circleci

The 2025 State of Software Delivery

// Contents

// Executive summary

TODAY'S ENGINEERING TEAMS are under immense pressure to find an edge. Advancements like CI/CD automation, infrastructure as code, and AI-powered developer tools have raised the floor for what it means to be good at delivering software. At the same time, the gap between good and great has never been wider. True competitive advantage now lies in how effectively teams can make the leap to great by translating technical capabilities into measurable business outcomes.

This year's data reveals a stark divide between high and low performers. Here's how teams in the top 25% separate themselves from the pack:

- They ship updates 3x as fast as teams in the bottom quartile, turning deployment velocity into market advantage.
- They save millions in annual development costs by completing critical workflows 5x faster than low performers, freeing resources for strategic initiatives.
- They reclaim thousands of hours of innovation time each year by debugging in minutes rather than days, keeping developers in flow and focused on value creation.

These performance gaps aren't theoretical — they represent real competitive advantages that compound quarter over quarter.

"For too long, these metrics have been siloed in engineering dashboards. It's time to connect them directly to the outcomes that matter most to the business."

– JIM ROSE, CIRCLECI CEO

THIS YEAR, WE'RE TAKING A FRESH APPROACH to our annual analysis. While we'll examine the fundamental metrics that define elite performance — duration, throughput, mean time to recovery, and success rate — we're also framing performance metrics in terms that resonate with business stakeholders and clearly demonstrate ROI.

To make this concrete, we'll use **RecurShip**, a hypothetical AI-powered logistics company, as a case study to illustrate the business impact of improving software delivery. While fictional, RecurShip faces complexity and challenges that will feel familiar to many teams delivering mission-critical software at scale.

We'll use the following assumptions as baselines for the calculations we'll make throughout the report:

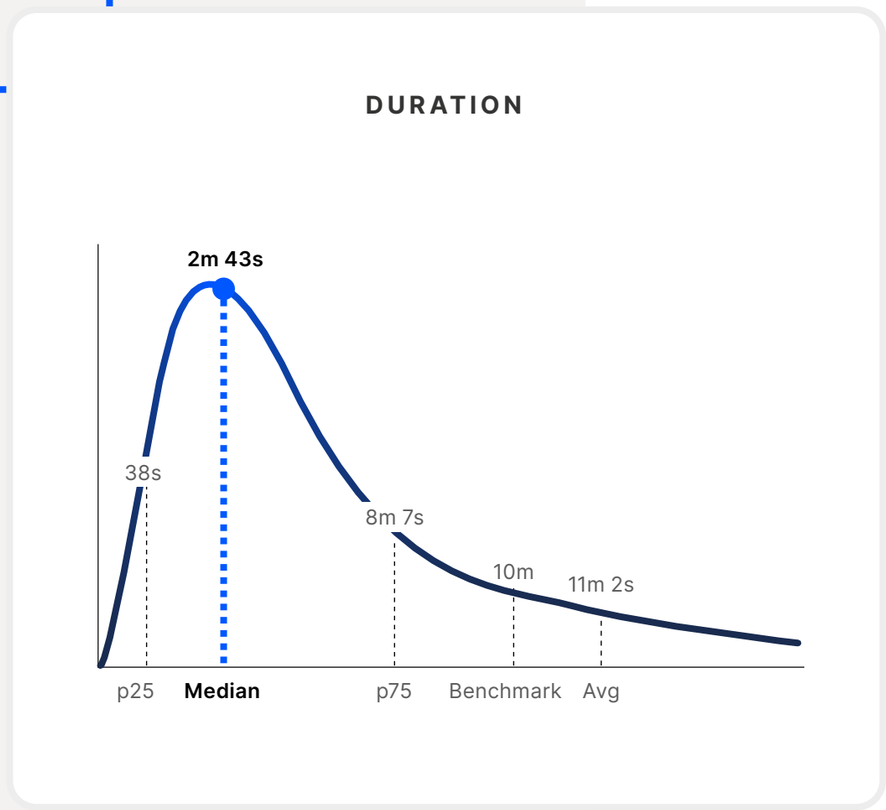
- **500 developers** working across global teams to maintain and enhance the core platform
- **3 commits per developer per week**, reflecting frequent updates and optimizations
- **$180,000 in total compensation per developer**, translating to **$1.50 per developer minute**

Using RecurShip as our proxy, we'll demonstrate how even small improvements in development efficiency translate into significant business value.

Let's explore each of these four key metrics, looking at not just where the bar for excellence stands today, but what achieving it means for your bottom line. Whether you're an engineering leader making the case for increased investment or an executive weighing competing priorities, you'll find clear evidence for the returns that elite software delivery performance can generate.

The four metrics that drive software ROI

Engineering benchmarks to boost performance and value



Duration

Duration measures the time from when an individual workflow is triggered until all of its jobs and steps are complete. It indicates how efficiently automated processes — like test suites, security scans, or deployment steps — execute within the broader software delivery lifecycle.

Long-running workflows create costly bottlenecks, negatively impacting both developer experience and productivity. When developers have to wait for feedback, they are either blocked from moving forward, or begin working on something else, losing valuable context in the process. These delays add up, directly affecting development velocity and deployment frequency.

This year's data shows a median duration of 2 minutes 43 seconds, with 25% of teams completing their workflows in under 38 seconds and 75% completing in approximately 8 minutes or less. This wide spread between quartiles reflects the diverse nature of workflow configurations: faster times often indicate lighter workflows with minimal validation steps, while longer durations suggest more complex testing and deployment processes. However, at the 95th percentile, workflows run for more than 25 minutes, revealing an overall 11-minute average in duration across runs. This means that many teams run workflows that far exceed our recommended 10-minute benchmark for balancing robust validation with developer productivity targets.

Let's examine the potential impact of optimizing durations for RecurShip, the hypothetical business we laid out at the beginning of this report:

- Current average duration: 20 minutes
- Target duration: 10 minutes

By reducing its average workflow duration from 20 to 10 minutes, RecurShip would save 10 minutes per run. With 300 daily runs across 250 working days, this improvement represents **750,000 minutes** of recovered developer time annually. At $1.50 per developer minute, optimizing workflow duration would yield **$1.1 million** in annual productivity gains.

