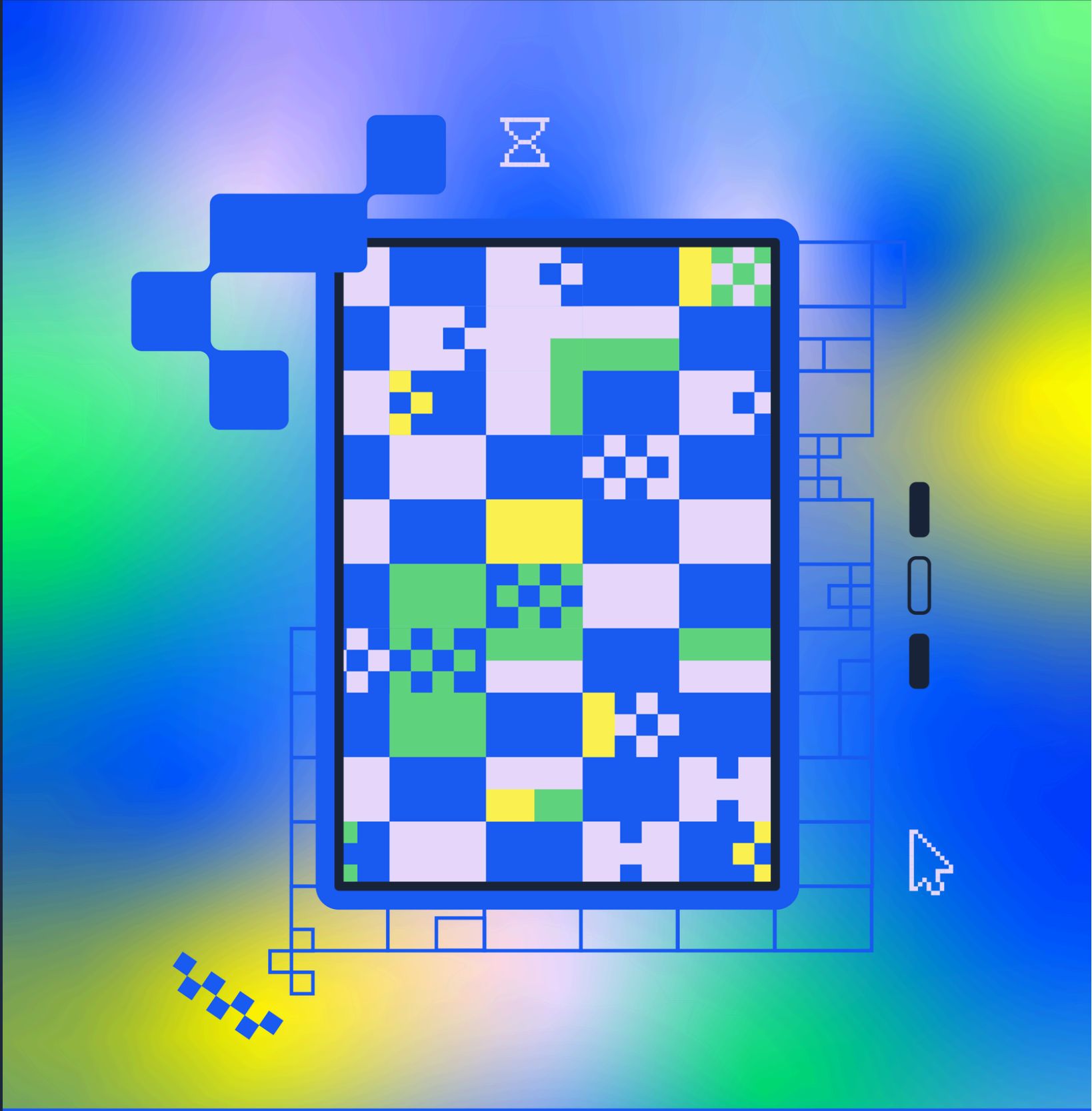


2026

State of Software Delivery

circleci

Sponsored by thoughtworks

// Contents

02 Executive summary

03 Some teams are winning in the AI era. Others are falling behind.

07 What's creating the AI performance gap?

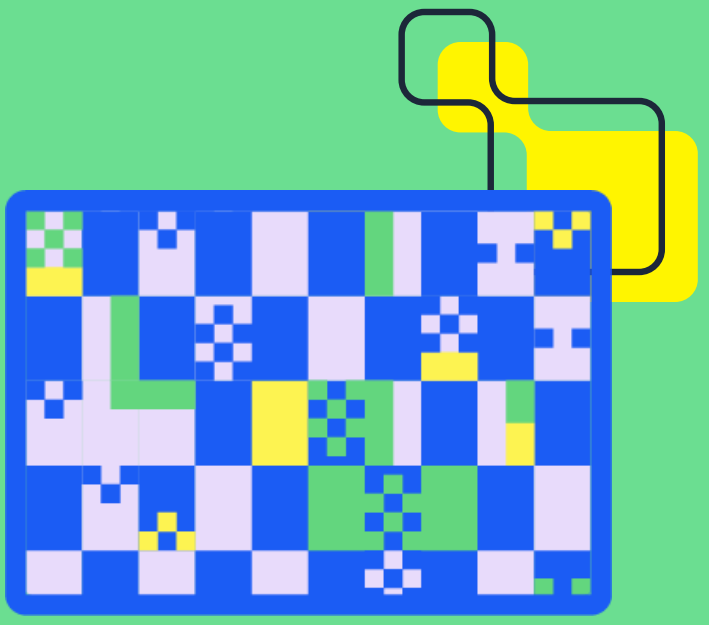
10 Complexity is the momentum killer

15 What winning teams look like

22 Break the AI bottleneck with autonomous validation

24 Summary of 2026 software delivery metrics

25 Methodology



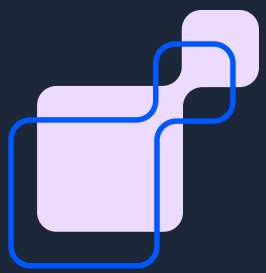
// Executive summary

Claims about AI transforming software development have circulated for years, often without clear proof. **TODAY WE CAN CONFIDENTLY SAY THAT ITS IMPACT IS VISIBLE IN REAL DELIVERY DATA.**

Analysis of more than 28 million CI workflows across thousands of teams shows a sharp increase in development activity. Teams are producing more code than ever before, driven largely by growing adoption of AI-assisted development and autonomous coding agents.

But AI acceleration is not benefiting all teams equally.

This year's data shows a widening gap between organizations that can effectively transform AI-driven acceleration into measurable business results, and those that cannot. A small group of top performers is shipping faster and more reliably than ever, but most teams are experiencing the opposite. Rising complexity is slowing code review, increasing failure rates during integration, and extending recovery cycles when things break. As a result, despite more code being written, a smaller number of changes are reaching customers.



THERE IS GOOD NEWS: The data also shows that this performance gap is not inevitable. Top-performing teams showed that higher change volume can translate into more shipped changes when validation keeps pace with the speed, scale, and context of AI-driven code generation.

Key Findings

- Average throughput increased 59% year over year, reflecting a sharp rise in AI-driven code generation.
- Throughput gains were heavily concentrated among top-performing teams, while the bottom half saw little to no improvement.
- Key stability indicators show that AI-driven changes are breaking more often and taking teams longer to fix, making validation and integration the primary bottleneck.
- A small minority of teams demonstrated that change volume and delivery stability can scale together, proving that with the right systems in place, AI can accelerate work across the entire delivery pipeline.

The data points to a clear conclusion: success in the AI era is no longer determined by how fast code can be written. The decisive factor is the ability to validate, integrate, and recover at scale.

The sections that follow unpack these findings and point toward how delivery systems must evolve to turn AI speed into real business outcomes.

//Some teams are winning big in the AI era.

Others are falling behind.

AI is turning up the volume on change

Across all projects building on CircleCI, average throughput (the number of daily workflows run) grew **59% year over year**. AI-powered code generation and agent-driven workflows are enabling teams to explore ideas, generate features, and iterate at unprecedented speed.

