State of AI Impact in Engineering: Q2 Report

Published July 2026 | Reporting period: April - June 2026

REPORT

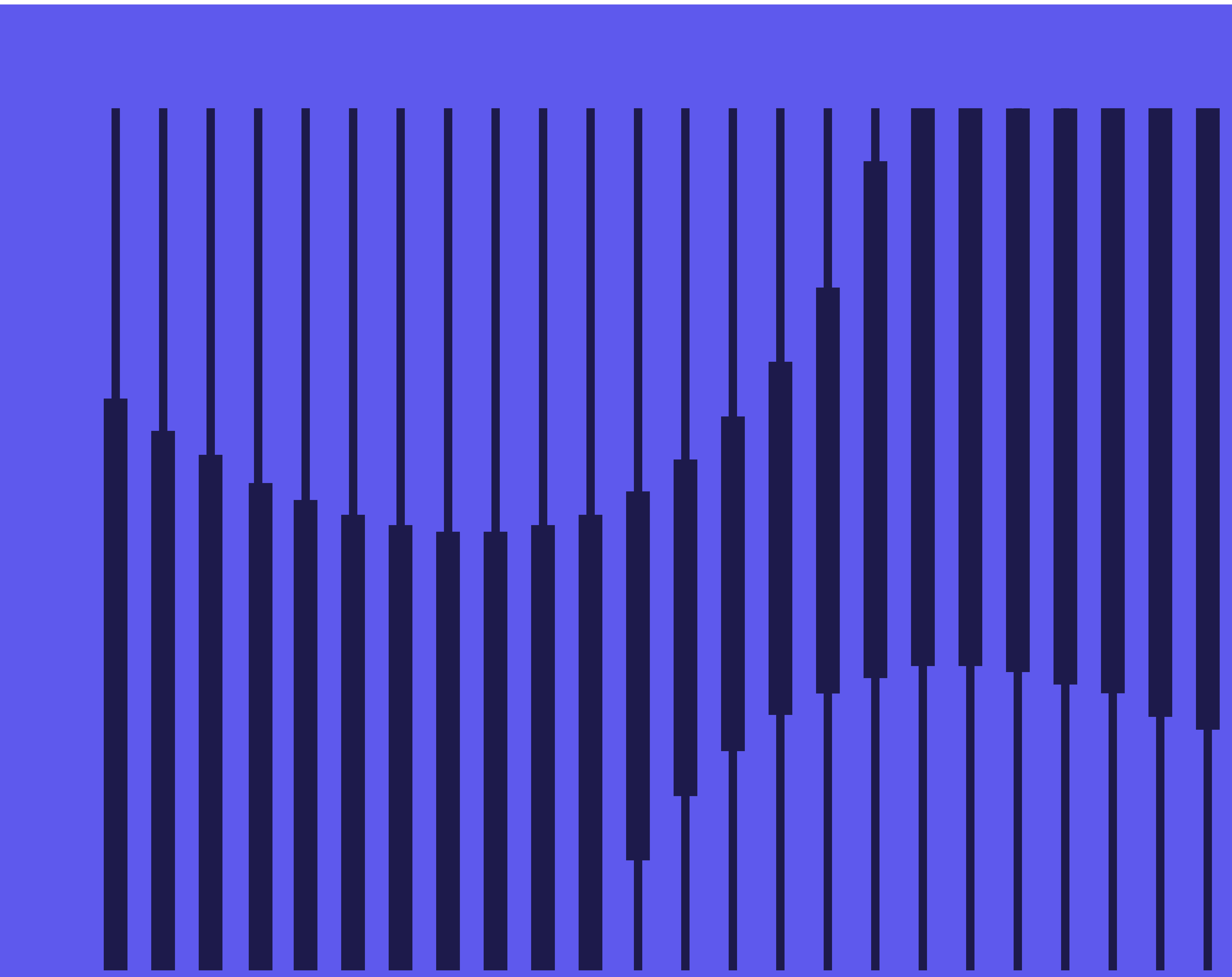


Table of contents

State of AI Impact in Engineering: Q2 Report

DX Research Team

Executive summary

Engineering organizations have entered a critical transition phase, shifting their focus from measuring baseline AI adoption to evaluating concrete return on investment. With AI utilization now above 90% across the industry, the question engineering leaders face is no longer "are we using AI?" but "is it paying off?"

This quarter's data shows a system in tension: code output is climbing, cost is climbing faster, and the human experience of building software has not kept pace with either. While AI tools are accelerating individual output, the surge in AI-authored code is testing the limits of existing infrastructure and human review gates.

This report measures 500+ organizations against the DX Core 4 (Speed, Effectiveness, Quality, Impact) and the DX AI Measurement Framework (Utilization, Impact, Cost). Four distinct themes have arisen from the data this quarter:

1. **Speed gains are real but uneven**

- Median weekly TrueThroughput rose 37% over four quarters (1.42 to 1.94 PRs/eng/week), and deployment frequency is up double digits across most segments.
- Gains are concentrated among small organizations (less than 100 engineers) and tech-sector teams, pulling away from larger and traditional-industry teams, and the gap is widening.
- Despite the throughput gains, developers' perceived rate of delivery has been flat for a full year. Pipeline acceleration and developer-perceived speed have seemingly decoupled.

2. AI is improving some aspects of developers’ work while creating new bottlenecks in others

- Documentation quality, code maintainability, and production debugging all improved, and represent the clearest, least-contested wins in this report.
- Incremental delivery, local iteration speed, and review turnaround all declined, and the average Developer Experience Index (DXI) slipped from 67 to 65. PR size nearly doubled over the same period, consistent with AI-driven inflation rather than disciplined, testable code.
- AI is making individual tasks faster, but the surrounding system is absorbing the slack rather than converting it into new value. This is the same "AI efficiency paradox" now being documented industry-wide: individual output surges while human gates become the new bottleneck.

3. Code is easier to read but harder to trust

- Code maintainability improved, but change confidence fell in the same period. These metrics are historically correlated, but are now in tension. AI can help engineers understand and change the code in front of them, but faith in the output and trust in the code is slipping.
- Change failure rate volatility increased, with many organizations now reaching +/-3 percentage points against a 4% industry benchmark. AI has not created this volatility, but it has amplified it.
- Perceived software quality tells a counterintuitive story: Traditional industries and Financial Services, the slowest-moving segments on throughput, report the highest perceived quality. Speed and perceived quality are not moving together, and leaders should not assume they will.

4. Spend is outrunning the return it's meant to justify

- Median quarterly AI spend grew from ~$1.5K to ~$44K in a year, roughly a 28x increase in the Tech sector alone, while the innovation ratio (time spent on creating new features vs. maintenance) has stayed essentially flat over the same four quarters.
- Time savings are real and growing, now over 6 hours a week on average, but saved hours are not yet visibly converting into new value creation at the portfolio level.
- Smaller organizations pay more per seat for AI tools, since they lack the negotiating leverage larger companies get from volume licensing. Yet they're getting more output per dollar spent than larger companies. This challenges the common assumption that scaling up AI spend at the enterprise level will automatically improve ROI. The data suggests the opposite: bigger AI budgets don't guarantee better returns, and may even come with diminishing ones.

The bottom line for leaders

AI is unambiguously accelerating code-writing but is not yet translating into faster felt delivery, higher innovation capacity, or spend that has proven its return. The organizations best positioned going into the remainder of the year are treating AI adoption as the starting line, not the finish line. Leaders should pair every throughput or spend metric with a quality and experience counterweight (change confidence, DXI, innovation ratio, etc.), and invest as much in the surrounding environment and platform as in the AI tooling itself.

Introduction: The ubiquity era of AI

Welcome to our Q2 AI impact report, the third edition of our quarterly AI industry analysis. When we began this project, the primary goal was to study and publish what happens to a team's output after they adopt generative AI. For the past two quarters, answering that question meant measuring AI cohorts against historical baselines, representing engineering performance pre- and post-AI adoption.

With industry-wide adoption hovering well above 90%, we've reached an inflection point where we can no longer treat AI utilization as a binary condition, and as a result, this structure for comparison is no longer viable. Because utilization is essentially ubiquitous, we lack a control group, so comparing impacts across cohorts tells us very little now.

This quarter, we are uncovering entirely new insights by shifting our lens toward deeper firmographic and demographic segmentation, measuring exactly how AI impact is shifting across different organization sizes, geographies, and industry verticals. Additionally, we are able to report on a more even blend of qualitative insights and system metrics than in previous reports. We have also updated the format of this report to better represent how AI augments various aspects of the SDLC.

As AI becomes infrastructure rather than experiment, engineering leaders are under pressure to justify massive AI budgets. The instinct is to reach for new metrics like token counts and lines of AI code, but new metrics only indicate how work is changing, not whether it's producing better outcomes. A 90% AI adoption rate means nothing if delivery stalls or system stability crashes.

This is why outcome-oriented measurement frameworks matter more, not less, in the age of AI. While AI represents a shift in how software is built, it does not alter what engineering organizations are ultimately trying to accomplish. The questions engineering leaders need to answer remain the same: How quickly is value delivered? How easy is it for developers to do their work effectively?

Answering these questions demands measurement frameworks that are outcome-driven and grounded in peer-reviewed research, not reactive dashboards built around the latest hype cycle. The two frameworks underpinning this report, the DX Core 4 and AI Measurement Framework, were designed to meet those standards.

PART 1

DX Core 4

Speed | Effectiveness | Quality | Impact

The DX Core 4 focuses our evaluation on the structural pillars of engineering health: Speed, Effectiveness, Impact, and Quality. This ensures we aren't just looking at isolated coding habits, but instead validating whether AI investments are driving gains in organizational productivity.

DX Core 4

	Speed	Effectiveness	Quality	Impact
Key metric	Diffs per engineer* (PRs or MRs) *Not at individual level	Developer Experience Index (DXI) Measure of key engineering performance drivers, developed by DX	Change failure rate	Percentage of time spent on new capabilities
Secondary metrics	Lead time Deployment frequency Perceived rate of delivery	Time to 10th PR Ease of delivery Regrettable attrition* *Only at organizational level	Failed deployment recovery time Perceived software quality Operational health and security metrics	Initiative progress and ROI Revenue per engineer* R&D as percentage of revenue* *Only at organizational level

Speed

Velocity continues to increase, extending a steady quarter-over-quarter growth trend observed since Q3 2025. However, the data highlights a divergence between objective system metrics and subjective developer perception. Measurable pipeline acceleration has not yet translated into developers reporting a higher perceived rate of delivery.

Furthermore, these speed improvements are distributed unevenly across market segments. Smaller organizations and Tech-industry teams are currently recording the highest throughput gains, whereas larger and Traditional-industry teams are demonstrating slower, more incremental progress.

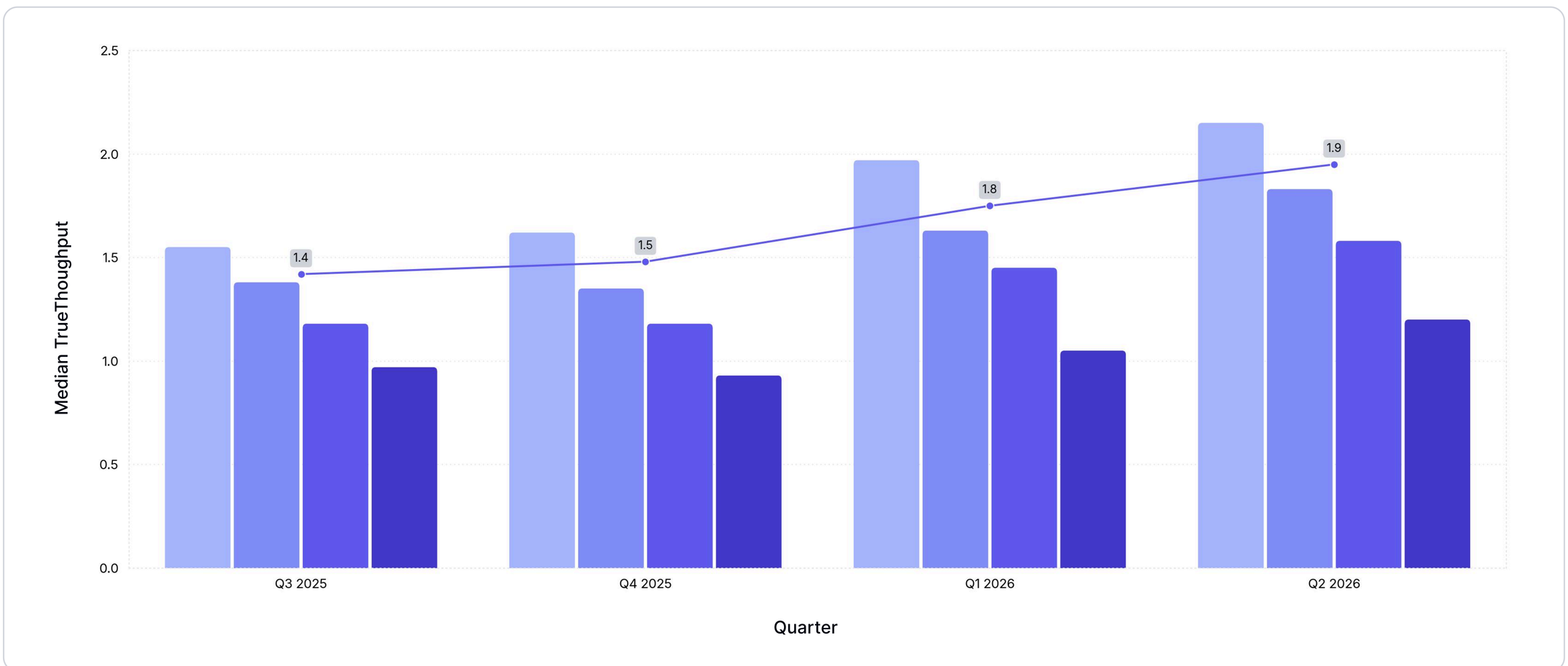
While these metrics demonstrate a clear increase in output, velocity alone does not fully indicate the total value generated by AI investments.

Primary metric: Diffs per engineer

To understand directional changes in engineering velocity, DX utilizes TrueThroughput™, which incorporates code diff complexity analysis. By weighing the resulting metric based on this complexity, it provides a much more accurate reflection of developer effort than treating all pull requests equally. TrueThroughput generally skews lower than traditionally calculated pull request (PR) throughput.

Median weekly TrueThroughput by organization size

Sample from 500+ companies from July 2025 to June 2026



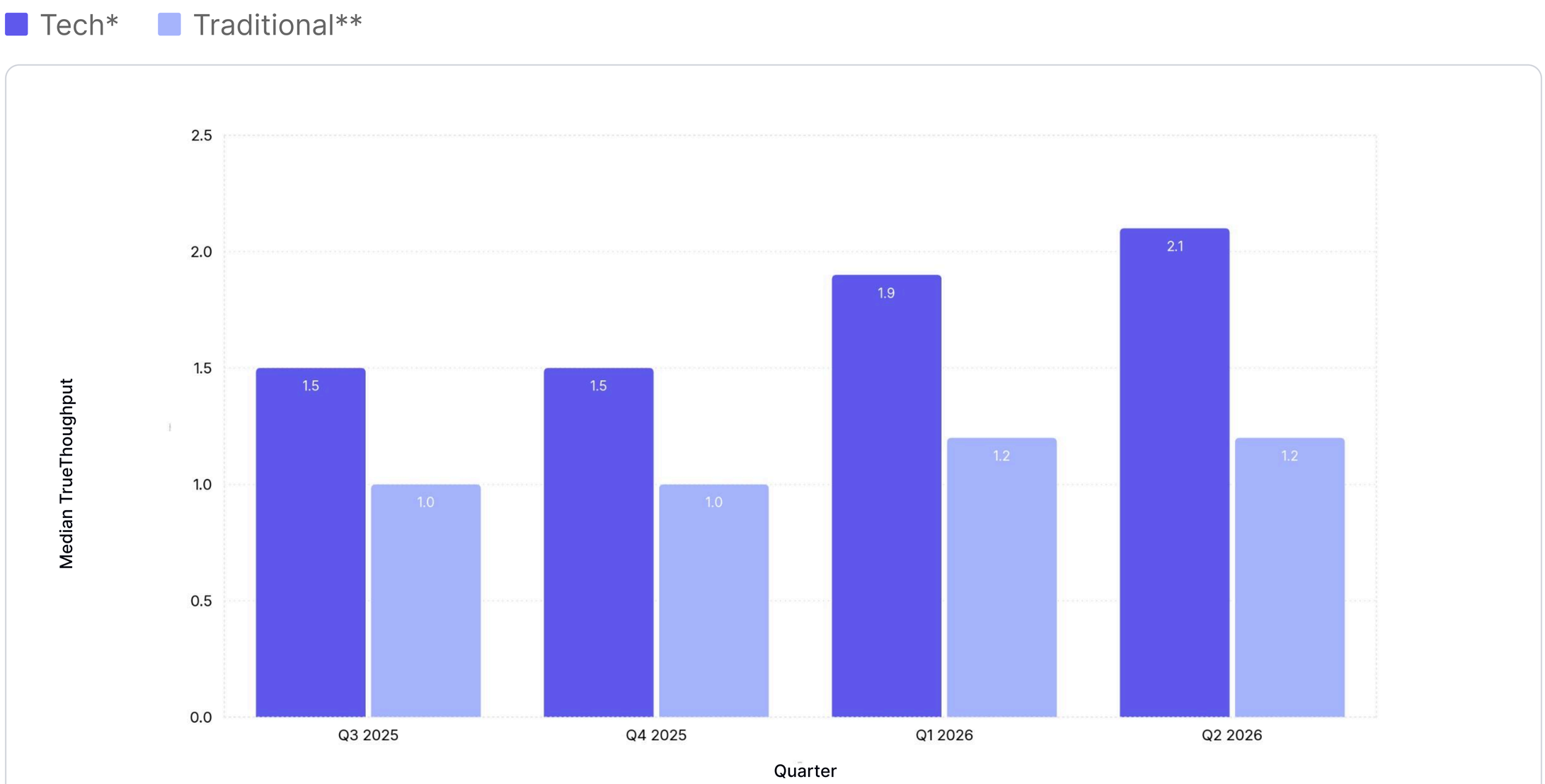
For DX users: Reports > Core 4 > PR throughput

Developers are shipping more code. Median weekly TrueThroughput increased from 1.42 to 1.94 weekly PRs per engineer (PRs/eng/week) over four quarters, spiking particularly in the last two, representing a 37% increase in output. While the absolute change appears to be small, that increase becomes meaningful at scale.

However, the gap between the smallest and largest organizations is widening, indicating that scale and process overhead continue to dampen throughput gains for larger teams. Small-to-midsize (SMB) organizations (15-99 engineers) lead the pack, hitting ~2.2 PRs/eng/week in Q2 2026, while larger environments (750+ engineers) lag at 1.2 PRs/eng/week. Medium-sized organizations (100-749 engineers) saw the sharpest acceleration between Q4 2025 and Q1 2026, pointing to a potential inflection point for mid-tier teams.