Throughput

Throughput measures the average number of workflow runs on a project per day. It is a valuable indicator of productivity that can help organizations identify potential bottlenecks in the development process.

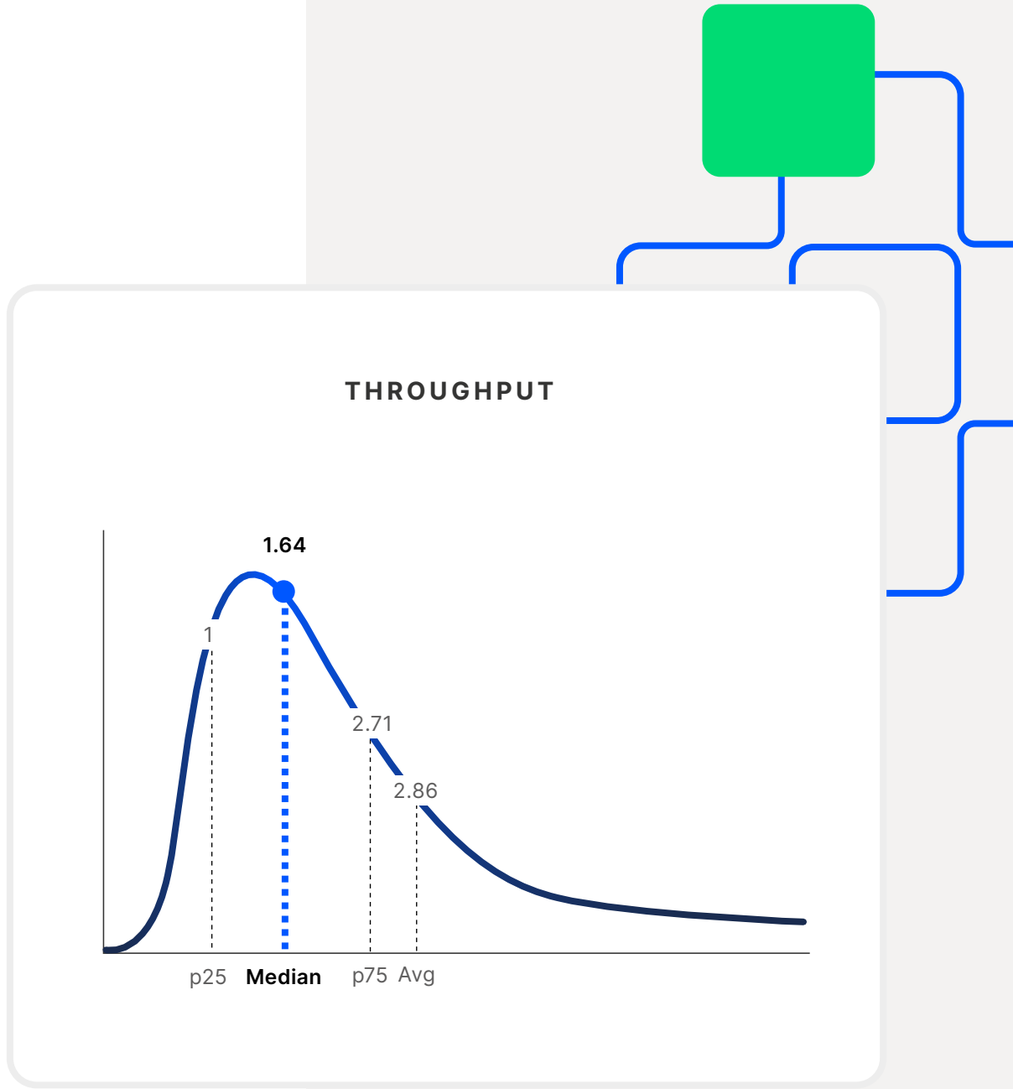
By tracking this metric over time, teams can understand if their tooling and practices are helping or hindering the developer experience, and identify where improvements can reduce friction in the development lifecycle. Normalizing throughput by team size allows organizations to focus on individual productivity trends independent of headcount changes or resource reallocations.

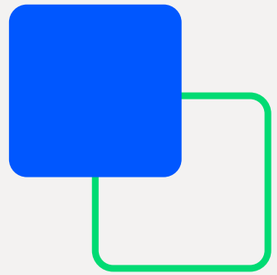
This year's data shows a median throughput of 1.64 workflow runs per day, with 75% of projects running at least one workflow and 25% executing 2.7 runs or more. At the upper end of the distribution, teams run hundreds, even thousands of workflows every day. Among the 20 most productive organizations on CircleCI, average daily throughput reaches 3,762.

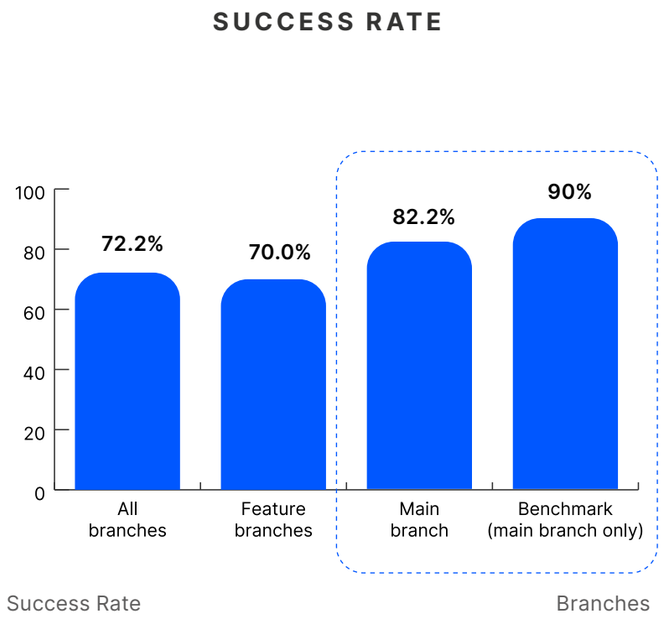
The delta between average and top performers suggests significant untapped potential in most software teams. Here's how a targeted investment in developer productivity can help close that gap:

Assume RecurShip, with its 500 developers running 300 workflows per day (0.6 per developer), invests in a dedicated platform team of 25 engineers to remove friction from development pipelines. Through their efforts, the platform team boosts throughput to 394 workflows (0.75 per developer) per day.

