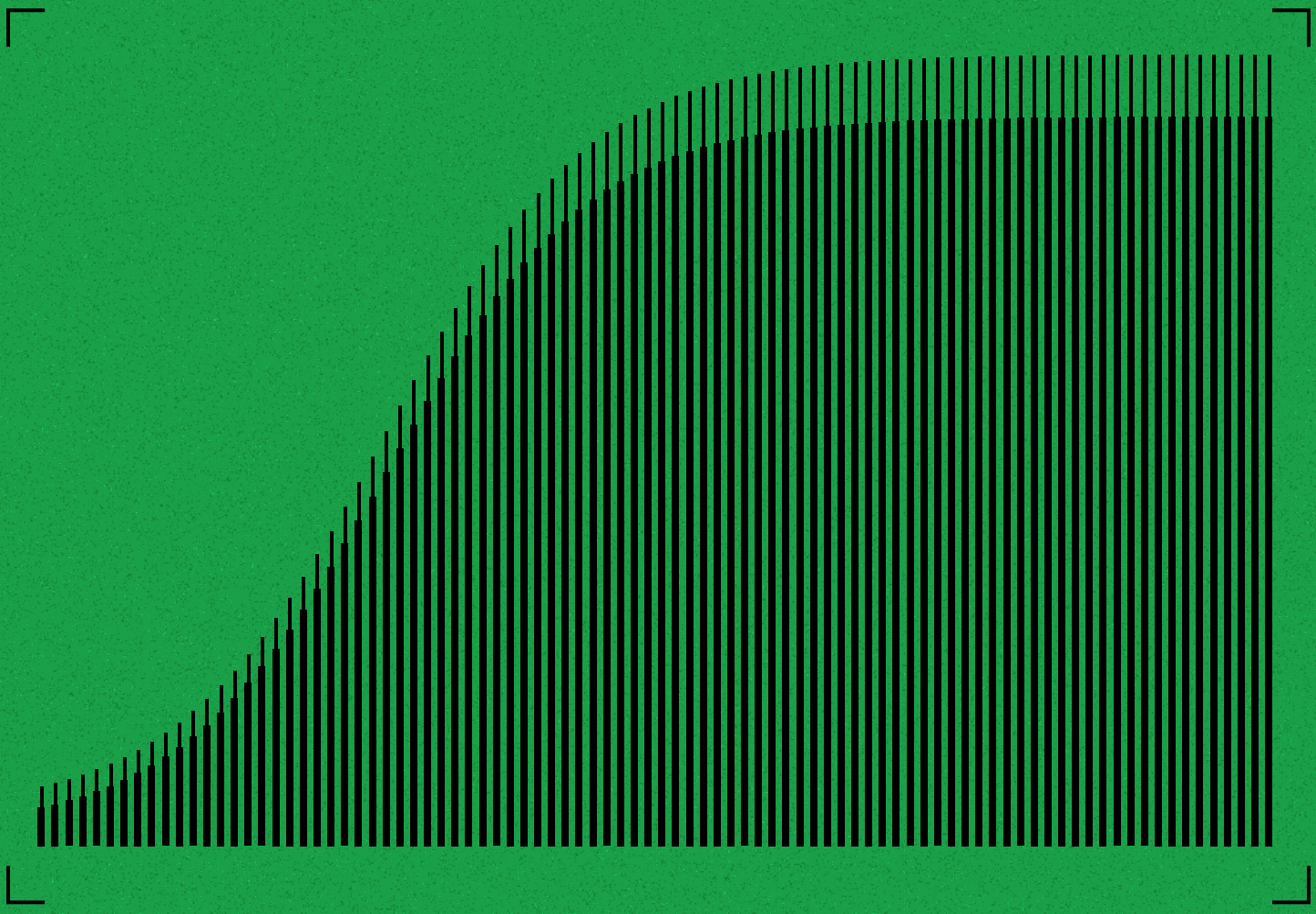
The AI Productivity Plateau

91% OF TEAMS SAY AI MADE THEM MORE PRODUCTIVE.

