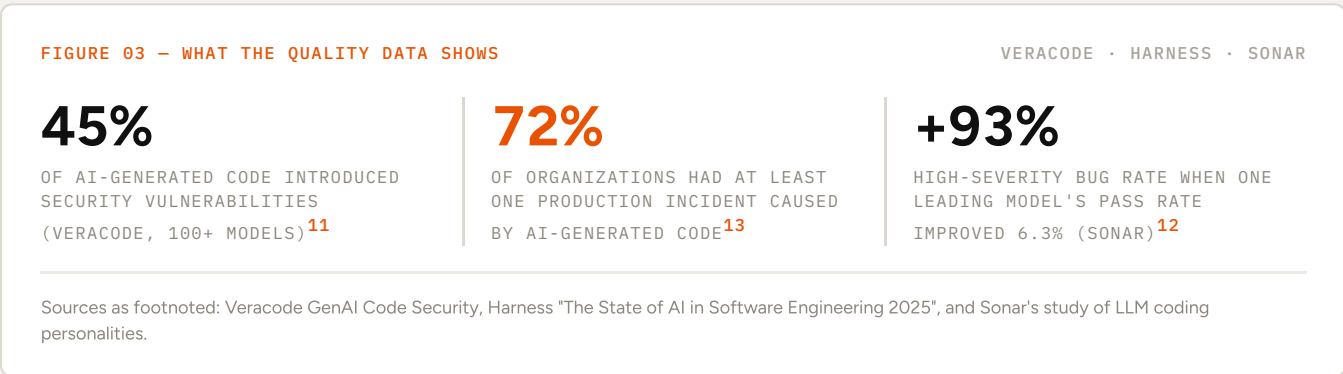
⁸ Jellyfish, "2026 State of Engineering Management Report", May 2026, 600+ engineering leaders.

⁹ Bain & Company, Technology Report 2025; BCG, "Executive Perspectives: AI-Enabled Engineering Excellence Transformation", Apr 2025; METR, Jul 2025 (arXiv:2507.09089).

3 Quality and Stability

GitClear's analysis of 211 million changed lines documents how the code itself changed during 2024: an eightfold increase in blocks of duplicated code, the share of new code churned within a month of authorship climbing from 5.5% in 2020 to 7.9% in 2024, and 2024 as the first year on record in which copy-pasted lines exceeded moved, that is refactored, lines. Its 2026 follow-up, spanning 623 million changes, found refactoring down 70% against 2022 levels while duplicated blocks rose 81%.¹⁰ Across both datasets, volume rose while maintainability metrics declined.

The security data follows the same pattern. Veracode found 45% of AI-generated code introduced security vulnerabilities, with an 86% failure rate on cross-site scripting, and its Spring 2026 update, spanning 150+ models, concluded that two years of model releases left the security pass rate essentially unchanged at roughly 55%.¹¹ CodeRabbit's analysis of 470 real pull requests found AI-co-authored PRs carrying 1.7x more issues; Sonar found one leading model improving its benchmark pass rate 6.3% while its high-severity bug rate rose 93%.¹²



These defects are reaching production: 72% of organizations report at least one production incident caused by AI-generated code, and 45% of deployments involving AI-generated code lead to problems.¹³ This is the mechanism behind DORA's persistent instability finding, and it compounds the queue problem: every incident pulls senior attention back out of the review pipeline that was already the constraint.

¹⁰ GitClear, "AI Copilot Code Quality", Feb 2025, 211M changed lines; "The Maintainability Gap: AI Code Quality in 2026", Jan 2026, 623M changes.

¹¹ Veracode, "2025 GenAI Code Security Report" (80 tasks, 100+ LLMs) and "Spring 2026 GenAI Code Security Update" (150+ models).

¹² CodeRabbit, "State of AI vs. Human Code Generation", Dec 2025, 470 open-source PRs; Sonar, "The Coding Personalities of Leading LLMs", Aug 2025.

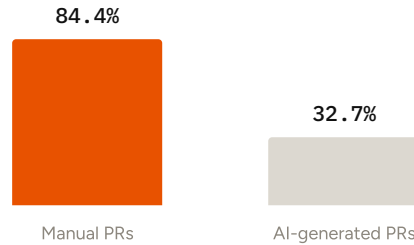
¹³ Harness, "The State of AI in Software Engineering 2025", Coleman Parkes, n=900, fielded Aug 2025.

4 The Review Bottleneck

The single clearest dataset of the year is LinearB's telemetry across 8.1 million pull requests from 4,800 organizations. It shows exactly where the generated volume piled up.