Median weekly TrueThroughput by sector

Sample from 500+ companies from July 2025 to June 2026



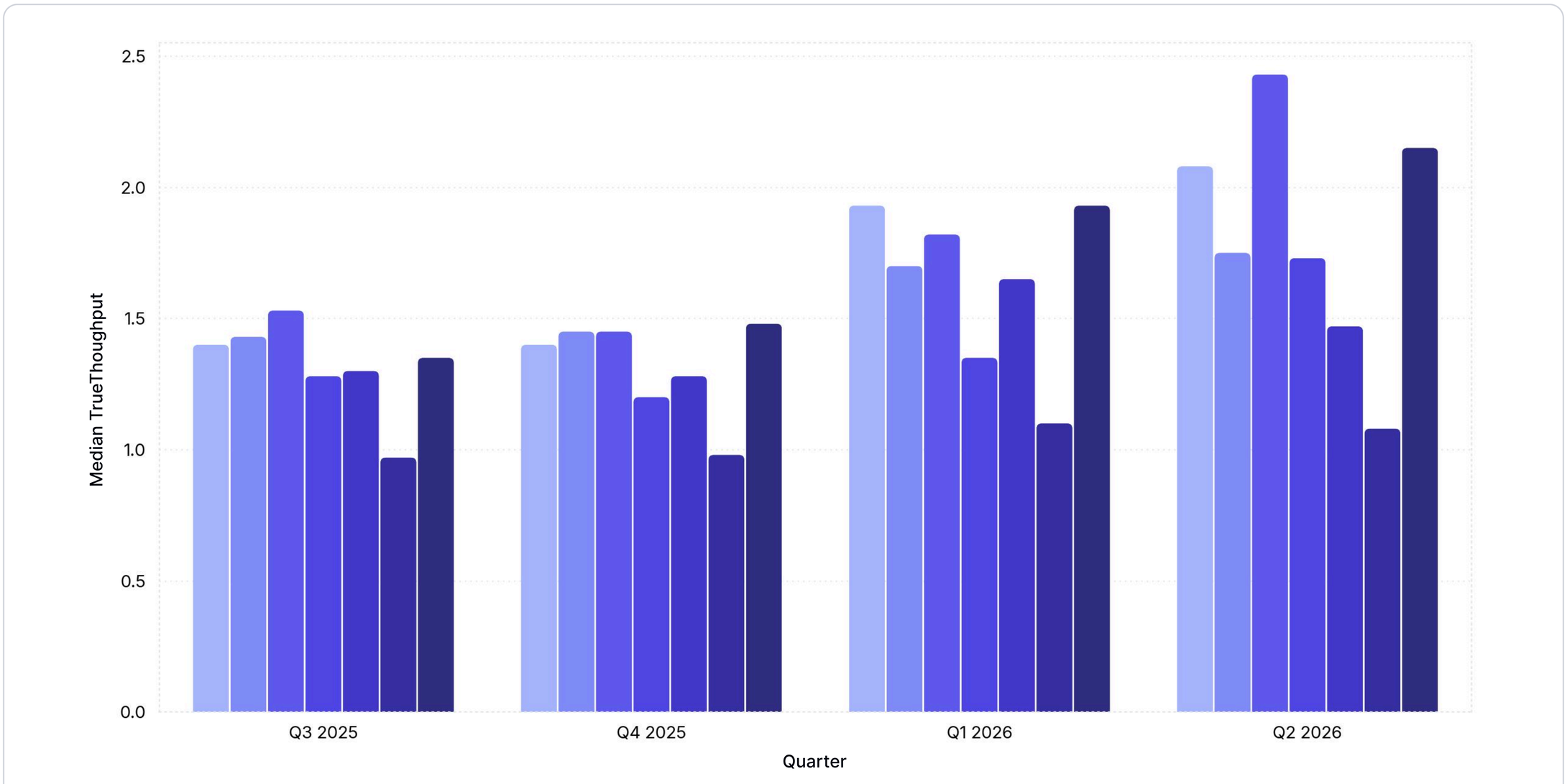
*Tech companies: Organizations whose primary business is building and selling software or technology products and services. This includes software development, IT services, cloud infrastructure, and platform companies.

**Traditional companies: Organizations in non-technology-native industries where software engineering supports the core business. Examples include Retail/E-Commerce and Manufacturing.

There are measurable differences across industry verticals. Tech companies consistently outpace Traditional industries, and the gap between them is growing. While Traditional industries are improving, their rate of progress is roughly half that of the Tech sector. Tech climbed from 1.5 to 2.1 PRs/eng/week, whereas Traditional moved from ~1.0 to 1.2 PRs/eng/week during the same timeframe.

Median weekly TrueThroughput by industry

Sample from 500+ companies from July 2025 to June 2026



Breaking this data down further by specific industry verticals highlights that while growth is broad-based across all segments, the baseline TrueThroughput highly depends on the specific operational realities of the sector. Healthcare has emerged as this quarter's throughput leader, outpacing historically fast-moving segments like Software Development & IT. Conversely, highly regulated sectors such as Financial Services continue to post the lowest overall throughput, reflecting the systemic constraints and compliance requirements inherent to their release pipelines.

What other metrics are capturing velocity in the age of AI?

PR throughput reflects how much code is moving through a system, not necessarily how fast value reaches users. We report it here as a well-established leading indicator of development activity, while necessitating teams interpret throughput gains in concert with other metrics.

Uber’s new AI-native measurement framework combines velocity and value to point towards a new north star: feature velocity.

> "PRs measure activity, features measure value." — Abhishek Tibrewal, Uber

Feature velocity – number of features shipped per engineer per unit of time – is the company's method to measuring value, not just speed, over time. It is supported by flow efficiency, quality, and capability expansion to form an agent-proof way to measure value.

Listen to Uber's DX Annual talk to learn how they measure AI impact.

Secondary metric: Deployment frequency

Median number of deploys per week

Sample from 500+ companies from July 2025 to June 2026

Deployment frequency, a primary DORA metric, has increased by double-digits across most segments, confirming the upward trajectory of actual output.

For DX users: Reports > DORA > Deploy frequency

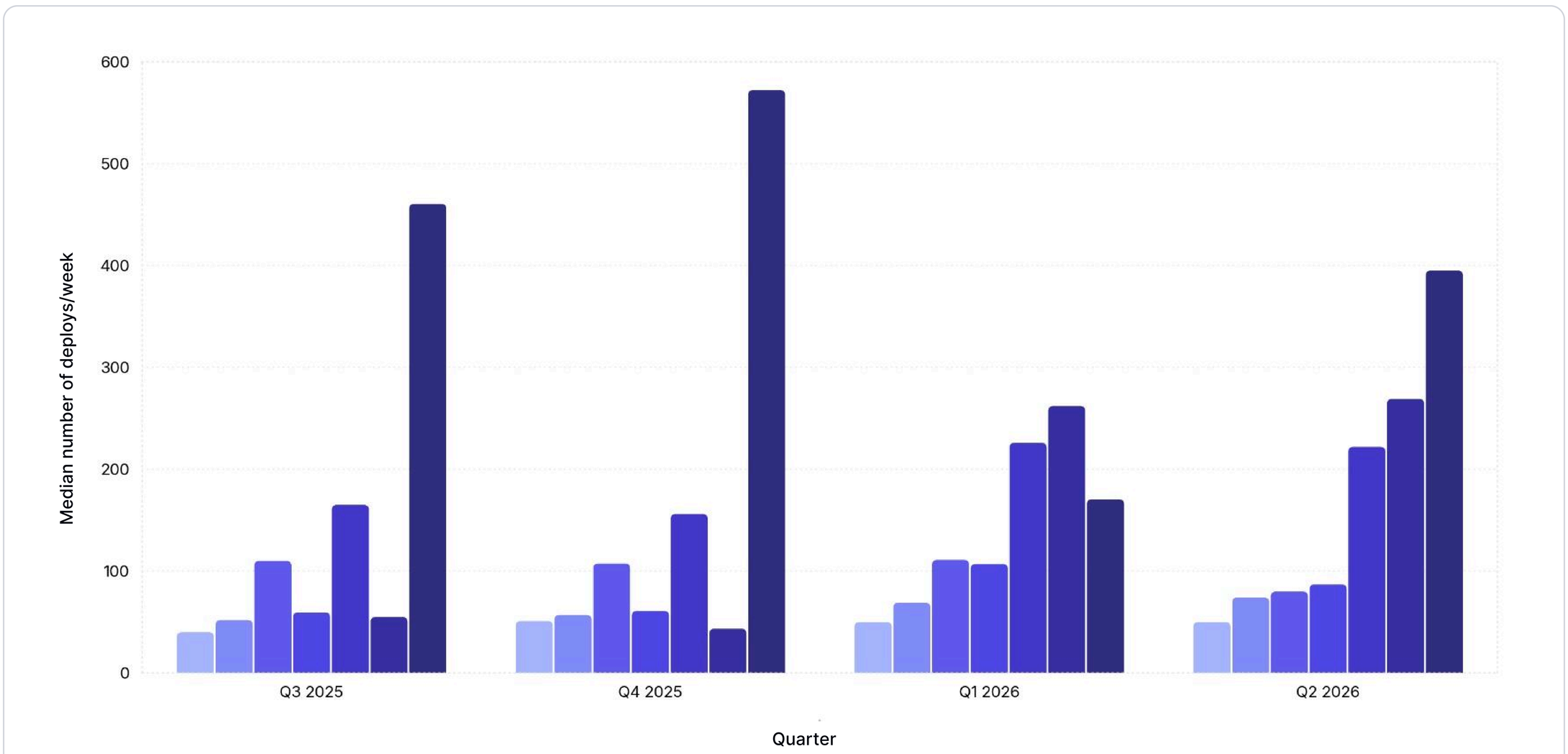
Median number of deploys per week by organization size

Sample from 500+ companies from July 2025 to June 2026

Medium-sized organizations are the clear acceleration story, sustaining outsized gains in deploy cadence across three consecutive quarters (+26% to +68% to +61% QoQ). Larger teams showed a rebound to +14% in Q2 2026, signaling potential progress in overcoming scale-related release management bottlenecks. In contrast, SMB's deploy frequency growth is modest and nearly flat in Q2 2026, suggesting these teams may have already reached their practical deployment ceiling.

Median number of deploys per week by industry

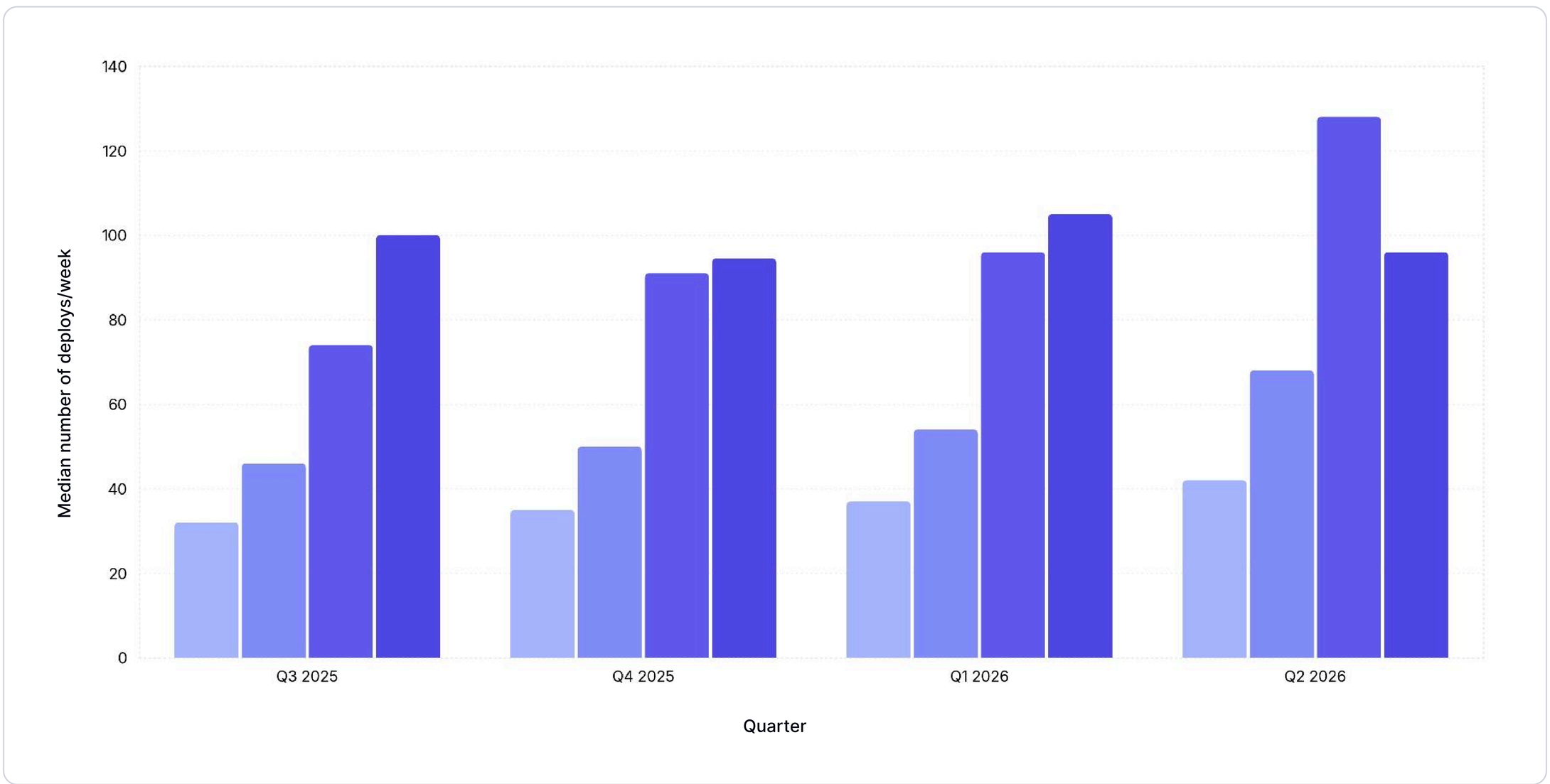
Sample from 500+ companies from July 2025 to June 2026



This industry breakdown illustrates how different verticals manage release frequency. The massive acceleration in deployment pipelines varies heavily depending on the regulatory and architectural constraints inherent to each specific sector.

Median number of deploys per week by region

Sample from 500+ companies from July 2025 to June 2026



Regional deployment metrics show a divergence from the metrics segmented by size and region, displaying acceleration only across some tracked geographic segments. Silicon Valley-based organizations exhibited the sharpest increase, showing a dramatic spike in their deployment cadence by Q2 2026. Conversely, regions such as Europe began the tracking period with a higher baseline frequency but have remained relatively flat over the last two quarters. Meanwhile, Asia-Pacific and non-Silicon Valley North American teams continue to demonstrate steady, incremental growth.

Secondary metric: Perceived rate of delivery

Despite objective gains in throughput and deployment frequency, developers' internal sense of their delivery speed has not meaningfully shifted. The median perceived rate of delivery has held remarkably steady at roughly 67% to 70% across all four quarters. Most developers cluster tightly in the 60% to 75% range, showing that measurable pipeline acceleration does not automatically translate to a frictionless developer experience. Persistent outliers on both ends of the spectrum seem to reflect individual or team-level variations in workload and blockers rather than market-wide sentiment shifts.

Median perceived high rate of delivery

Sample from 500+ companies from July 2025 to June 2026

For DX users: Reports > Efficiency > Perceived rate of delivery

Effectiveness

The primary metric used for engineering effectiveness is the Developer Experience Index (DXI), which is a weighted average of qualitative drivers which together represent a comprehensive view of the developer experience in an organization. The DXI can be used as a check against the assumption that more code output equals a better overall developer experience.

In Q2 2026, we saw a measurable drop in this score overall, explained by a cumulative drop in developer experience dimensions such as cross-team collaboration, review turnaround time, and incremental delivery.

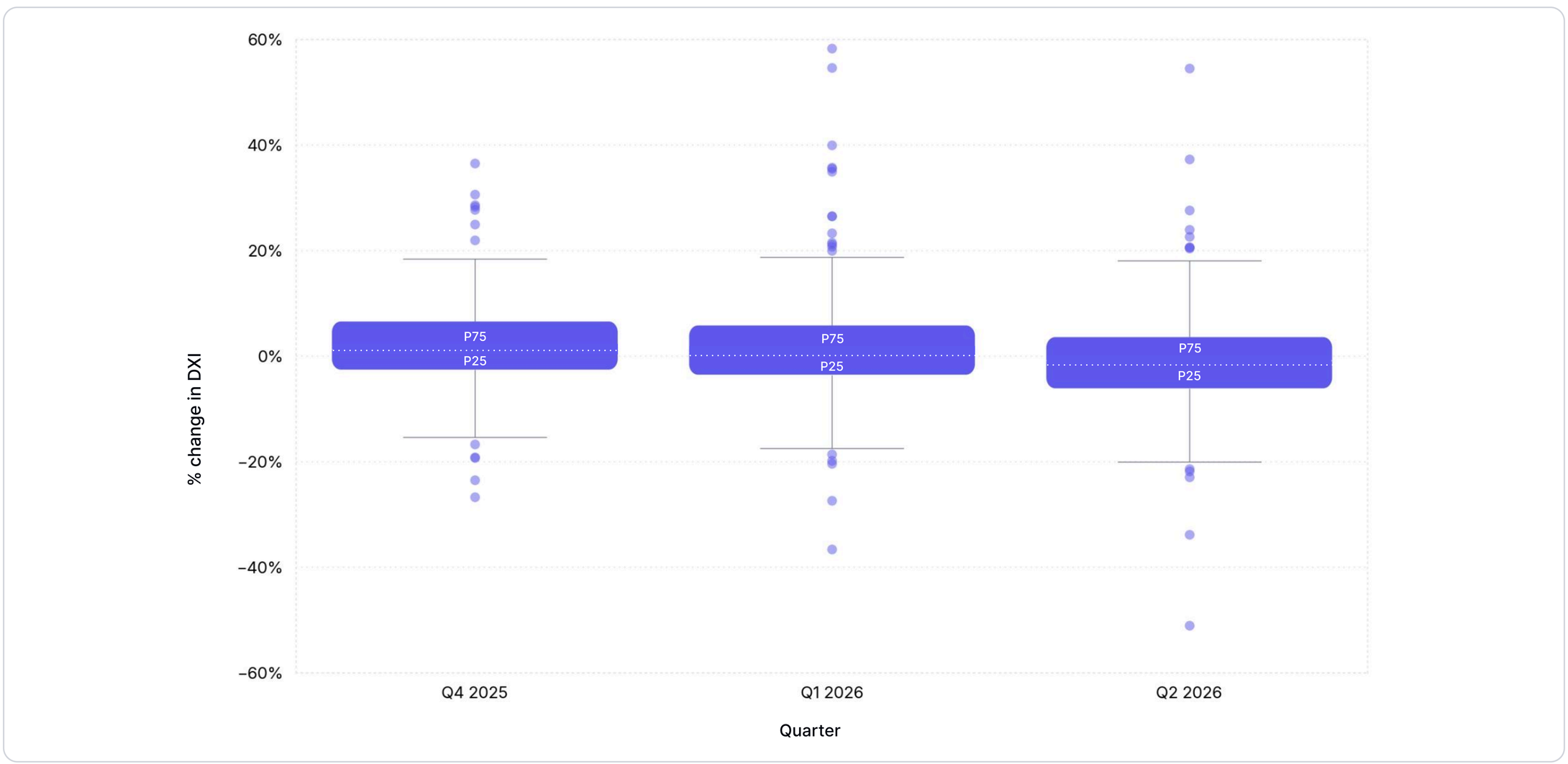
A single-percent drop in the DXI represents the loss of roughly 10 hours annually per engineer to sources of friction, toil, and bottlenecks in the engineering process. These losses nullify aspects of the velocity gains, leading to outcomes that aren't always aligned with increases in speed.