At this rate, RecurShip achieves a 25% improvement in individual productivity, far outpacing the 5% growth in headcount. Even accounting for the platform team hires, the productivity gains alone are equivalent to adding **156 full-time developers** at the original efficiency level — delivering the impact of **$28.4 million** in additional engineering headcount through a focused investment in CI/CD optimization and developer experience.







Success rate

Success rate measures the percentage of runs that complete without failure. While failures are expected (even useful) during development on feature branches, maintaining high success rates on the main branch is crucial for deploy-readiness and preventing team-wide bottlenecks.

Failures on the main branch are costly because they affect downstream teams waiting on changes and often require immediate attention from multiple developers to resolve. Success rates below our benchmark of 90% indicate potential gaps in testing and validation that can let problems slip through.

Our data shows an average main-branch workflow success rate of 82.15%. This suggests that teams are effectively catching most issues early through pre-merge checks, stable test suites, and consistent build processes. However, there is still room for improvement, as top-performing teams reach success rates closer to 90%.

Let's assume RecurShip, now running 525 workflows per day after a successful platform engineering initiative, has an average success rate of 75%. By improving its success rate from 75% to 90%, RecurShip would reduce failed builds from 131 to 53 daily. While this improvement saves tens of thousands in annual compute costs by eliminating the workflow reruns associated with the recovery effort, the real value comes from keeping development teams moving forward.

With each miscue potentially triggering a prolonged recovery process, eliminating 79 failed builds per day saves RecurShip thousands of engineering hours that would otherwise be spent debugging and resolving pipeline issues. Assuming an average recovery time of 4 hours, improving from 75% to 90% success rate would result in **78,750 reclaimed engineering hours** annually, at a value of **$7.1 million**.

Mean time to recovery (MTTR)

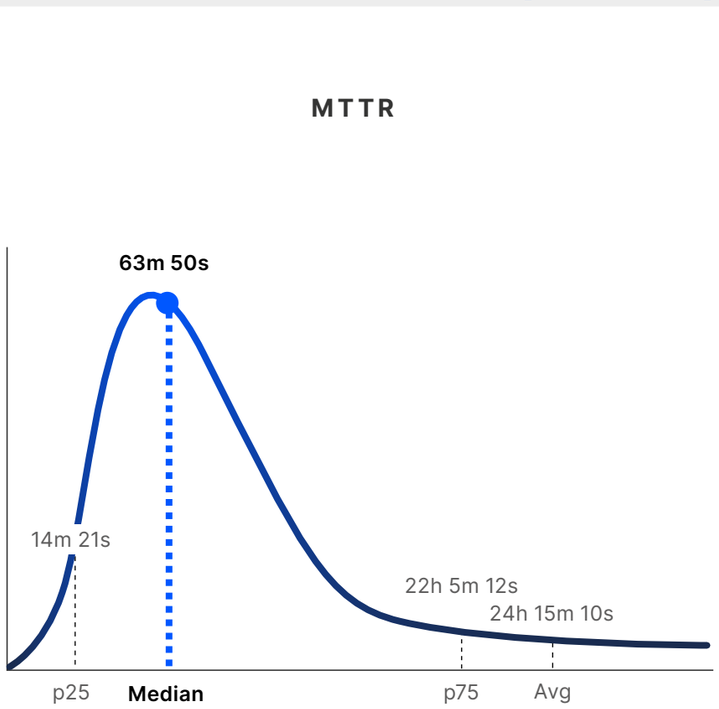
MTTR measures how long it takes teams to resolve workflow failures, from the moment a workflow fails until a subsequent workflow succeeds. When development comes to a halt due to failing workflows, the entire team's productivity suffers. Long recovery times not only delay the immediate work but can create cascading delays as changes queue up behind the blockage.

This year's data shows a median MTTR of 63.8 minutes, clustering near our recommended benchmark of 60 minutes or less. High-performing teams demonstrate that rapid recovery is achievable, with 25% of teams resolving failures in under 15 minutes. However, the data also shows a long tail of extended recoveries, with some incidents stretching into days or even weeks.

This pronounced rightward skew pulls the average MTTR up to 24.3 hours, nearly 23 times longer than the median. Failed builds lasting longer than a day often point to deeper technical or cultural issues: complex dependencies, insufficient feedback mechanisms, or a lack of shared ownership, which can turn straightforward failures into prolonged investigations.

Addressing these underlying challenges is crucial for maintaining business velocity. While some teams quickly diagnose and fix issues, others lose days of productivity to prolonged debugging. Here's how investing in CI/CD observability and automated recovery processes can help bridge this gap:

RecurShip has now increased its throughput to 525 workflows per day and has a success rate of 90%. That means engineers face approximately 53 build failures per day. By reducing recovery time from 4 hours to 90 minutes, RecurShip's developers would save 2.5 hours of debugging per failure. Across 250 working days, this amounts to **33,125 hours reclaimed for innovation** - equivalent to nearly **$3 million** in annual productivity gains.



Summary of 2025 software delivery metrics

METRIC	P25	P50	P75	AVG	BENCHMARK
DURATION	38s	2m 43s	8m 7s	11m 2s	10m
THROUGHPUT	2.71	1.64	1	2.86	1
MEAN TIME TO RECOVERY	14m 21s	63m 50s	22h 5m 12s	24h 15m 10s	60m
SUCCESS RATE	—	—	—	82.15%	90%

Since success rate reflects a binary outcome (pass or fail), its distribution lacks the variability that percentile data provides; therefore, only the average and benchmark are reported for this metric.

// The effect of team and company size

Strategies to improve development outcomes

How does company size affect software delivery success?

Company size (the total number of employees) can affect software delivery performance through broader dynamics like resource availability, cross-departmental dependencies, and organizational complexity.

As companies grow, factors such as approval processes, compliance requirements, and business unit coordination can significantly influence how pipelines are implemented and managed to as they deliver on different company priorities.

Small organizations (2-20 employees) achieve the highest workflow success rates at 85%, but they struggle with recovery when failures occur, with median MTTR exceeding 100 minutes.

Mid-sized companies (51-100 employees) demonstrate balanced pipeline performance, combining the fastest recovery times (38 minute MTTR) with high throughput (1.68 runs per day) and solid success rates (83.5%).