FIGURE 04 — PULL REQUEST ACCEPTANCE RATE

HUMAN VS AI-GENERATED



Source: LinearB, 2026 Software Engineering Benchmarks Report, 8.1M+ pull requests.

FIGURE 05 — WHAT REVIEW DOES WITH AI-GENERATED VOLUME

LINEARB TELEMETRY

4.6x

LONGER WAIT FOR FIRST REVIEW;
AGENTIC-AI PRS WAIT 5.3X LONGER¹⁴

2x

FASTER ONCE PICKED UP, SO
REVIEWERS TRIAGE RATHER THAN
READ DEEPLY¹⁴

2.6x

LARGER AT THE 75TH PERCENTILE,
408 VS 157 LINES¹⁴

Each review therefore costs more attention than a human-written one. Source: LinearB, 2026 Software Engineering Benchmarks Report, 8.1M+ pull requests across 4,800+ organizations.

Developer surveys are consistent with the telemetry. The top frustration in Stack Overflow's survey, cited by 66%, is AI solutions that are "almost right, but not quite"; 45% say debugging AI-generated code is more time-consuming.¹⁵ The checks at the end of the pipeline also miss a substantial share of defects: 43% of AI-generated code that passes QA and staging still requires manual debugging in production, with an average of three redeploy cycles to verify a single AI-suggested fix.¹⁶

WHY THIS DOESN'T SELF-CORRECT

Review capacity is the one input AI didn't multiply. As long as every machine-written change consumes scarce human attention on its way to production, code generation scales and delivery does not. The organizations escaping this constraint (section 7) are reducing the amount of change that requires human review in the first place.

¹⁴ LinearB, "2026 Software Engineering Benchmarks Report", 8.1M+ pull requests, 4,800+ organizations.

¹⁵ Stack Overflow Developer Survey 2025, n=49,000+.

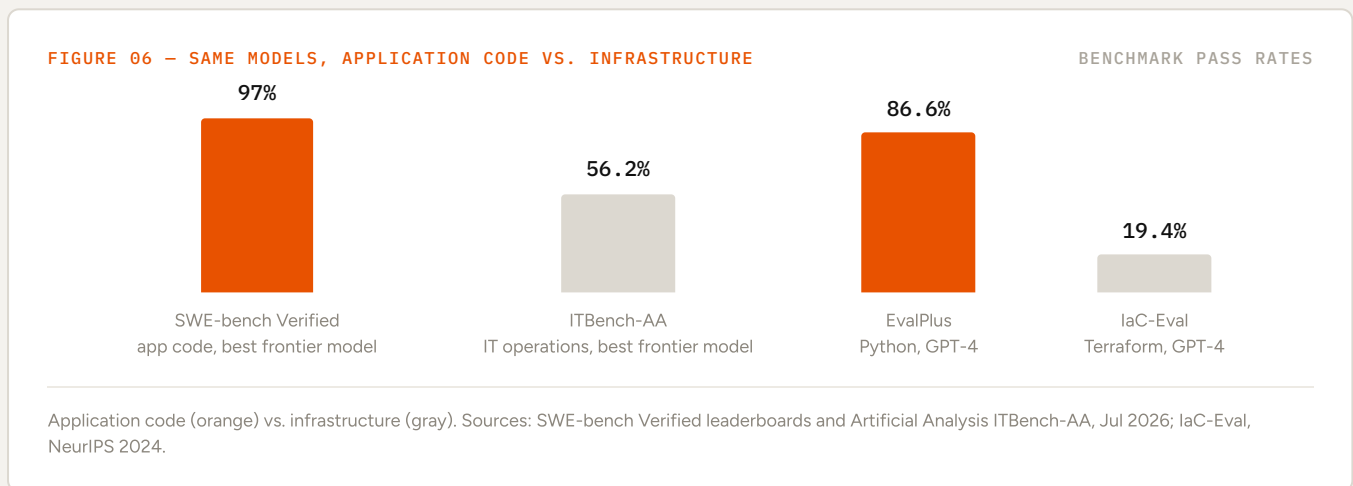
¹⁶ Lightrun, "The State of AI-Powered Engineering 2026", Global Surveyz, n=200 senior SRE/DevOps leaders.