Primary metric: Developer Experience Index (DXI)

Overall median DXI shifted from 67 in Q3 2025 to 65 in Q2 2026, falling consistently from Q4 2025 to our present dataset. On its own, this is concerning, but the potential risk is further elucidated when we analyze each individual DXI factor.

Quarter-over-quarter percent change in DXI

Sample from 500+ companies from July 2025 to June 2026

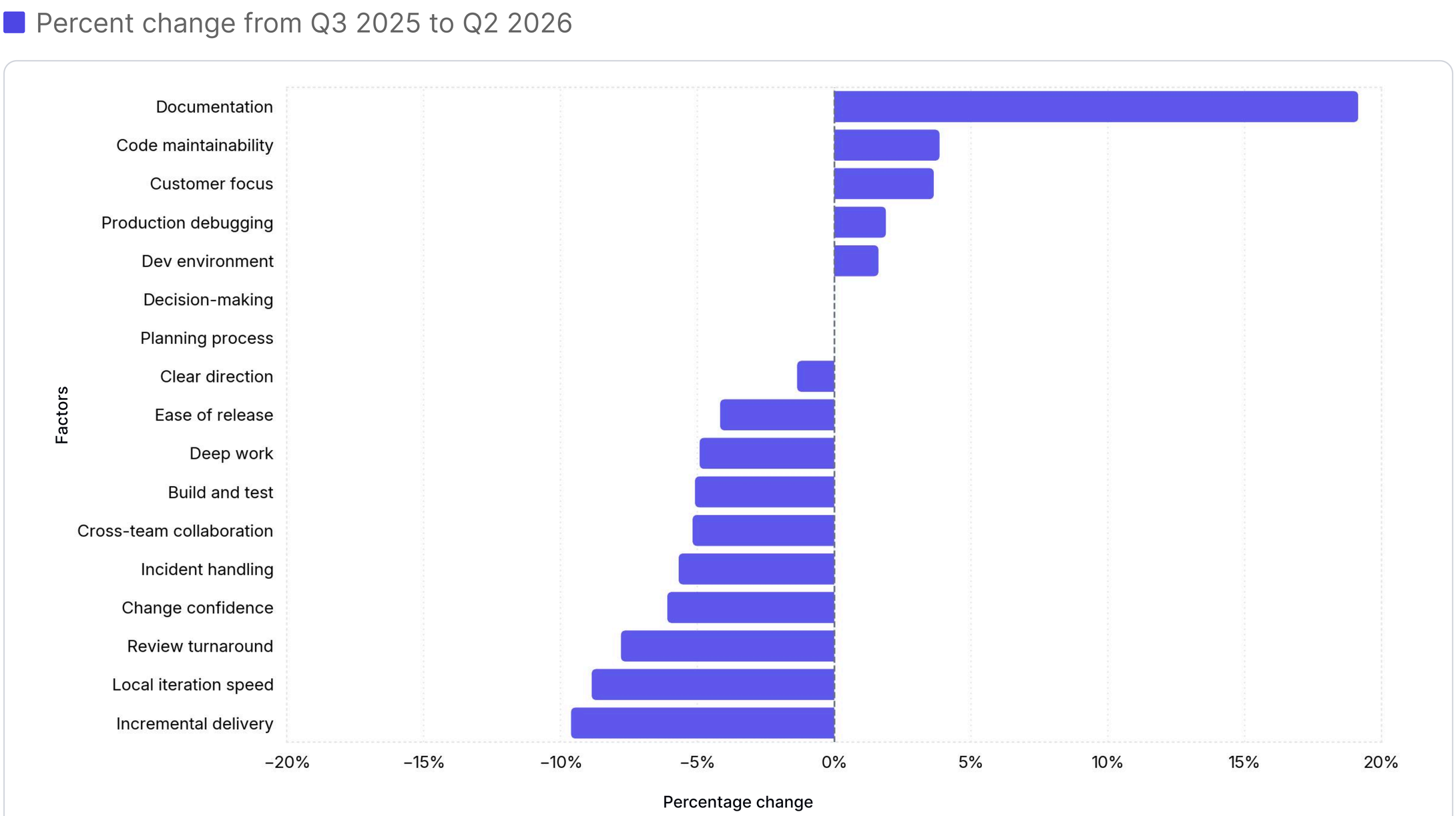


Individual developer experience factors

We have broken down the DXI into its individual factors, and calculated the impact to those factors over the same period, from Q3 2025 to Q2 2026. Studying this data, we see disparity across all factors, with some areas impacted positively, but many more impacted negatively.

Percent change in DXI factors

Sample from 500+ companies from July 2025 to June 2026



Gains in several factors

The drastic increase in documentation quality is clear. Given the strong utility of AI for assisting in documentation generation, a gain this substantial is expected, and leaders should continue to build documentation-generation into their AI strategy. There are also notable gains in areas such as code maintainability and production debugging.

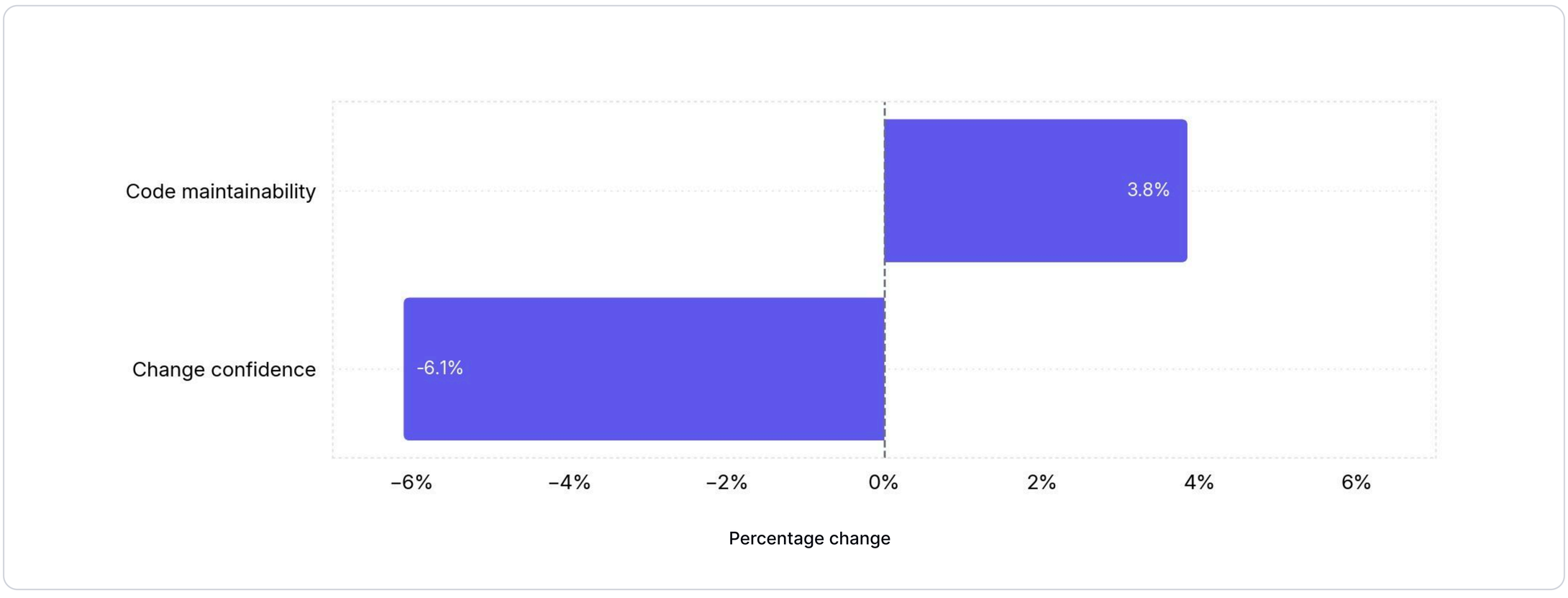
These gains, however, are eclipsed by degradation in other key factors such as incremental delivery, local iteration speed, and review turnaround time.

Varied impact on quality

DXI tracks software quality through two primary metrics: change confidence (developers' trust that their changes won't break things) and code maintainability (how easy the codebase is to understand and modify). During this period, the two told opposite stories. Code maintainability improved by 3.8%, while change confidence fell by 6.1%.

Percent change in change confidence and code maintainability

Sample from 500+ companies from Q1 (Jan-Mar) 2026 → Q2 (April-June) 2026



Code maintainability indicates the ease with which code can be understood by developers, something augmented by coding assistants and agents, and that unsurprisingly has improved as a result.

By contrast, change confidence measures whether developers feel confident making changes to that code without accidentally breaking things. While AI is making it easier for developers to understand what's in front of them, the data suggests that they have less trust in the code they are actually pushing into production.

Why change confidence is dropping as output rises

> "Engineers feel deeply accountable for what ships. When something is high-risk or hard to undo, they pull back on how much they leverage agentic velocity. That's not irrational, it's human behavior that's hard to get over." — Eirini Kalliamvakou, Research Advisor, GitHub

Listen to the full AI and Engineering Productivity panel debate from DX Annual.

Worsened incremental delivery

The sharpest decrease is in incremental delivery, which asks developers whether they work on small, incremental changes to the system. This metric measures healthy PR throughput, following a practice of creating PRs that are scoped to a single, testable problem. When cross-referenced with the quality → PR size section later in this report, an increase in this metric points to a broader trend of PR bloat.

This is a combined result of multiple behaviors such as:

- Verbose code generated by AI models
- A temptation to increase batch size because of AI code generation
- Agent-generated PRs leading to more functionality added per PR
- “Over-engineering” of solutions by agents and assistants

To add to this problem, the data also shows increases in build and test time, as well as a sharp decrease in local iteration speed. These are factors that research from multiple organizations associated with shipping larger batches. If a developer is anticipating an hour-long build cycle, they will naturally include more functionality per build, rather than waiting on multiple hour-long builds to complete.

A larger, less incremental PR means a longer review cycle, a more complicated merge and test set, and a more difficult time debugging and reverting changes when necessary.

Validating AI investments with developer sentiment

BNY used DX survey data to pinpoint where engineers experienced the most friction across their software delivery lifecycle, ensuring investments targeted real pain points rather than chasing trendy AI use cases. This sentiment-driven prioritization gave the team the confidence to codify AI into every step, from planning and peer review to testing, change management, and compliance, because they could validate which processes were bottlenecks.

By grounding their strategy in developer experience data, BNY moved beyond isolated coding assistants to a systematic approach to AI-assisted delivery across 8,000+ engineers. Read more about their process.

For DX users: Reports > Core 4 > DXI

Secondary metric: Time to 10th PR

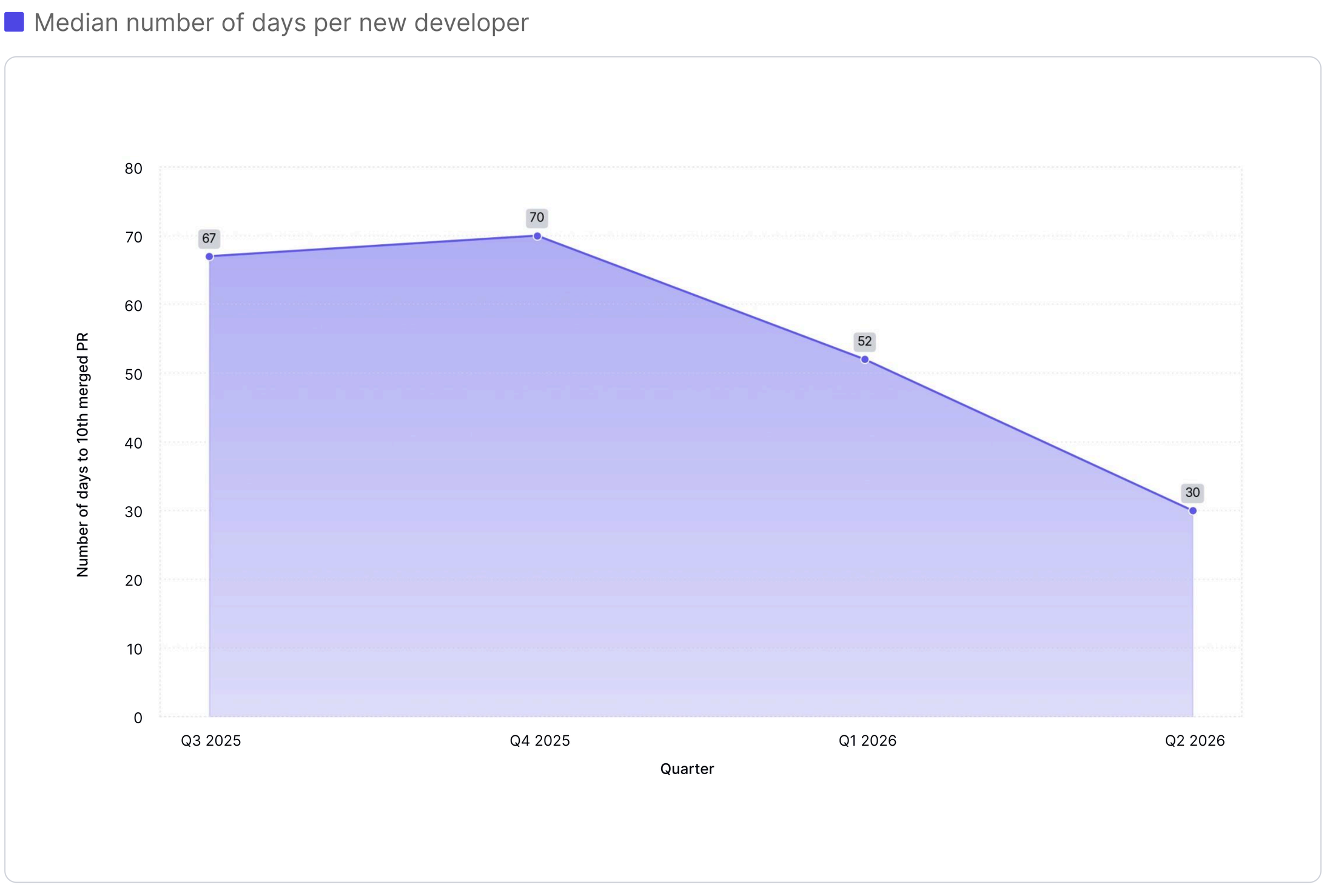
In Q1, we reported that time to 10th PR, an onboarding signal measuring how long it takes new developers to merge their first 10 pull requests, had dropped from 39 to 33 days between Q3 and Q4 2025. This metric has been reduced by more than half since Q1 2024, before the use of AI was widespread in the industry.

Developer ramp-up time is one of the clearest, least-contested areas of AI impact. AI assistants and agents can help developers navigate unfamiliar codebases, interpret documentation, and produce acceptable code faster. As more agentic solutions are devised for onboarding processes, we may see this figure continue to decline. However, diminishing returns are inevitable as onboarding requires many steps that aren't currently impacted by AI.

For engineering leaders, this is one of the strongest ROI arguments for AI investment: reduced ramp time directly translates to faster time-to-productivity for every new hire.

Median time to 10th merged PR

Sample from 500+ companies from July 2025 to June 2026



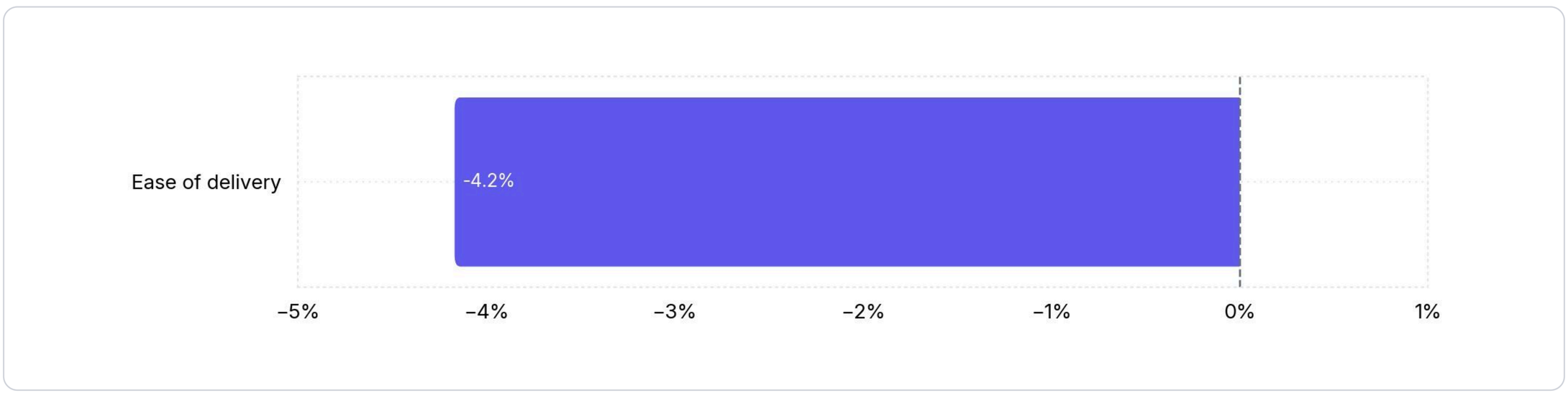
For DX users (privileged users only): Tools > Onboarding

Secondary metric: Ease of delivery

Both a DXI driver and a secondary Core 4 metric, ease of delivery captures how frictionless it is for developers to move work from "ready to code" to "shipped."

In an AI-augmented environment, this metric separates organizations that have embedded AI into their full SDLC from those that have only optimized the code-writing phase.

As the Q1 report made clear, non-AI factors such as CI wait times, meeting overhead, and unclear requirements still outweigh AI time savings. Ease of delivery is where those systemic inefficiencies show up.



AI can't fix an inefficient CI/CD pipeline, approval chain, or misaligned staging environments. The overhead of software delivery, which the Q1 report framed as "non-AI factors still outweighing AI benefits," remains a dominant drag on this metric. Organizations seeing the highest ease of delivery scores tend to be those that have also invested in platform engineering, developer portals, and process simplification alongside their AI rollout.

Bottlenecks shifting downstream

> "If your code throughput goes up by 3x tomorrow, would your SDLC be able to absorb it? We have seen some early signs it is not the case. It breaks everywhere." — Uma Namasivayam, Dropbox

Listen to the full talk from DX Annual on How Dropbox is rethinking productivity in the agentic era.