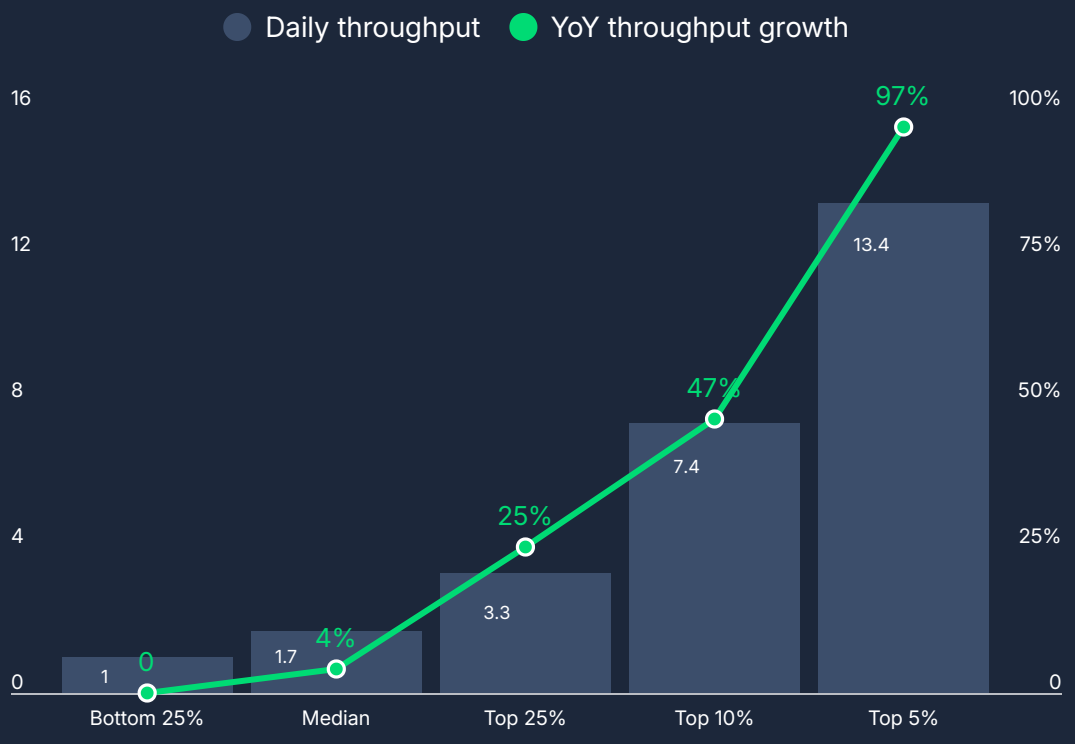
But most productivity gains are concentrated at the top

A closer look at the throughput data shows that year-over-year improvements are most noticeable among the **top 25% of teams.**

59%

increase in average throughput

Daily throughput and YoY growth trend by performance band

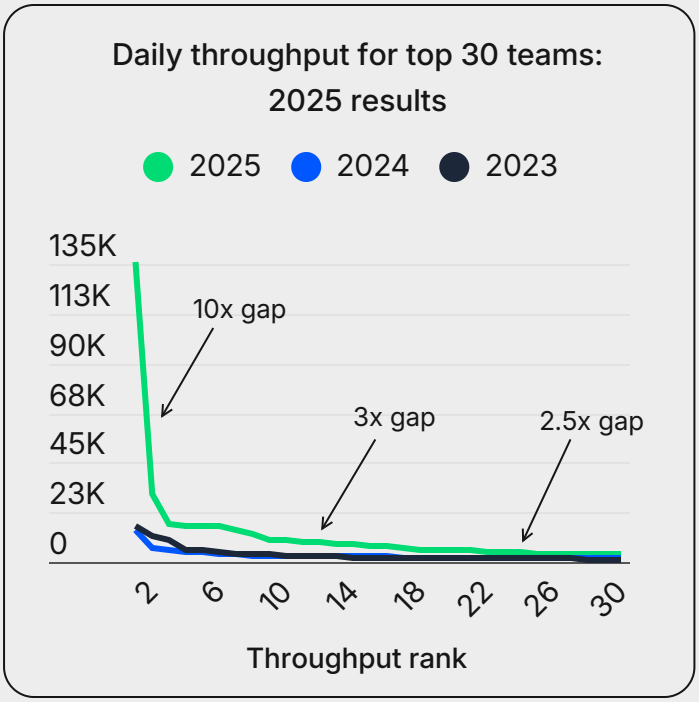


The top 5% of teams nearly doubled their throughput year over year, from 6.8 to 13.4 daily workflow runs. The top 10% and 25% of teams saw smaller but still significant increases of 47% and 25%, respectively.

The median team, meanwhile, increased throughput by just 4%, while the bottom 25% saw no measurable increase.

Even among elite performers, the gap is widening

Looking at a small sample of ultra-high-performing teams, productivity gains continue to be highest among the top performers.



All 30 of the most productive teams on CircleCI outperformed last year's equivalent. That finding is consistent with the overall increase in throughput seen at the top end of the data.

However:

- This year's most productive team delivered roughly 10x the throughput of 2024's leader
- At positions 5-15, throughput was approximately 3x the previous year's
- At positions 20-30, throughput was up ~2.5x

These results mirror the broader dataset, but in a more extreme form: AI-driven acceleration is amplifying existing delivery strengths, allowing the highest-ranked teams to pull even further ahead.

Scale separates winning teams from the rest

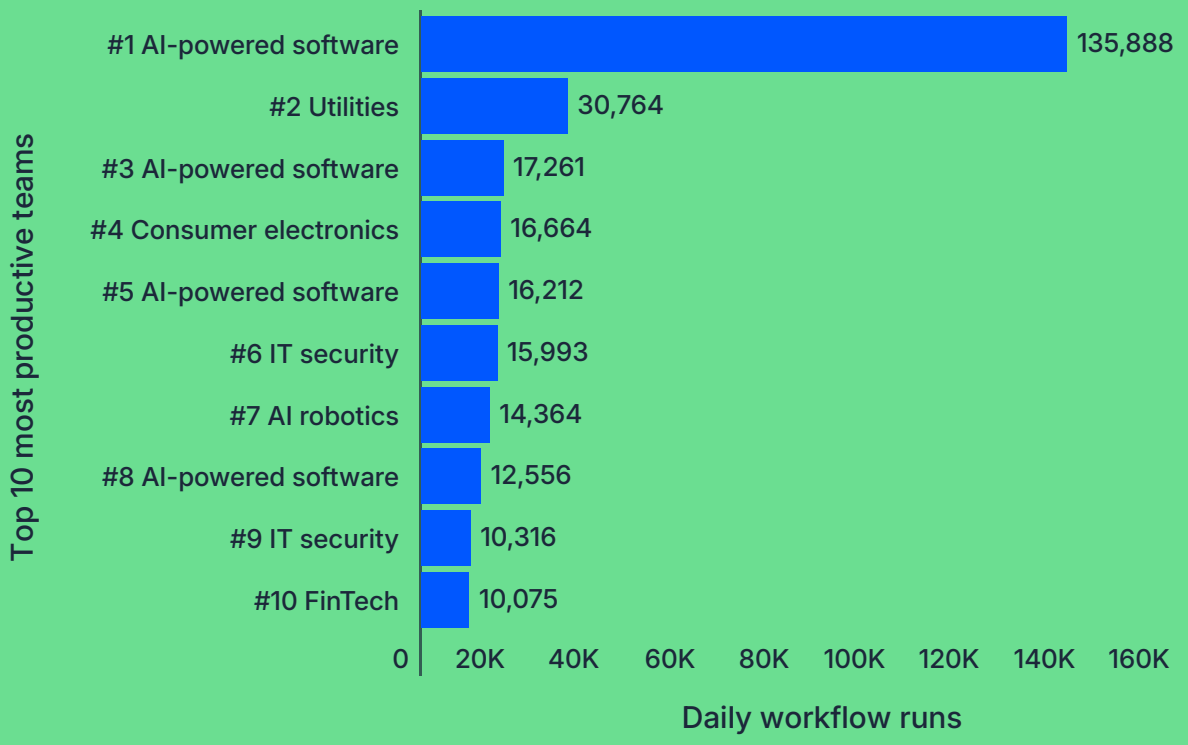
Elite teams are now operating at levels that would overwhelm many traditional delivery pipelines.