5 The Infrastructure Gap

Below the review queue sits the layer the industry automated least, and the 2026 data made the mismatch explicit. In a 2026 survey of 406 IT and platform leaders, 67% say development is moving ahead of infrastructure in AI adoption, and 93% of organizations have experienced at least one AI-caused infrastructure incident. A third of infrastructure teams say they would apply AI-generated HCL directly to production without any review; only 19% have built governance foundations for AI.¹⁷

The infrastructure estate was strained before AI arrived: 35% of teams tie configuration drift to multiple costly production incidents in the past year, 90% agree their IaC orchestration falls short, and only 34% would trust AI agents to make autonomous production changes.¹⁸

The capability gap in benchmarks



Frontier models resolve up to 97% of real application-code issues, while the best published result on realistic IT-operations scenarios is 56.2%, and most models remain below 50%. Within a single model the split is starker: GPT-4 passes 86.6% of standard Python tasks and 19.4% of Terraform generation tasks.¹⁹ Agents write applications well and operate infrastructure poorly, which means the infrastructure layer either stays human-gated, and remains the bottleneck, or gets restructured so agents never need to touch it.

WHY HUMAN-LEVEL QUALITY IS THE WRONG BAR

Human-written infrastructure code fails at some rate too, and standing practice catches what slips through in production: apply, watch the alerts, roll back. That loop holds because human throughput caps the error volume at a level operations teams can absorb. The point of AI adoption is to multiply change throughput, and at 10x the throughput an agent that matches human quality delivers 10x the escaped errors. Sustaining that through the production loop would take near-perfect correctness per change, far beyond anything above. The alternative is validation before production: infrastructure derived from typed declarations, checked before deployment, environments identical by construction.

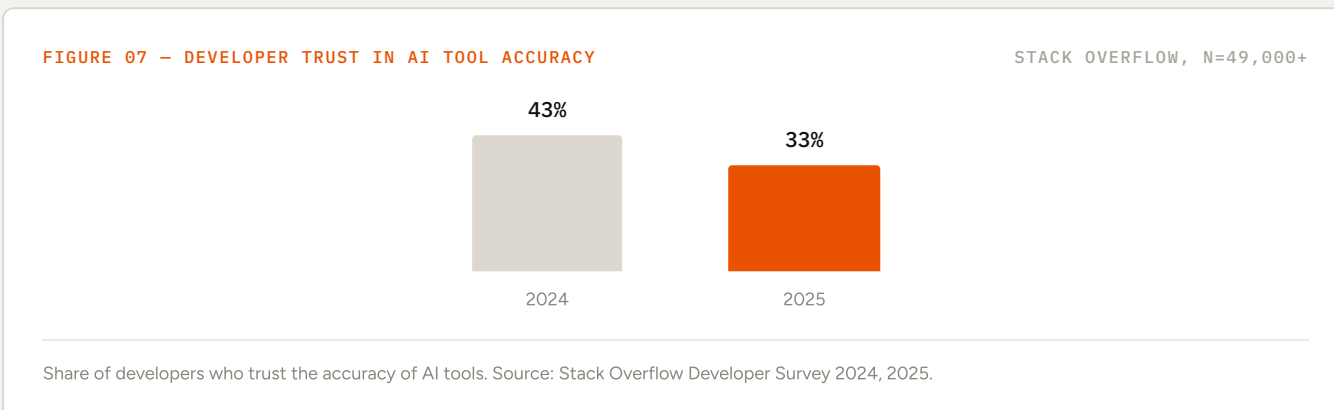
¹⁷ Spacelift, "The Infrastructure Automation Report 2026: The AI Readiness Gap", n=406, fielded by Panterra Group.

¹⁸ Firefly, "State of IaC", 2025 and 2026 editions.

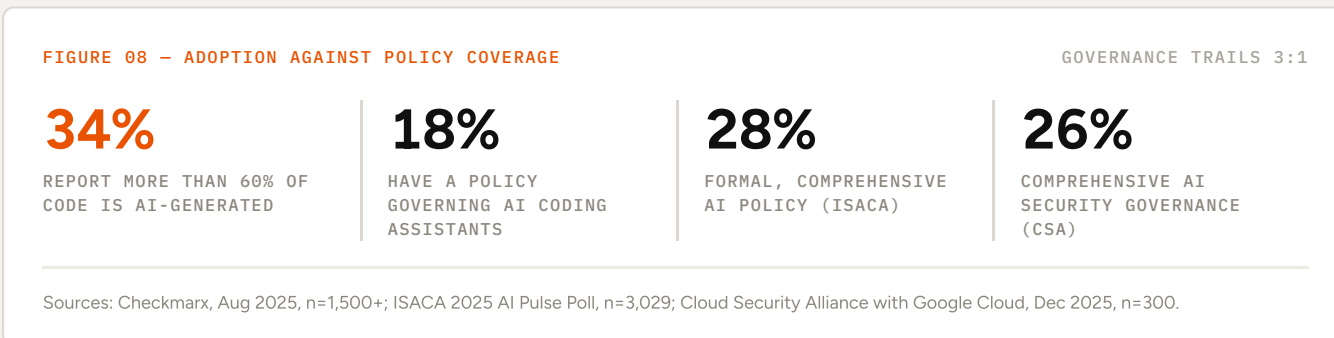
¹⁹ SWE-bench Verified leaderboards (vals.ai, llm-stats.com), Jul 2026; IBM Research, ITBench (ICML 2025) and Artificial Analysis ITBench-AA, 2026; Kon et al., IaC-Eval, NeurIPS 2024.