For DX users: Reports > Efficiency > Ease of delivery

Quality

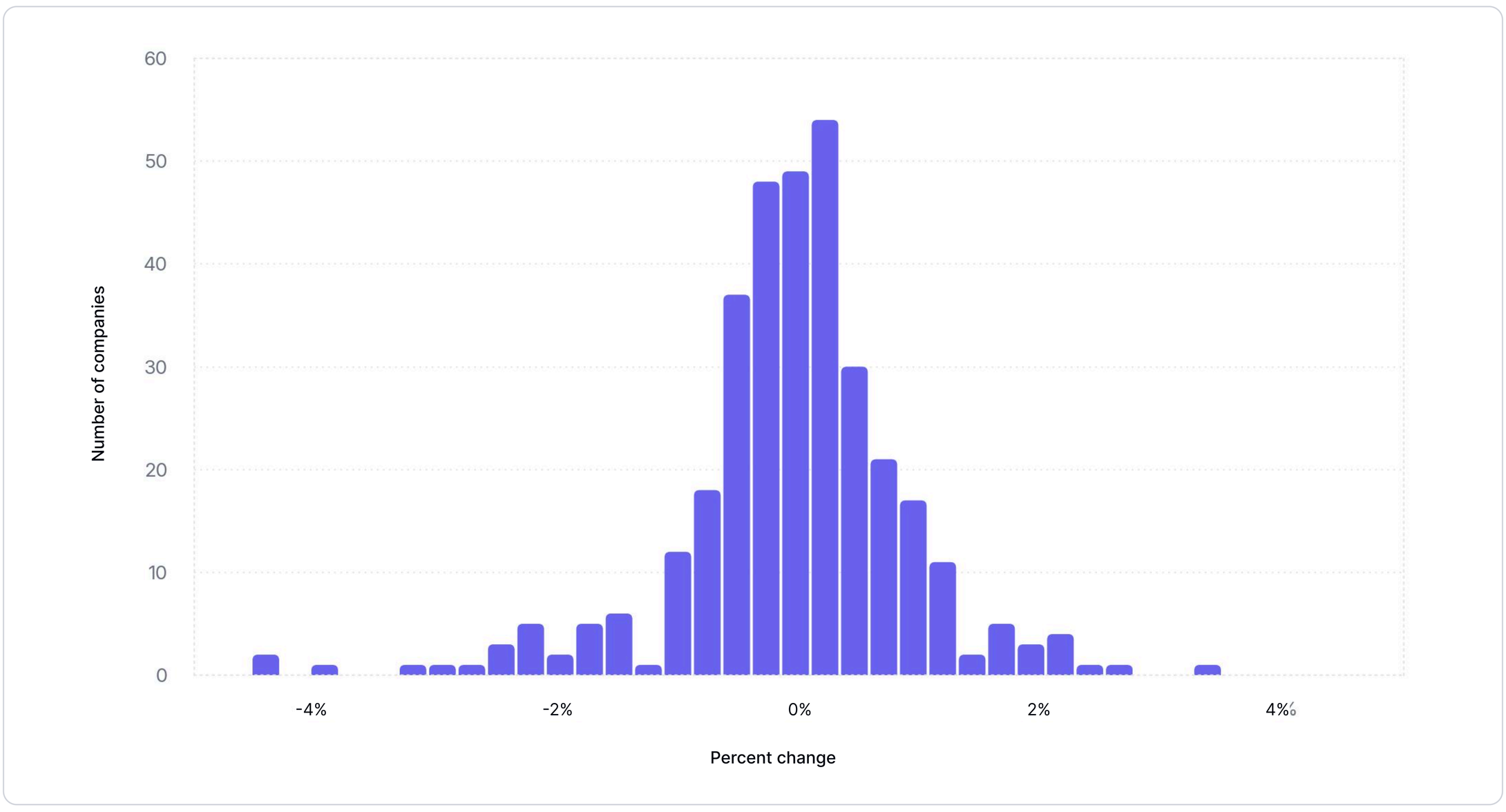
In Q1, we flagged quality as "varied and volatile," and that characterization has not softened. Change failure rate, which DORA describes as "the percentage of changes that result in degraded performance requiring immediate remediation," remains the most unpredictable metric in the report.

Primary metric: Change failure rate

The data in our previous report showed some companies seeing change failure rate increases of nearly 2 percentage points. Positioned against an industry benchmark of 4%, that represents as much as a 50% increase in production defects. The Q2 data sees this trend continue, with new outliers extending into the +/-3% range.

Distribution of percent change in change failure rate %

Sample from 500+ companies from Q1 (Jan-Mar) 2026 → Q2 (April-June) 2026



For DX users: Reports > DORA > Change fail percentage

Quality metrics are considered metrics that are oppositional to speed metrics, and that's highlighted in the tension this creates with the speed data above. Deploy frequency is rising across nearly every segment. TrueThroughput is up 37%. But if change failure rate is simultaneously climbing for a meaningful subset of companies, then some organizations are simply shipping defects faster. This is the uncomfortable truth that the industry's velocity-first narrative tends to gloss over: speed without quality discipline is not acceleration, it's recklessness.

When visualized company by company, we see how wildly volatile this metric continues to be. It bears mentioning that this general pattern in change failure rate has been observed prior to AI initiatives, but AI has increased the amplitude of this pattern.

Engineering leaders should be pairing every speed metric with a quality counterweight. If your team's deploy frequency is up 60% but you're not also tracking change failure rate, recovery time, and rollback frequency, you have an incomplete picture of overall productivity.

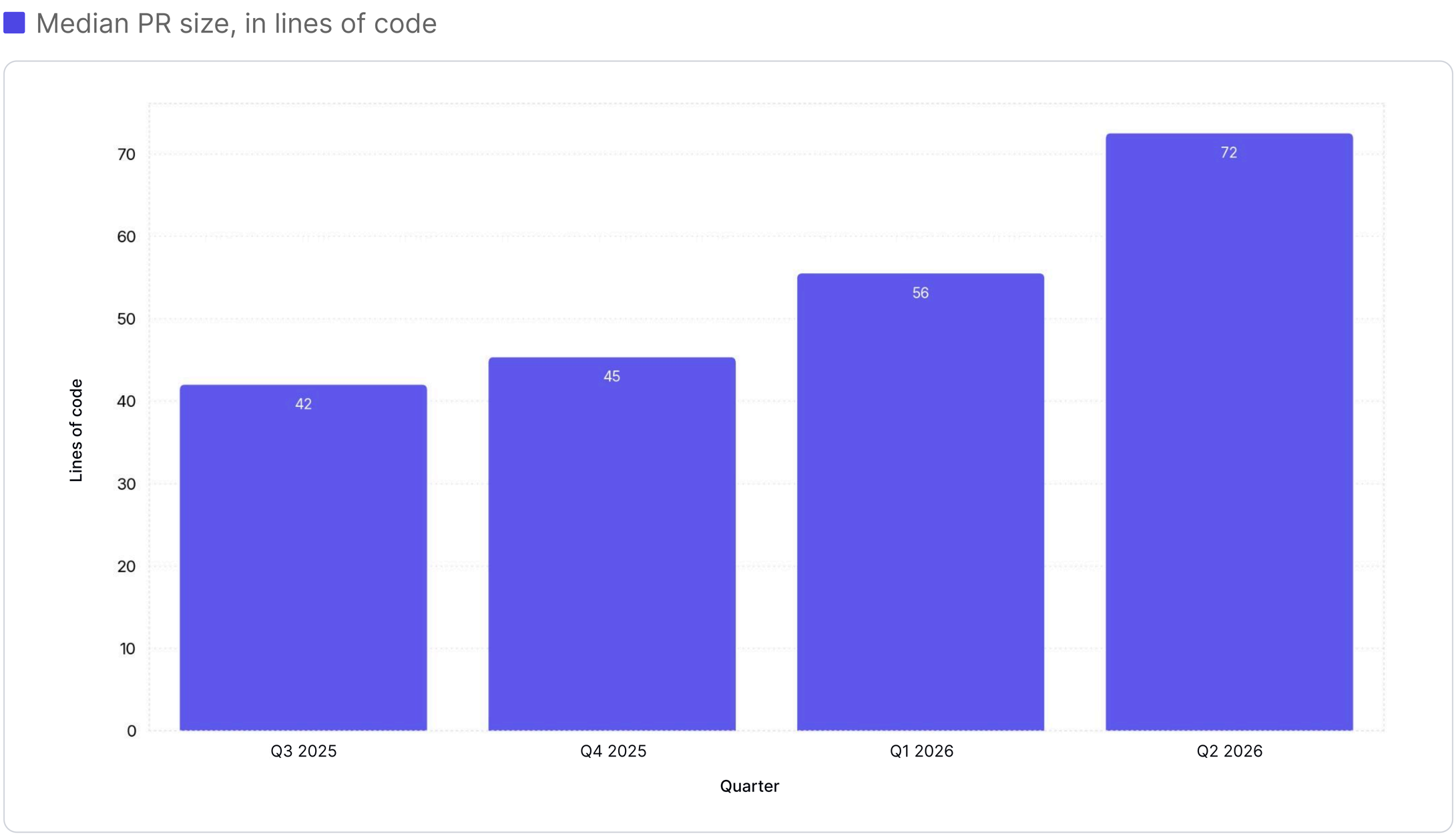
Secondary metric: PR size

In the same period of time that we've seen AI adoption explode, we've also seen PR size nearly double. This metric can serve as an early indicator of impending tech debt, and so this inflation is an immediate cause for concern.

Generally, more code can equal more complexity, less portability, and a greater potential for bugs and vulnerabilities. More verbose code can also be more difficult to review and maintain. One of the traits of a skilled engineer is the ability to fully implement a use case with exactly as much code as needed to perform the task. When AI undermines that instinct at scale, the result is not just technical debt. It is compounding cognitive debt across the team as engineers struggle to understand code they did not write.

Median PR size over time

Sample from 500+ companies from July 2025 to June 2026



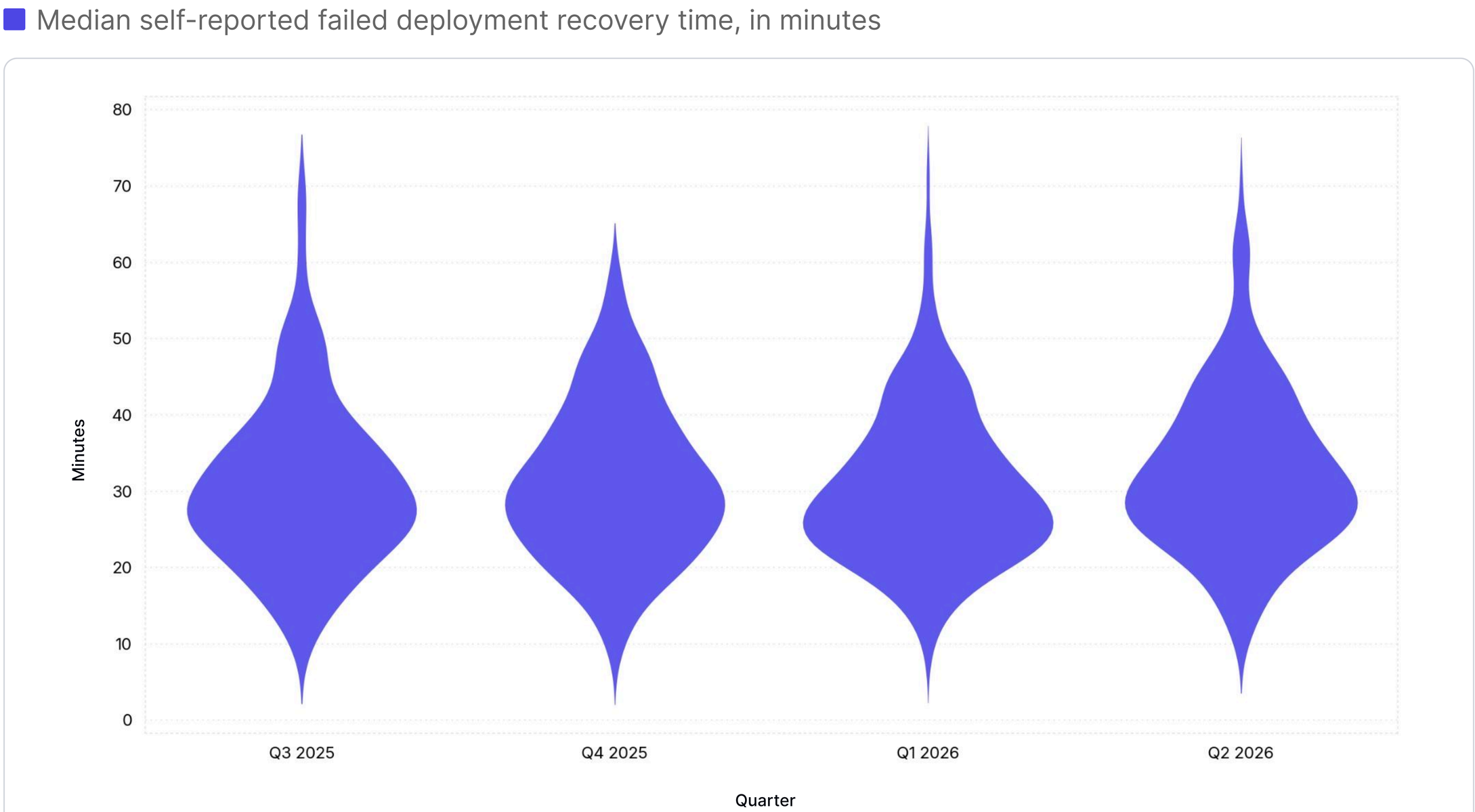
For DX users: Reports > PR size

Secondary metric: Failed deployment recovery time

Recovery time represents how quickly a team can restore service after a failed deployment. We've seen the distribution of recovery time lean towards longer recovery times, despite the qualitative signal that the Production Debugging DXI driver included above has improved by a small measure, as detailed above.

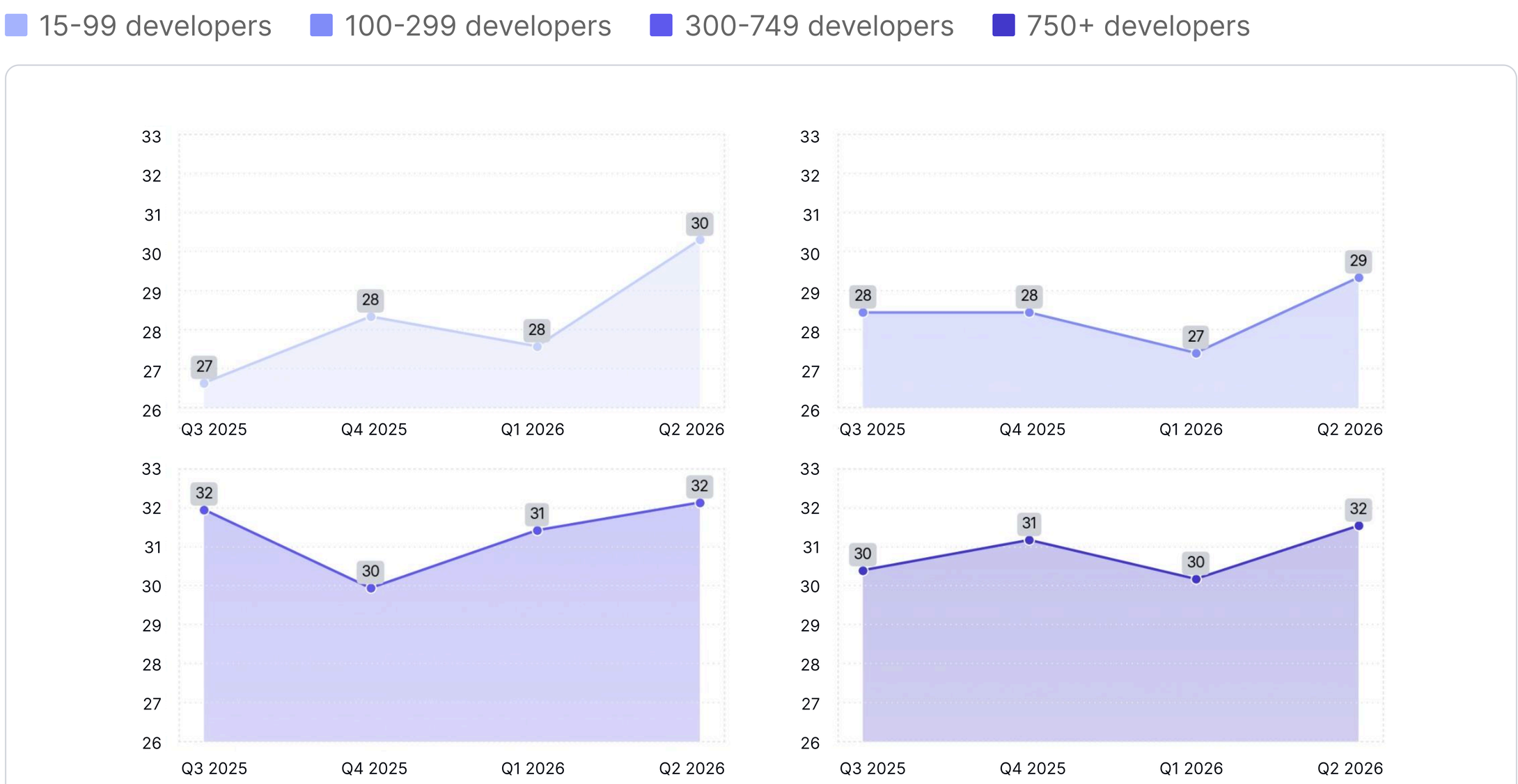
Distribution of failed deployment recovery time

Sample from 500+ companies from July 2025 to June 2026



Median failed deployment recovery time by organization size

Sample from 500+ companies from July 2025 to June 2026



A team can tolerate a higher failure rate if recovery is fast and automated; conversely, even a low failure rate becomes costly if each incident requires hours of manual intervention.

As AI-authored code constitutes a growing share of production changes (now over 52% — see AI Utilization section below), the character of failures may also be shifting. Failures caused by AI-authored code may be more diffuse and harder to trace, particularly when the developer who merged the PR didn't fully understand the code the model produced. This is a hypothesis worth tracking in future quarters.

Building an agent-first organization can reduce defects

Intercom's AI-first strategy doubled PR throughput in nine months and led to measurable improvements in code quality. As teams gained efficiency, they reinvested time into reducing technical debt and fixing defects. They were able to cut their defect backlog by over 50% in approximately four months.

Read about how Intercom increased their code quality while increasing AI adoption and throughput.

Secondary metric: Perceived software quality

Like perceived rate of delivery, perceived software quality captures developer sentiment and serves as an early warning system. If developers begin reporting that code quality is declining even as throughput rises, it's a strong signal that AI-authored code is introducing maintenance debt that will compound over time. This is especially critical to monitor as AI-authored code crosses the majority threshold.