Each of the top ten teams on CircleCI validates more than 10,000 changes every day. The most productive team ran approximately 136,000 workflows per day, an order of magnitude increase over the highest daily throughput previously observed in the dataset.

136K

workflows per day

Highest throughput organizations on CircleCI



At this scale, the bottleneck is no longer how fast developers can work. When teams are validating tens of thousands of changes per day, delivery success depends on whether the underlying systems can keep up.

//So what's creating the performance gap?

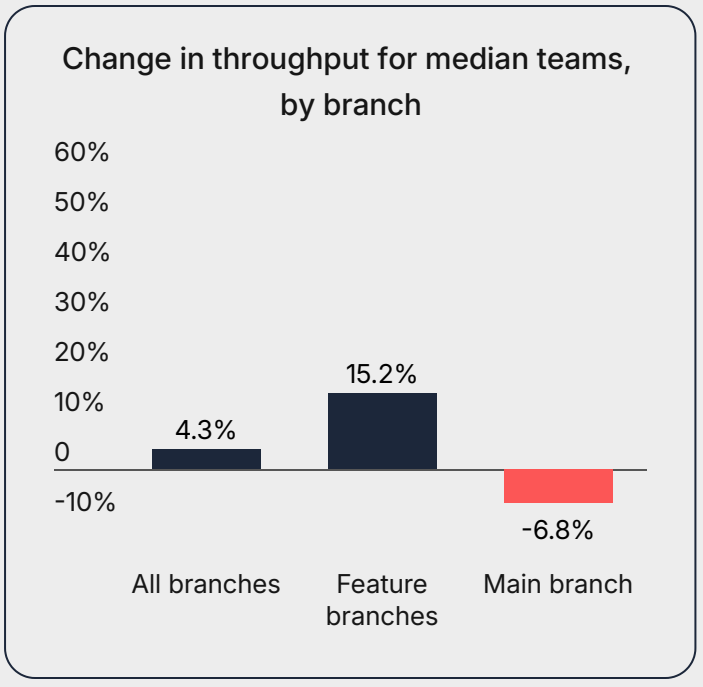
And more importantly, **how can you get ahead?**

For most teams, integration is the biggest bottleneck

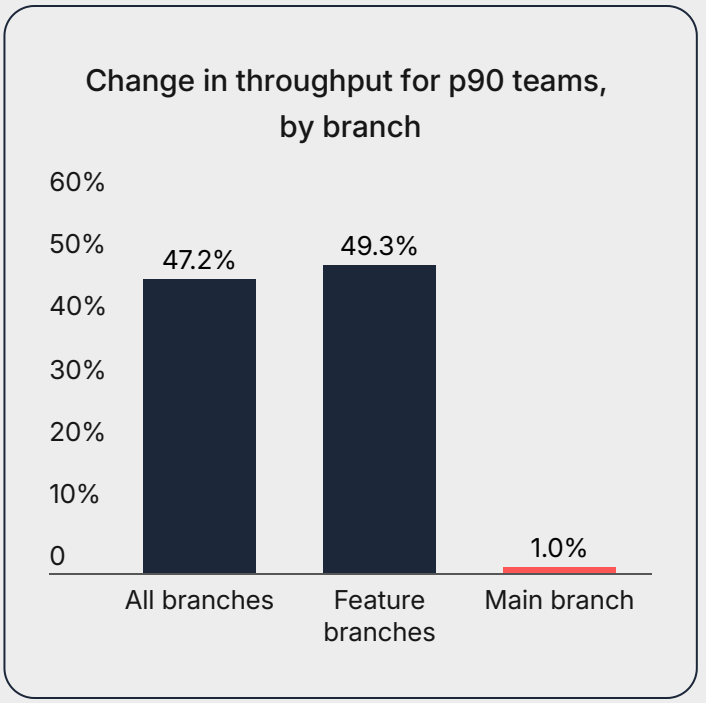
Although the majority of teams saw little to no increase in overall throughput, it's important to note that doesn't mean they are getting no benefit whatsoever from AI-driven development. In fact, most teams saw a noticeable increase in throughput on feature branches, where AI can support rapid experimentation, prototyping, and iteration.

The challenge starts when those changes need to be integrated into mainline production code.

The median team, for example, saw a 15% increase in throughput on feature branches. Throughput on the main branch, however, declined 7% from last year, indicating new difficulties reviewing, validating, and promoting AI-generated changes into shared code.



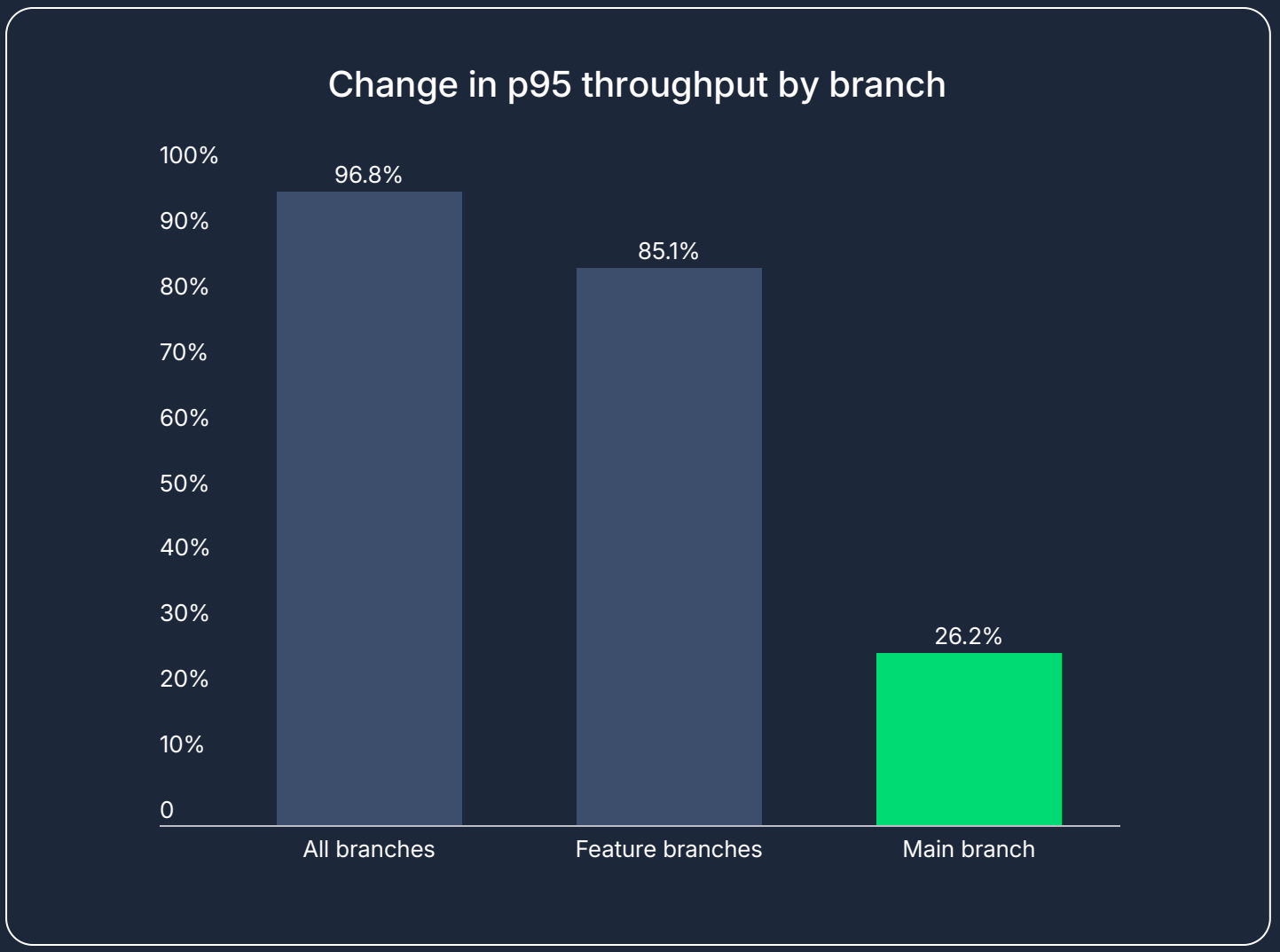
Even teams in the top 10% struggled to move changes into production. Throughput on feature branches increased by almost 50% for this group, while main branch activity was essentially flat.



