circleci

The 2024 State of Software Delivery

// Contents

// Executive summary

// It’s no secret that the software industry is in the midst of a massive transformation, driven by two powerful undercurrents.

On the one hand, elevated interest rates have curtailed investment in the tech sector, leaving teams feeling the pressure of smaller headcounts and a heightened focus on efficiency and productivity. On the other, the emergence and widespread adoption of generative AI has unlocked new tools and revenue streams, infusing the industry with a renewed sense of optimism and opportunity.

What this adds up to is a pervasive feeling that software teams can and should be doing more with less. Is the industry up to the task? As the data in this report shows, the answer is a resounding yes.

We analyzed nearly 15 million data points from teams building on CircleCI’s cloud CI/CD platform and found that development teams are more productive in 2024 than ever before. They are generating more code, at a higher quality, and solving problems faster than at any point prior.

SOME KEY FINDINGS:

- Throughput, a measure of team productivity, **is up 11% across all branches** and an incredible **68% on production branches**, indicating a sharp uptick in new feature delivery.
- For the first time, the median team recovers from errors in **under 60 minutes**, exceeding our industry benchmark and freeing up developers to spend more time innovating.
- The most successful teams are running **longer workflows** on production branches and **adding new security and code quality tools to their pipelines**, underscoring the strong correlation between automation, feedback, and productivity.

*Sourced from The Total Economic Impact™ of CircleCI

The productivity boom

THE FINDINGS OF THIS REPORT represent a watershed moment for the software industry. The productivity boom taking shape across the broader economy already has materialized among tech teams, with tangible benefits being realized by those building on CircleCI.

At the same time, software delivery faces an unprecedented level of risk. Change is no longer limited to the code repository; it's happening all throughout the software ecosystem — in the foundational models and prompts that power our AI-enabled software, in the external libraries and frameworks that support today's distributed application architecture, and in the cloud infrastructure that keeps our applications online and accessible. Upstream and downstream changes now have outsized effects on your ability to predict and control application performance.

These new ways of building require a more dynamic and comprehensive approach to managing change. The old, repo-centric model is gone, and with it goes the traditional CI practice of monitoring only the code base for changes. That means it's also curtains for CI theater, where teams could automate small portions of their workflows, cover up the rest with inconsistent and unreliable manual processes, and call it CI. The margin for error today is small and shrinking. Implementing robust, continuous feedback at all stages of your development cycle is mission critical.

In this report, we lay out the trends, practices, and technologies driving growth and efficiency in software teams, giving you actionable intelligence you can use to optimize your processes and stay competitive.

We also make the data-driven case that there is a real separation happening between teams who are adapting and thriving in today's environment and those who are being left behind.

Which type of team are you on?

> “As the software world continues to adapt to new tools, processes, and market forces shaping the industry, the best organizations are rethinking not just how they deliver software but how they define and measure success. When efficiency is paramount, CI/CD gives teams the ability to deliver more value in less time, setting the foundation for stable, sustainable growth in an era of rapid change.”
>
> — **JIM ROSE**, CircleCI CEO

// What is an elite team?

THROUGHOUT THIS REPORT, we use terms like "elite" and "high-performing" to describe teams that consistently outperform benchmarks on the four key software delivery metrics shown to the right.

These benchmarks are not arbitrarily chosen. They are based on insights by Paul Duvall in his foundational 2007 book *Continuous Integration* and adapted from the groundbreaking research by Nicole Forsgren and the DevOps Research and Assessment (DORA) team's 2018 book *Accelerate: Building and Scaling High Performing Technology Organizations*. As outlined in these works, **organizations that meet or exceed these DevOps benchmarks** are significantly more likely to achieve superior outcomes in software delivery and operational performance.

METRIC	DEFINITION	BENCHMARK
DURATION	Average length of a CI/CD workflow	10 minutes
THROUGHPUT	Number of workflows per day	1 per day
MEAN TIME TO RECOVERY	Average time between a failed build and the next successful run	60 minutes
SUCCESS RATE	Percentage of passing builds	90% on the main branch

QUANTIFYING AND CATEGORIZING software delivery performance is notoriously difficult because teams often lack the resources and capabilities to capture accurate and comprehensive data. When they do capture data, it's challenging to interpret it in a meaningful way due to variations in project scopes, technologies, and business domains. It is difficult to make "apples to apples" comparisons to other organizations, as contextual differences can significantly affect performance indicators.

Other data reports, the most notable being the DORA team's annual *State of DevOps* report, attempt to overcome this by surveying a broad sample of teams and looking for shared characteristics in the self-reported data. While this yields valuable insights into the adoption and impact of DevOps practices across industries, results must be interpreted with caution due to self-reporting bias and challenges in capturing and interpreting data. Respondents may have incentives to overstate their success, and the subjective nature of self-assessment can introduce inaccuracies.

// What sets the State of Software Delivery apart from other studies is our access to one of the world's largest datasets on the actual behavior of software organizations as observed on our cloud CI/CD platform.

This data provides concrete, quantitative insights into development practices, workflow frequencies, recovery times, and other metrics. Unlike survey-based research, analyzing CI/CD data removes the subjectivity and variance of self-reported information, offering a more accurate and scalable way to assess software delivery performance. By leveraging this data, we can derive insights that are grounded in real-world practices and offer a picture of elite performance driven by tangible achievements measured on our platform.

// How do you compete?

A roadmap for success at scale

THIS YEAR'S DATA represents a clear call to action for technology leaders and organizations. The best organizations are **shipping more features and recovering from errors faster** than ever before. At the same time, teams struggling to adapt to the new paradigm are quickly falling behind the curve.

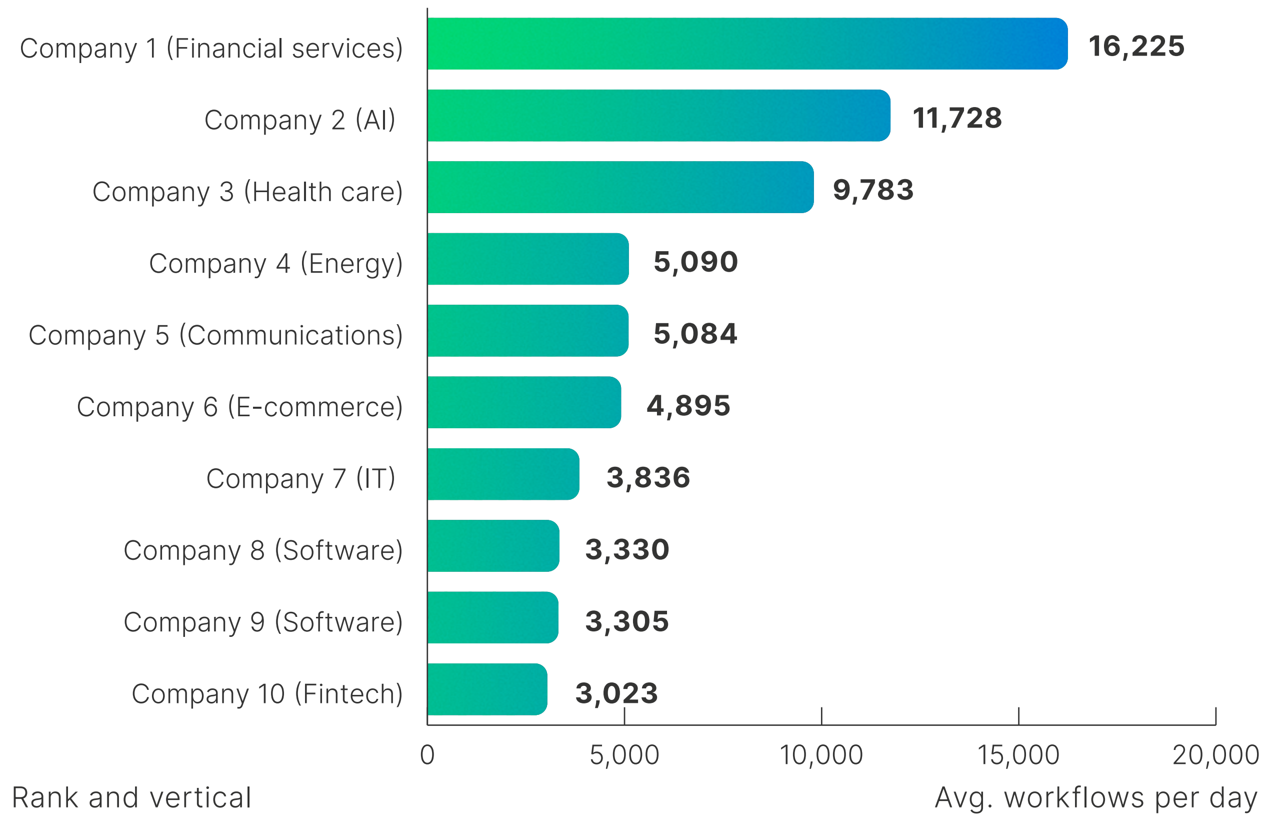
To remain competitive, organizations must adopt the technologies and practices that drive efficiency, productivity, and value-focused delivery. For many, that will involve:

- shifting to smaller, more agile teams
- responsibly integrating AI-powered tools to accelerate code generation and issue resolution
- embracing DevOps and continuous delivery models to streamline deployment processes

But while these changes may help you meet the challenges of today, it's equally important to plan a roadmap to compete two, ten, even twenty years into the future.