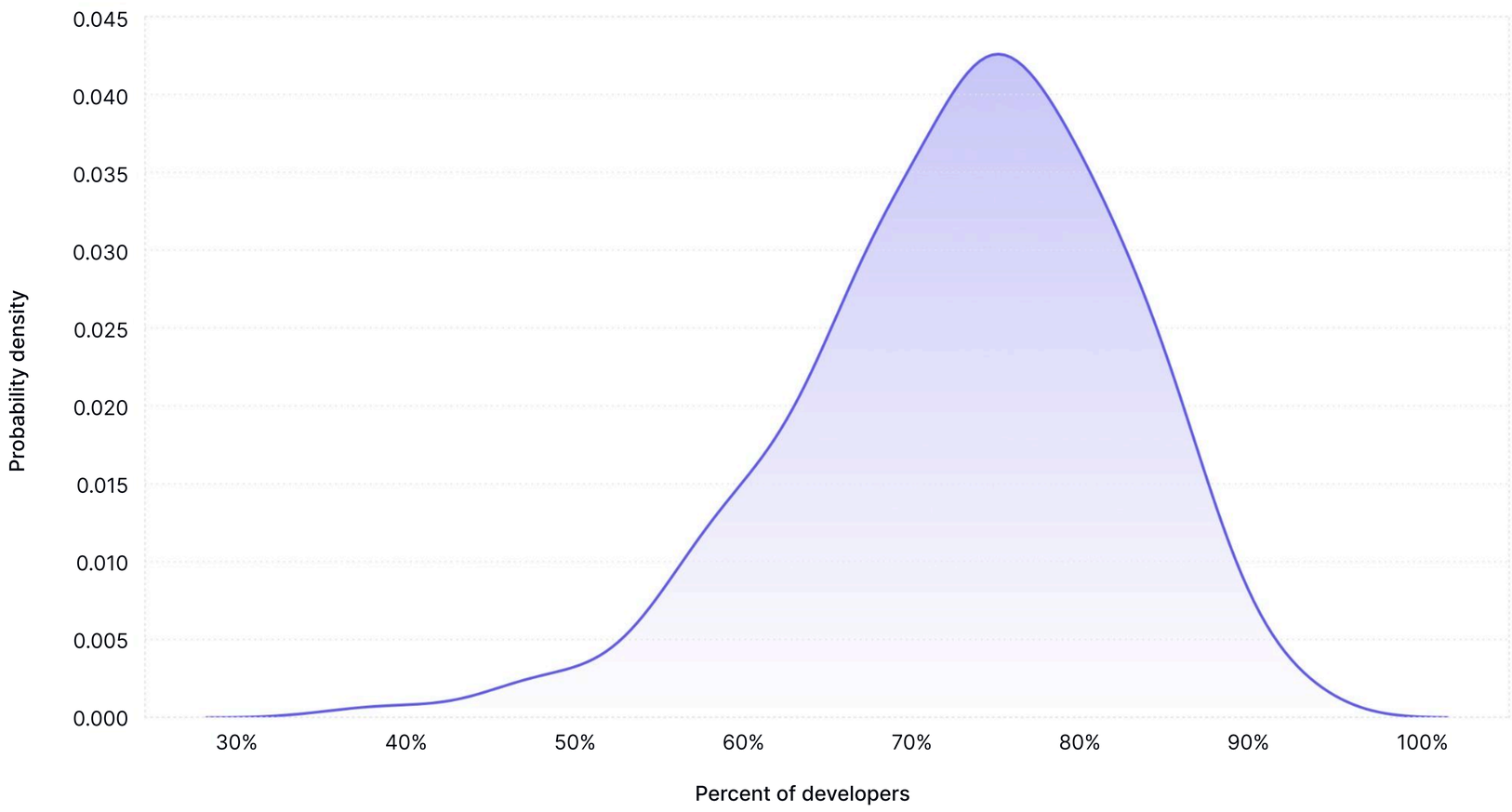
Median perceived high software quality

Sample from 500+ companies from July 2025 to June 2026

Distribution of perceived high software quality

Sample from 500+ companies from July 2025 to June 2026

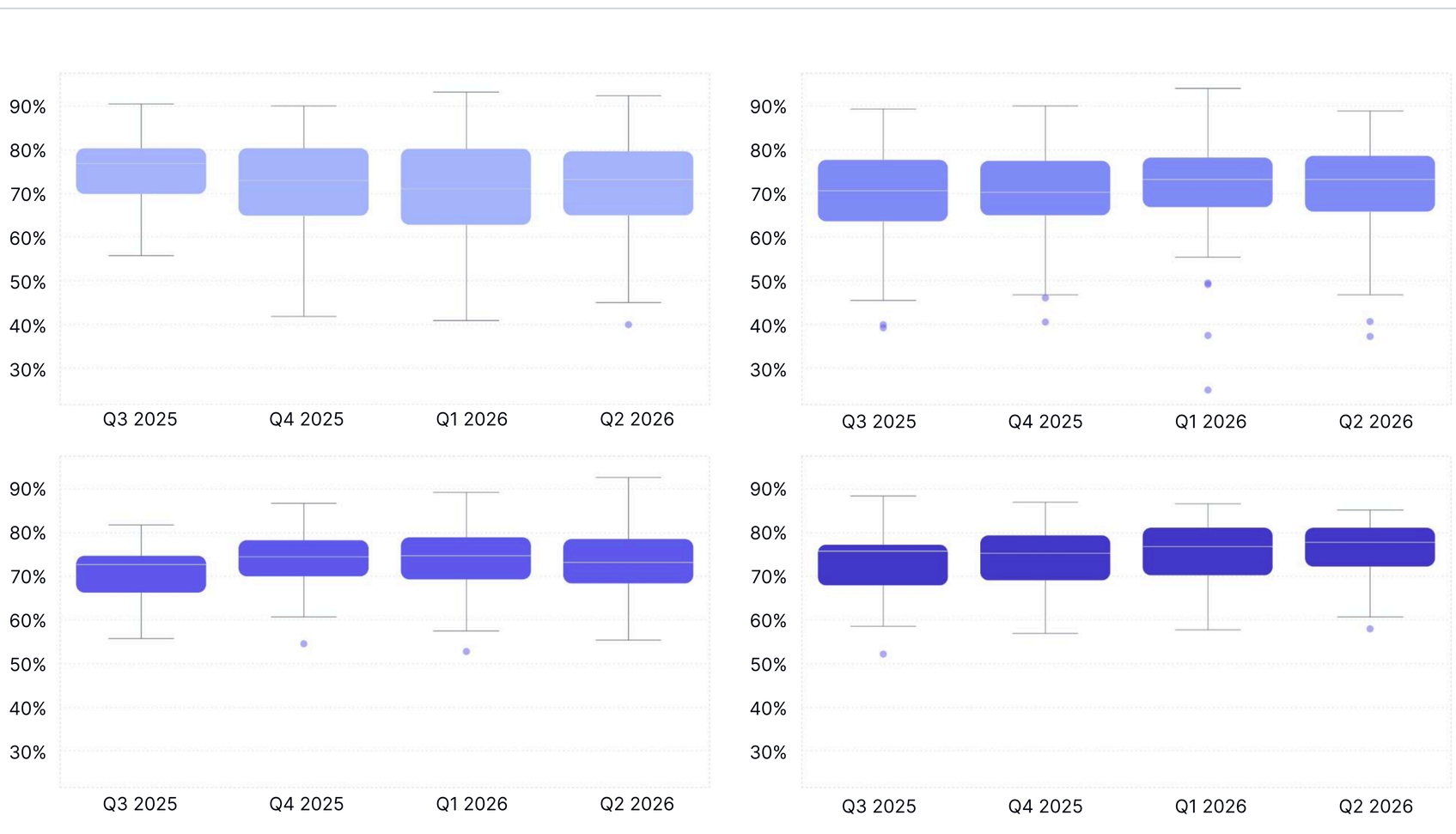
Percent of developers reporting high software quality



Distribution of perceived high software quality by organization size

Sample from 500+ companies from July 2025 to June 2026

15-99 developers 100-299 developers 300-749 developers 750+ developers



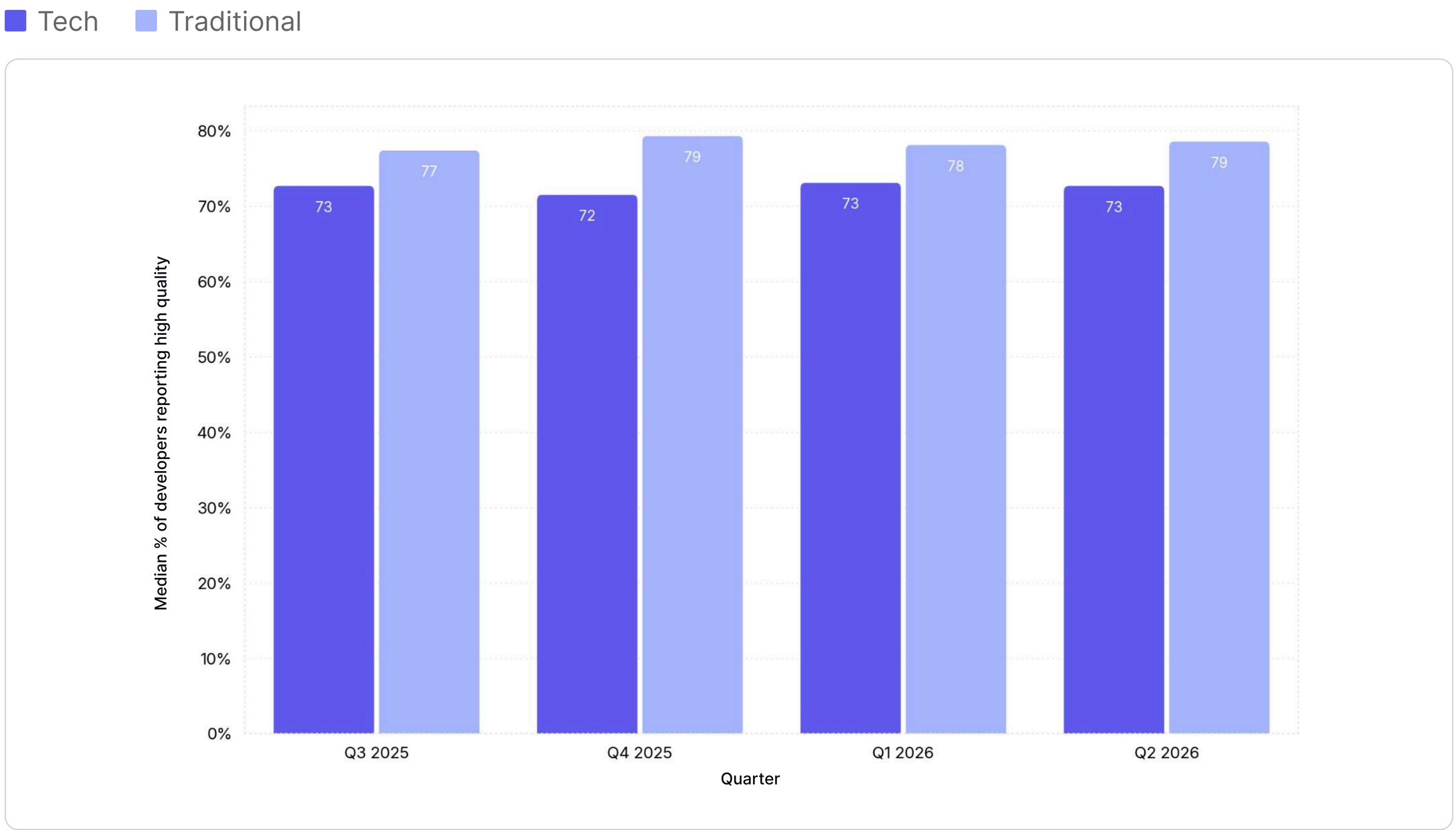
An interesting tension point is that the segments leading on Speed are not the segments leading on perceived quality. Traditional industries report higher perceived software quality than Tech in every single quarter (~77–79% vs. ~72–73%), despite Tech's lead in TrueThroughput. The same pattern holds at the vertical level: Financial Services posts the highest perceived quality (~77–81%) while also posting the lowest throughput, and Healthcare, this quarter's throughput leader, sits on the lower end of the perceived-quality range.

This is a clean, data-backed counter to the assumption that AI-driven speed comes without organizational cost. It doesn't prove AI is degrading quality in fast-moving segments, but it's exactly the kind of finding that should make engineering leaders ask the question of organizational impact before assuming the answer.

Knowing that traditional companies place a narrower focus on structured rollouts, AI included, and understanding that structured rollouts are positively associated with success in AI initiatives, it is perhaps unsurprising to see a greater perception of quality observed in traditional companies.

Median percent of developers perceiving high software quality by sector

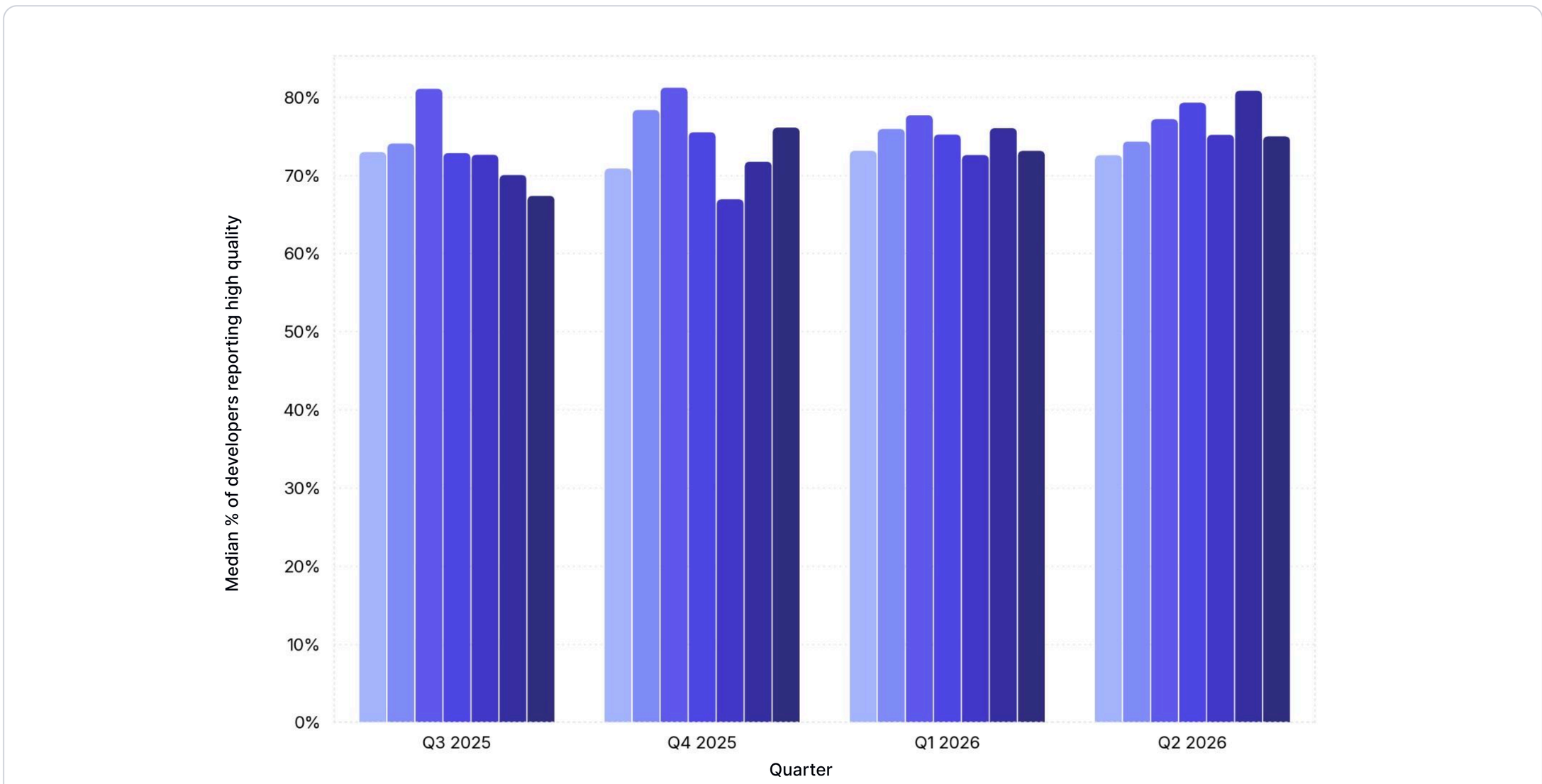
Sample from 500+ companies from July 2025 to June 2026



For DX users: Reports > Quality > Perceived quality

Median percent of developers perceiving high software quality by industry

Sample from 500+ companies from July 2025 to June 2026

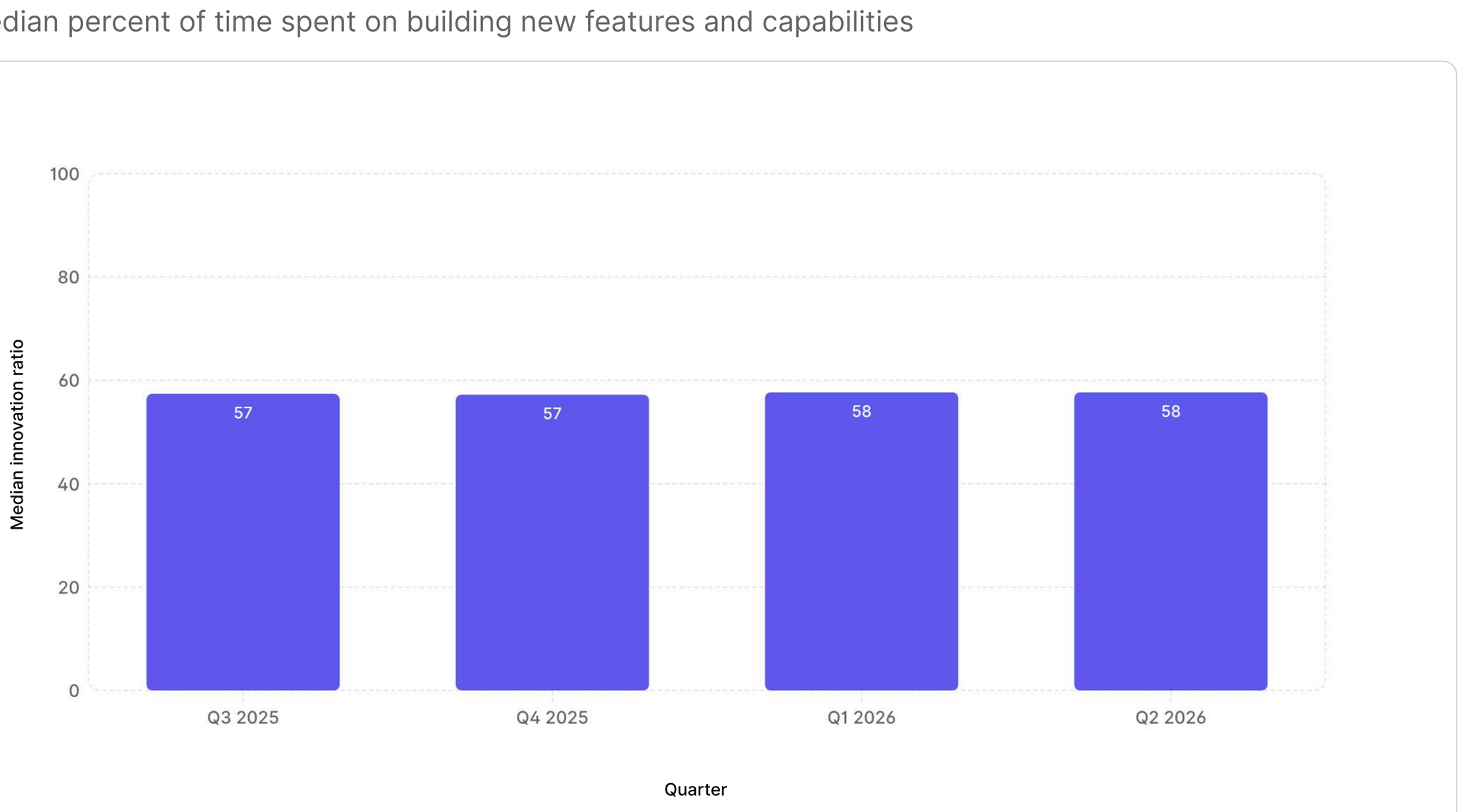


Impact

Primary metric: Innovation ratio

Median innovation ratio over time

Sample from 500+ companies from July 2025 to June 2026



Innovation ratio, the allocation of engineering effort spent on new feature work versus maintenance, toil, and operational overhead, is a critical output metric that connects AI investment to business outcomes. It's also where the hype meets reality.