The vast majority of teams are struggling to translate increased development activity into shipped changes.

The highest performing teams, however, are accelerating everywhere

In contrast to the bottom 95%, the top 5% of teams are writing and shipping substantially more code.

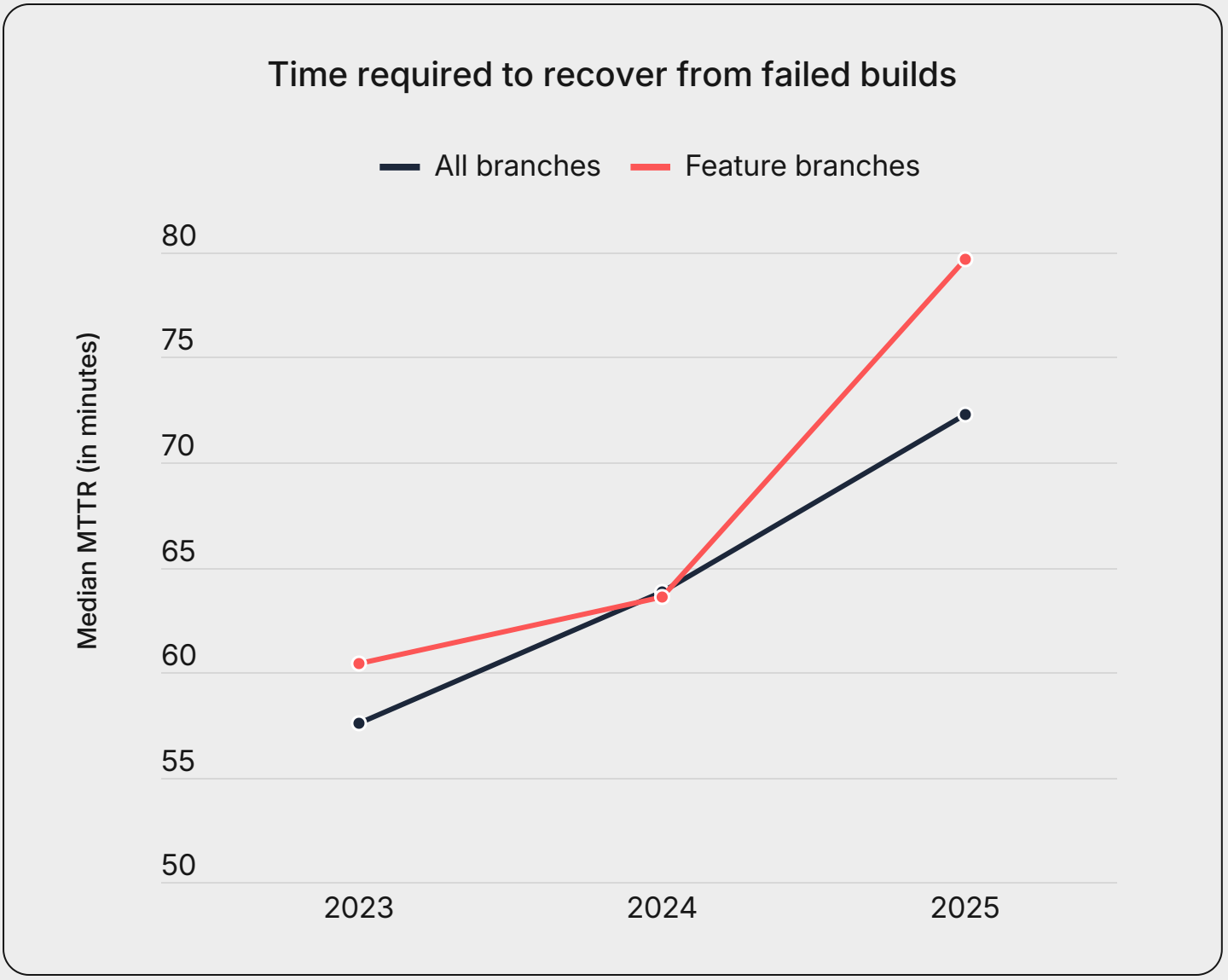


For this group, throughput on the main branch increased 26% while feature branch activity grew by 85%.

These teams are converting raw coding speed into real, deployable changes. But they're the exception, not the rule. Fewer than 1 in 20 teams are currently able to absorb AI-driven acceleration without slowing down their delivery pipelines. Their results show what's possible but also highlight how far the majority of teams still have to go to turn AI-enabled output into shipped software.

//Complexity is the momentum killer

One of the biggest challenges teams face in integrating AI-generated changes is debugging unfamiliar code when a build fails. Since generative AI began to gain traction at the end of 2022, we've been tracking an increase in the time required to recover from failures.



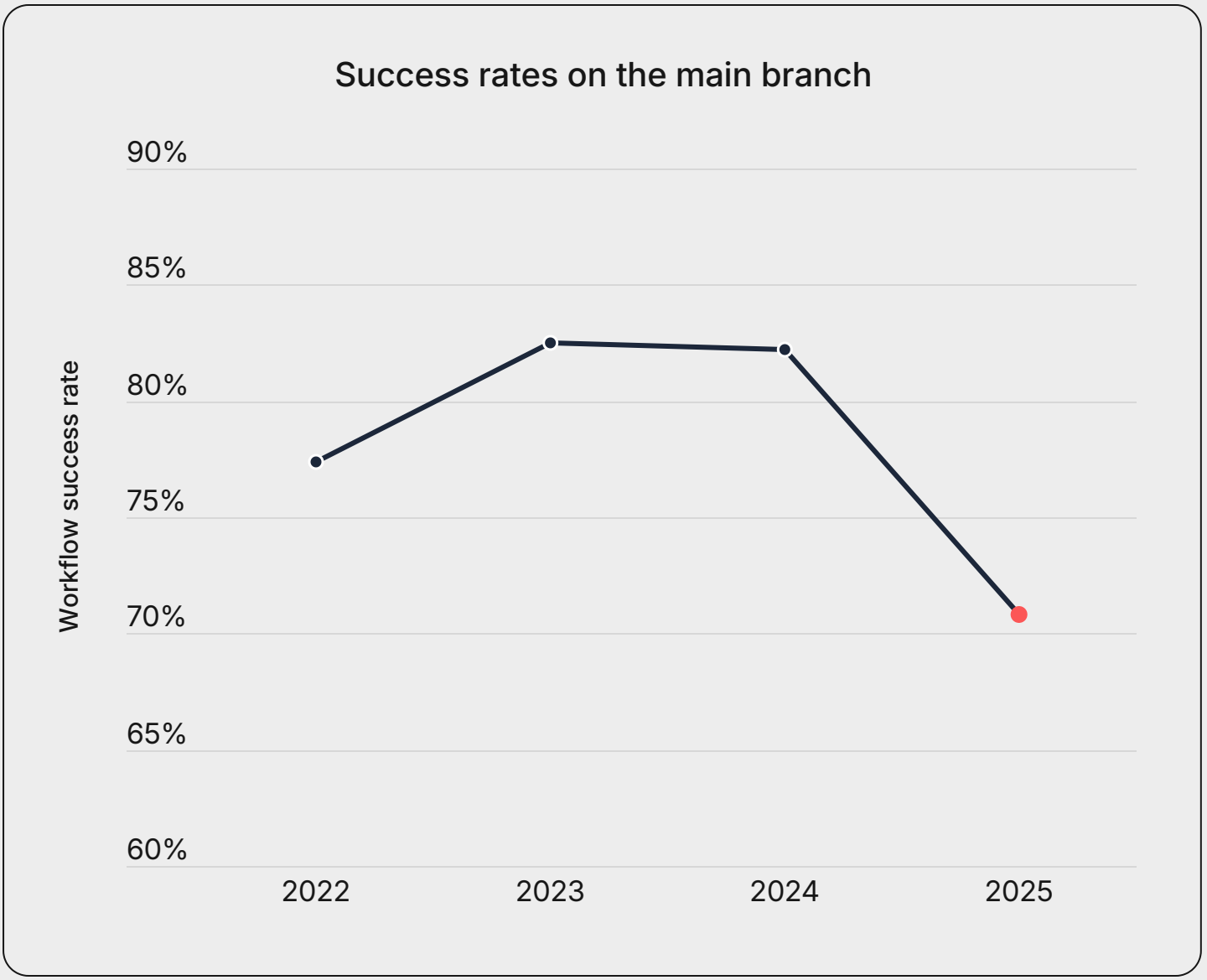
This year, the typical team takes **72 minutes** to get back to green, a **13% increase** in recovery times from last year. Recovery times are even longer on feature branches, where test suites generally deliver less comprehensive feedback. MTTR on those branches increased 25%, from 64 minutes in 2024 to nearly 80 minutes in 2025.