6 Trust and Governance

Usage and trust are moving in opposite directions. 84% of developers use or plan to use AI tools, up from 76%. Trust in AI accuracy fell from 43% to 33% in the same year, more developers now actively distrust it (46%) than trust it, and only 3% report high trust. DORA measures the same unease: 30% of developers report little or no trust in AI-generated code, while 90% use it.²⁰



Governance trails adoption roughly three to one. 34% of organizations report more than 60% of code is AI-generated; 18% have policies governing AI coding assistants. ISACA puts formal, comprehensive AI policy at 28% of organizations worldwide; the Cloud Security Alliance puts comprehensive AI security governance at 26%.²¹ The exposure is measurable: commits assisted by a leading AI coding agent showed a 3.2% secret-leak rate against a 1.5% baseline across all public commits, AI-service credentials leaked grew 81% year over year, and 2025 set the all-time record for exposed secrets on public GitHub.²²



Regulatory deadlines add to the pressure: the EU AI Act became fully applicable on August 2, 2026, and the Commission's draft Article 50 guidelines drew the first regulatory line through AI-generated code specifically, exempting source code from content-marking while documentation from the same tools stays in scope.²³

²⁰ Stack Overflow Developer Survey 2024 and 2025; Google Cloud DORA, 2025.

²¹ Checkmarx, Aug 2025, n=1,500+; ISACA, 2025 AI Pulse Poll, n=3,029; Cloud Security Alliance with Google Cloud, Dec 2025, n=300.

²² GitGuardian, "The State of Secrets Sprawl 2026".

²³ Regulation (EU) 2024/1689; European Commission, draft Guidelines on Article 50 transparency obligations, May 8, 2026.

7 What Separates the Leaders

Every source that segments its respondents finds the same shape: a small group converting AI into organizational results, and a majority stuck at individual gains. McKinsey's top quintile achieves 16 to 30% improvements in productivity, time to market, and customer experience, with 31 to 45% gains in software quality; only about 5% of surveyed companies meet its "AI high performer" bar. Bain's process-transformers reach 25 to 30% against the typical 10 to 15%.²⁴

The common factor among the leaders is platform quality rather than model choice. DORA 2025 states it directly: "A high-quality platform amplifies the effects of AI adoption on organizational performance." Platform adoption is now near-universal, at 90% of organizations, with 76% running dedicated platform teams; quality is what varies.²⁵ Harness quantifies the same lever at the delivery end: moving from low to moderate continuous-delivery automation more than doubles the likelihood of realizing velocity gains from AI, from 26% to 57%.¹³

	THE MAJORITY	THE LEADERS
AI investment	Tools for individuals	Tools plus end-to-end delivery transformation (Bain)
Realized gains	10-15%, often not redirected	25-30% (Bain); 16-45% across metrics (McKinsey top quintile)
Delivery automation	Low; 6% describe CD as fully automated (Harness)	Moderate-to-high; velocity realization doubles (26%→57%)
Platform quality	Platform exists; impact unmeasured (29.6% of platform teams measure nothing)	Platform treated as the multiplier on the AI budget (DORA)
Review model	Humans review everything, queues grow	Guardrails by construction; human review spent on business logic

Sources as footnoted: Bain Technology Report 2025, McKinsey, Harness, DORA 2025 and the Platform Engineering Community survey.

Gartner's prediction that 80% of large engineering organizations would run platform teams by 2026 arrived on schedule.²⁶ The question for 2026 is whether the platform converts AI speed into delivery or queues it.

²⁴ McKinsey, "Unlocking the value of AI in software development", Nov 2025, and "The State of AI in 2025", n=1,993; Bain & Company, Technology Report 2025.