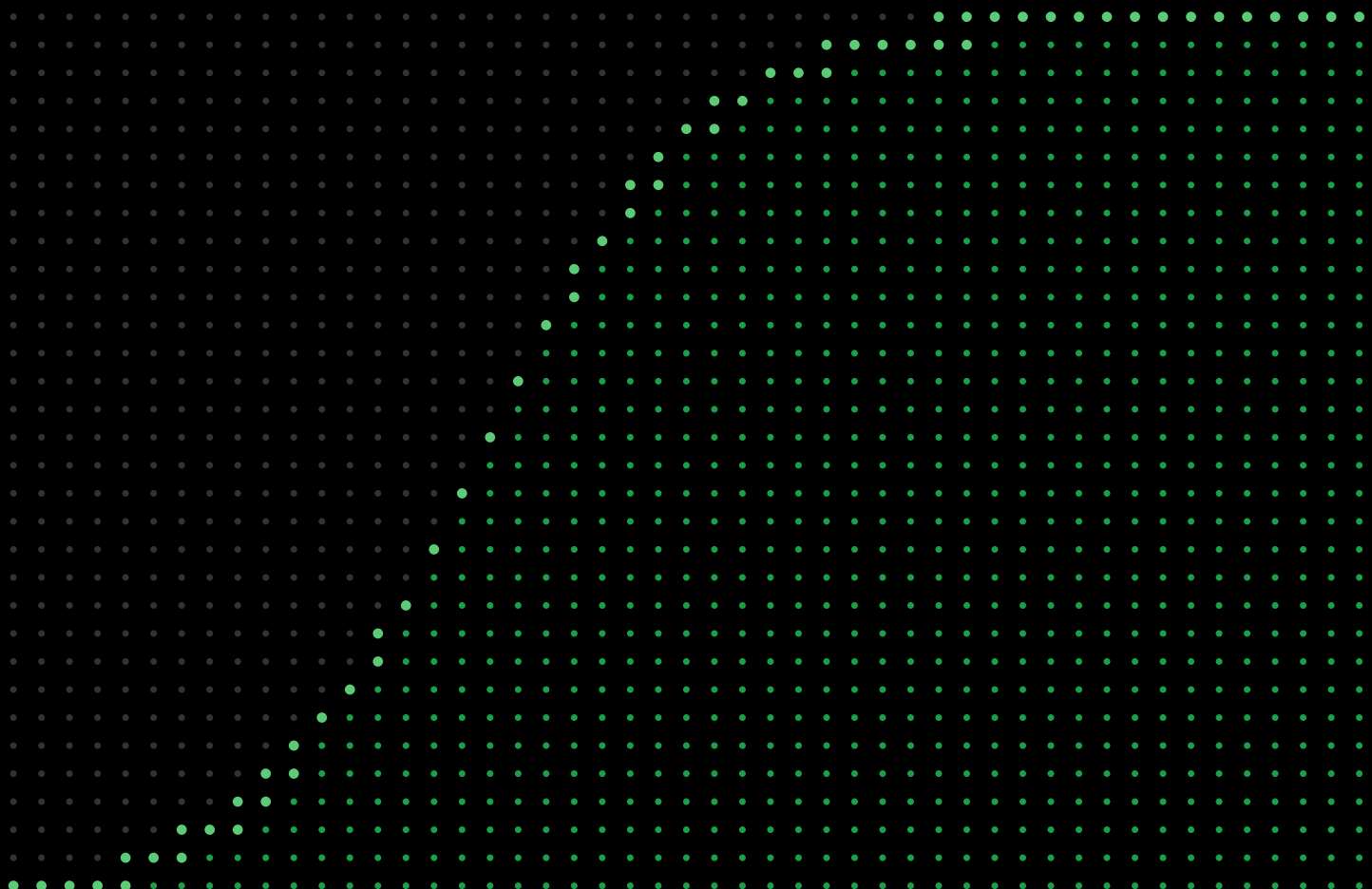
Only 12% ARE EXTREMELY CONFIDENT IN THE METRICS.



A benchmark of AI adoption, productivity, and measurement from 427 responses from senior engineering leaders.

augment code

CONTENTS



We asked 427 senior engineering leaders about AI:

Their adoption of AI tools, budget plans, how AI was impacting team productivity, and whether they were able to confidently measure their results.

Engineering leaders expected more from AI than their organizations have realized. 49% believe effective adoption could produce gains of 51% or more. Only 23% report gains at that level today.

The tools are widely available. The promised productivity has not arrived at the same scale.

81% report widespread adoption or beyond, and 77% expect spending to increase over the next six months.

The budget has arrived. Now leaders have to explain what it returned.

Most AI rollouts began one engineer at a time: buy seats, encourage experimentation, and let teams find what works. That produced local speed, but it did not redesign review queues, testing, release processes, or the metrics used to judge the system. Adoption scaled faster than the operating model around it.

The result is the productivity plateau: AI speeds up parts of engineering work, but much of that gain is lost before it becomes organization-wide output. 91% feel AI has improved productivity, yet only 45% report gains spreading across the team. Perceived acceleration is common. Sustained delivery gains are not.

Teams further along the adoption curve report stronger results. 71% of advanced teams report proven value, compared with 47% of teams at the widespread stage, where use is broad but less consistent. The data shows an association: coordinated workflows are more likely to produce organization-wide gains than isolated tool use.

Investment is continuing before measurement catches up. Even among respondents who explicitly report a value plateau, 88% plan to increase AI spending. The next step is to establish a baseline, measure whether gains survive review and production, and decide what evidence will justify expanding or stopping a rollout.

BY THE NUMBERS

EXPECTATION VS. REALITY

49% vs 23%

Believe adoption could deliver 51%+ gains vs actual gains today

THE PLATEAU

91% vs 45%

say AI made teams more productive vs actual widespread gains.

ADOPTION & SPEND

81%

Report widespread adoption or beyond.

77%

Expect spend to increase over the next six months.

ADVANCED VS. WIDESPREAD

71% vs 47%

Advanced teams report proven higher value versus widespread-stage teams.

STILL SPENDING

88%

Plan to increase AI spending even while reporting a value plateau.