AI-assisted code isn’t just harder to fix. It breaks more often, too.

Another measure by which we can tell complexity is killing AI productivity for most teams is that success rates on the main branch fell to their lowest level in over 5 years, to 70.8%. In other words, attempts at merging changes into production code bases now fail 30% of the time.

While CI failures aren’t always negative (we want to catch real issues before they reach production, after all), this result is well below our recommended benchmark of 90%, a level that indicates you're catching issues efficiently without being overwhelmed by bad code.

A rapid decline like this suggests code quality is starting to suffer.



What's the impact?

Declining success rates and rising recovery times can cause a wide array of negative impacts on software teams, ranging from wasted engineering time and missed delivery deadlines to higher burnout, runaway costs, and frustrated customers.

For a team pushing 5 changes to the main branch per day, going from a 90% success rate to 70% is the difference between one showstopping breakage every two days to 1.5 every single day (a 3x increase).

At just 60 minutes recovery time per failure, you're looking at an **additional 250 hours in debugging and blocked deployments** every year. And that's at a relatively modest scale. Teams pushing 500 changes per day would lose the equivalent of **12 full-time engineers.**

90% → 70% at 60 minutes recovery time per failure

=

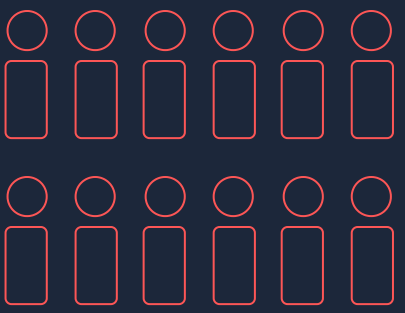
+250h

in debugging and blocked deployments every year

500

changes / day

=

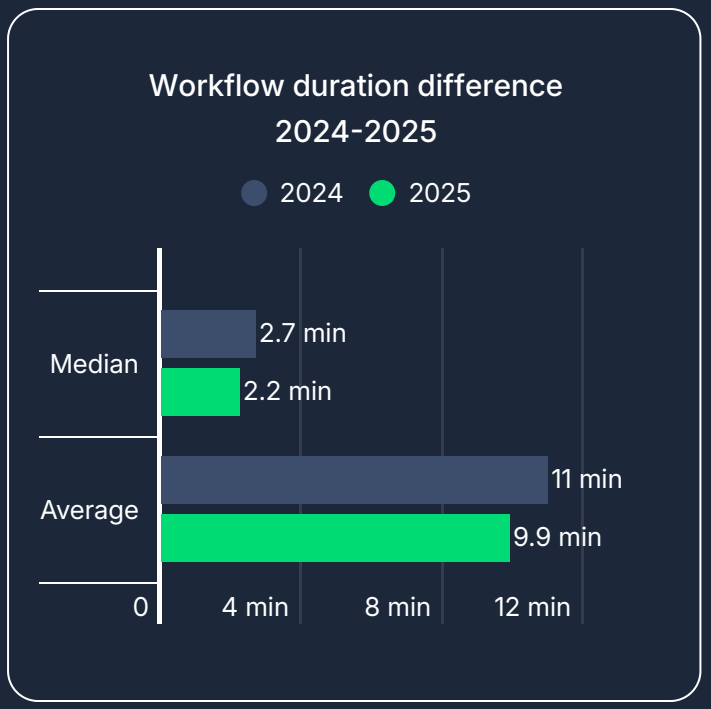


The loss of 12 full-time engineers

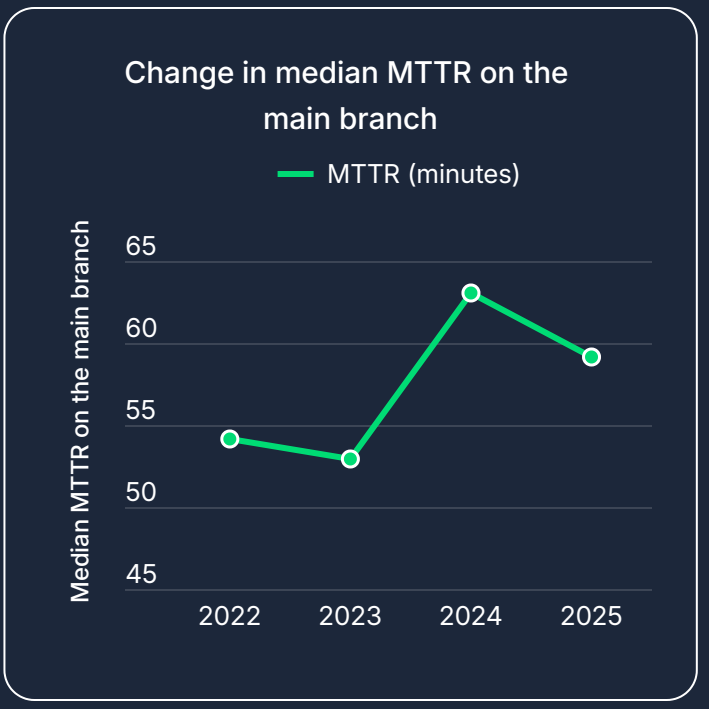
Early signs of adaptation

While the data clearly shows teams are feeling the strain of rising complexity, there are also signs that some are beginning to make targeted improvements to keep work moving.

Workflows are faster overall this year, with the median duration standing at 2.2 minutes (18% faster than last year) and the mean clocking in at 9.9 minutes (10% faster than last year).

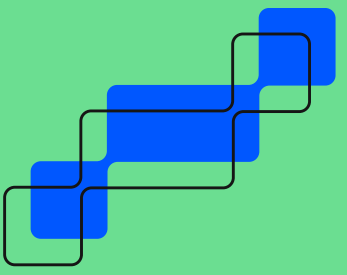


And that faster feedback is paying off where it matters most, helping teams focus their recovery efforts on the critical path to production. Even as MTTR increased overall, it improved slightly on the main branch, falling back to 59.2 minutes—below the 60-minute benchmark—after rising sharply last year.



Shorter feedback and recovery cycles can help alleviate the strain created by higher change volume. Even modest improvements can make a meaningful difference as AI continues to increase both the pace and complexity of development.

What winning teams look like



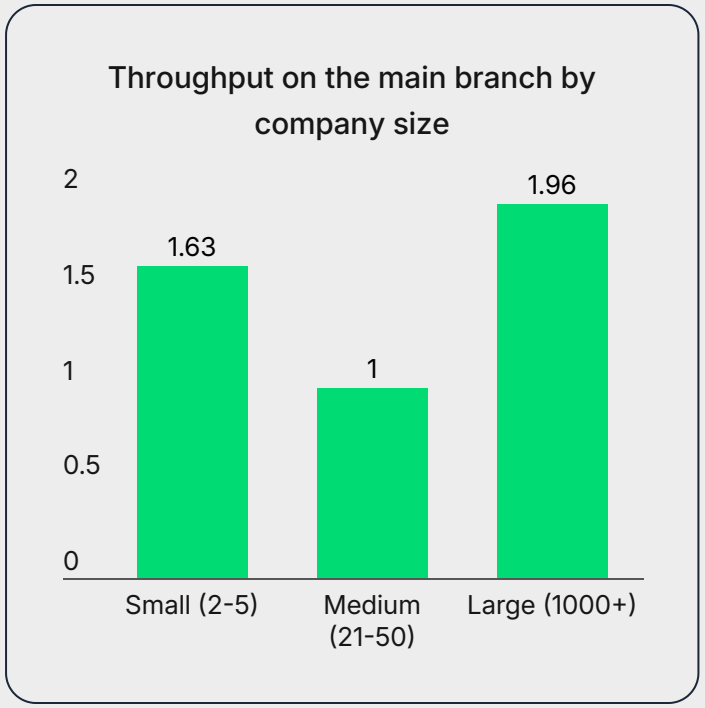
AI is reshaping patterns of software delivery performance. The data shows that success is not evenly distributed, nor strictly a function of team size, headcount, or geography. Instead, performance varies based on how teams are structured to manage complexity, validate change, and recover quickly when things break, producing clear differences across organizational contexts.