The innovation ratio (time on net-new value creation vs. keep-the-lights-on work) has been essentially flat for a year: ~57% in Q3 and Q4 2025 and slightly increasing to ~58% in Q1 and Q2 2026. Given how much of the AI narrative centers on "freeing developers up to innovate," a flat-to-slightly-up trend over four quarters is a sobering data point worth stating without spin. The time AI is saving doesn't yet appear to be reliably converting into a shift toward innovation work at the portfolio level.

For DX users: Reports > Core 4 > Innovation ratio

AI's promise has always been to free up developer time for higher-value work. The Speed data confirms that AI is accelerating the code-writing phase. But the Q1 report's finding that non-AI factors (meetings, CI wait times, unclear requirements) still eclipse AI time savings raises a pointed question: is the time AI saves actually being reinvested in innovation, or is it being absorbed by the same organizational overhead it was supposed to eliminate?

The Atlassian Teamwork Lab refers to this as the “AI efficiency paradox” in their 2026 State of Teams report. This report surveyed 12,000+ knowledge workers and 170+ executives at Fortune 1000 companies and found that only 6% of executives feel confident that they can point to specific organization-wide AI ROI.

The Teamwork Lab found that “as AI helps individuals work faster, the surge of output backs up at reviews, approvals, and other human-judgment gates, slowing down the flow and wiping out the speed gains.”

Engineering leaders should be tracking innovation ratio longitudinally. If it isn't moving upward despite throughput gains, the organization has a prioritization problem, not a tooling problem.

PART 2

AI Measurement Framework

Utilization | Impact | Cost

The DX AI Measurement Framework provides the vendor-agnostic architecture required to calculate true ROI by tracking tool adoption, measuring impact, and guiding AI investments at scale. When paired with the DX Core 4, this framework allows leaders to directly trace how individual AI utilization translates into broader organizational performance. Developed in collaboration with industry researchers, vendors, and enterprise leaders, this data-driven approach has a proven track record of guiding strategy at scale.

DX AI Measurement Framework

* Metrics for autonomous AI agents

Utilization How much are developers adopting and utilizing AI tools?	**Impact** How is AI impacting engineering productivity?	**Cost** Is our AI spend and return on investment optimal?

• AI tool usage (DAUs/WAUs)
• Percentage of PRs that are AI-assisted
• Percentage of committed code that is AI-generated
• Tasks assigned to agents *
	• AI-driven time savings (dev hours/week)
	• Developer satisfaction
	• DX Core 4 metrics, including:
	  • PR throughput
	  • Perceived rate of delivery
	  • Developer Experience Index (DXI)
	  • Code maintainability
	  • Change confidence
	  • Change fail percentage
	• Human-equivalent hours (HEH) of work completed by agents *
		• AI spend (both total and per developer)
		• Net time gain per developer (time savings - AI spend)
		• Agent hourly rate (HEH / AI spend) *

Utilization

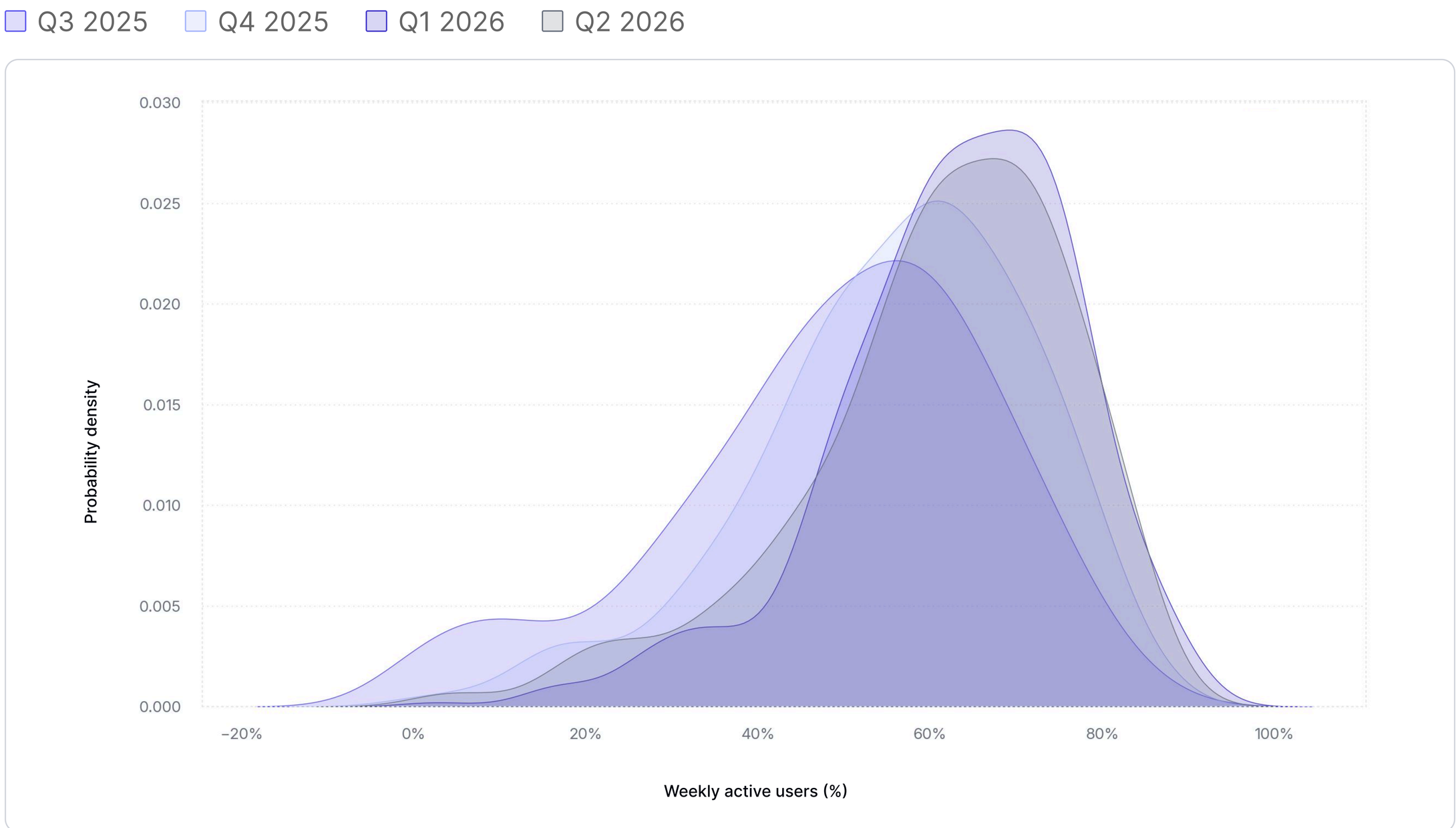
AI tool usage (DAUs/WAUs)

With adoption above 90% industry-wide since Q1 2026, the utilization story has shifted from "are developers using AI?" to "how deeply is it embedded in daily workflows?" The density of weekly active users in this study provides the clearest picture yet of how saturated AI tool usage has become.

The strategic question is no longer adoption, it's targeting and optimization. Leaders should be looking at the ratio of daily to weekly users within their own organizations. A high DAU/WAU ratio indicates that AI is embedded in routine workflows; a low ratio (high weekly, low daily) suggests that developers are experimenting but not yet integrating AI into their core development loop.

Density distribution of weekly active users

Sample from 500+ companies from July 2025 to June 2026



For DX users: Reports > AI utilization > AI adoption

Encouraging AI adoption and use with structured enablement

Indeed used structured enablement to increase AI adoption from 25% to 97%. They designed a self-paced course, partnered with engineering leadership to encourage training completion, and created spaces for community and continued engagement.

> "We never set a mandate that said you must use AI. We did, however, set a suggested target for all of our engineers for the training itself...The idea was that we can't force you to use these tools, but if you do start using them through this training and find that you like them, you will use them"
> — Michael Redding, Indeed

View Indeed's AI Training Playbook and listen to their DX Annual talk to learn more about how they did it.

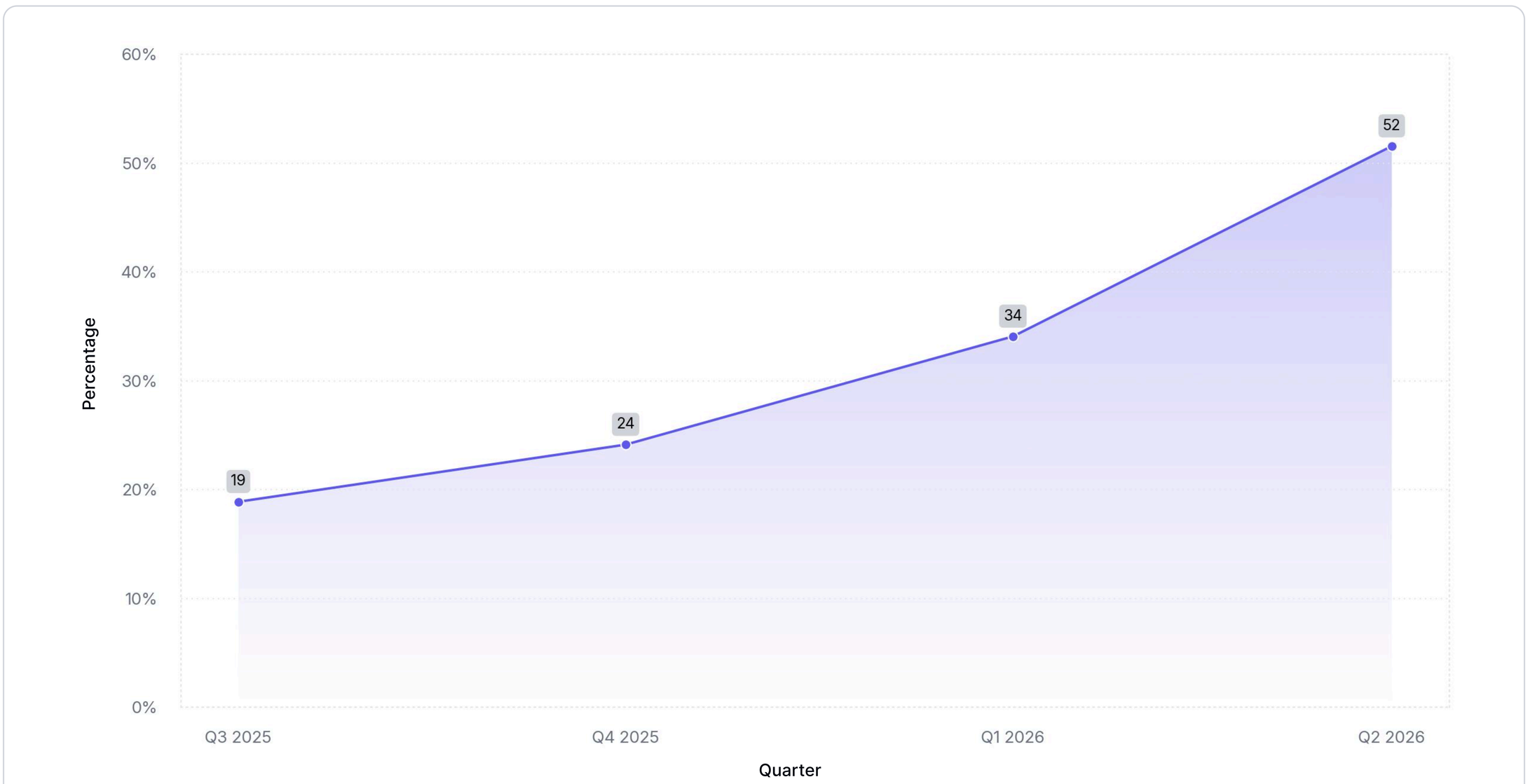
Percentage of code that is AI-generated

Across all developers in the study, 52.7% of code is now AI-authored. What is most notable is the sharp acceleration in this metric over the past three quarters. Looking at the distribution of the average percentage of AI-authored code, the volume has steadily surged from 24% in Q4 2025, to 34% in Q1 2026, before sharply spiking to 52% in Q2 2026. This steep trajectory suggests that once AI tools are available, the code they produce rapidly permeates the codebase, likely because AI-authored code flows through reviews, shared libraries, and collaborative workflows.

Consequently, organizations must recognize that AI-authored code is a collective, codebase-wide phenomenon rather than an isolated metric. This rapid saturation requires scaling quality infrastructure, code review norms, and testing protocols comprehensively across the engineering organization. Developers across the organization are now frequently reviewing, integrating, and maintaining code originally generated by AI, meaning the downstream impact on maintainability and technical debt applies to the entire team.

Distribution of average percent of AI-authored code

Sample from 500+ companies from July 2025 to June 2026



Using AI to reduce the code review bottleneck

As engineering teams scale with AI, code review is becoming more of a bottleneck than before. Dropbox is using agents to review pre- and post-commits, automating routine checks and validating code before human reviewers.

> "We have agents looking at commits and providing a risk scale, so that the reviewer and the developer can look at it and make a call."
> — Uma Namasivayam, Dropbox

Dropbox is using agents for around 5 steps of the SDLC. Learn more about what they are doing and how they are doing it.

For DX users: Reports > AI Utilization > AI code percentage

Impact

AI-driven time savings

In Q1 2026, daily AI users saved an average of 4.72 hours per week, with the overall average at 3.9 hours. As models continue to improve and agentic workflows mature, we expect this number to continue its upward trend. This rate of increase may begin to plateau as organizations exhaust the easiest automation targets (code completion, boilerplate generation, test scaffolding) and move into more complex, judgment-intensive tasks.

Median time saved from coding AI assistants

Sample from 500+ companies from July 2025 to June 2026

Time savings are climbing steadily across every usage cohort, with no sign of plateauing yet. Heavy users are tracking toward roughly 6+ hours/week by Q2 2026, up from the 4.72 hours/week reported for daily users in Q1. However, time savings are only valuable if they translate into business outcomes. An engineer who saves 5 hours a week but spends those hours in additional meetings has produced no net value from the AI investment. Leaders need to pair time savings tracking with downstream metrics such as innovation ratio and customer-facing feature velocity to understand whether saved hours are being reinvested or merely reallocated.

For DX users: Snapshots > Workflows > AI time savings

Developer satisfaction

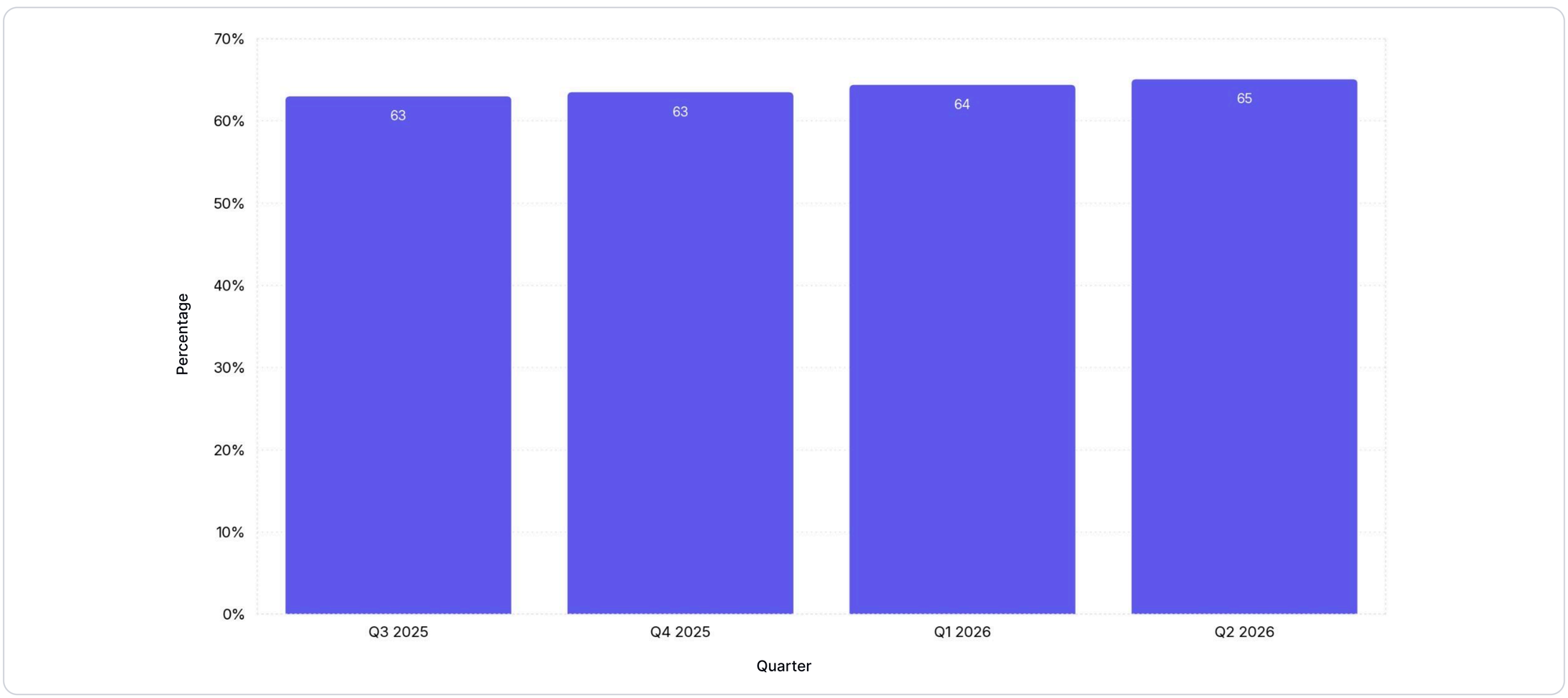
Developer satisfaction is a highly multifaceted concept. A developer's daily satisfaction is influenced by numerous factors ranging from team culture and management practices to psychological safety and broader organizational investments in developer experience.

For this reason, in this analysis we use the percentage of developers who report being "highly energized by their work" as a generic proxy. This metric serves as a reliable indicator of whether developers continue to feel engaged in their daily tasks, regardless of the shifting toolscape around them.

The data paints a picture of stabilization. Despite the compounding downstream challenges detailed throughout this report, overall developer energy has demonstrated resilience, holding steady between 63% and 65% from Q3 2025 through Q2 2026.

Median percent of developers who are highly energized by their work

Sample from 500+ companies from July 2025 to June 2026



While this report doesn't demonstrate AI actively fixing or elevating engineering culture on its own, this data strongly suggests that AI is, at the very least, not making it worse. It appears that the localized benefits of AI are effectively offsetting the new systemic complexities it creates, ensuring that the baseline developer experience is successfully maintained.

For DX users (privileged users only): Reports > Delivery > Attrition

Cost

AI spend (both total & per developer)

Cost is a dimension which will receive a lot of attention in the latter half of 2026 and beyond. As the initial hype begins to wear off, increasingly leaders will seek to understand whether the new R&D spend, in the form of AI tokens and subscriptions, will be in line with the desired outcomes such as improved feature velocity and an increased capacity for innovation, as discussed earlier in this report.

With median quarterly organizational AI spend data now available across 400+ organizations segmented by type, size, and region, leaders can benchmark their investments for the first time with real industry data.