As your organization grows, your competition will evolve along with you. To illustrate the challenge of competing at scale, consider the difference in throughput between our median performers and the **10 most productive** organizations on the CircleCI platform:

HIGHEST THROUGHPUT ORGANIZATIONS ON CIRCLECI



Organizations in the 95th percentile (P95) do nearly 5x the amount of work as those in the median, running 7.6 workflows each day.

// Those in the top 10 overall run more than 6,000 workflows per day on average.

This is not only several orders of magnitude more productive than the median performer, but it also represents a **97% year-over-year increase** in throughput among top-performing teams. The competition is tough, and getting tougher.

To compete at this level requires a strategic focus on speed and agility optimizations.

How are they doing it?

One of the critical areas where high-throughput teams excel is in their monitoring of queuing and concurrency. Because of the high rate of change in their projects, these teams often face merge conflicts that cause slowdowns and build failures. By monitoring these issues and incorporating merge queues, auto-scaling infrastructure, and parallel processing in response, they are better able to manage the volume and complexity of work required to compete at the top levels of performance.

As throughput levels increase, elite teams also need platforms that can manage the complexity of continuous delivery at scale. This includes efficient handling of releases and rollbacks, which are inevitable parts of the lifecycle of any application. The agility to track releases across environments, quickly identify issues, and revert changes or push updates without disrupting the user experience unlocks the true value of high throughput and sets elite performers apart.

Elite organizations understand that their CI/CD infrastructure is a foundational element of their success. They invest in platforms that not only support high levels of automation and integration but also provide tools to effectively optimize every stage of the delivery process. You can set your organization on a similar path to success today by building pipelines that are scalable and responsive to the needs of your business today, tomorrow, and well into the future.

// The four key metrics

How do today's teams perform against industry benchmarks for duration, throughput, mean time to recovery, and success rate?

2m 50s

MEDIAN DURATION

10m

BENCHMARK

Duration

KEY TAKEAWAYS:

- Workflows are 13% faster year over year, averaging 2 minutes and 50 seconds across all branches
- Durations increased on production branches and decreased on feature branches
- Teams are balancing the need for rapid value delivery with the requirement for stable, secure deployments

Duration is a pivotal metric that directly influences development speed, code quality, and the overall agility of a team. Shorter durations reduce friction and tighten feedback loops, but they can also obscure problems such as insufficient test coverage or manual toil outside of the pipeline. The best teams balance speed and stability by tailoring their workflows to the needs of each development stage.

In 2024, we observed a divergence in duration trends that underscores the need for a more nuanced approach to team velocity. While the median duration fell 13% to 2m 50s, the speed gains were entirely concentrated on feature branches, which were almost a full minute (26%) faster year over year. In contrast, workflows on production branches were 11% slower, increasing by an average of 17 seconds.

These changes indicate teams are implementing effective strategies to increase both the velocity and quality of their output: faster feedback on development branches to drive innovation and more robust checks on the default branch to limit production failures. Notably, longer workflows on production were coupled with a dramatic increase in the number of workflows run.

// Robust testing is a productivity multiplier.

For the optimal balance in your pipelines, focus on rapid improvements to code quality early in the development process through practices such as linting, unit testing, and static application security testing (SAST). Reserve more comprehensive checks such as dynamic application security testing (DAST), performance testing, and end-to-end testing for QA and production branches. Invest in fast, scalable CI/CD and drive efficiency through parallelization, caching, and consistent build environments. These improvements will accelerate development cycles, eliminate costly rework, and ensure a high level of product quality from the outset.

Throughput

KEY TAKEAWAYS:

- Compared to last year, teams are doing more work overall, and particularly on their production branch
- Gains are concentrated in the top 50%; lower performers saw no growth
- Innovative teams are finding ways to do more with less while the rest get left behind

Throughput is a pivotal indicator of a team's ability to deliver software efficiently. It measures the total number of workflows run, directly impacting how quickly a team can respond to changes and deliver value to users. A higher throughput signifies more efficient processes and a more productive and responsive team.

This year's data shows a notable increase in throughput among top performers, particularly on production branches. Median throughput was 1.68 workflows per day on both main and feature branches, representing a 10% increase in throughput overall and a surprising 68% increase in activity on main. These improvements, however, were concentrated in the top 50% of teams. The bottom half of teams saw no growth in activity year over year.

What this suggests is an emerging split in teams' ability to adapt and thrive in the increasingly resource-constrained world of software delivery. Teams who can successfully navigate the technical, cultural, and economic challenges of today are finding ways to do more with less, shipping more value at a faster pace. Those who struggle to adapt are finding themselves left behind.

More effective automation with CI/CD is an important driver of team productivity, but it represents just one piece of the puzzle. Emerging practices such as the use of platform engineering to optimize developer experience and scale effective practices across teams, as well as the safe and strategic use of AI-powered tools for code generation, error resolution, and predictive analysis, give innovative teams an edge. By focusing on efficiency, adaptability, and the wellbeing of their engineers, these teams are setting new standards for productivity in the face of adversity.

1.68

WORKFLOWS / DAY

MEDIAN THROUGHPUT

1+

BENCHMARK

Mean time to recovery (MTTR)

FIRST TIME BELOW BENCHMARK

57m 33s

MEDIAN MTTR

60m

BENCHMARK

KEY TAKEAWAYS:

- Average recovery times are faster than the industry benchmark for the first time
- Teams recovered from failed feature branch workflows 17+ minutes faster than last year
- Faster recovery supports quicker iteration cycles and drives increased throughput

Mean time to recovery (MTTR) is a critical metric for evaluating the resilience and efficiency of CI/CD pipelines. A lower MTTR indicates a team's ability to quickly address and resolve issues, minimizing downtime and its impact on operations.

This year, teams on CircleCI achieved a significant MTTR milestone: average recovery times across all branches fell below the 60-minute benchmark for the first time in our reporting history, dropping 11% to 57m 33s. Teams recovered from errors 1m 17s (2.4%) faster on the default branch and an impressive 17m 17s (22.6%) faster on feature branches. The emphasis on velocity in feature development mirrors a similar finding in this year's duration numbers: teams are accelerating early development workflows to drive efficiency in innovation.

These results continue a multiyear trend of accelerated recovery times and suggest a growing focus on maintaining deploy-ready code. Rapid error recovery is a prerequisite to increasing deployment frequency, and the notable increase in throughput identified in this year's report provides more evidence that fast MTTR supports the broader objective of keeping the delivery pipeline flowing smoothly.

To improve your MTTR and eliminate bottlenecks in your organization, integrate robust testing, with verbose and actionable error reporting, in all stages of your development cycle. Use monitoring tools and AI-driven insights to help identify, diagnose, and resolve issues more swiftly. Embrace core DevOps principles including smaller, more frequent commits, trunk-based development, and maintaining a deploy-ready default branch. By making your development process more efficient and your codebase more resilient, you can significantly reduce downtime and enhance the reliability of your software, ultimately leading to improved customer satisfaction and sustainable competitive advantage.

Success rate

KEY TAKEAWAYS:

- Success rates on main rose 5 percentage points year over year, nearing our benchmark of 90%
- Lower success rates on feature branches (70.1%) are expected given higher levels of experimentation
- Improved performance on main supports higher levels of throughput

Success rate is a crucial metric indicating the percentage of builds or deployments that pass through the CI/CD pipeline without errors. With sufficient test coverage, a higher success rate points to better code quality, more efficient development practices, and an improved flow of value to users.

This year's data indicates an upward trend in the quality of code being generated by teams. Overall, success rates increased from 69.5% to 72.3%. On the main branch, where successful deployments are most critical, success rates increased from 77.4% to 82.5%, approaching our recommended benchmark of 90%. On feature branches, where experimentation and early-stage development make errors more likely, success rates held steady at 70.1%

Higher success rates on the main branch are especially noteworthy given the corresponding increase in both throughput and workflow durations on main. Taken together, these trends indicate teams are not only generating more code, but they are subjecting it to higher levels of scrutiny and implementing more effective ways to ensure its quality before deployment. This comprehensive approach to software development suggests that organizations are becoming more sophisticated in their use of CI/CD pipelines, emphasizing not just the quantity of work produced but also its quality.

To capitalize on these positive trends, organizations should continue prioritizing code quality. This could involve adopting nuanced testing strategies, such as the use of feature toggles for isolating new code during testing phases, and shifting test coverage left to capture errors earlier. Platform teams can automate policy enforcement to ensure that all contributions align with organizational best practices regarding coding standards, tool choice, and compliance with security protocols. For company leadership, investing in advanced CI/CD tools and AI-powered coding assistants can further support these efforts, providing the necessary resources and infrastructure to maintain high success rates.

82.5%

AVERAGE SUCCESS RATE (MAIN BRANCH ONLY)

90%

BENCHMARK

// Additional findings

What are the tools, processes, and strategies that contribute to DevOps success?

*Sourced from The Total Economic Impact™ of CircleCI

How do team and company size affect software delivery success?

IN THIS SECTION, we explore how the number of contributors and the size of a company influence CI/CD performance. These observations provide insights into strategic priorities and emerging opportunities across diverse organizational scales.

// We share them as a tool you can use to identify optimal resource allocation and process adjustments for your organization.