Key strategies by company size:

FOR SMALL COMPANIES

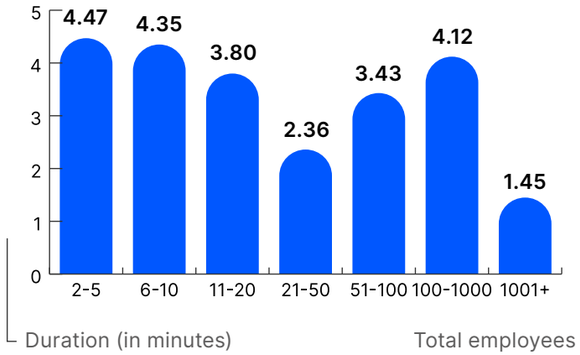
Focus on building resilient pipelines that can run independently when limited staff are pulled into other priorities. Invest in automated testing to reduce the debugging burden.

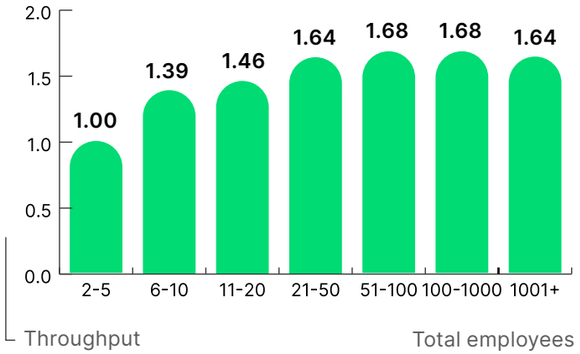
FOR MID-SIZED COMPANIES

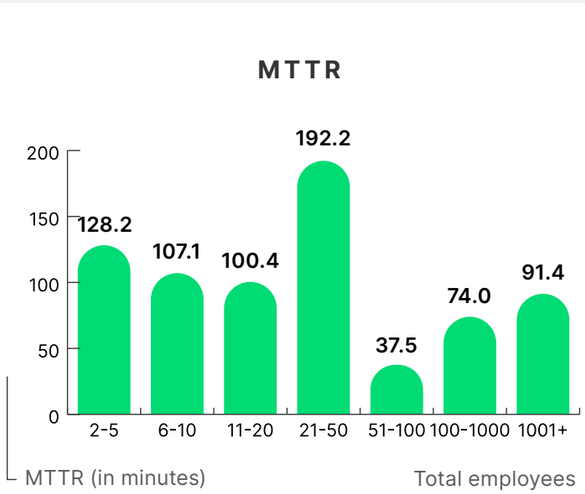
Maintain the advantages of quick recovery times while standardizing processes. Identify and replicate practices from high-throughput teams across the organization.

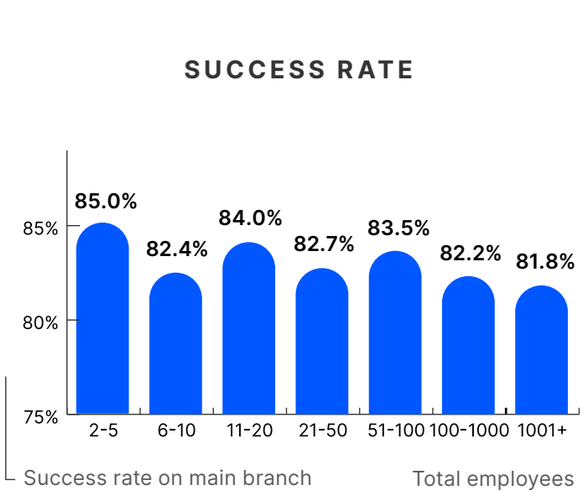
FOR LARGE COMPANIES

Streamline change management and approval flows without compromising control. Balance build speed optimization with processes that scale across business units.









How does team size affect software delivery success?

Engineering team size has significant implications for delivery performance. It shapes critical dynamics such as communication patterns, role specialization, and workflow complexity.

Smaller teams tend to move quickly with less coordination overhead, while larger teams must navigate dependencies, standardization, and process complexity as they scale.

Understanding these trade-offs is key to optimizing development velocity and reliability.

Teams of 6-10 engineers achieve the fastest recovery speed with a median MTTR of 29 minutes. Median throughput is 1 run per day, but an average of 3.91 runs — the highest among all team sizes — indicates that smaller teams have the greatest potential advantage in agility and productivity.

As teams grow to around 50 engineers, performance begins to suffer. Throughput remains flat but recovery times lengthen dramatically, reaching a peak of 170 minutes. This suggests increasing complexity may be a productivity bottleneck for mid-sized teams.

Teams of 100+ developers show effective scaling practices: their median MTTR of 77 minutes is significantly better than 51-100 person teams, while throughput stays similar at 1.71 runs per day. This suggests that larger teams invest in tools and processes that help manage complexity in both regular workflows and failure scenarios.

Based on these patterns, engineering managers should prioritize these strategies:

LIMIT TEAM SIZE

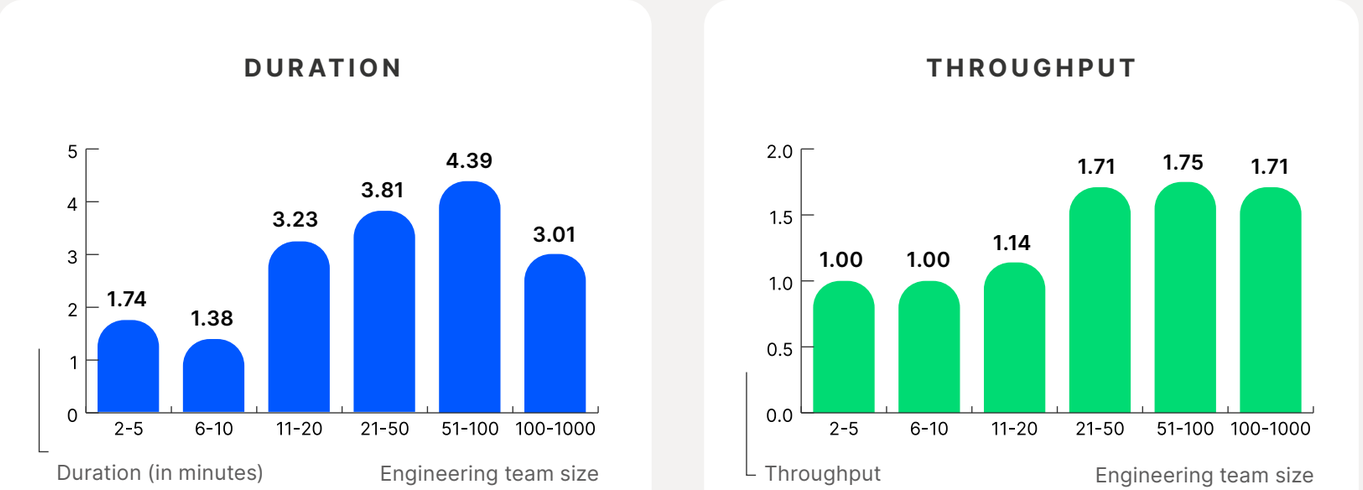
Organize development teams into autonomous units of 5 to 10 engineers. Consider breaking larger teams into smaller, cross-functional pods that maintain autonomy while sharing infrastructure and standards.