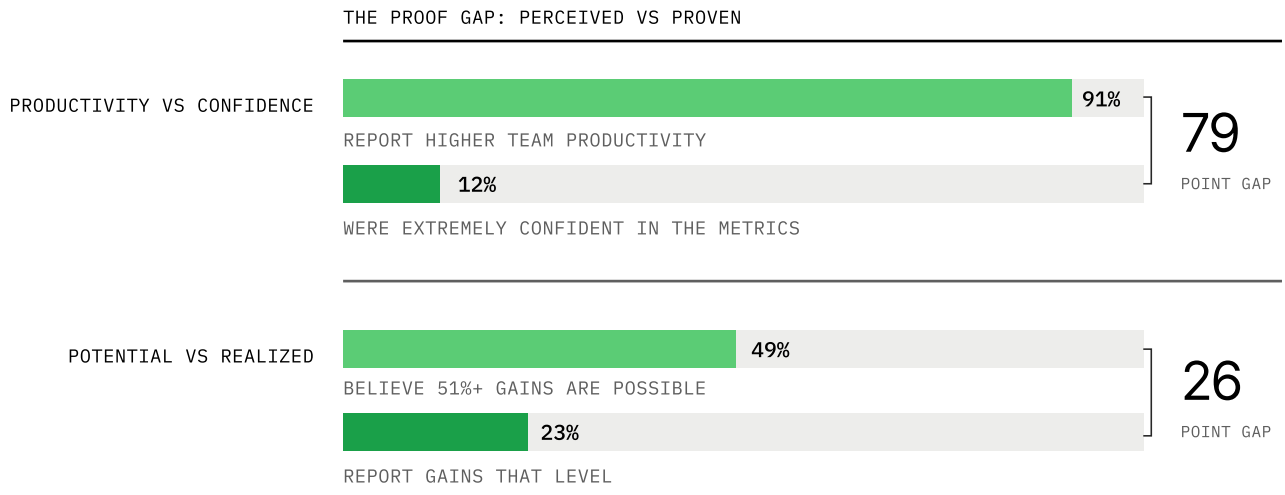
The question has shifted from how AI changes the job to how leaders prove its value.

Our spring report focused on how AI was changing engineering work. **Nearly half of respondents' code was AI-generated, but only 32 of 219 leaders had formally changed** how they measured productivity.

This survey asks whether that adoption produced the gains leaders expected. The answer is: not yet, and not consistently. Trust remains unresolved: 39% raised confidence concerns in the spring, and 42% now say errors and hallucinations create too much correction and review. Measurement is part of the problem, but it is not the whole problem. Teams also report that gains remain isolated, review becomes a bottleneck, and quality can decline. The question has shifted from how AI changes the job to why its local gains have not translated into the expected organizational return.

The productivity plateau in numbers

AI adoption is widespread, and investment continues to rise. The harder question is whether teams can turn that activity into measurable business value.



81%

have reached widespread AI adoption or beyond

AI is no longer limited to isolated experiments. Most organizations report widespread, advanced, or cutting-edge use.

77%

Expect AI spending to increase

Investment continues even as teams struggle to measure the return.

28%

Report a plateau, unclear impact, or pivot

More than one in four teams have yet to translate adoption into sustained, measurable value.

Where the productivity plateau shows up

The plateau is not a claim that AI stopped working. It is the gap between engineers feeling faster and organizations being able to show the gain in team throughput, quality, and financial results.

[1]

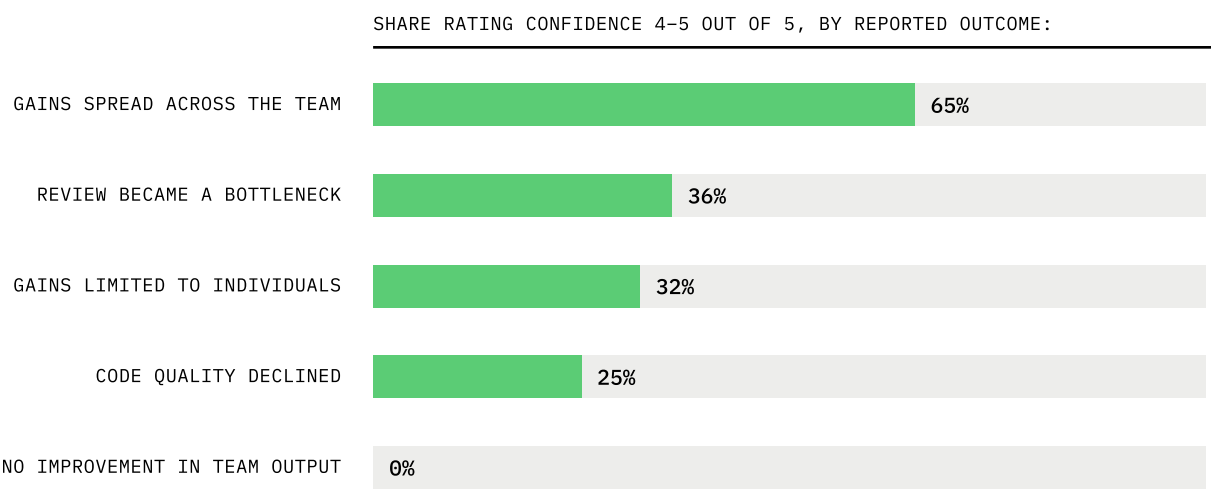
The bottleneck moved downstream

AI often speeds up code production first. When the rest of the delivery system stays the same, work piles up in context gathering, review, testing, integration, or deployment. A team can produce more code without shipping more value.