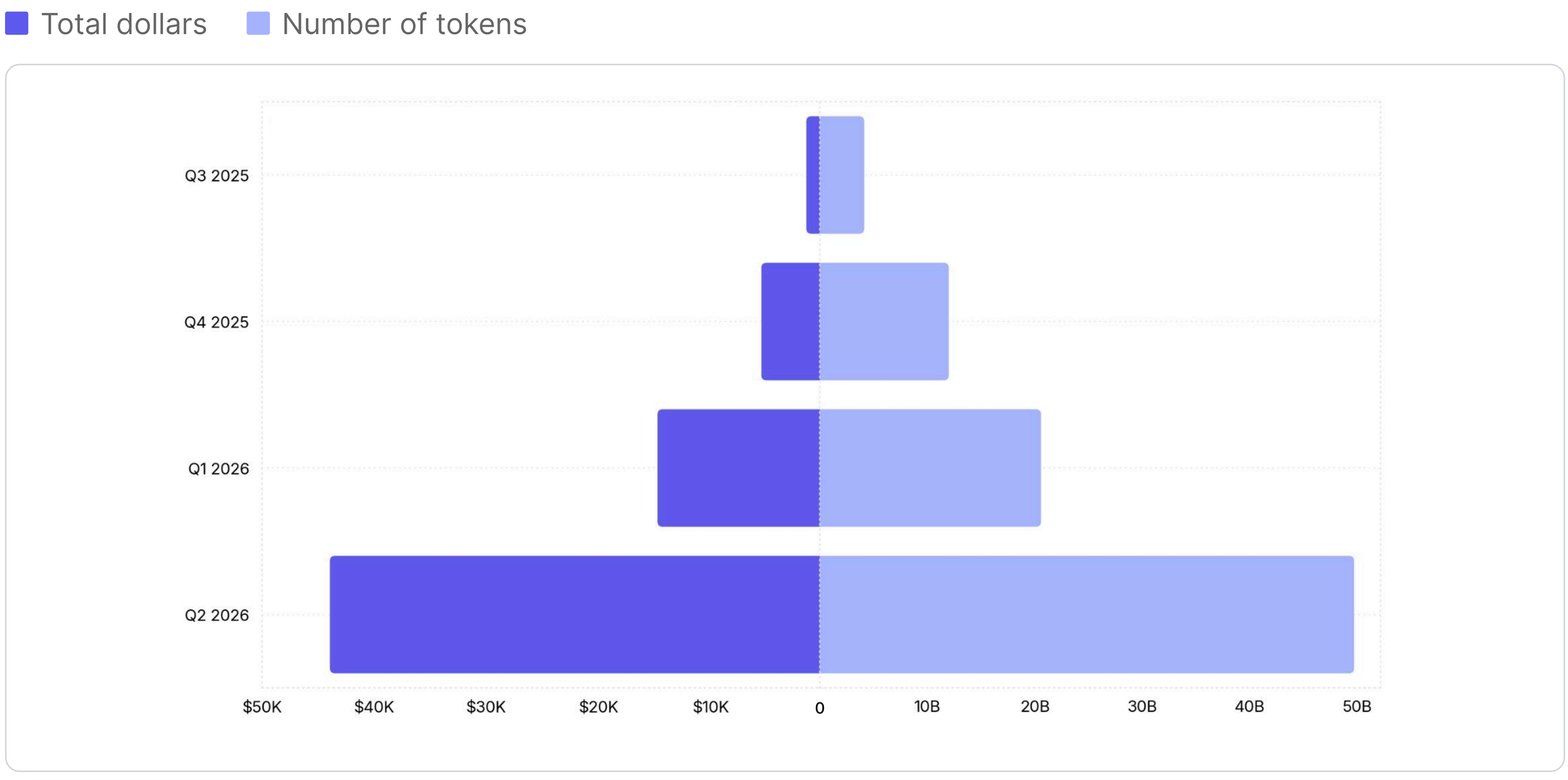
The data reveals a dramatic spike in total spending, heavily skewed by the Tech sector, alongside a counterintuitive market inversion: the smaller organizations paying the highest premium per developer are currently extracting the greatest value.

Total organizational spend

Total organizational AI spend is accelerating rapidly as token usage expands across the software development lifecycle. Median quarterly organizational spend climbed from roughly $1.5K in Q3 2025 to nearly $44K by Q2 2026.

Median quarterly organizational AI dollar and token spend

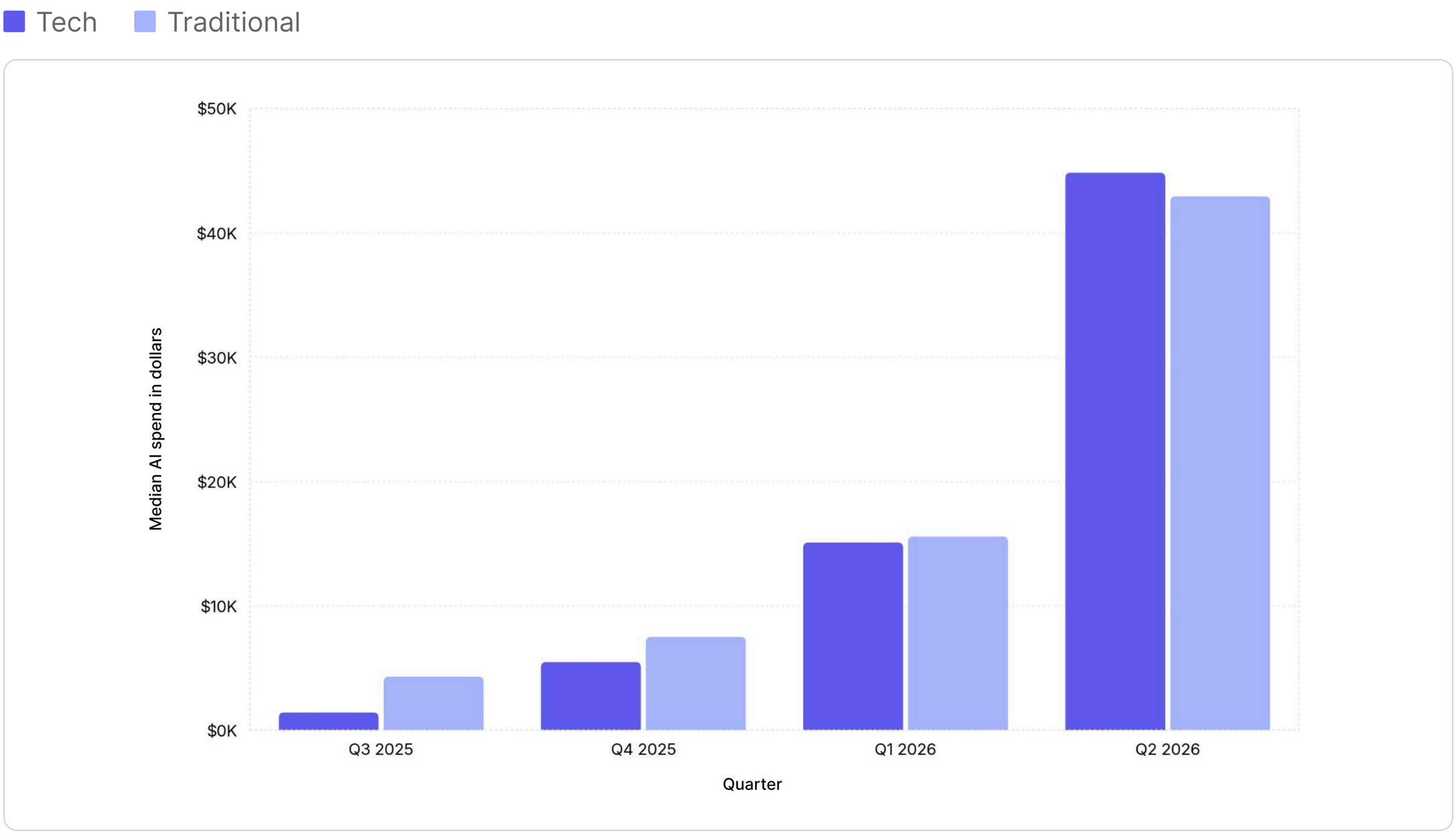
Sample from 500+ companies from July 2025 to June 2026



The growth in this AI investment is highly uneven across sectors. Tech-sector organizations are driving the spending spike, with quarterly spend jumping from around $1.4K to $45K over four quarters (an approximate 28x increase). Conversely, traditional industries grew far more modestly, reaching roughly $43K, an 8x increase from $4.3K from Q3 2025.

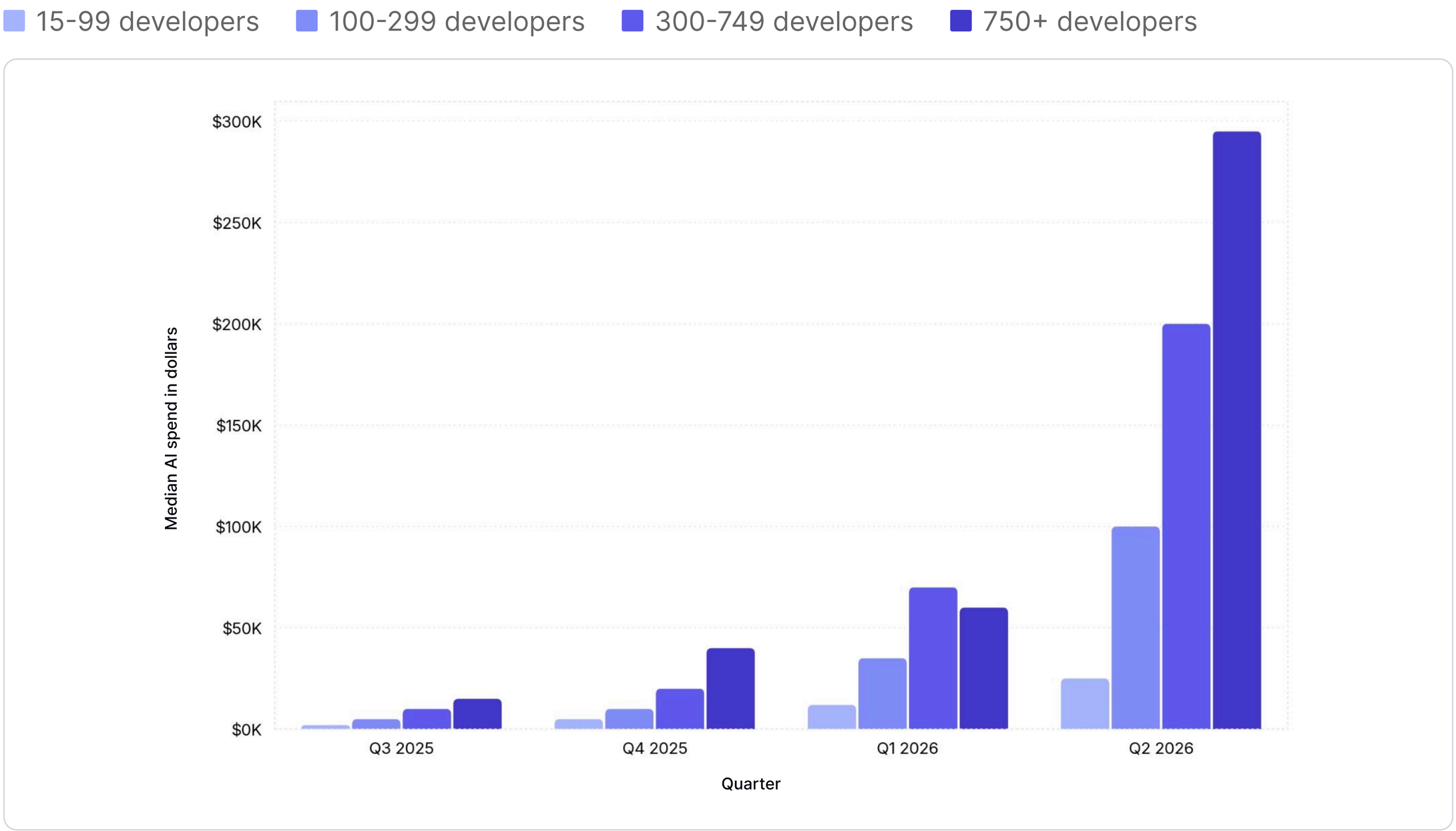
Median quarterly organizational AI spend by organization type

Sample from 500+ companies from July 2025 to June 2026



Median quarterly organizational AI spend by organization size

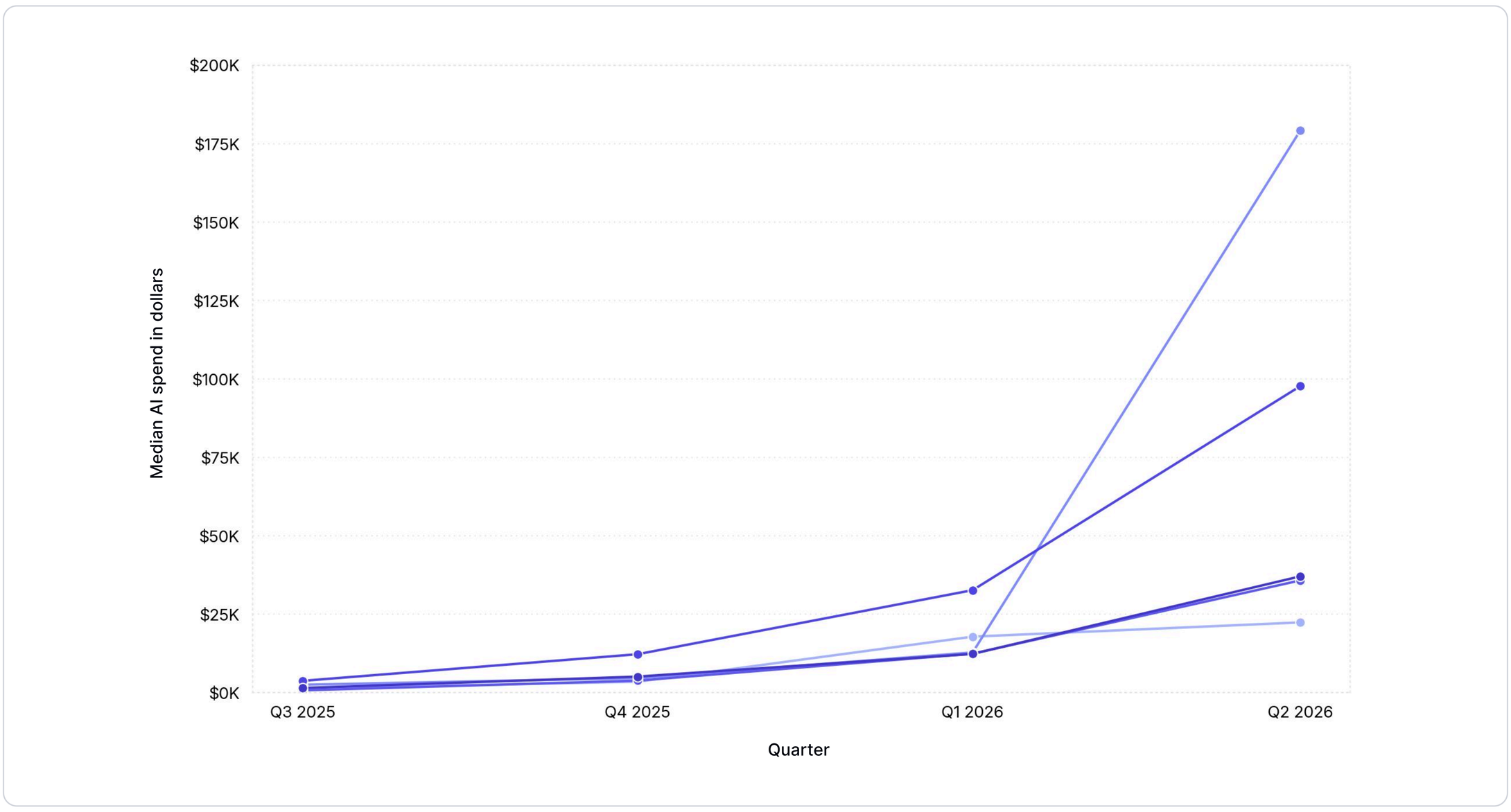
Sample from 500+ companies from July 2025 to June 2026



Larger organizations benefit from volume licensing and negotiated enterprise agreements, resulting in lower per-developer costs. Smaller organizations pay a premium on a per-seat basis but, as the Speed data shows, are extracting more throughput per dollar. This creates an interesting dynamic: the organizations paying the most per developer are getting the most value per developer.

Median quarterly organizational AI spend by region

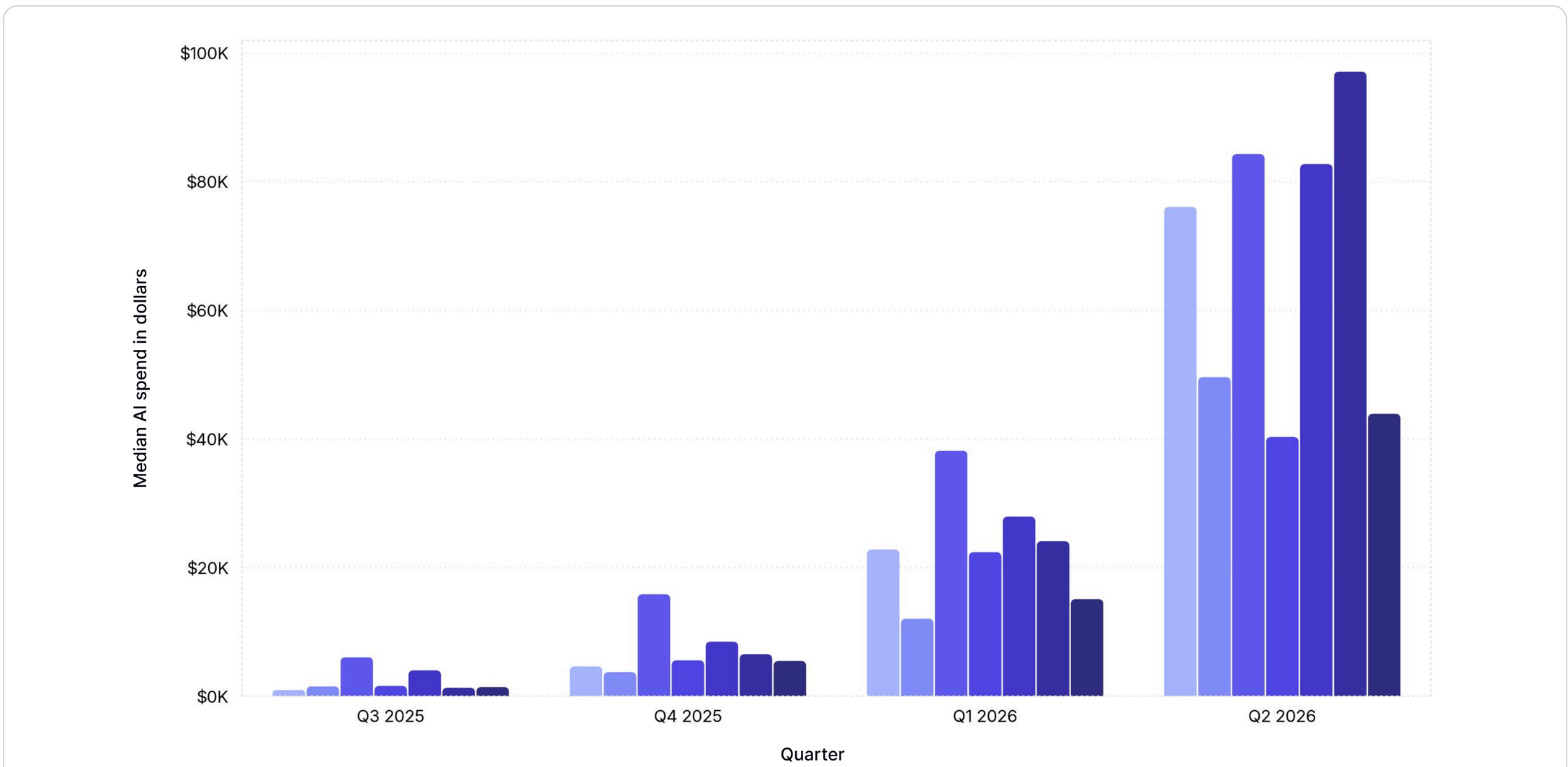
Sample from 500+ companies from July 2025 to June 2026



Regional spending patterns highlight that AI costs are heavily influenced by local labor cost dynamics, vendor pricing strategies, and regional purchasing power. Global engineering leaders must ensure they are benchmarking their AI expenditures against regional output metrics, rather than relying solely on global averages that may skew their ROI calculations.