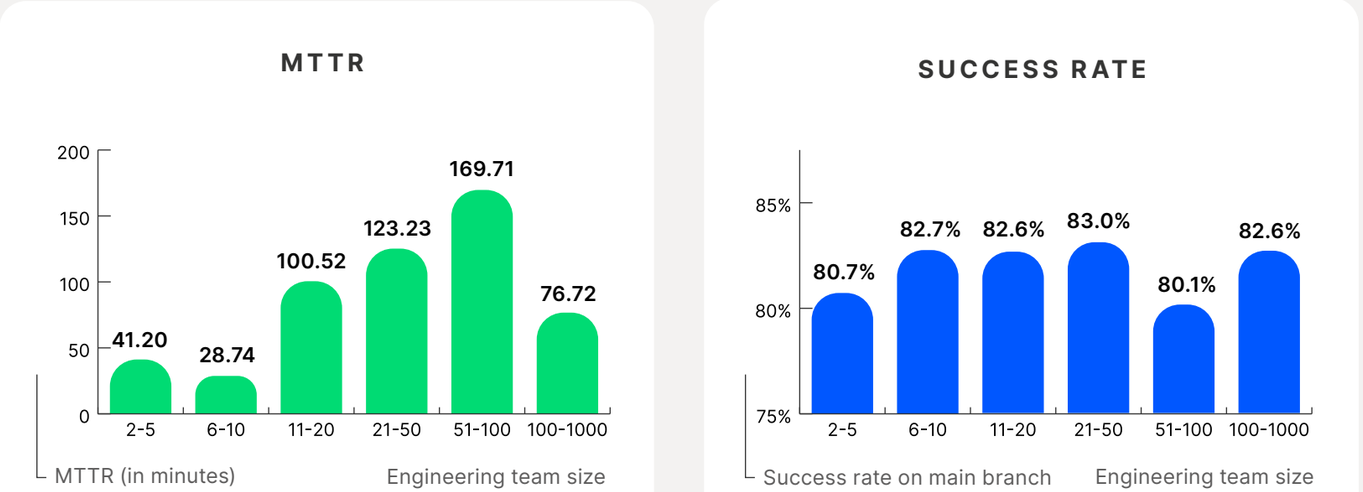
BREAK THROUGH COMPLEXITY

Mid-sized teams should invest in robust automation to unlock higher throughput and overcome the complexity barriers common at this size.

SCALE WITH STANDARDS

When scaling past 100 engineers, implement the standardized tooling and processes that allow large teams to maintain fast MTTR despite their size.
A dedicated platform team can create shared tooling and workflows that help product teams ship frequently and recover quickly.





// Industry results

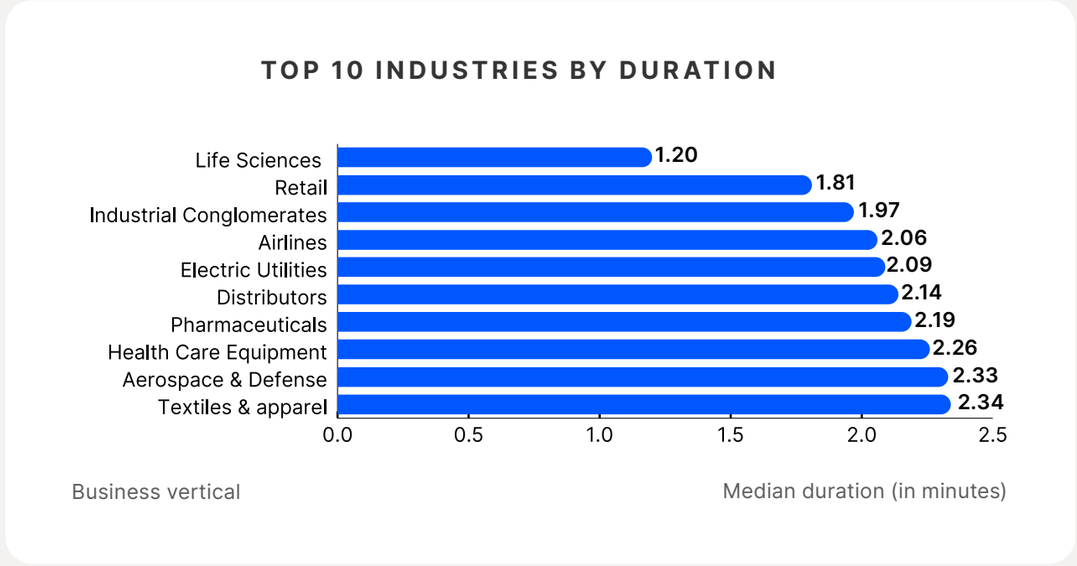
Comparing engineering success across sectors