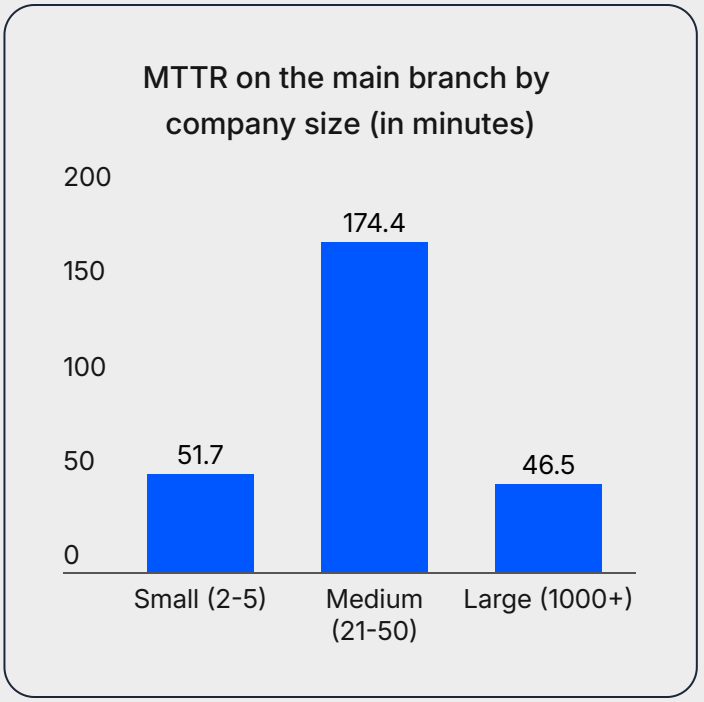
Looking at delivery outcomes across factors like company size, team size, industry, and region helps clarify what high-performing teams have in common, and why some groups are better positioned to benefit from AI-driven acceleration than others.

Company and team size

Both company size (total number of employees) and team size (total number of contributors on a given project) can affect an organization's ability to meet its business goals. For both characteristics this year, performance appears to be roughly U-shaped.

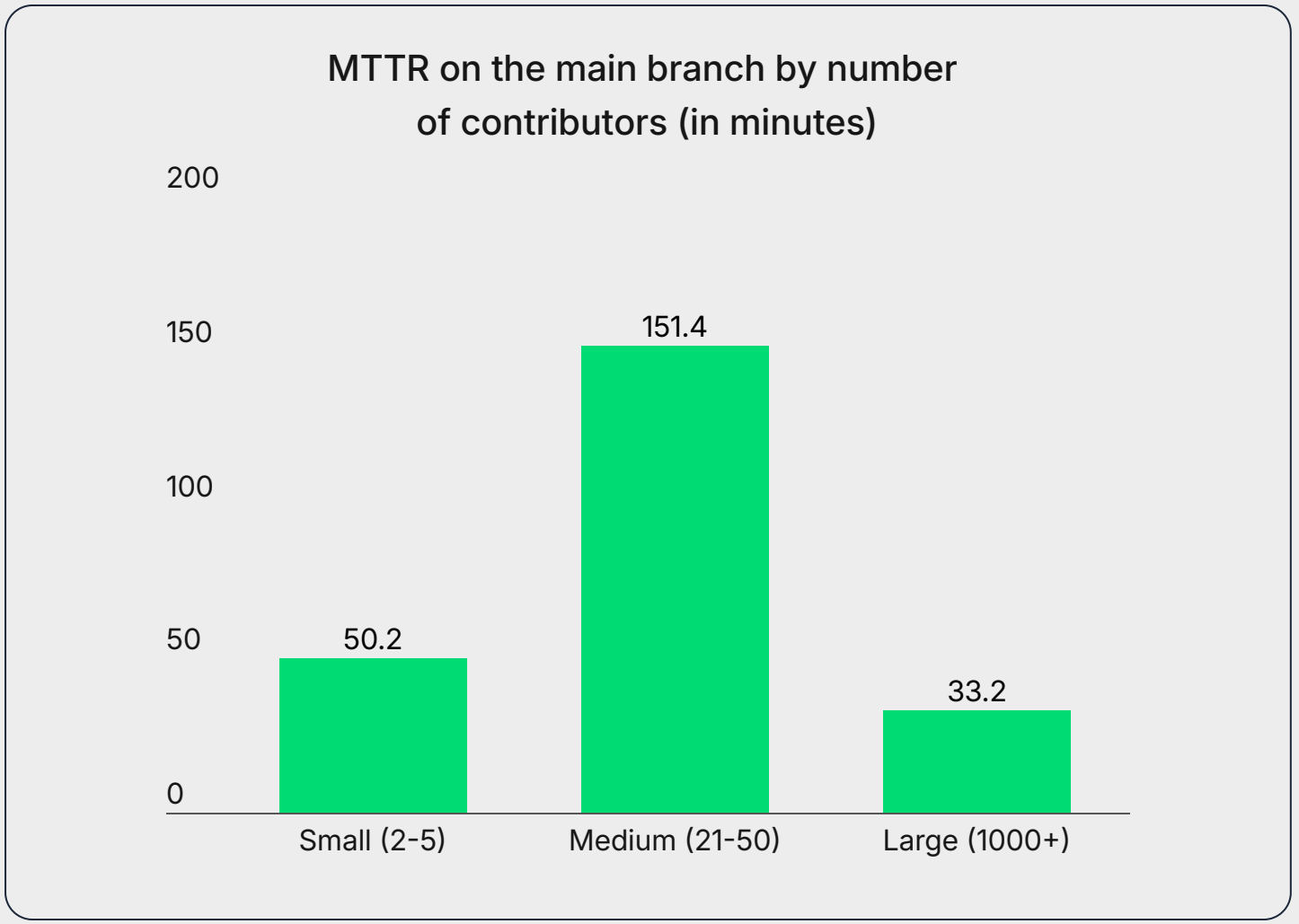
Company size: The smallest companies (2-5 employees) and the largest enterprises (1000+ employees) perform best overall. They have the highest levels of main-branch throughput and the lowest recovery times on the main branch.





Mid-sized companies (21-50 employees) struggle the most, with the lowest throughput of any segment and recovery times of nearly 3 hours. This suggests a potential "messy middle" problem, wherein companies have outgrown the speed and simplicity of small teams but have not yet built the systems and practices needed to operate effectively at larger scale.

Team size: While the number of contributors to a project has a linear effect on throughput (that is, more contributors tends to mean more pipelines run), when it comes to recovery times, team size has an effect similar to company size: small and larger teams significantly outperform mid-size teams on MTTR.