²⁵ Google Cloud DORA, "State of AI-assisted Software Development", 2025; Platform Engineering Community, Vol. 4, Dec 2025, n=518.

²⁶ Gartner, platform engineering trend prediction, 2023.

Outlook: 2027

Three developments will shape next year's edition of this report.

Agents move from assistants to actors

Agent adoption is early and wary: 52% of developers either don't use agents or stick to simpler AI tools, and 81% have security or privacy concerns about them.¹⁵ Gartner expects over 40% of agentic AI projects to be canceled by the end of 2027, on escalating costs, unclear business value, or inadequate risk controls.²⁷ The projects that survive will be the ones whose delivery systems were built for machine-speed change, with bounded autonomy and guardrails enforced by the platform rather than by review.

Governance rules start to arrive

Formal requirements are arriving from several directions: NIST's control overlays for AI systems, including agent deployment, are in draft; the EU AI Act's transparency regime is live; and customer security questionnaires are already asking AI-governance questions that most organizations cannot yet answer.²⁸ The 18% of organizations with a policy governing AI coding assistants is likely to be the fastest-moving statistic in next year's data.

The infrastructure bottleneck

Every dataset in this report points the same direction: the constraint on AI-era software delivery is shipping code rather than writing it, and the infrastructure layer is where the queue is longest and the automation weakest. Organizations that restructure that layer, so infrastructure derives from the code agents already write well, convert the volume into delivery. Organizations that respond by adding review capacity will keep the constraint and see queue times grow through 2027.

COMPANION GUIDES IN THIS SERIES

This report describes the state of things. Three companion guides cover what to do about it: **The AI-Ready Infrastructure Playbook** (closing the infrastructure gap), **The Compliance Guide to AI-Written Code** (governing what agents ship), and **The CTO's Guide to Post-AI Engineering Economics** (the budget case). All at encore.dev/library.

²⁷ Gartner, press release, Jun 25, 2025.

²⁸ NIST COSAIS drafts, 2025-2026; Schellman, Apr 2026; Regulation (EU) 2024/1689.

About Encore

Encore is a development platform for the AI era, built for a world where AI multiplies software delivery 10-100x rather than making each developer slightly faster. In a Terraform-based workflow, much of the infrastructure code cannot be meaningfully validated until it is applied to a real cloud environment, so mistakes surface in staging or production, and at 10-100x change volume that loop breaks: generated infrastructure would need to be effectively perfect rather than merely human-level.

Encore changes the model instead. Developers and AI agents declare infrastructure requirements directly in ordinary TypeScript or Go, and Encore derives the configuration, permissions, environments, and deployment setup from those declarations, provisioning the resources in your own AWS or GCP account. The same application model runs locally, in preview environments, and in production, so changes are validated with real infrastructure before they reach the production loop.

DECLARE IN CODE	DEPLOY ANYWHERE	OBSERVE EVERYTHING
- Type-safe APIs, databases, queues, caches, and cron jobs - Standard language patterns, no DSLs or YAML - Infrastructure semantics checked at compile time	- AWS or GCP, in your own cloud account - Preview environments per pull request - Production deployments automated end to end	- Distributed tracing with zero configuration - Metrics, logging, and service catalog built in - Architecture diagrams generated from the code

Why this matters for the findings in this report

- **Finding 3 (review bottleneck):** platform guardrails absorb infrastructure review, so human attention goes to business logic.
- **Finding 4 (staging leaks):** local, preview, and production environments are structurally identical, removing the works-in-staging class of failure.
- **Finding 5 (infrastructure gap):** agents never touch IaC, because there is none. Infrastructure derives from the application code they already write well.
- **Finding 6 (governance):** credentials and infrastructure state live in the platform, outside the repository and outside AI reach, with an automatic audit trail.
- **Finding 7 (the platform multiplier):** the delivery automation that doubles realized AI gains is the default, not a program.

"Encore is our foundation for all new development. Since adopting it, we've seen a 2–3x increase in development speed and 90% shorter project lead times. What used to take days or weeks of back-and-forth between developers and infra teams is now automated and completed in minutes."

Josef Sima, Engineering Director

GROUPON

"Encore lets us scale both our product and infrastructure footprint, without additional hiring. The ROI we've seen is outstanding, easily 10x."

Daniel Stocks, CTO

CARLA · THE CASE STUDY REPORTS ANNUAL SAVINGS OF OVER \$200,000 ON HEADCOUNT COSTS

Start building with Encore

Infrastructure that moves at the speed of the code. See the delta on one service.

Get started free

Book a demo