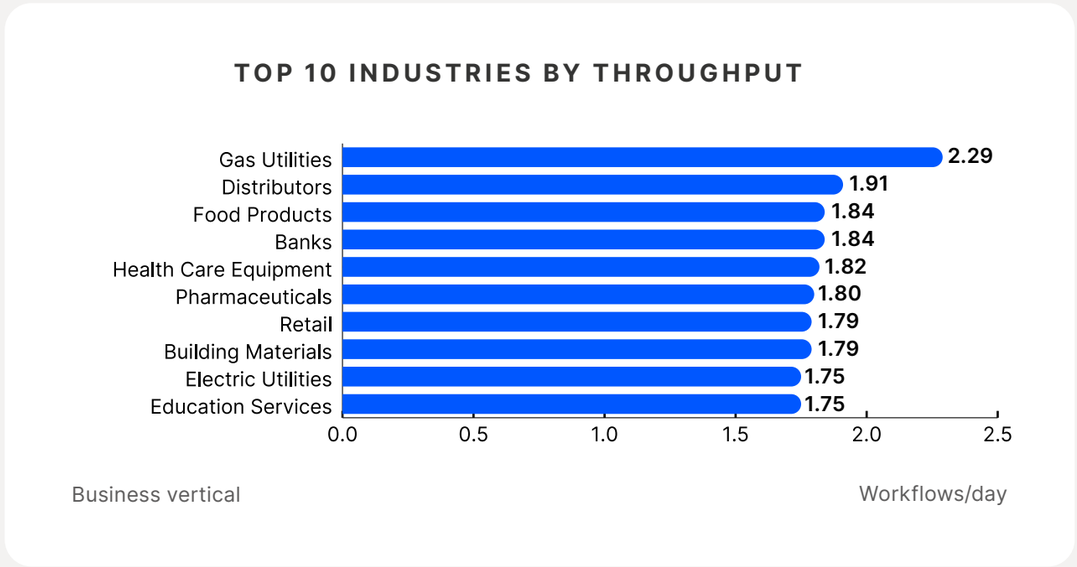
Industry performance

SOFTWARE TEAMS FACE different operational challenges based on their industry requirements. Finance, healthcare, and government sectors must adhere to strict security and compliance protocols, ensuring data integrity and privacy at every step.

Meanwhile, industries like e-commerce and entertainment prioritize rapid deployment and scalability to meet the changing preferences of their customers.

Despite these differences, one constant remains: the ability to deliver high-quality software quickly is a competitive advantage.





The top performers in workflow duration span regulated sectors like healthcare, defense, and airlines, as well as infrastructure-critical industries like utilities, distribution, and heavy industry. All complete workflows in under 2.5 minutes, beating the global median. Similarly, the highest throughput rates are achieved across diverse sectors, from utilities to banking, retail, and education.

The presence of retail, pharmaceuticals, and utilities among top performers in both duration and throughput shows that with proper tooling and automation, even industries with strict operational requirements can achieve high development velocity.

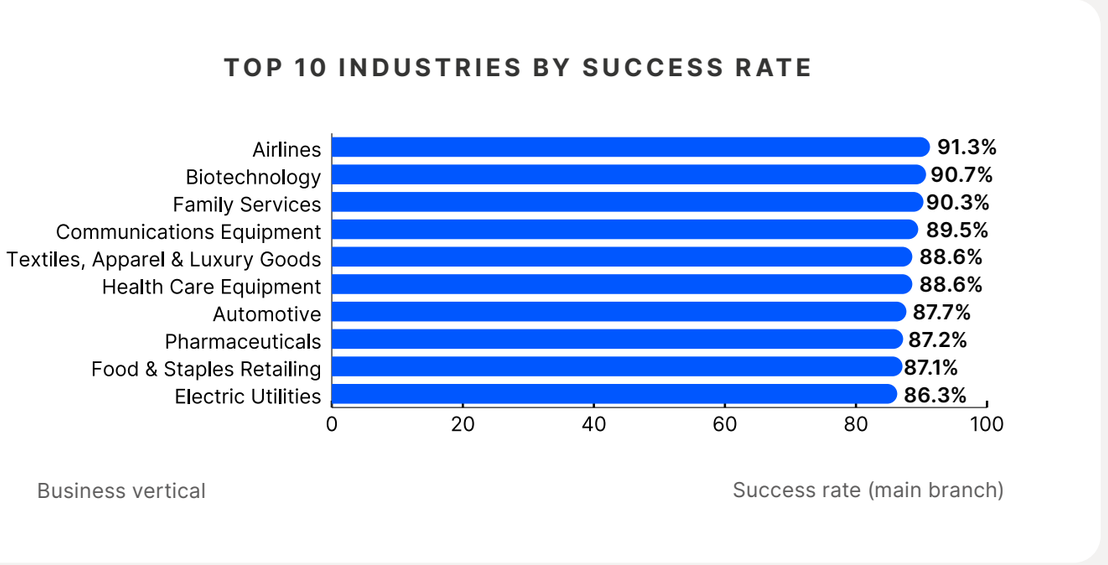
In fact, longer, more robust workflows tend to be a net positive for team productivity. Education services (5m 25s) and food products (5m 1s), for example, had the 4th- and 5th-longest durations of all industries we measured, yet are among the highest throughput industries on average.

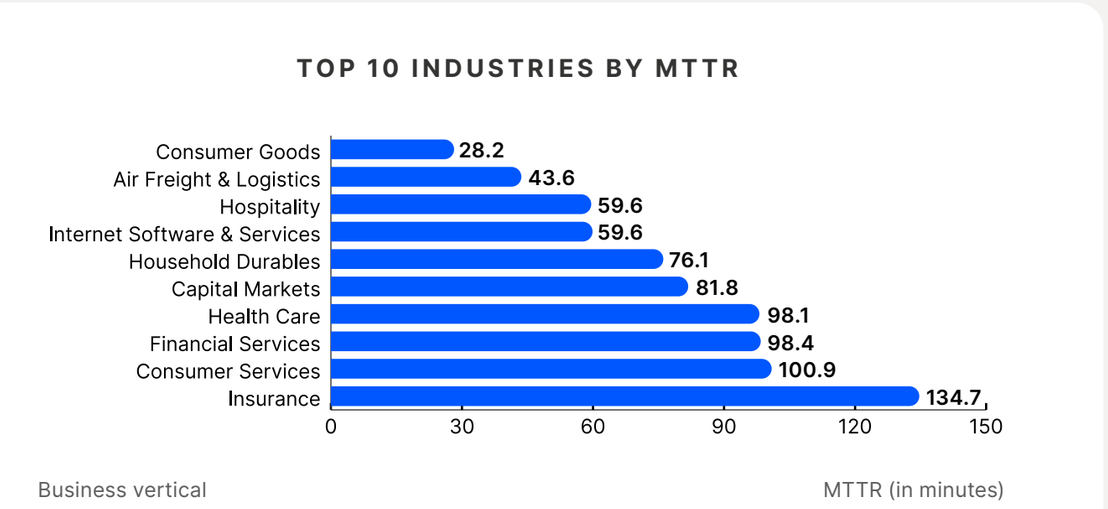
Airlines and biotech lead in success rates, exceeding the 90% benchmark on main-branch workflows. Industries such as these, where software failures can endanger human lives or lead to costly fines, often set up extensive quality gates to catch issues earlier in the development cycle, before they are merged to production branches.