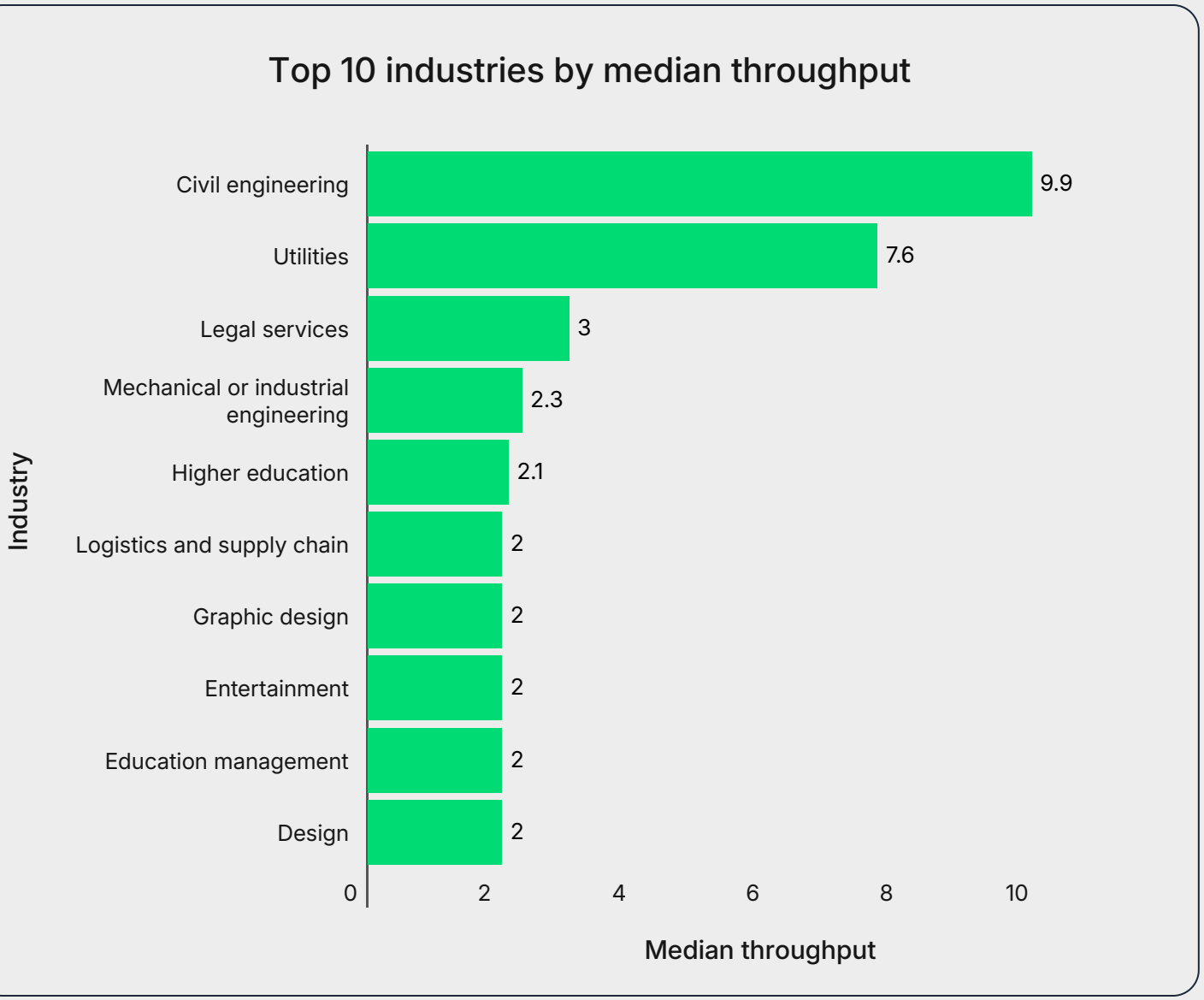


Teams with 2-5 contributors repair failed builds on the main branch in 50 minutes. Those with 1000+ contributors recover in just 33 minutes. Those with 21-50 contributors, however, take more than 150 minutes.

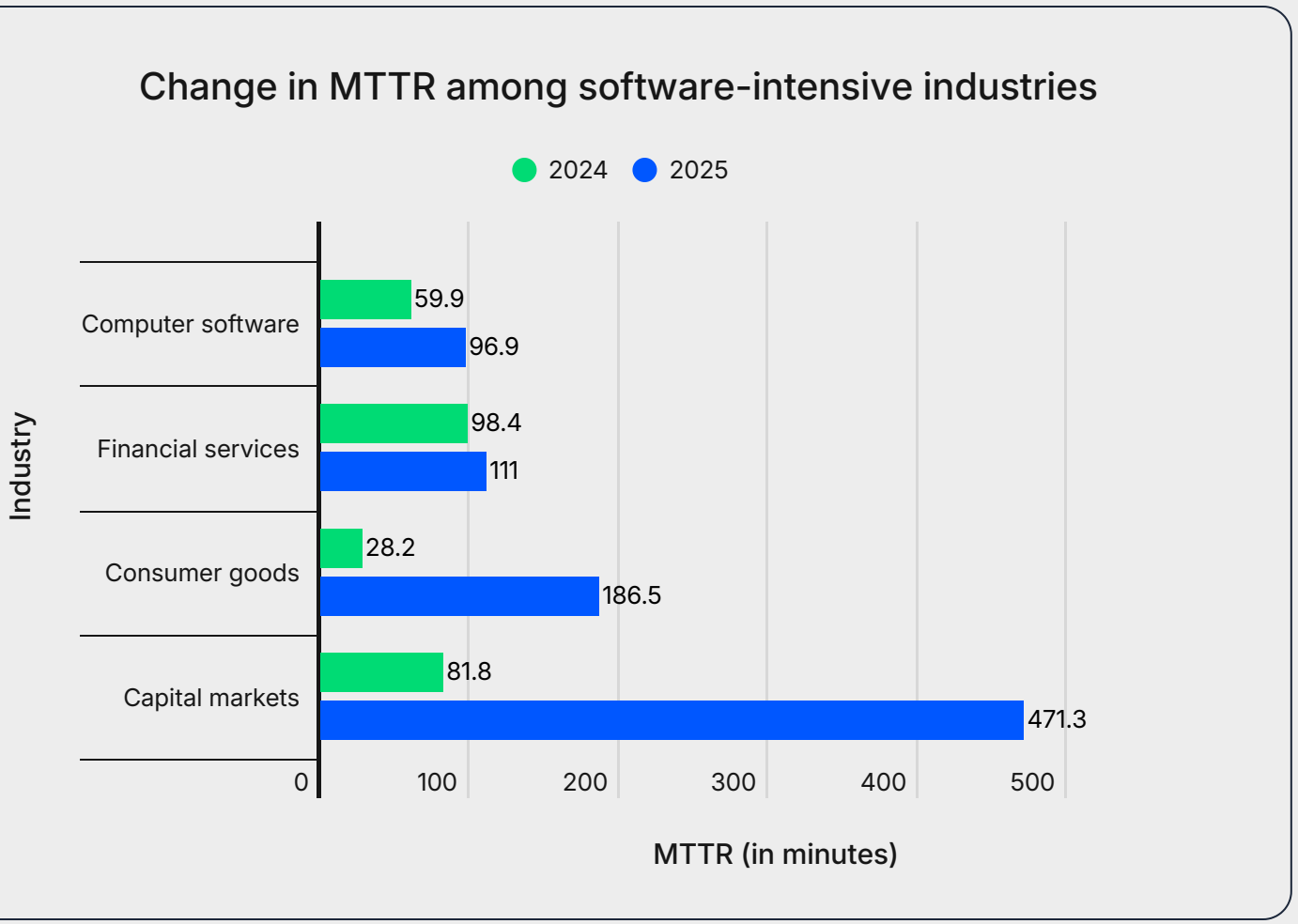
Industry

Industry data supports the idea that performance begins to suffer as code complexity grows.

High-throughput industries, for example, are not necessarily the most software-focused. Industries like civil engineering, utilities, logistics, education, and design rise to the top because much of their work involves small, frequent, predictable workflows like checking structured data and automating outputs.



Sectors with larger, more complex workflows like computer software, financial services, and capital markets all run fast workflows and produce steady throughput, but they are also seeing the biggest jump in recovery time. Many of these teams recovered in under an hour last year but now take 90 minutes or more as AI-generated changes reach deeper into their systems and create failures that take longer to address.