[2]

Team-level gains are easier to prove.

Faster coding alone does not create a credible ROI story. Leaders report greater confidence when AI gains show up across the team and persist through review and production.



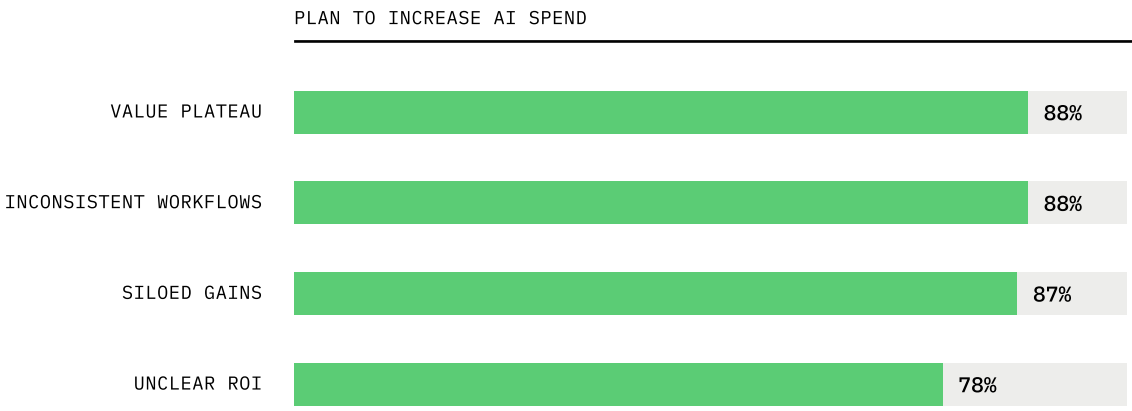
Nearly two-thirds of teams reporting widespread gains rated their confidence at 4 or 5. When gains remained limited to individuals or review became a bottleneck, only about one-third did. Confidence fell to 25% when code quality declined and to 0% when AI produced no measurable improvement in team output.

Individual speed is easy to feel; system-level delivery is what leaders can defend. Gains have to carry through review, testing, and production without adding rework, defects, or new bottlenecks.

[3]

Companies are spending through the plateau

Uncertainty is not slowing investment. Planned spending increases remain high among respondents reporting the clearest barriers:



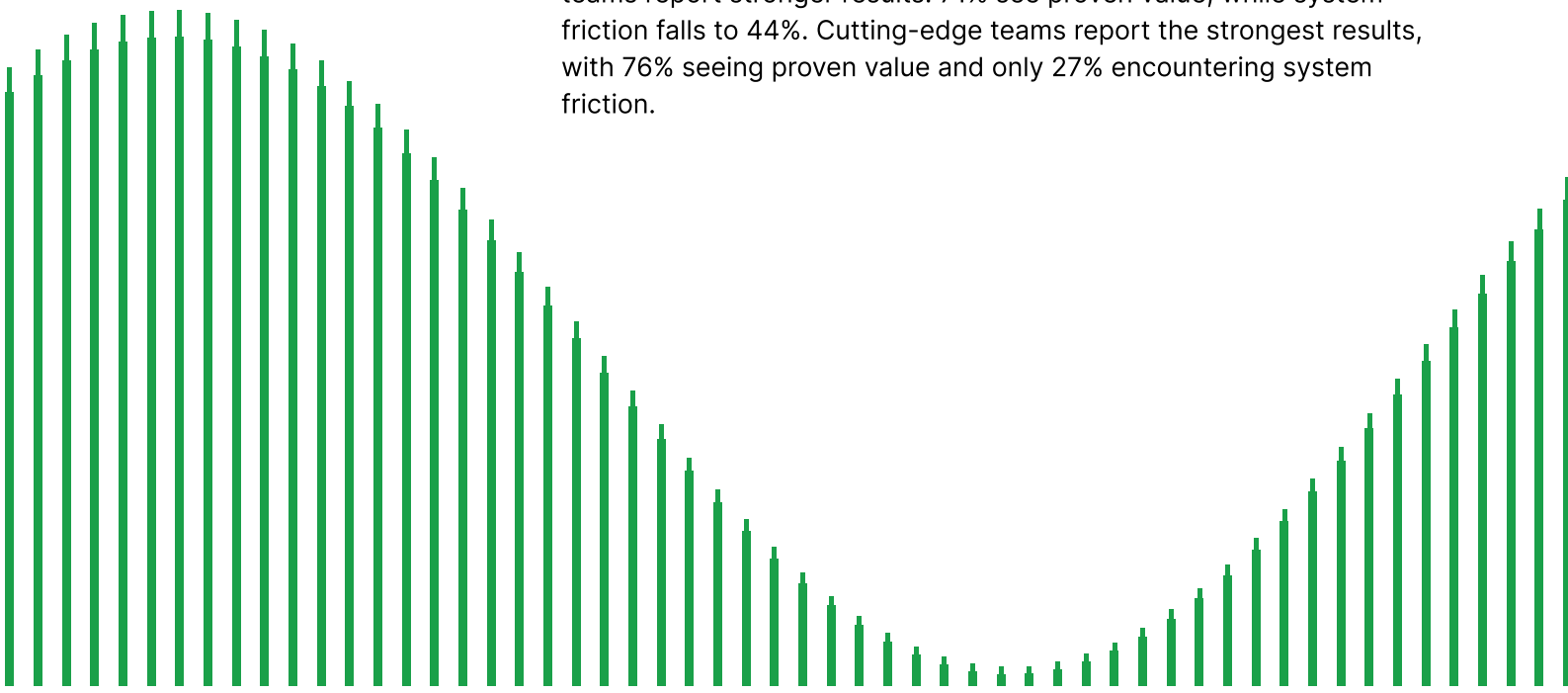
Companies are not waiting for perfect evidence before expanding AI. That raises the cost of weak measurement: each budget cycle commits more money before the last one has been fully explained.