TEAM SIZE

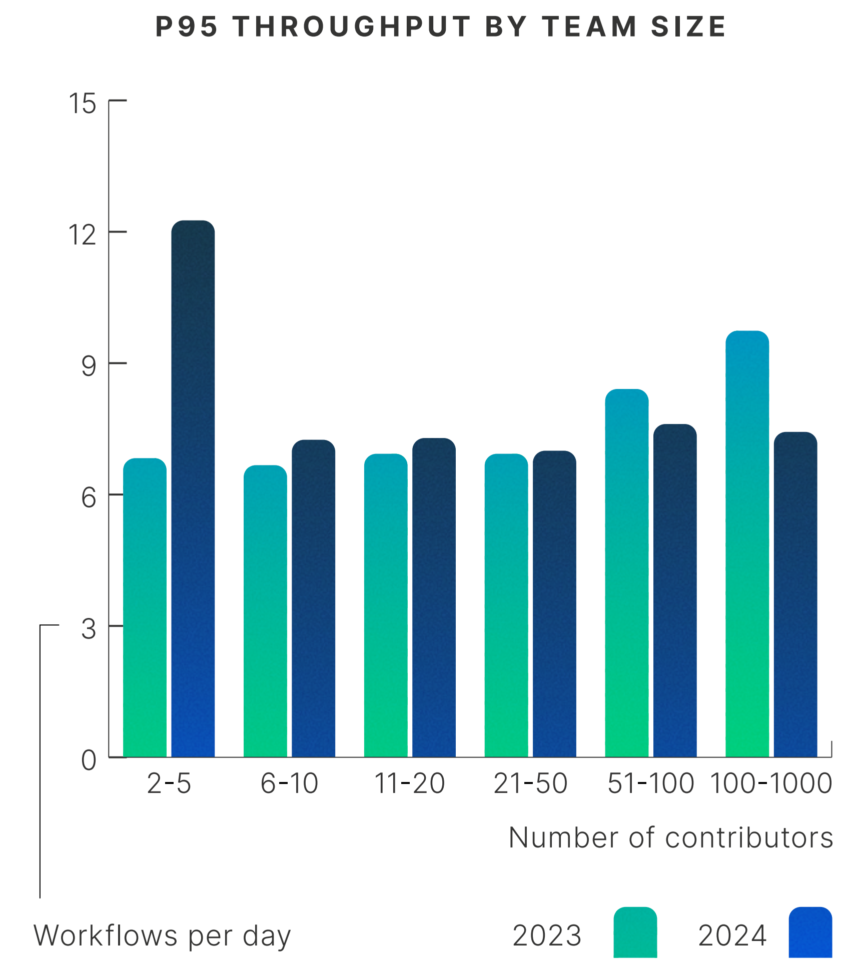
Here are some key findings related to team size, as measured by the number of contributors to a given project.

Smaller teams ran longer workflows on main; larger teams opted for speed: Teams with 50 or fewer contributors increased durations on the main branch by an average of 19%. Those with 100 or more contributors cut durations by more than 50% and now run shorter workflows than all but the smallest teams (2-5 contributors).

Smaller teams see bigger productivity gains: The team size with the biggest overall increase in throughput was the 11–20 member group, which saw a 53% uptick in workflows run. When looking at the top performers (P95), the team size with the highest level of throughput by far is teams with 2–5 contributors, at 12.3 workflows per day. That is a significant change from last year, when teams sized 100–1000 were far and away the most productive. This year, large teams saw a decline in throughput at the P95 level.

Smaller teams recover faster everywhere, larger teams prioritize main: Teams with 11–20 contributors experienced the most substantial reduction in recovery times compared to last year, nearly halving them across all branches to reach 37m 40s on main and 38m 50s on feature branches. On the main branch, teams sized 100–1000 recovered fastest, at 33m 17s. However, recovery times on feature branches for teams of this size exceeded 3.5 hours, an increase of 65% year over year.

The improvements in throughput and stability observed among teams with 20 or fewer contributors further supports the hypothesis that software organizations are doing more with less, and may suggest that the best positioned organizations are shifting resources to optimize smaller, more agile teams rather than scaling up their workforce.



COMPANY SIZE

Now here are some some of the trends happening among different company sizes, as measured by the total number of employees (not restricted to engineering teams).

Productivity is leveling out on average: Last year, we reported a positive correlation between company size and throughput. In this year's data, we found a flattening in throughput performance, as small and medium-sized businesses are running a number of workflows approximately on par with their larger competitors. While throughput increased year-over-year (YoY) across all organization sizes, companies with 11-50 employees had outsized gains and now match the average throughput of companies with 100 to 1000 employees, at 1.68 workflows per day. Those with 5 or fewer are only slightly behind at 1.64 per day.

Top performers are separating from the pack: Despite the convergence in median productivity, when we consider the top performers (P95) by company size, throughput does still generally increase as company size grows. The highest throughput, 7.5 workflows per day on average, happens at companies with greater than 100 employees, and the lowest (6.64) occurs at companies with five or fewer.

Smaller companies are running longer workflows: Duration increased year over year in companies with fewer than 100 employees. Similar to what we observed for team sizes, organizations with hundreds or thousands of employees (or more) have continued to accelerate their workflows and now run the fastest workflows on our platform at 2m 40s on average. Smaller organizations are trending in the opposite direction, now running longer workflows at approximately 4 minutes on average.

// Based on this data, we are reporting a trend in which the traditional correlation between company size and throughput is diminishing.

Smaller, more agile organizations appear to be finding new ways to drive productivity. Notably, these organizations are running longer, more complex workflows, which may suggest productivity gains are driven by an increased use of automation and more comprehensive testing and quality assurance processes.

KEY TAKEAWAYS ON TEAM AND COMPANY SIZE

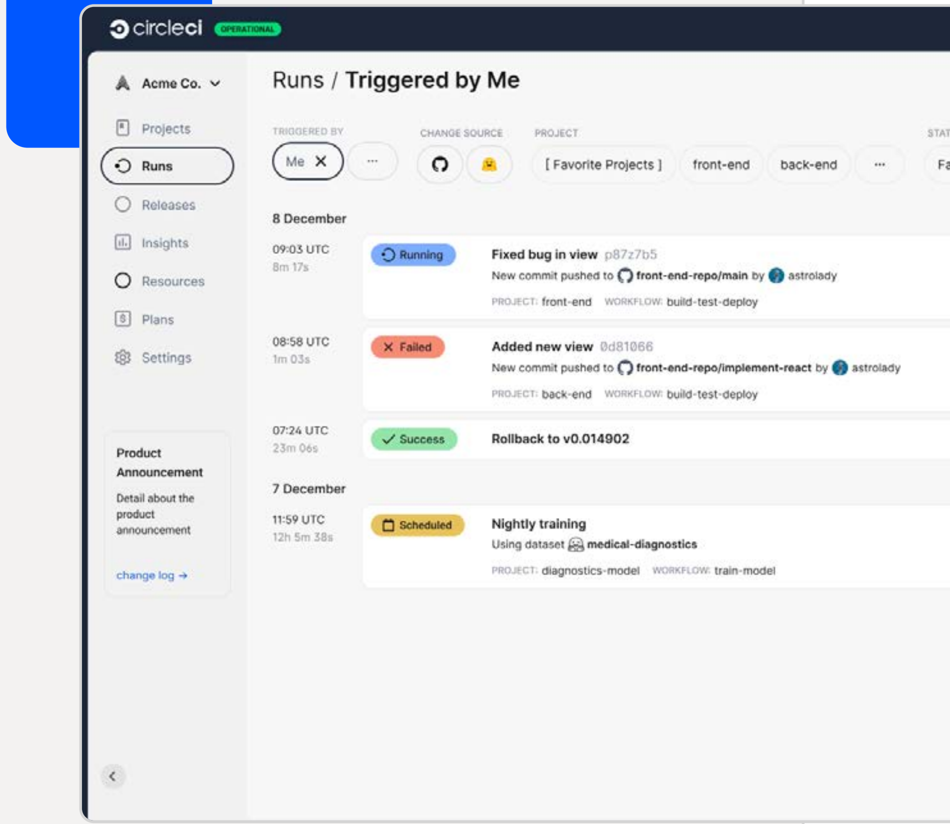
Software delivery performance is shifting in ways that challenge conventional wisdom about team and organizational size. Smaller teams and businesses have shown remarkable increases in productivity, closing the gap with their larger counterparts, while simultaneously increasing the length and complexity of their workflows.

> // The growing divide between high and low performers is not just a matter of company or team size ...

... more often it reflects a willingness and ability to embrace new technologies and methodologies. Smaller companies that are agile and forward-thinking can compete effectively, sometimes even outperforming larger organizations by being more innovative and responsive to change. Larger organizations may find benefits in reorganizing around smaller, more nimble stream-aligned teams.

Success in today's tech landscape is not solely about scale but also about adaptability and the strategic use of technology to improve performance. As the tech industry continues to evolve, the gap between the companies that thrive and those that struggle to keep up is likely to widen.

Those able to harness the power of CI/CD and other technological advancements will continue to lead, reshaping the competitive landscape in their favor.