While airlines and biotech experience fewer failures, their recovery times are among the longest, at 18 hours and 8 hours, respectively. This pattern makes sense, as the rigorous validation processes that prevent most issues also make resolving the rare failures more complex and time-intensive. In contrast, industries like consumer goods, hospitality, and software excel in rapid recovery, likely due to their "ship fast, fix fast" approach and the lower cost of failure in these industries.

Based on this survey of industry performance, here are three takeaways for business leaders:

- **Industry ≠ destiny:** Top performers span regulated and unregulated sectors, showing that proper tooling and automation matter more than industry requirements.
- **Quality gates drive productivity:** Longer validation cycles correlate with higher throughput. Invest in testing to accelerate, not inhibit, delivery speed.
- **Let risk inform response:** Match your CI/CD strategy to failure impact. Aim for early issue detection in safety-critical systems and frequent deployments and fast fixes in customer-facing services.





Summary of 2025 software delivery metrics by industry

INDUSTRY	DURATION (in seconds) p50 / p05	THROUGHPUT p50 / p95	MTTR (in minutes) p50 / p05	SUCCESS RATE Mean
Aerospace & Defense	139.9 / 13.0	1.1 / 7.8	369.9 / 20.0	83.3%
Airlines	123.3 / 7.5	1.0 / 4.2	1081.6 / 413.2	91.3%
Automotive	368.9 / 37.6	1.8 / 5.4	384.8 / 77.6	87.7%
Banks	261.4 / 41.9	1.8 / 13.1	137.2 / 6.7	74.3%
Biotechnology	231.1 / 20.7	1.1 / 5.9	477.1 / 30.8	90.7%
Capital Markets	201.4 / 8.4	1.5 / 7.2	81.8 / 1.9	81.8%
Communications Equipment	437.7 / 18.1	1.3 / 5.9	524.9 / 25.9	89.5%
Distributors	128.3 / 19.6	1.9 / 11.5	2598.8 / 2253.1	79.7%
Diversified Financial Services	193.1 / 8.0	1.6 / 7.0	98.4 / 3.5	80.1%
Education Services	331.3 / 18.0	1.8 / 6.9	458.4 / 7.4	81.7%
Electric Utilities	125.3 / 5.9	1.8 / 7.0	218.6 / 32.7	86.3%
Gas Utilities	251.9 / 20.1	2.3 / 12.6	218.6 / 32.7	77.4%
Health Care Equipment & Supplies	135.7 / 11.2	1.8 / 6.0	810.2 / 703.5	88.6%
Hotels, Restaurants & Leisure	176.7 / 12.2	1.6 / 7.4	59.6 / 2.9	82.0%
Insurance	227.0 / 10.3	1.4 / 7.4	134.7 / 3.9	83.4%
Internet Software & Services	141.2 / 4.4	1.6 / 7.0	59.9 / 0.8	82.5%
Pharmaceuticals	131.3 / 7.3	1.8 / 7.6	252.8 / 94.7	87.2%
Real Estate	207.1 / 9.9	1.6 / 7.3	231.6 / 5.4	72.4%
Retailing	108.4 / 7.9	1.8 / 5.9	405 / 15.9	82.5%
Textiles, Apparel & Luxury Goods	140.7 / 7.4	1.4 / 8.4	202 / 6.4	88.6%