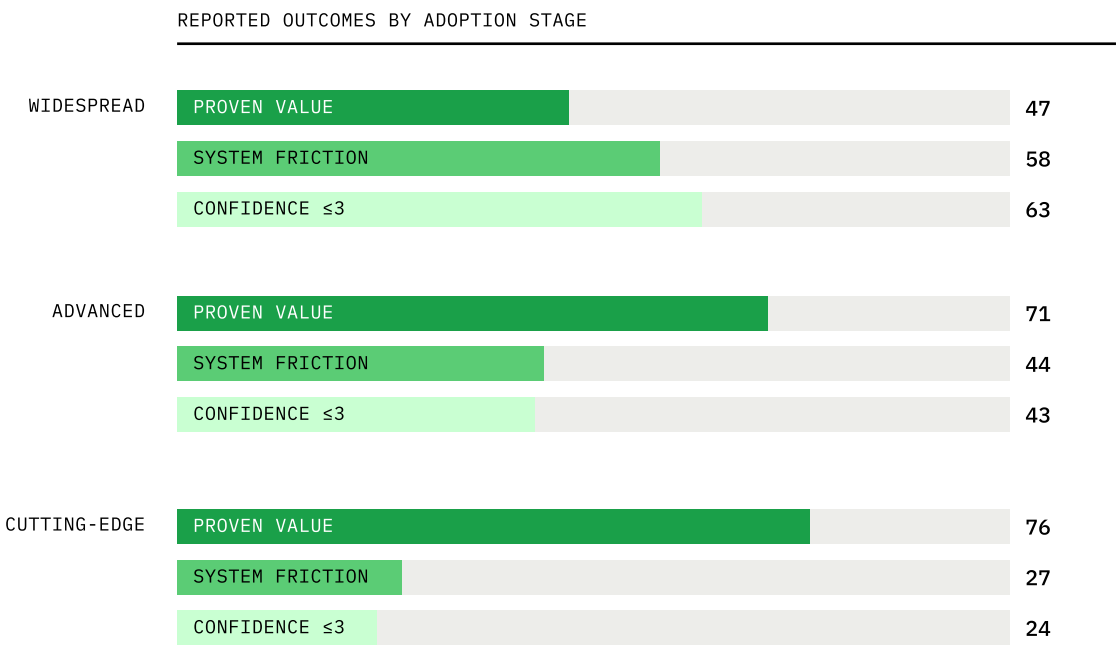
[4]

Widespread adoption is the hardest stage

At the widespread stage, AI creates enough additional output to stress the delivery system, but teams have not yet standardized how that work moves through it. More changes enter the system through different tools and practices, while review, testing, and release processes remain largely unchanged.

Among teams with widespread AI use, 63% rate confidence in their metrics at 3 or below and 58% encounter system friction. Advanced teams report stronger results: 71% see proven value, while system friction falls to 44%. Cutting-edge teams report the strongest results, with 76% seeing proven value and only 27% encountering system friction.





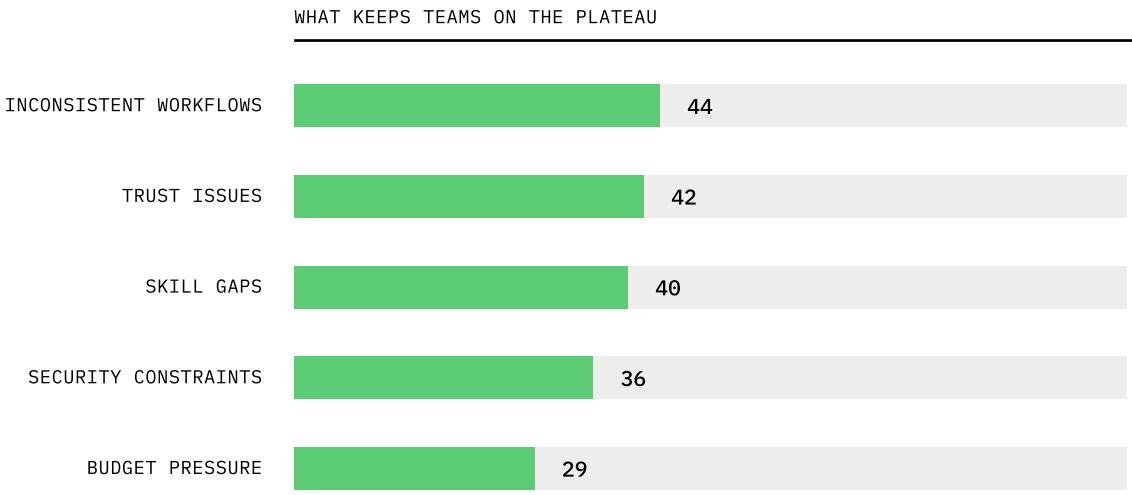
These results show an association, not proof that workflow maturity caused the improvement. The pattern is still useful: teams report more measurable value when AI is part of a coordinated delivery system rather than a collection of individual tools and shortcuts. Progress requires redesigning how work moves from context and implementation through review, testing, and production.