8 December

09:03 UTC 8m 17s — Running — Fixed bug in view p87z7b5
New commit pushed to front-end-repo/main by astrolady
PROJECT: front-end WORKFLOW: build-test-deploy

08:58 UTC 1m 03s — ✕ Failed — Added new view 0d81066
New commit pushed to front-end-repo/implement-react by astrolady
PROJECT: back-end WORKFLOW: build-test-deploy

07:24 UTC 23m 06s — ✓ Success — Rollback to v0.014902

7 December

11:59 UTC 12h 5m 38s — Scheduled — Nightly training
Using dataset medical-diagnostics
PROJECT: diagnostics-model WORKFLOW: train-model

Product Announcement
Detail about the product announcement
change log →

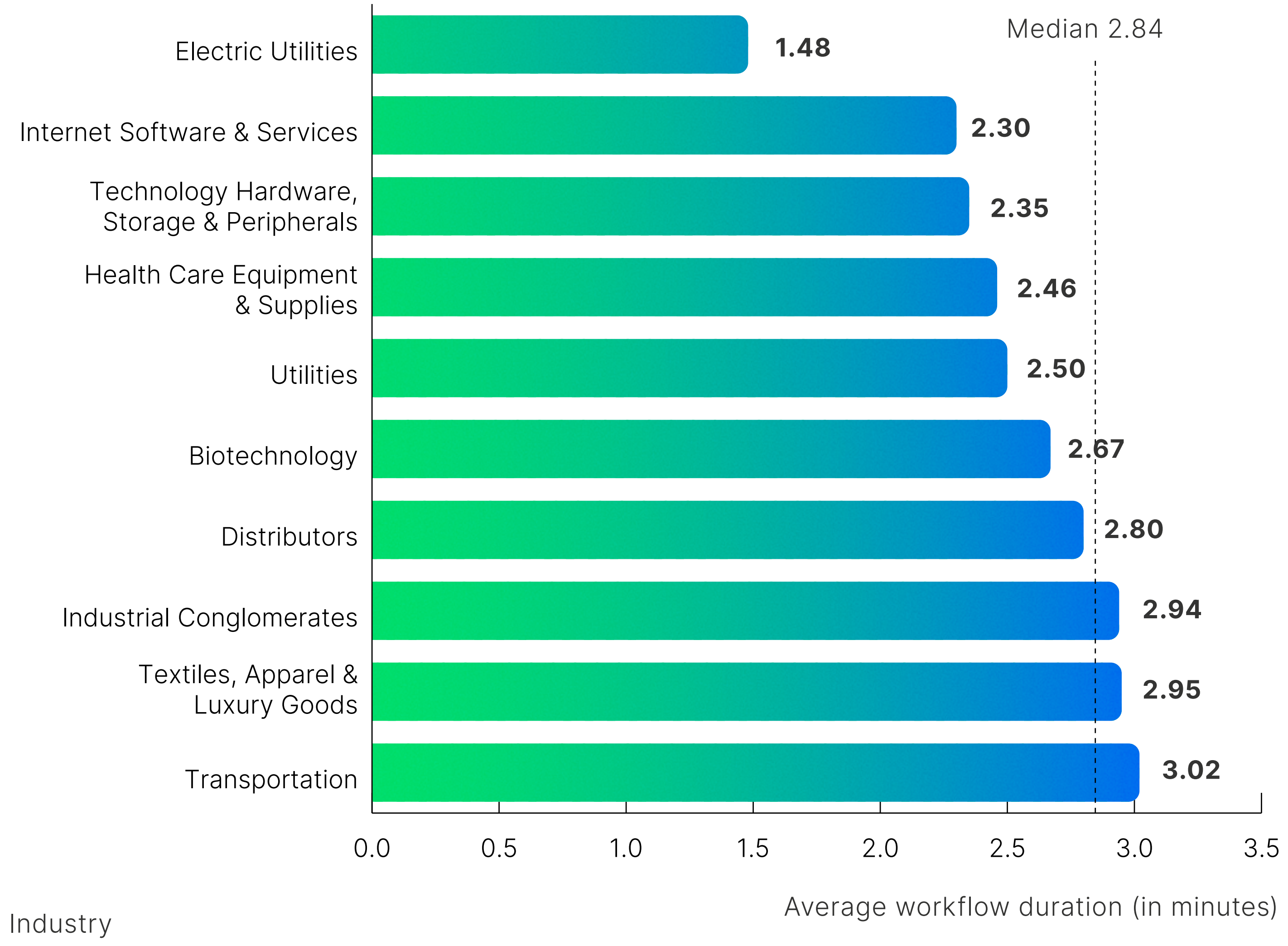
How does your industry stack up?

UNDERSTANDING broad trends in software delivery performance can lead to valuable insights, but when making decisions that affect your business, industry-specific metrics can help you make better-informed decisions about how to compete.

In this section, we'll reveal the top ten business verticals ranked by their performance across our four key delivery metrics.

// These metrics offer a clearer picture of performance standards, customer expectations, and competitive benchmarks in specific markets, helping you measure your team's performance against competitors and uncover opportunities for innovation or efficiency gains.

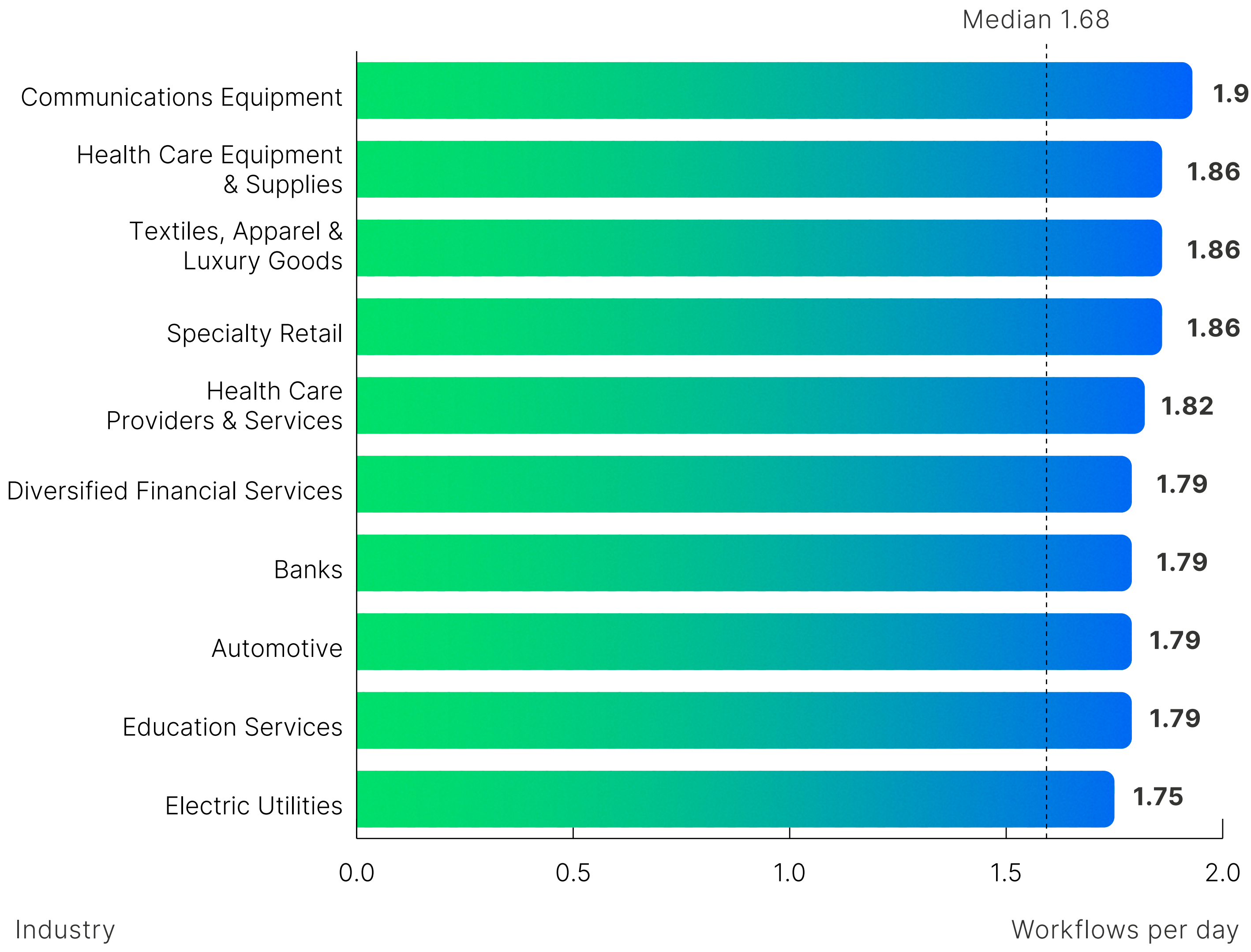
TOP 10 INDUSTRIES BY DURATION



DURATION

Electric utilities leads the list with the shortest average duration of 1m 29s (48% faster than the overall average). This might reflect a high level of automation and efficiency in their software deployment processes, which is crucial for industries — including utilities, health care, and transportation — where responsiveness and uptime are critically important. However, shorter durations can sometimes compromise the depth of testing and quality assurance, so it is important to balance speed and thoroughness based on the specific operational requirements and risk tolerance of your sector.