Median quarterly organizational AI spend by industry

Sample from 500+ companies from July 2025 to June 2026



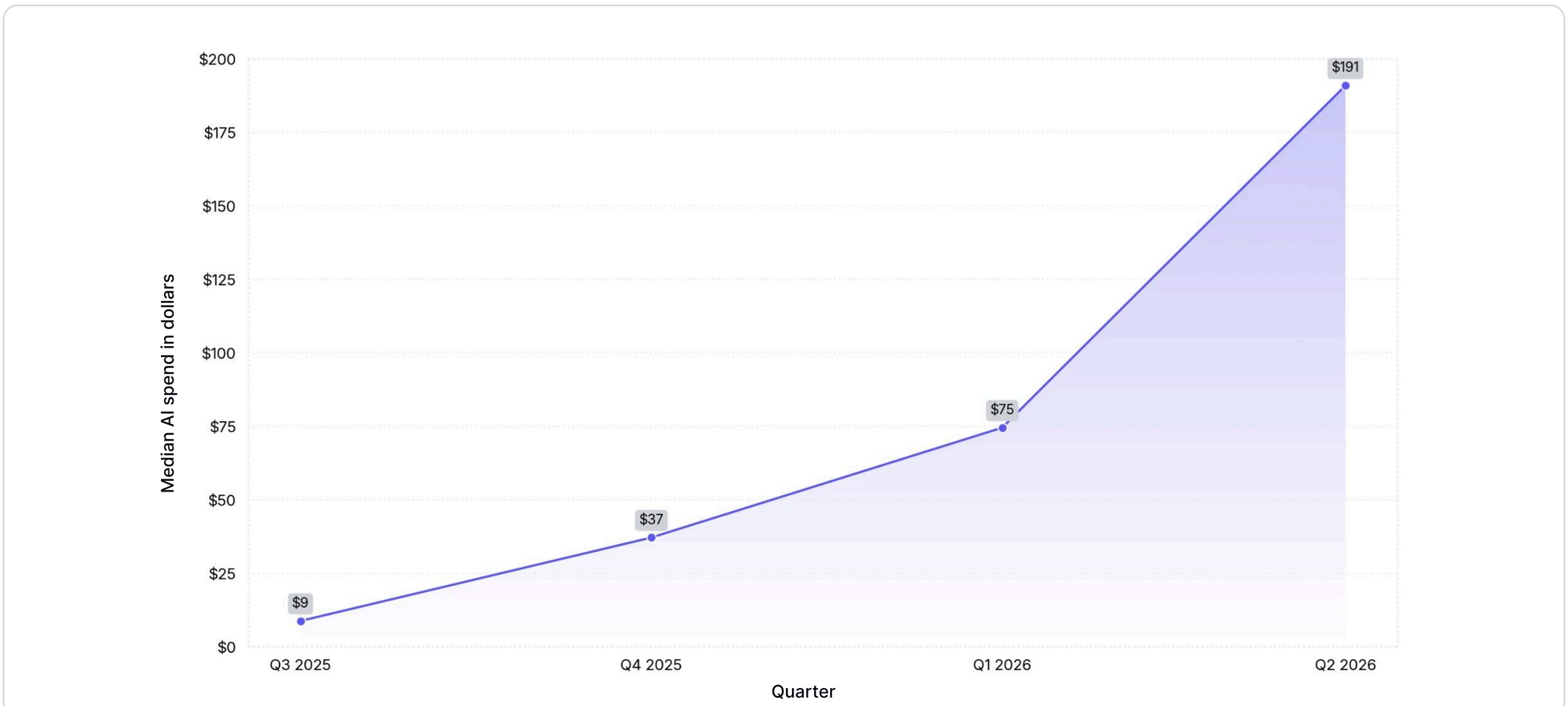
Viewing absolute spend by industry vertical highlights the variance in how aggressively different markets are capitalizing on AI, with specific sectors financially committing at significantly higher rates.

Per-developer spend

Breaking down cost on a per-seat basis reveals a critical inversion in the market. While smaller organizations lack enterprise negotiating power and are forced to pay a premium per seat, earlier Speed data confirms they are extracting more throughput per dollar spent.

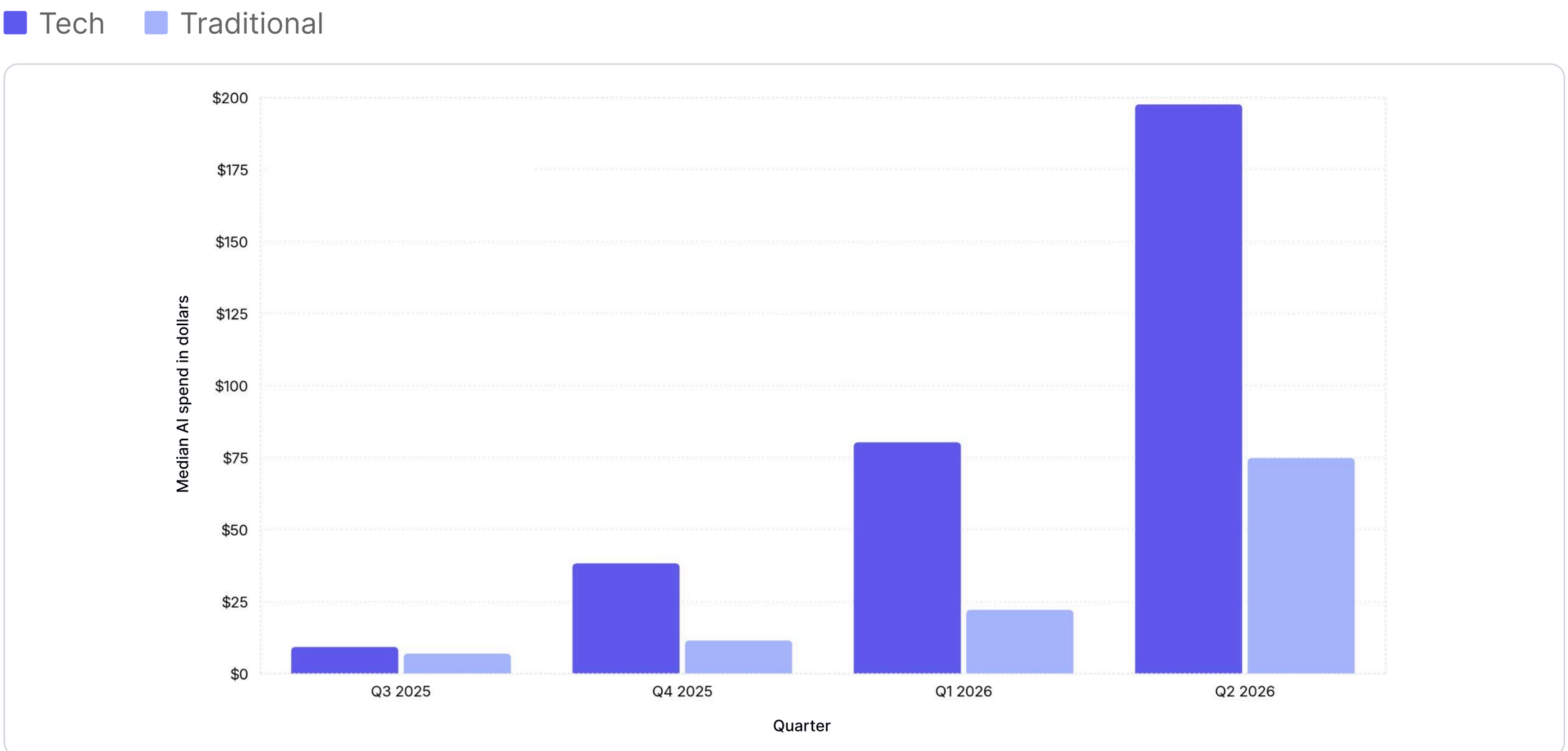
Median AI spend per user

Sample from 500+ companies from July 2025 to June 2026



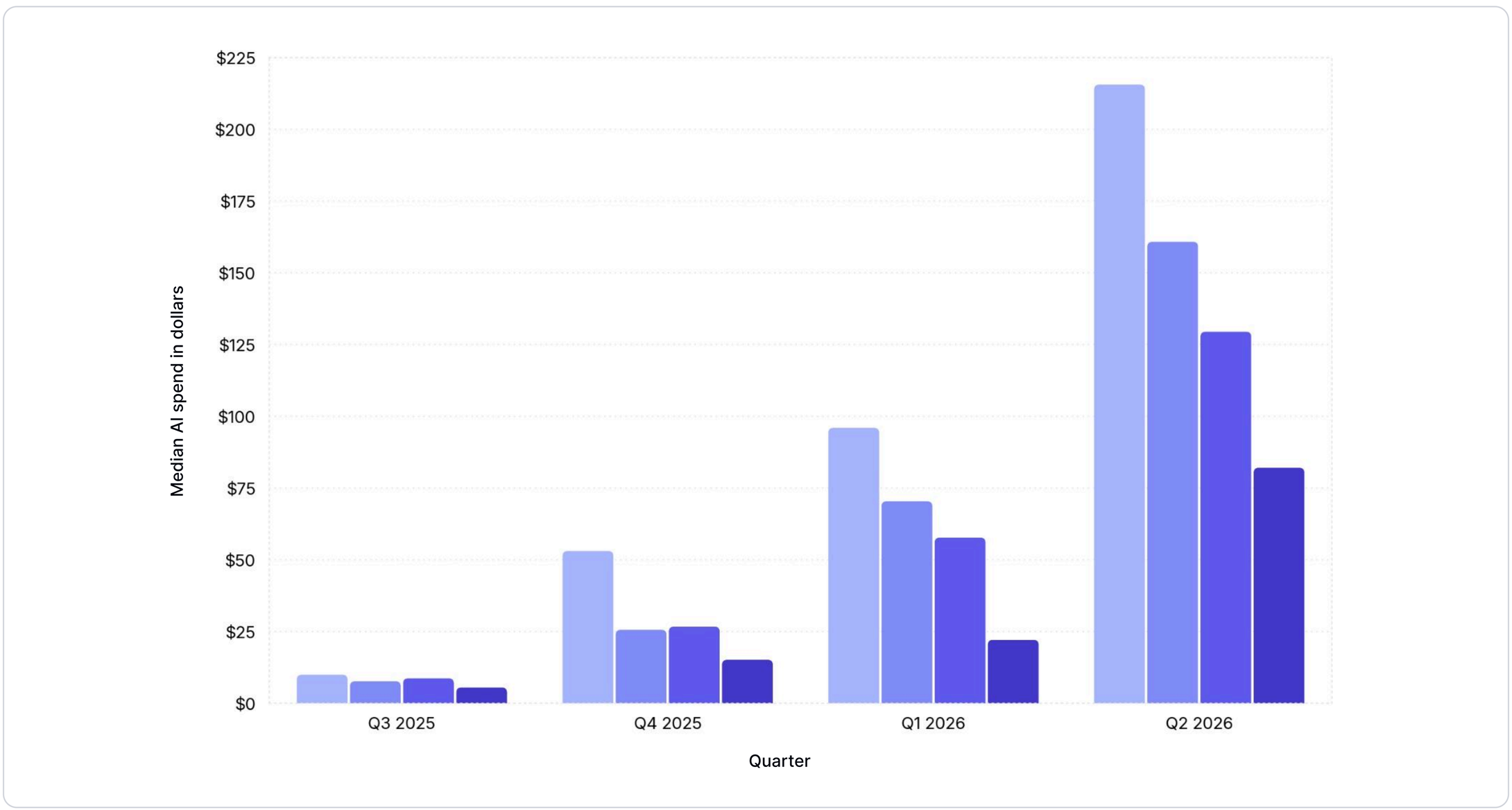
Median AI spend per user by organization type

Sample from 500+ companies from July 2025 to June 2026



Median AI spend per user by organization size

Sample from 500+ companies from July 2025 to June 2026



For DX users: Reports > AI cost > AI cost management

Closing thoughts: Moving from adoption to ROI

The Q2 2026 data indicates a transition phase for AI adoption, shifting from base deployment to evaluating the concrete return on investment. While AI assistants are saving developers an estimated 4 to 6 hours per week, these time savings have not yet reliably converted into higher-level business outcomes.

Specifically, the innovation ratio has remained essentially flat over the past year, plateauing around 58%. Furthermore, the financial investment in AI has escalated significantly, with Tech sector quarterly spend increasing nearly 29x from $3M to $87M. This cost acceleration is outpacing the proportional gains in system throughput.

Engineering leaders must now focus on tracking whether saved hours are being reallocated to product innovation or if they are simply being absorbed by existing organizational friction.

Managing code volume and quality

With 52.7% of all codebase contributions now authored by AI, the characteristics of software quality are measurably shifting across the industry. Developers report that while AI improves overall code maintainability, their confidence in making changes to the system without breaking things has actively declined.

This bifurcation in key Developer Experience Index drivers aligns with objective system metrics demonstrating that pull request sizes have nearly doubled over the past year. The combination of larger batches of code and increasingly diffuse code provenance means organizations are facing evolving forms of technical debt.

While deployment frequencies have increased across most segments, change failure rates remain highly volatile, suggesting some teams are shipping defects at a proportionally faster rate. To address this, leaders should scale their quality infrastructure to match code generation speeds by incorporating AI-driven testing, standardized prompts, and updated code review protocols.

Optimizing flow and delivery

Despite objective acceleration in TrueThroughput and deployment frequency, developer perception of delivery speed remains flat at approximately 70%. This plateau indicates that the downstream phases of the software development lifecycle, such as CI/CD wait times and review latency, have become the primary constraints on delivery.

Interestingly, organizational scale also presents a friction point for AI ROI. Smaller organizations (SMBs), despite paying a premium per developer for AI seats, are currently realizing the highest throughput gains per dollar spent. In contrast, enterprise organizations continue to lag in throughput growth, indicating that scale and process overhead ultimately dampen the effectiveness of AI tooling. For engineering leaders, the strategic focus must shift from simply acquiring AI tools to optimizing the surrounding platform and development pipelines to ensure the generated code can be delivered and maintained efficiently.

Methodology

Sample

The dataset represents a combined sample of developers across 500+ engineering organizations that use the DX platform. Companies range in size from fewer than 50 to over 10,000 employees, represent a wide geographic spread from Europe to Asia to North America, and encompass many industries, including Finance, Retail/E-Commerce, and Healthcare. Data covers the period of April 1, 2026 to June 30, 2026. Where trend comparisons are shown, historical data extends back to Q3 2025.

Company and industry classification

Sector

- **Tech companies:** Organizations whose primary business is building and selling software or technology products and services. This includes software development, IT services, cloud infrastructure, and platform companies.

- **Traditional companies:** Organizations in non-technology-native industries where software engineering supports the core business. Examples include Retail/E-Commerce and Manufacturing.

Geographic regions

Data is disaggregated by the following regions based on company headquarters location: North America (broken down by Silicon Valley and non-Silicon Valley), Europe, Asia-Pacific, and Latin America.

Organization Size

- **Small to mid-sized organizations:** Fewer than 100 engineers
- **Medium-sized organizations:** 100–749 engineers
- **Large organizations:** 750+ engineers

Where finer segmentation is shown (e.g., per-developer spend), ranges such as 15-99, 100-299, 300-749, and 750+
are used.

Methodology

Key metrics and data sources

This report draws on two complementary data sources:

- **System metrics:** Quantitative metrics collected automatically from source control, CI/CD pipelines, AI coding tools, and other engineering systems. Includes TrueThroughput™, deployment frequency, PR size, time to 10th merged PR, and weekly active AI tool users.
- **Survey metrics:** Developer feedback and experience data collected through DX's research-based survey instruments. Includes the Developer Experience Index (DXI) and its 14 driver dimensions, self-reported time savings, perceived rate of delivery, perceived software quality, change confidence, code maintainability, ease of delivery, and innovation ratio.

Analytical frameworks

Findings are organized around two frameworks:

- DX Core 4: Measures engineering productivity across four outcome dimensions: Speed, Effectiveness, Quality, and Impact. It synthesizes key principles from DORA, SPACE, and DevEx into a unified methodology designed for executive decision-making.
- AI Measurement Framework: Provides diagnostic context for AI-specific telemetry across three categories: Utilization (adoption and AI-generated code share), Impact (time savings and developer satisfaction), and Cost (total and per-developer AI spend).

Limitations

1. Self-reported time savings and developer experience data are subject to response bias. To mitigate this, we triangulate survey responses with system-level telemetry to validate directional trends, rather than relying on any single data source.
2. Company-level aggregation means individual team variation within organizations is not captured. Where possible, we segment by organization size, industry, and region to surface meaningful differences and avoid overgeneralizing across dissimilar contexts. In other research insights, we also examine the effect of AI on individual behavior to view user-level trends.
3. The sample skews toward organizations that have invested in developer productivity measurement platforms. While this means the findings may not generalize to all engineering organizations, it does ensure a high-quality dataset with consistent measurement practices, making the data especially relevant for organizations evaluating or already using similar platforms.

About DX

DX is an engineering intelligence platform designed by leading researchers to help organizations thrive in the AI era. By unifying system metrics with real-time developer and AI agent feedback, DX empowers engineering leaders and platform teams to continuously improve software health and productivity. DX serves hundreds of the world's most innovative companies, including Dropbox, Block, Pinterest, and BNY, to drive higher ROI across their entire engineering workforce.

getdx.com

DX Research team

Justin Reock

Deputy CTO

Justin is an engineer, speaker, writer, and software practice evangelist with over 20 years of experience.

Gratiana Fu

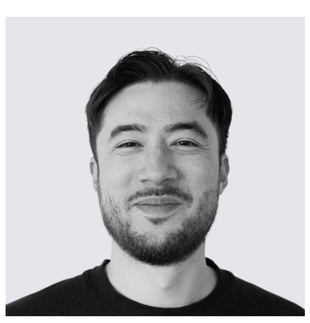
Research Analyst

Grace is a data scientist and analyst at DX with nearly 10 years of experience turning data into actionable insights.

Brian Houck

Distinguished Scientist

Brian co-authored the SPACE framework, now one of the most widely adopted approaches for measuring developer productivity.



Abi Noda

CEO

Abi is the Co-Founder and CEO of DX, where he leads the company's strategic direction and R&D efforts.

Engineering intelligence

Learn more at getdx.com

Copyright © 2026