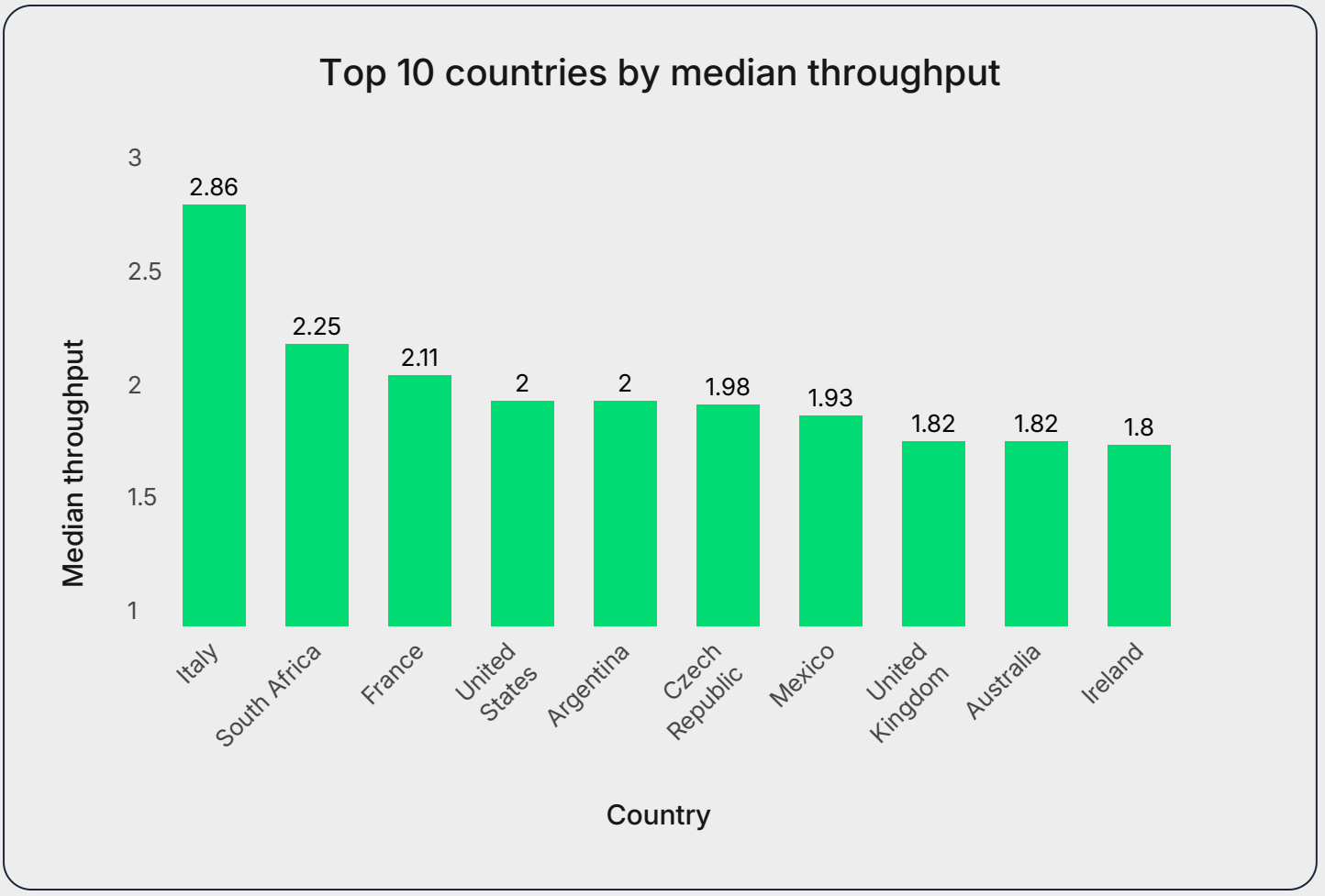


Region

Regional data shows that AI has made it easier for teams everywhere to write code faster, but delivery results still vary by region.

The highest levels of throughput (p95) are still concentrated in major tech hubs like the US, UK, Germany, Japan, and Canada.

But emerging regions are leveling the global playing field at the median (p50) level, with countries like Italy, South Africa, Argentina, and the Czech Republic outperforming major hubs. Smaller ecosystems often mean fewer legacy systems and a higher proportion of modern, cloud-native pipelines, so the “typical” team often moves faster even if the extremes don’t reach as high.



Where regions still differ is in recovery. The US, UK, and Australia fix broken builds far faster (often within 60–90 minutes), while teams in Europe, Asia, and Latin America take two to four times longer. AI is leveling access to development speed globally, but the ability to recover from failures quickly still determines whether teams can keep changes flowing into production.

// Benchmark yourself against the best

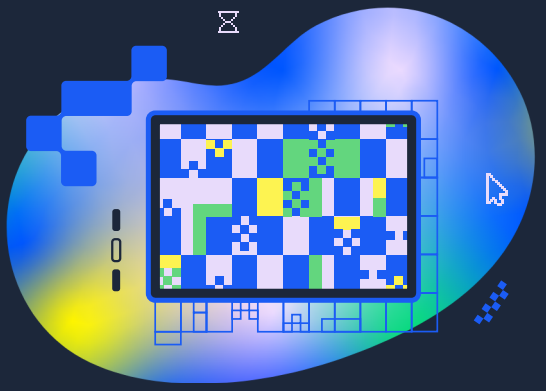
As AI reshapes engineering performance, knowing how your team compares is an essential first step for planning your next investments.

We created the Software Delivery Data Explorer so you can access the same dataset used in this report and see how your delivery performance compares to the industry.

Compare results by team size, industry, or region, and plot your own results for an at-a-glance view of where you stand relative to peers and top-performing teams.

Visit the data explorer to get started.

Explore the data →



//Break the AI bottleneck with autonomous validation

AI has accelerated how fast code can be created, but the systems responsible for building, testing, and deploying that code have not kept pace. The mismatch is now clearly visible in delivery metrics across the industry: more change entering pipelines, more workflows failing, and far fewer teams able to turn that activity into delivered software.

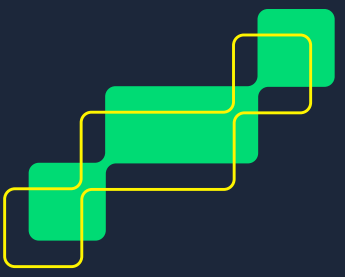
A small group is proving that the gap can be closed, but most are still struggling with slow feedback, manual workflows, and pipelines that can't adapt to today's volume or complexity.

CircleCI addresses this bottleneck with autonomous validation, moving beyond static scripts to bring context and intelligence directly to your CI/CD pipelines.

Rather than relying solely on fixed rules, autonomous validation uses signals from your build history to validate only what matters, identify sources of risk and friction, and work autonomously to resolve them before they slow your team down. The burden of maintenance shifts from engineers to the system itself, helping your team keep change moving even as volume and complexity increase.

Built on CircleCI's fast, cloud-native platform, autonomous validation operates at the scale AI-driven development demands. Teams can validate more changes per day, recover faster when things break, and focus engineering time on shipping instead of maintaining pipelines.

Ready to see what this looks like in practice? Contact CircleCI for a personalized demo, or sign up for a free account and start building the future today.



// Summary of 2026 software delivery metrics

Metric	Median	Average	Top 5%	Benchmark
Throughput	1.71	4.54	13.36	1+
Mean time to recovery	1hr 12m 18s	26h 32m 21s	1m 36s	1hr
Success rate	-	70.82%	-	90%
Duration	2m 13s	9m 56s	6s	10m

//Methodology

To create this report, we pulled data from over 28 million CircleCI workflows within the first 28 days of September 2025. In an attempt to restrict our analysis to real companies and repeatable workflows, we restricted the dataset to CircleCI projects that have at least 2 contributors (all time) and workflows that ran at least 5 times on CircleCI during the analysis period.

When analysis focuses on the default branch of the project, it is using the current value for the default branch, possibly missing some older data for projects that changed their default branch during the analysis window. Industry data is sourced from Clearbit and is not available for all organizations.

Data details:

- Every day between September 1, 2025 and September 28, 2025
- Only projects with more than one contributor
- Only workflows that ran at least 5 times
- 28,738,317 workflows

+28M

CircleCI workflows

Contributors

Report authors: Ron Powell, Jacob Schmitt
Report editor: Gillian Kieser
Report designers: Toni Cater, Birdy Wheeler-Wolowicz
Acknowledgments: Bijan Samimi