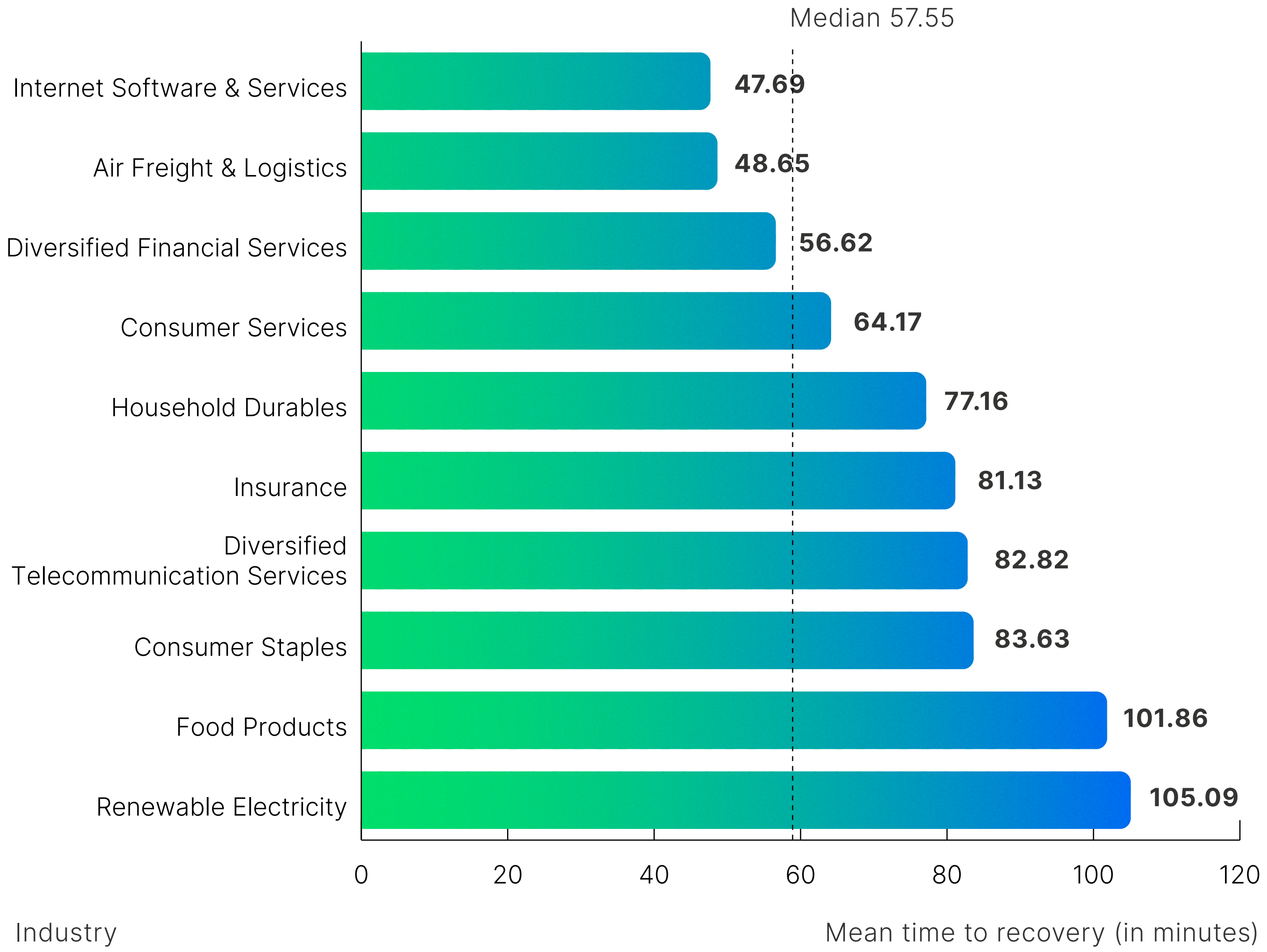
TOP 10 INDUSTRIES BY THROUGHPUT



THROUGHPUT

As reflected in the overall increase in throughput seen in this year’s data, the presence of both technology-oriented sectors (like communications and health care equipment) and traditional sectors (like banking and automotive) suggests that operational excellence and high throughput are recognized as competitive advantages across the board.

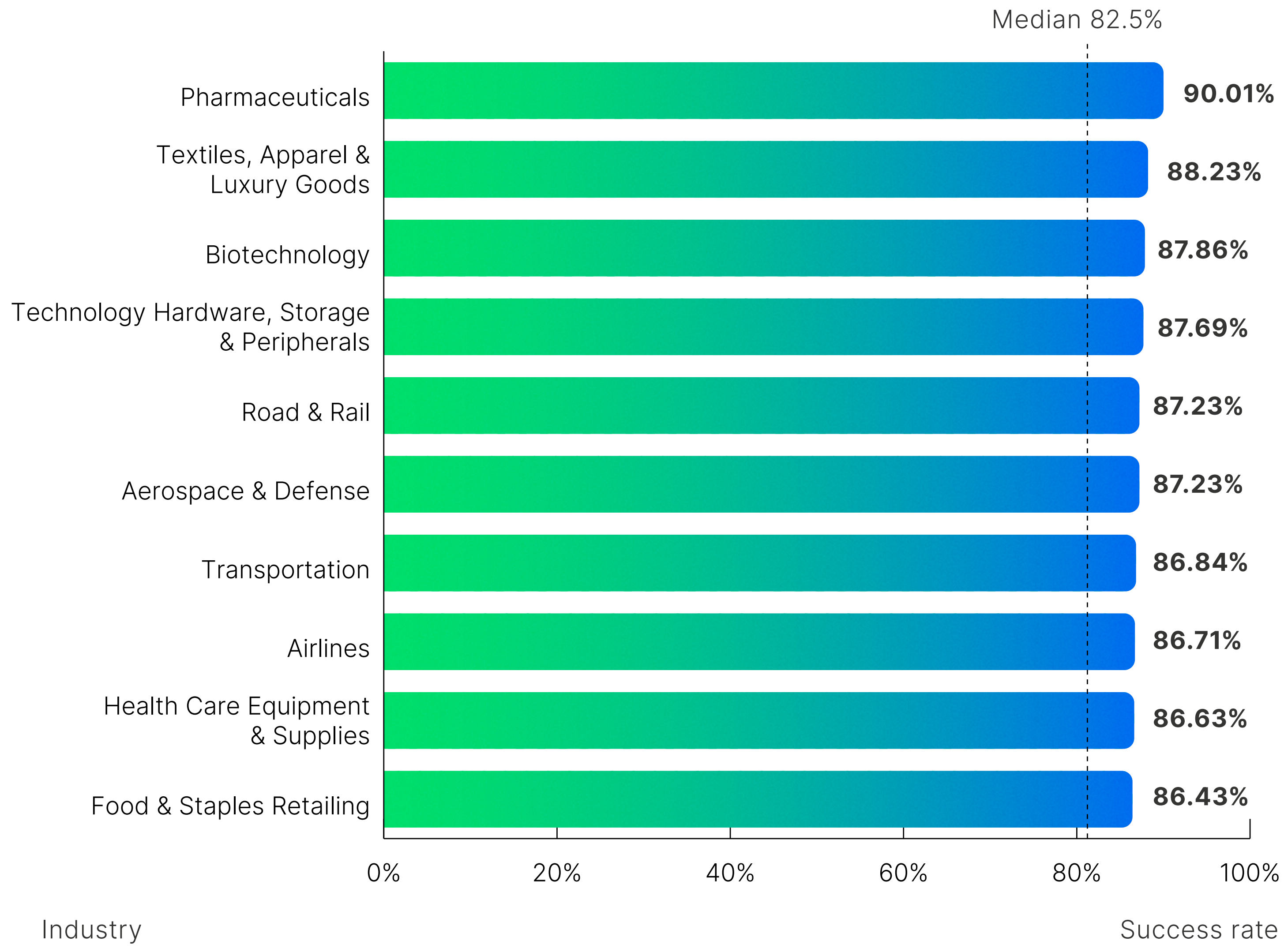
TOP 10 INDUSTRIES BY MEAN TIME TO RECOVERY



MEAN TIME TO RECOVERY

Internet software and services leading the list with the shortest recovery time (17% faster than the average) underscores the sector's focus on high availability and the ability to quickly address issues. Similarly, air freight & logistics and financial services are particularly sensitive to downtime and are the only other industries to recover faster than the 60-minute benchmark.