// Scale your success with CI/CD

Tools to unlock software-driven business growth

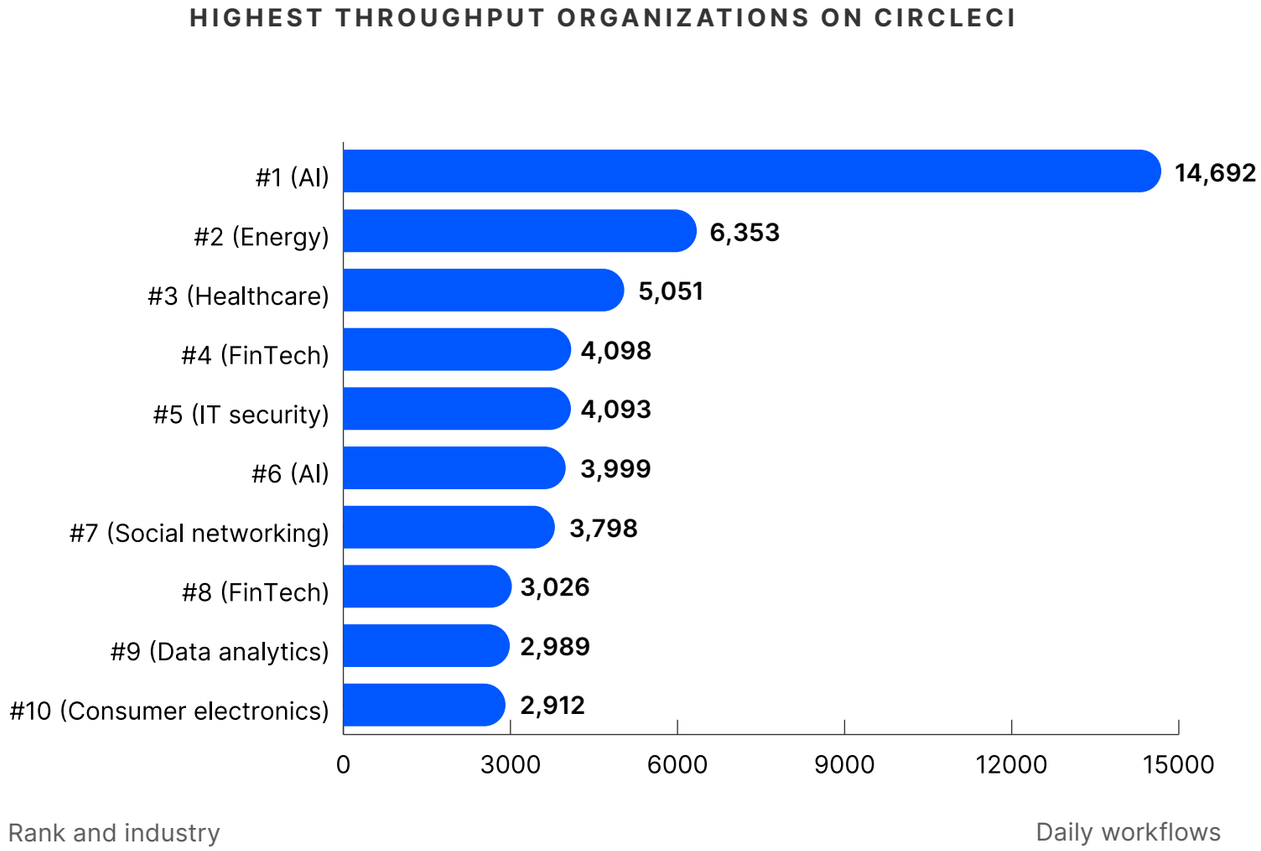
Software excellence starts with CI/CD

THE DATA IN THIS REPORT offer a useful baseline for teams to evaluate current capabilities and identify areas for growth. But market leaders don't aim for achieving industry baselines — they measure themselves against organizations pushing the boundaries of what's possible.

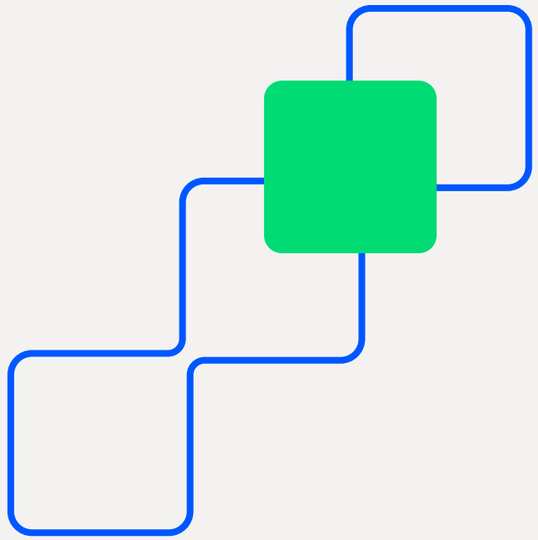
The most successful companies ship software thousands of times faster than their competitors, using CI/CD to transform software delivery from a cost center into a value multiplier.

Take top-performing teams on CircleCI as an example:

While most teams measure daily deployments in single digits, elite performers ship thousands, even tens of thousands, of changes each day.



This type of scale simply isn't achievable with manual processes or basic automation tools.



YOU NEED ROBUST INFRASTRUCTURE that can deliver continuous feedback at every stage without impeding velocity. With the right CI platform, teams can consistently deliver value at a pace that sets them apart from the competition.

If your organization wants to achieve this level of performance, it's time to invest in the tools and practices that make it possible.

CircleCI offers industry-leading speed, world-class support, and the flexibility to integrate with any tech stack. No matter where your team stands today, you can start quickly, scale effortlessly, and deliver measurable improvements in the metrics that matter most to your business.

Want to learn more? In the next section, we'll explore some of the features that make CircleCI a top choice for elite performers and teams striving to unlock their full potential.

CIRCLECI IS THE INDUSTRY'S LEADING CI/CD PLATFORM, empowering high-performing engineering teams to deliver exceptional software with speed, efficiency, and control. By automating complex workflows, smartly optimizing resources, and providing actionable insights, CircleCI transforms good engineering teams into elite performers. Here's how CircleCI enables engineering excellence.

Speed

Shorten development cycles and ship features faster

- The industry's **fastest build environments** across Docker, Linux, macOS (M1), Windows, GPU, and Arm, with flexible machine sizing for optimal cost-performance balance
- **Pre-built Docker images optimized for CI** to reduce build times and eliminate maintenance overhead
- **Advanced parallelization** features including test splitting and matrix jobs to accelerate testing without compromising coverage
- Sophisticated caching options including **Docker layer caching** to minimize workflow duration
- Webhooks to **trigger pipelines from external events**, enabling rapid response to changes across the software ecosystem

Efficiency

Get the most value from your engineering investment

- **Insights dashboard** and **usage API** to track key delivery metrics, test flakiness, resource usage, and spending
- Advanced **release orchestration features** with automated rollbacks and full visibility into deployment health
- **Granular rerun options** to minimize resource usage on failed workflows
- **SSH debugging** access for rapid issue diagnosis and remediation
- **VS Code extension** to provide real-time pipeline visibility in the development environment
- **Centralized configuration management** to eliminate redundant pipeline code across repositories

Control

Maintain security and standardization at scale

- Config policies enable **automated governance** of configuration elements at the organization level
- Reusable configuration packages (orbs) standardize workflows and share components across projects
- Enterprise-grade security with **granular access controls**, **IP range restrictions**, and **OpenID Connect**
- **Self-hosted runners** and **on-premises installation** options for sensitive workloads
- Detailed **audit logs** to enable forensic analysis of pipeline activities
- Premium support with **onboarding assistance** and **configuration reviews** to optimize security and performance

// Methodology

TO CREATE THIS REPORT, we pulled data from over 14 million CircleCI workflows within the first 28 days of September 2024. We filtered this data to include only projects that use GitHub as their VCS. 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 restricts to 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, 2024 and September 28, 2024
- Only GitHub projects
- Only projects with more than one contributor
- Only workflows that ran at least 5 times
- 14,146,319 workflows

REPORT AUTHORS Ron Powell
Jacob Schmitt

REPORT EDITOR Gillian Kieser

REPORT DESIGNERS Monica Russo-Dunphy
Birdy Wheeler-Wolowicz

ACKNOWLEDGMENTS Klimis Tsakiridis


circleci