Organization size adds another layer. Among teams with 50 or fewer engineers, 96% report higher productivity and 62% report proven value. At organizations with 501 or more engineers, 78% report higher productivity, but only 30% report proven value. Another 39% describe unclear impact, a plateau, or a strategic pivot.

Small teams can see faster work quickly. At enterprise scale, the gain must survive more repositories, handoffs, reviewers, policies, and dependencies. The challenge is coordinating the system around the additional output.

What gets in the way of scaling AI productivity

We asked respondents to select all the obstacles their teams faced when trying to turn AI adoption into sustained productivity gains. Because respondents could choose more than one answer, the percentages do not add to 100%.



The most common obstacle was inconsistent workflows: 44% said AI adoption was growing, but practices varied across the team. 42% cited trust issues, meaning errors or hallucinations created too much correction and review. 40% reported skill gaps that limited effective use of AI tools.

Other obstacles included security constraints, selected by 36% of respondents, and budget pressure, selected by 29%. Security constraints covered data privacy, legal, or compliance policies that limited tool use. Budget pressure meant tooling expenses were outpacing tangible business value.

These results point to a systems problem. Scaling AI productivity requires shared workflows, reliable review and quality controls, appropriate security guardrails, and the skills to use AI consistently across the SDLC.

> “AI IS TRYING TO SABOTAGE OUR CODE QUALITY.”
>
> - ONE RESPONDENT

Five write-in responses mentioned measurement design, review and CI/CD overhead, internal review policies, non-engineers submitting merge requests, and concern about AI-generated code quality.

How to measure ROI: a working guide

The baseline is easiest to capture before rollout and hardest to reconstruct once AI becomes part of everyday work. Without one, leaders can see more activity but cannot determine whether delivery improved. The first measurement decision should happen before the next procurement decision.

Half of respondents rate confidence in their metrics at 3 or below. The following framework draws partly from Pearl Technologies' Cosmos deployment.

AUGMENT CUSTOMER EXAMPLES [CUSTOMER-REPORTED COSMOS RESULTS, NOT CONTROLLED COMPARISONS]

	MEASURED	RESULT
Pearl Technologies	COMPLEX-CHANGE CYCLE TIME; PRS SHIPPED	Multi-day migrations became same-day work; 728 PRs shipped in 13 working days from a stalled baseline
A memory-hardware engineering team	TIME TO MERGE, AGENT-HANDLED VS. MANUAL	About 48 hours manually vs. 21 hours agent-handled; about 7.9 hours on the review path
A maritime-software team	MERGE-REQUEST TIME TO MERGE	22 days fell to 4 days; 38% MR adoption on a shared repository
An enterprise platform team	SINGLE-FEATURE BUILD TIME	A 60-hour feature took 15–16 hours; internal fixes cost $2–$3 each
A govtech engineering team	LONG-STANDING BUG; EPIC-GENERATION COST	A nine-month-old bug was fixed in about one hour for $35; a full epic was generated for about $7
A security-software support organization	SUPPORT LOAD ABSORBED WITHOUT HEADCOUNT	About 90% of support tickets were AI-handled; engineering escalations fell about 50%; ticket volume rose 20% year over year with no added headcount

A six-step framework

[1]

Establish a baseline

Record PR throughput, review latency, cycle time, rework, and incidents before rollout. When adoption has already started, reconstruct the baseline from repository history. Compare similar work over complete time windows. A number without a baseline is an anecdote.

[2]

Measure delivery, not tool activity

Seats, prompts, and tokens show use. Measure changes shipped per engineer, complexity-adjusted output, cycle time, and roadmap work completed. A useful test is whether the metric would still matter if AI were a new hire rather than a tool.

[3]

Track trends across comparable cohorts

Measure weekly after rollout. Compare similar repositories, work types, and engineer cohorts. Pearl's PRs per agent session rose from 3.2 to 4.7 over three weeks, a 47% gain after the initial adoption jump. Rising usage with flat output per session signals a plateau, not a return.

[4]

Pair speed with quality

Track rework, reverts, review burden, escaped defects, and incidents. Faster output that creates more downstream work is not a productivity gain. Pearl audited 21 agent-driven episodes: 71% succeeded fully, 29% were partial, and none caused downstream breakages.

[5]

Count full costs, translate to money

Translate saved hours using loaded labor cost. Pearl estimated 73-113 senior-developer hours saved across those 21 episodes by comparing agent time with a conservative estimate for the same work. Subtract licenses, compute, enablement, review, rework, and platform labor. Keep capacity created separate from costs actually avoided.