TOP 10 INDUSTRIES BY SUCCESS RATE (DEFAULT BRANCH)



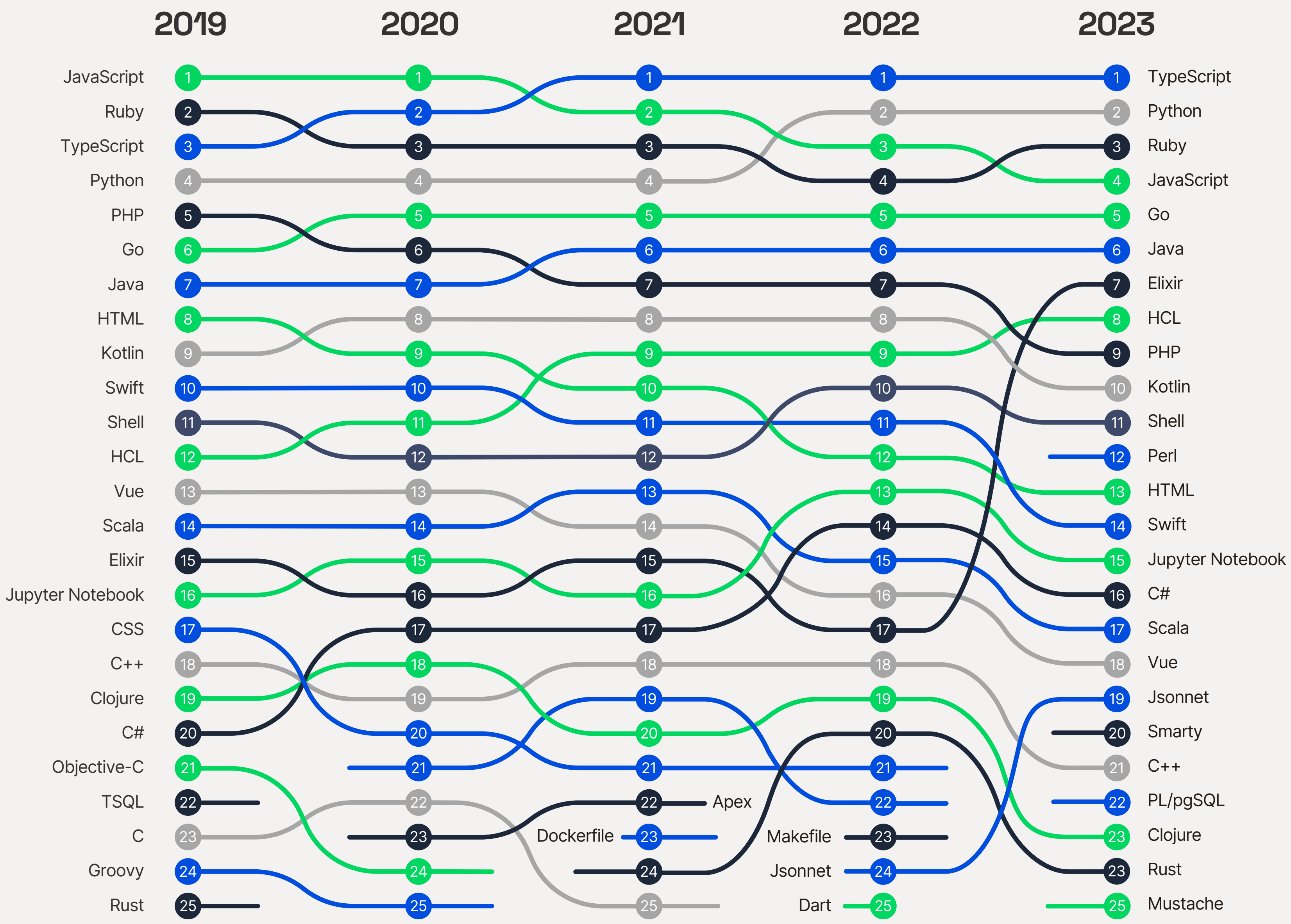
SUCCESS RATE

Pharmaceuticals stands as the only industry with a success rate on main above our 90% benchmark, reflecting the industry’s high cost of failure and emphasis on precision, quality control, and regulatory compliance. Other high-stakes industries, including biotechnology, aerospace and defense, and health care equipment, underscore the critical nature of reliability and accuracy in these fields, where successful deployments are vital to maintaining safety, compliance, and trust.

Language trends

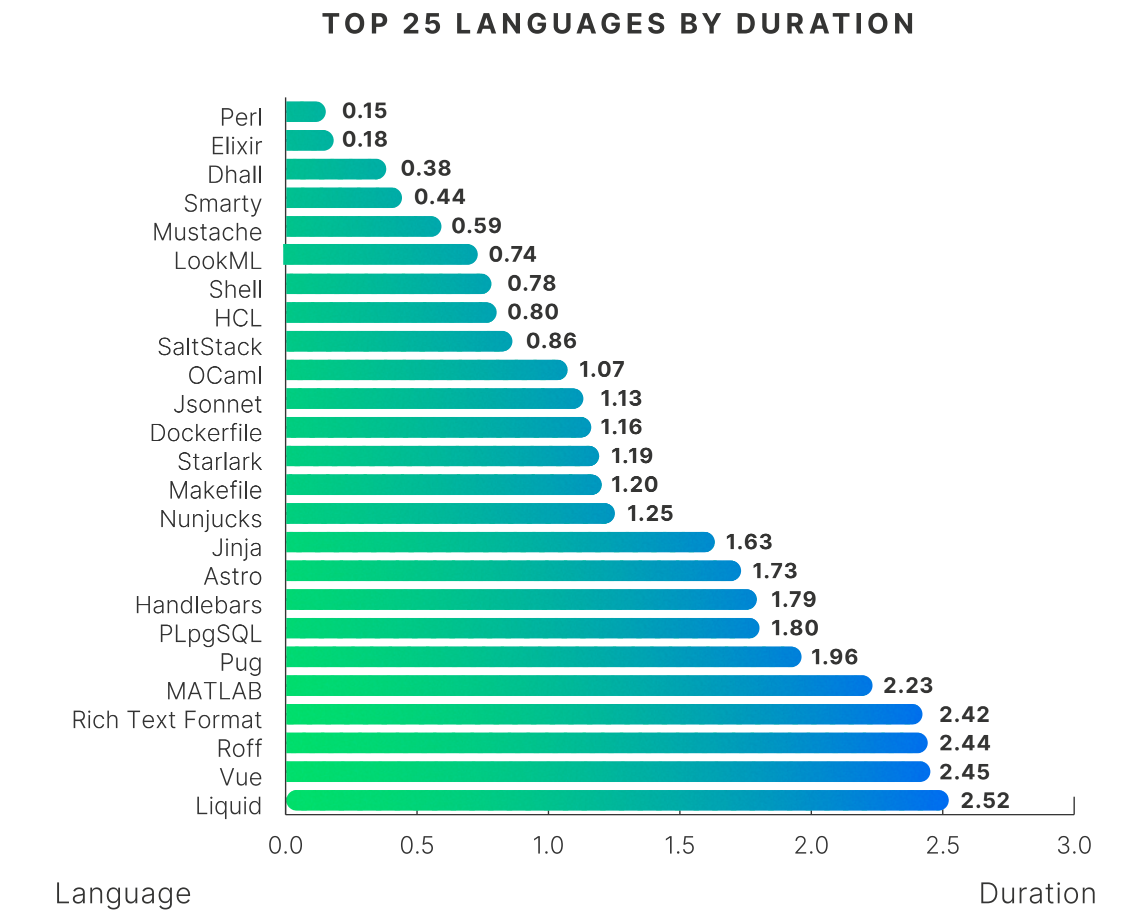
Although programming language is not the most important factor in the success of your CI/CD initiatives, it can influence your project's build times, testability, and ease of automation, along with the tools available for use in your pipelines. A well-chosen language can enhance CI/CD processes, leading to smoother deployments and faster feedback cycles, while a less suitable choice may introduce challenges that complicate these workflows.

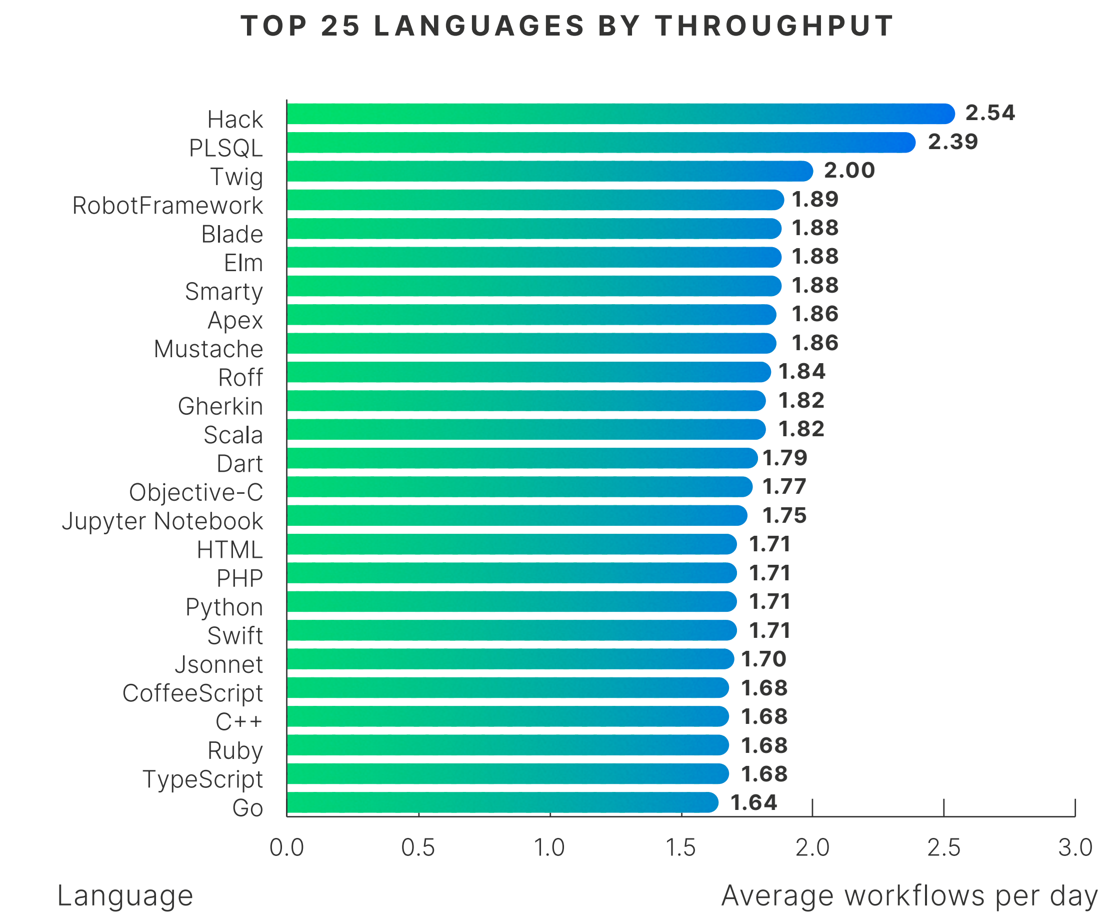
Below, we look at the most commonly used languages on our platform, as well as the top 25 languages ranked for their performance on each of the four key metrics. In exploring these trends, we aim to provide insights into how different languages can make CI/CD practices either more efficient or more cumbersome, highlighting the importance of thoughtful language selection in optimizing software delivery outcomes.