[6]

Connect results to business outcomes

Track roadmap delivery, support load, reliability, onboarding time, and revenue or cost effects. Define the evidence required to expand or stop the pilot before it begins.

A decision matrix for engineering leaders

WHAT YOU OBSERVE	LIKELY ISSUE	WHAT TO DO NEXT
High usage, isolated gains	Fragmented workflows	Standardize two or three high-value workflows
Faster coding, slower reviews	Bottleneck displacement	Measure review load and change size; add test and review capacity.
Output rises, confidence stays low	Weak attribution	Establish a baseline and comparison cohort
Rework or defects rise	False productivity	Pause expansion for that use case; fix the quality loop
Spend rises, impact unclear	Unbounded rollout	Set a time-boxed pilot with expansion and stop thresholds

Let an agent run the measurement for you

This work sounds straightforward until it spans hundreds of engineers and repositories. Establishing comparable cohorts, reconstructing baselines, and following changes through review and deployment is difficult to maintain manually.

Cosmos Analyst runs that analysis against GitHub or GitLab history and reports throughput, cycle time, revert rate, and Cosmos spend. It is read-only. The analysis shows repository activity and association with adoption; it does not declare ROI or establish causal business value on its own.

READ THE SETUP GUIDE →

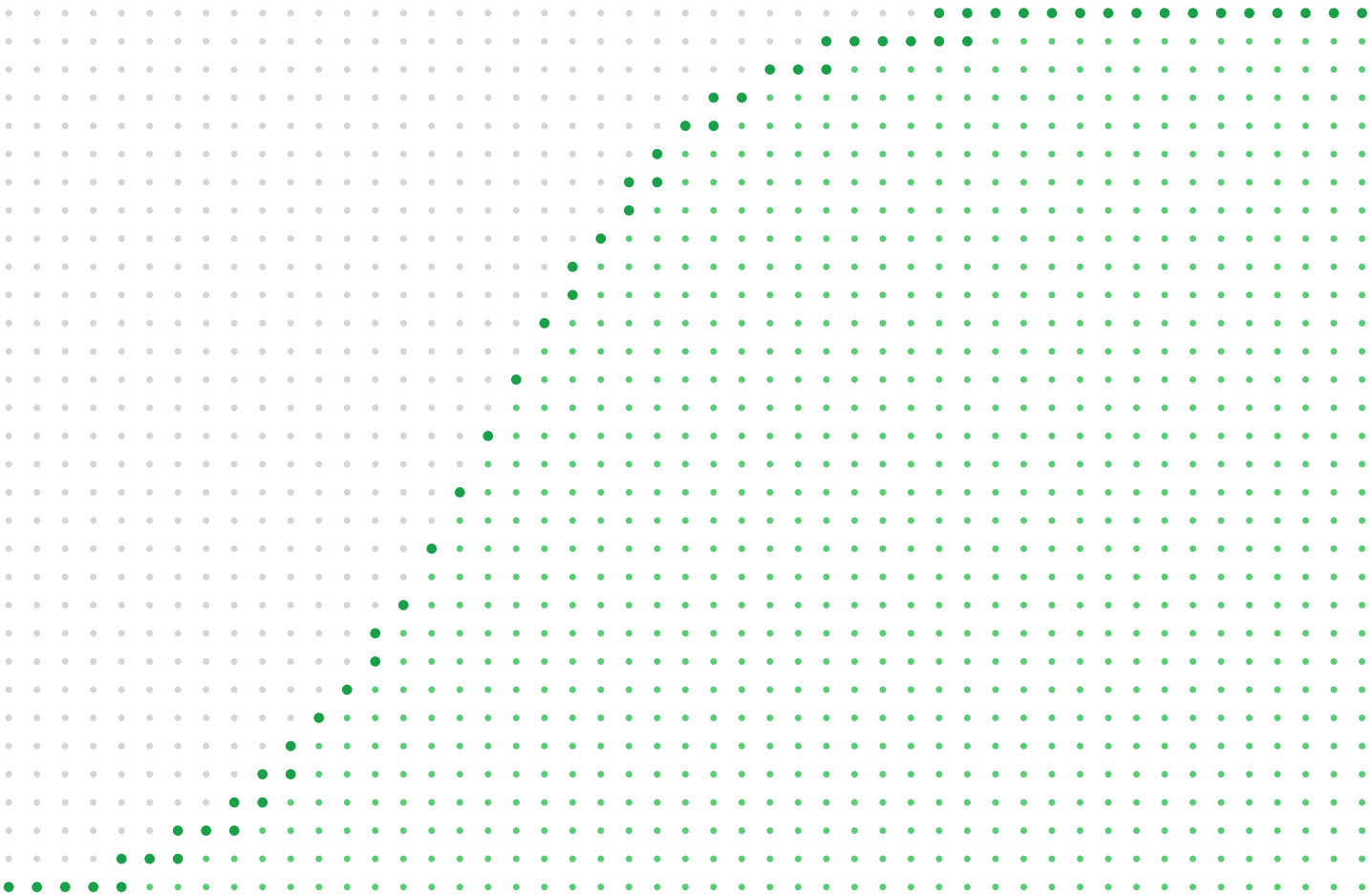
BOOK A DEMO →

Getting off the plateau is a systems problem.