DURATION

Scripting languages like Perl and Shell are known for their ability to quickly automate repetitive tasks and regularly appear at the top of our list of fastest languages. Many of the other entries, such as LookML, HCL (HashiCorp Configuration Language), SaltStack, and Dockerfile, are domain-specific languages (DSLs) that provide higher expressiveness and efficiency for certain CI/CD tasks, like deploying infrastructure or managing configurations.

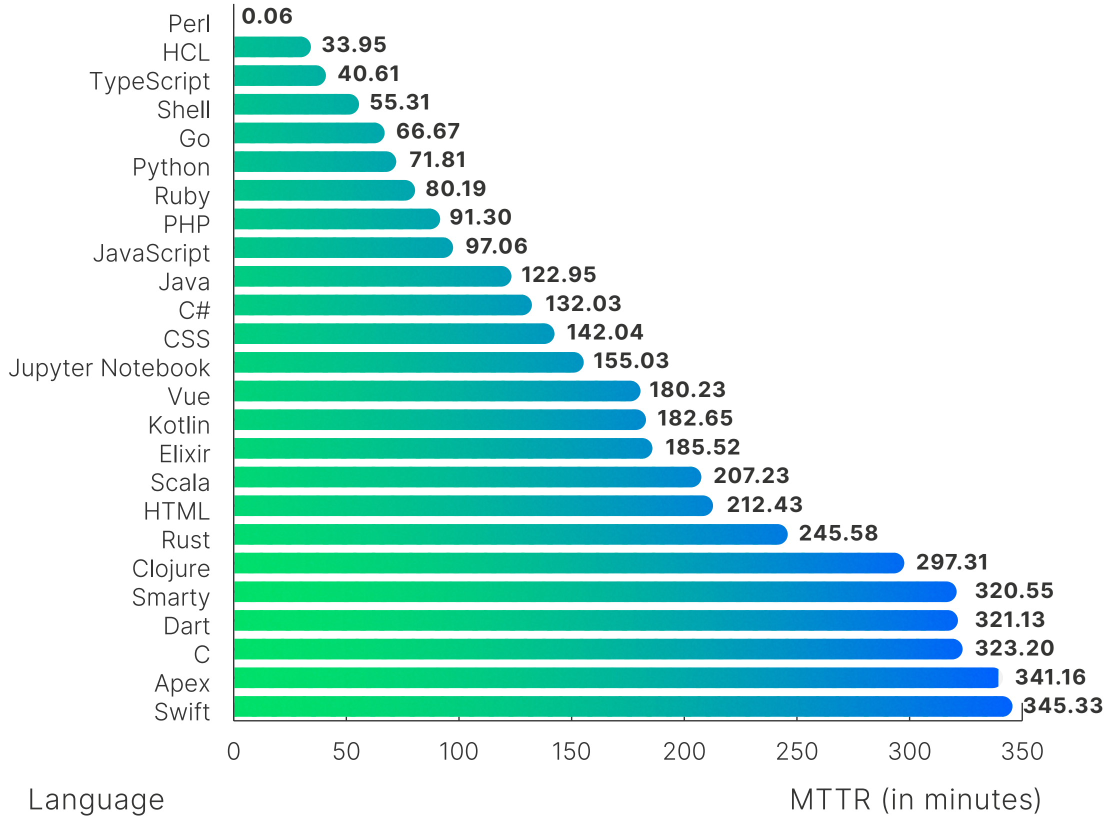




THROUGHPUT

The PHP dialect Hack continues its run as the language with the highest throughput, leading our list for the third straight year. The presence of templating engines (Twig, Smarty, Mustache) and DSLs (RobotFramework, Gherkin) further highlights the efficiency specialized languages bring to CI/CD pipelines.

TOP 25 LANGUAGES BY MTTR



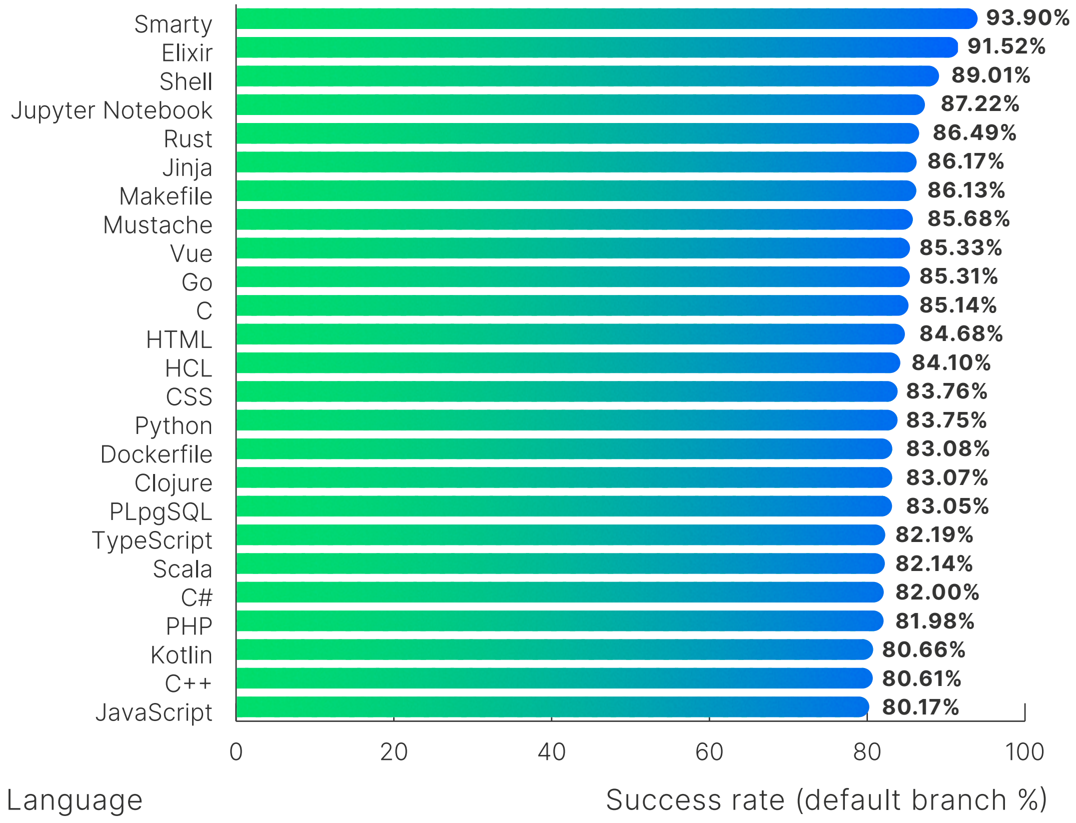
MTTR

Scripting languages like Perl and Shell, which also perform well on duration, have some of the fastest recovery times. Languages with large ecosystems and community support, such as TypeScript, Go, Python, and C# benefit from extensive libraries, tools, and active community forums that also aid in rapid issue resolution (though, notably, only TypeScript falls below the 60 minute benchmark for MTTR).

SUCCESS RATE

Elixir, which saw significant growth in overall usage this year, lends itself to high success rates due to its syntax, functional programming principles, fault tolerance, and system reliability. Compiled languages such as Rust, C, C++, and Go are also noted for their performance and safety features, including strong type systems and memory safety, which can help catch errors early in the development cycle and improve overall success rates.

TOP 25 LANGUAGES BY SUCCESS RATE



// Most popular DevOps tools

THIS SECTION LISTS some of the most popular and frequently used tools across projects on our platform. We measure popularity based on both the number of organizations that include them in at least one project (org adoption) and the total number of pipelines run (usage). We'll also show which tools are the fastest growing in these two categories.

By evaluating tool usage and adoption, engineering leaders can better understand which tools are becoming industry standards, which are emerging as new favorites, and how the landscape of development tools is evolving to meet the needs of modern software delivery.