Giving individual engineers better tools can increase local speed, but it cannot fix the bottlenecks that appear across review, testing, integration, and deployment.

Teams need a system that coordinates AI-assisted work across the SDLC: shared workflows, reliable context, quality controls, and metrics that follow work from implementation through production.

Individual gains become sustained delivery only when the system around them is redesigned.



Engineering output accelerated after Cosmos adoption.

ACME PAYMENTS · JAN 5 – MAY 3 2026 · COSMOS ADOPTED MAR 2 2026

ILLUSTRATIVE REPORT, NOT BASED ON ACME, AUGMENT, OR ANY CUSTOMER'S ENGINEERING OR BILLING DATA.

CHANGE AFTER COSMOS ADOPTION [% SHIFT IN BASELINE VS FINAL]

+63%

SIZE-ADJUSTED OUTPUT / ENG / WK

2.7 → **4.4**

+64%

LINES CHANGED / ENG / WK

890 → **1,460**

+46%

MERGED CHANGES / ENG / WK

2.8 → **4.1**

−41%

MEDIAN TIME TO MERGE

31.4h → **18.6h**

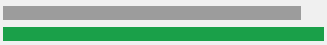
$18.1K

TOTAL COSMOS SPEND

$0 → **$18.1K**

+0.9

ACTIVE ENGINEERS



11.6 → **12.5**

PRE-COSMOS BASELINE FINAL RESULT

Engineering output rose materially after Cosmos adoption. Size-adjusted output increased 63% per active engineer, while the raw number of merged changes rose 46%. The difference suggests that the gain came from more substantial shipped work rather than splitting work into smaller changes. Median time to merge fell 41%, rework remained broadly stable, and tenant-wide Cosmos spend was $18.1K across the analysis window.

Interpretation

The size-adjusted output measure rose faster than the raw change count, so the result is not explained by splitting work into more, smaller changes. Delivery also accelerated: median time to merge fell from 31.4 hours to 18.6 hours. The rework rate increased by 0.3 percentage points, which is broadly stable but should continue to be monitored as the full 14-day quality window matures.

The report does not treat lines changed as a productivity result. It uses that measure only as a diagnostic because generated and vendored code can inflate it. Cost is shown beside delivery and quality so leaders can evaluate the full claim rather than looking at adoption or output alone.

Weekly trend findings

- Merged changes and size-adjusted output both rose after the 2 March adoption date.
- Median time to merge declined throughout the post-adoption period.
- Rework stayed within its prior range; the final 2 weeks remained an early read because their 14-day observation windows were incomplete.
- Weekly Cosmos spend increased as adoption grew, reaching $18.1K across the full reporting window.

Caveats

- Complete weeks: The window ended on a Monday, so no trailing partial week was included.
- Team size: Active-engineer counts provide context for the rates; they do not discount the per-engineer gain.
- Reverts: The rate is a lower bound because title matching misses some reverts. The final 2 weeks had not completed their full 14-day observation period.
- Cost scope: Cost covers tenant-wide Cosmos usage. CLI, IDE, and other product costs are excluded.
- Causality: The analysis shows an association between Cosmos adoption and repository outcomes. It does not prove that Cosmos alone caused the change.

Methodology and definitions

ITEM	DEFINITION
REPORTING WINDOW	5 January–3 May 2026, grouped into Monday-starting weeks.
ADOPTION DATE	2 March 2026, supplied by the customer.
REPOSITORIES	`acme/payments`, `acme/platform`, and `acme/checkout`.
COHORT	14 engineers who appeared as an author or merger during the window.
RECONCILIATION	226 merged changes: 214 attributed, 9 CI or infrastructure-bot changes excluded, and 3 agent-authored changes without a human merger excluded.
ATTRIBUTION	Human-authored changes are credited to the author. Agent-authored changes are credited to the human merger because merging is treated as shipping. CI and infrastructure bots are excluded.
NORMALIZATION	Weekly rates use active engineers, defined as people with at least one attributed change that week. Zero-change weeks are excluded so leave and inactivity do not dilute output.
BASELINE AND FINAL PERIODS	The baseline is the full pre-Cosmos period. The final period is the last 2 complete weeks. Per-engineer metrics use engineer-weighted pooling; time to merge uses the combined-window median.
MERGED CHANGES	Attributed merged changes divided by active engineers for that week.
SIZE-ADJUSTED OUTPUT	`min(changed files, 18) x log(1 + additions + deletions)` divided by the window-wide average complexity of 33.7. The 18-file cap is the dataset's 95th percentile and prevents a single mass refactor from dominating.
TIME TO MERGE	Median hours between creation and merge for changes merged that week.
REWORK OR REVERT RATE	Share of a week's changes reverted within 14 days, detected from platform-specific revert titles and mapped to the original merge week.
LINES CHANGED	Total additions plus deletions divided by active engineers; diagnostic only.
COST	Tenant-wide Cosmos-only billed amount in USD over the same reporting window. Weekly and total costs were fully paginated before aggregation.