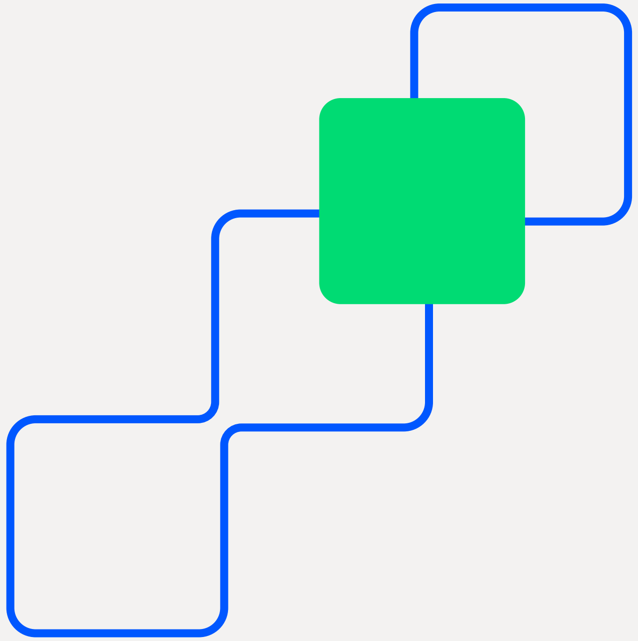
TOP TOOLS OF 2023

	By Org Adoption	By Pipeline Usage
1	Docker	Docker
2	AWS	AWS
3	Slack	Slack
4	Gradle	Cypress
5	Cypress	Helm
6	Terraform	Terraform
7	Codecov	GCP
8	Kubernetes	Kubernetes
9	GCP	Datadog
10	Maven	Gradle

FASTEST GROWING TOOLS OF 2023 (YOY%)

	By Org Adoption	By Pipeline Usage
1	Cloudsmith (+36%)	Sysdig (+318%)
2	StackHawk (+33%)	New Relic (+186%)
3	LaunchDarkly (+22%)	StackHawk (+122%)
4	Sysdig (+18%)	Pulumi (+95%)
5	Lacework (+15%)	Lacework (+61%)
6	Grafana (+12%)	Discord (+52%)
7	Bridgecrew (+6%)	LaunchDarkly (+48%)
8	Datadog (+5%)	Digital Ocean (+40%)
9	Honeybadger (+5%)	Datadog (+37%)
10	Artifactory (+4%)	Rollbar (+28%)

// Based on these results, we can draw a few conclusions, particularly around the core tools that appear on both the adoption and usage lists.



TOOLS SUPPORTING CLOUD computing (AWS, GCP), containerization and orchestration (Docker, Kubernetes), Infrastructure as Code (Terraform), build automation (Gradle), testing (Cypress), and collaboration (Slack) form the foundation of the modern DevOps toolchain.

Some tools, like Helm and Datadog, are used primarily by teams managing multiple complex deployments, limiting their adoption to a smaller number of teams that have heavier usage patterns. Others, like Codecov, enjoy broad adoption but less intensive use, highlighting their role in specific stages of the CI/CD process rather than continuous use throughout.

The overall growth of security, observability, and release management tools like Datadog, Lacework, StackHawk, LaunchDarkly, and Rollbar reflect a maturation of DevOps priorities in which teams are increasingly prioritizing the reliability, security, and user experience of the software being delivered. This is evidenced not only by the tools used in their workflows but also by the more robust workflows and improved recovery times observed throughout this year’s data.

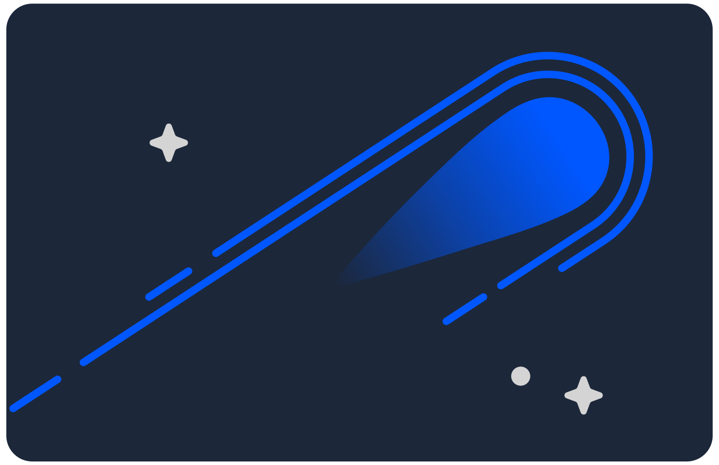
// To improve software delivery, start with CI

A ROBUST CI PRACTICE provides the foundation on which you can build a fast, secure, and sustainable software delivery process that will keep your organization ahead of the competition and your customers happy.

The data in this report shows that teams who invest in best-in-class CI tooling meet or exceed industry benchmarks for engineering performance. Average users of CircleCI rank among the highest-performing teams in the industry and save nearly $14 million dollars over a three-year period due to improvements in team speed, organizational efficiency, and quality assurance guardrails.

*Sourced from The Total Economic Impact™ of CircleCI

To achieve and even exceed these results at your organization, take advantage of the benefits that CircleCI can offer:

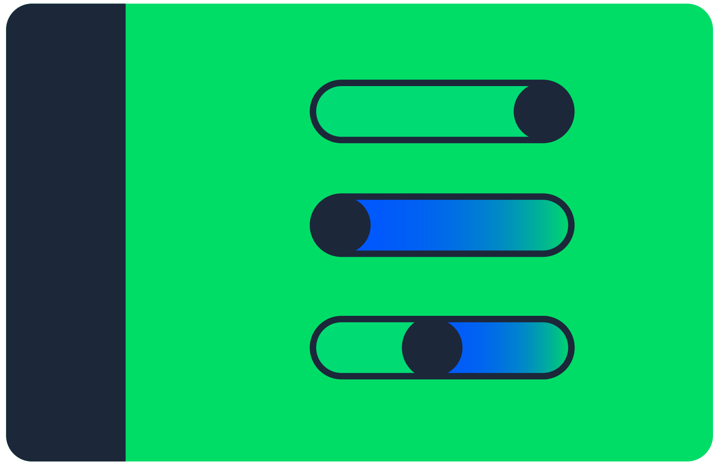


Speed

- CircleCI has the industry's fastest build environments, including Docker and remote Docker, Linux, macOS (including M1), Windows, GPU, and Arm. You can easily adjust the size of your machines to find the right balance of cost and workflow performance.
- Every user gets access to a fleet of custom-built Docker images. These images have been optimized for CI so that they are more deterministic and faster to load, cutting down on build minutes and saving you from having to build your own images.
- Intelligent test-splitting, matrix jobs, parallelization, and concurrency options significantly speed up test execution. By running multiple jobs simultaneously across separate build nodes, you can increase test coverage and success rates without sacrificing on duration.
- With advanced caching options, you can store and reuse data from previous jobs to reduce the duration of your workflows. Docker layer caching, for those building custom Docker images, can allow for an even greater reduction in workflow duration.

Efficiency

- Being able to record and monitor your team's performance across the four key metrics is an essential first step toward improving your delivery process. CircleCI users have access to the Insights dashboard, which measures the four key metrics for all of your workflows so you can optimize your team's projects. It also offers time- and money-saving data on credit spend, resource usage, and test flakiness.
- Powerful deploy and release features allow you to confidently roll out changes to Kubernetes environments, closely monitor their impact, and, if necessary, swiftly revert to previous states. With complete visibility throughout the entire delivery process, you'll gain a new level of operational flexibility and control.
- CircleCI offers the ability to quickly rerun a failed workflow or rerun only failed tests within a workflow, saving you time and helping you resolve issues with optimal resource usage.
- SSH debugging allows you to access the remote build environment, giving you the power to quickly reproduce, diagnose, and remediate issues, saving countless developer hours and shortening your MTTR.
- Using the CircleCI VS Code extension, developers get real-time feedback and visibility into their CI/CD pipelines directly from their development environment. This helps in quickly identifying and addressing issues, streamlines the commit-to-deploy process, and reduces context switching, leading to more focused and productive development workflows.



Control

- With config policies, you can create organization-level policies to impose rules and scopes to govern which configuration elements are required, allowed, or not allowed. By automating these rules, developers can innovate faster with access to exactly what they need, and nothing they don't.
- Orbs are sharable, reusable packages of configuration that you can use to standardize your workflows and share tools and components across projects. CircleCI users have access to a library of open source orbs created by trusted technology partners and community members, or you can create your own private orbs to share among authorized members of your organization.
- CircleCI supports role-based access controls, IP range restrictions, and OpenID Connect tokens, giving you a number of options for restricting who has access to critical project and configuration data.
- Self-hosted runners offer organizations the ability to run sensitive workloads on their private cloud. Organizations with heightened security and compliance concerns can also install CircleCI behind their own firewall, including support for air-gapped installation, with CircleCI Server.
- Premium support provides enterprise-grade consultations to help you continuously improve and optimize team productivity, including the option for personalized configuration reviews to improve pipeline security, eliminate bottlenecks, and maximize the value you get from your pipelines.

// Methodology

TO CREATE THIS REPORT, we pulled data from nearly 15 million CircleCI workflows within the first 28 days of September 2023. 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, 2023 and Sept 28, 2023
- Only GitHub projects
- Only projects with more than one contributor
- Only workflows that ran at least 5 times
- 14,833,746 workflows

REPORT AUTHORS	Ron Powell Jacob Schmitt
REPORT EDITOR	Gillian Kieser
REPORT DESIGNERS	Monica Russo-Dunphy Birdy Wheeler-Wolowicz
ACKNOWLEDGEMENTS	Can Wang

circleci