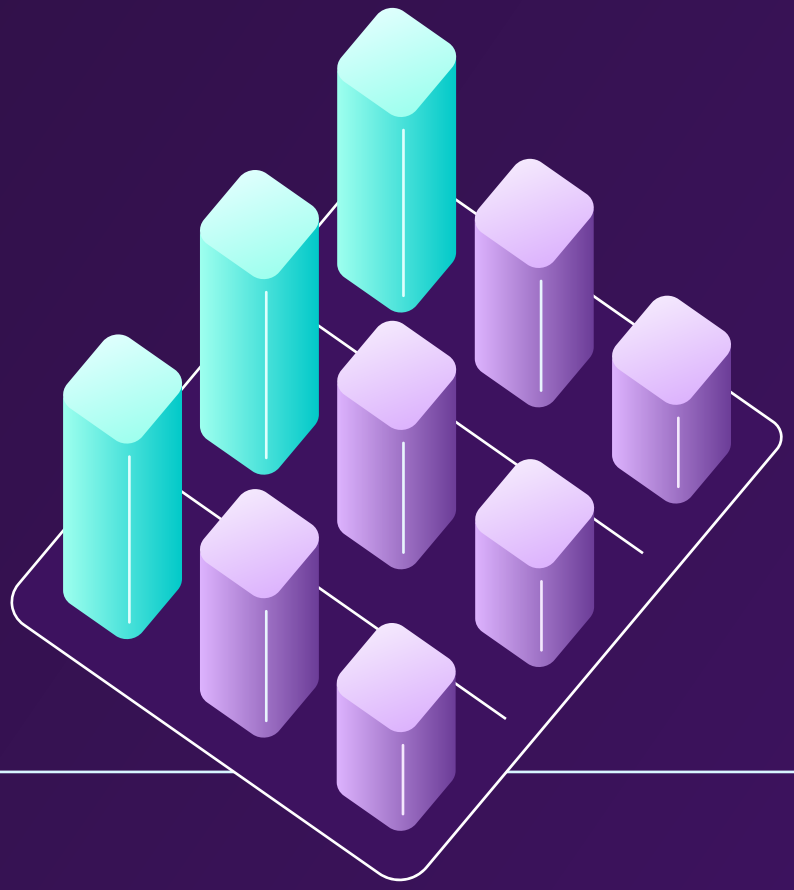
bitrise

Bitrise Mobile Insights 2025

Data-driven insights to help you build and release quality apps faster



Bitrise DevOps platform for mobile—trusted by over 8,500 brands and 400,000+ developers worldwide (and counting).

Executive summary

Mobile development is getting faster, more complex, and more demanding. This raises the question: how can mobile developers and engineering leaders tell if their build and release pipelines are best in class?

To answer that, we analyzed tens of millions of anonymized builds from thousands of teams between January 2022 and June 2025. The data revealed clear performance benchmarks, exposed gaps between laggards and leaders, and surfaced insights to help teams optimize efficiency, accelerate release cycles, and drive sustainable growth.

These are not theories—they're data-driven insights

to help you build quality apps at scale.

Key findings

Mobile app Continuous Integration (CI) pipelines are growing in complexity, yet build times have decreased: customers reduced average build times by 28% while workflow steps increased 23%.

Use of cross-platform frameworks is on the rise—30% of builds on Bitrise use cross-platform frameworks. Of all builds, 26% are React Native; Flutter accounts for 4%.

New Xcode versions become the most used by build volume in ~4 weeks; and reach majority usage by customer count in ~19–21 weeks.

Version control is consolidating, with GitHub accounting for 73% of builds on Bitrise in 2025.

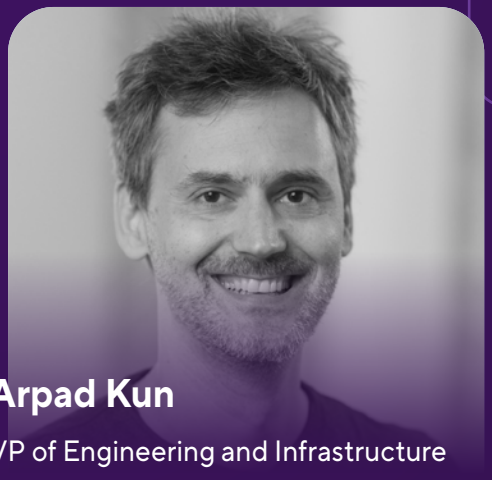
Parallelization is becoming the new normal with multi-workflow pipelines rising from 16% to 42% over the three year period. However, the median pipeline size remains small (3 workflows or less).

Flaky tests are increasing. However, teams using monitoring tools (like Bitrise Insights) enjoy 25% fewer flaky reruns and higher build success rates.

When teams use a structured release framework (like Bitrise Release Management), they are twice as likely to ship every two weeks or faster—the industry best practice.

Please note: All findings are based on aggregated, anonymized Bitrise customer data.

Contributors



Arpad Kun

VP of Engineering and Infrastructure

bitrise

Arpad leads Engineering and Infrastructure at Bitrise, helping businesses of all sizes realize the power of automation and technology advancements. With previous leadership roles at Meta, IBM, and Ustream (acquired by IBM), he combines deep technical expertise with a customer-first mindset.

Connect with Arpad on LinkedIn

Ben Boral

Staff Solutions Engineer

bitrise

Ben is an expert in applying automation strategies across the entire mobile DevOps lifecycle, helping countless mobile engineering organizations unlock new levels of efficiency and achieve automation nirvana.

Connect with Ben on LinkedIn

István Rechner

Principal Data Analyst

bitrise

István is a Principal Data Analyst at Bitrise, helping mobile teams apply analytics, experimentation, and automation to improve product and commercial outcomes. Previously he worked on applied Machine Learning research and production analytics for live video.

Connect with István on LinkedIn

Acknowledgements

Bitrise Mobile Insights 2025 was made possible by the contributions of many across Bitrise and the wider mobile ecosystem, with special thanks to the Bitrise Data Science team and our growing customer community.

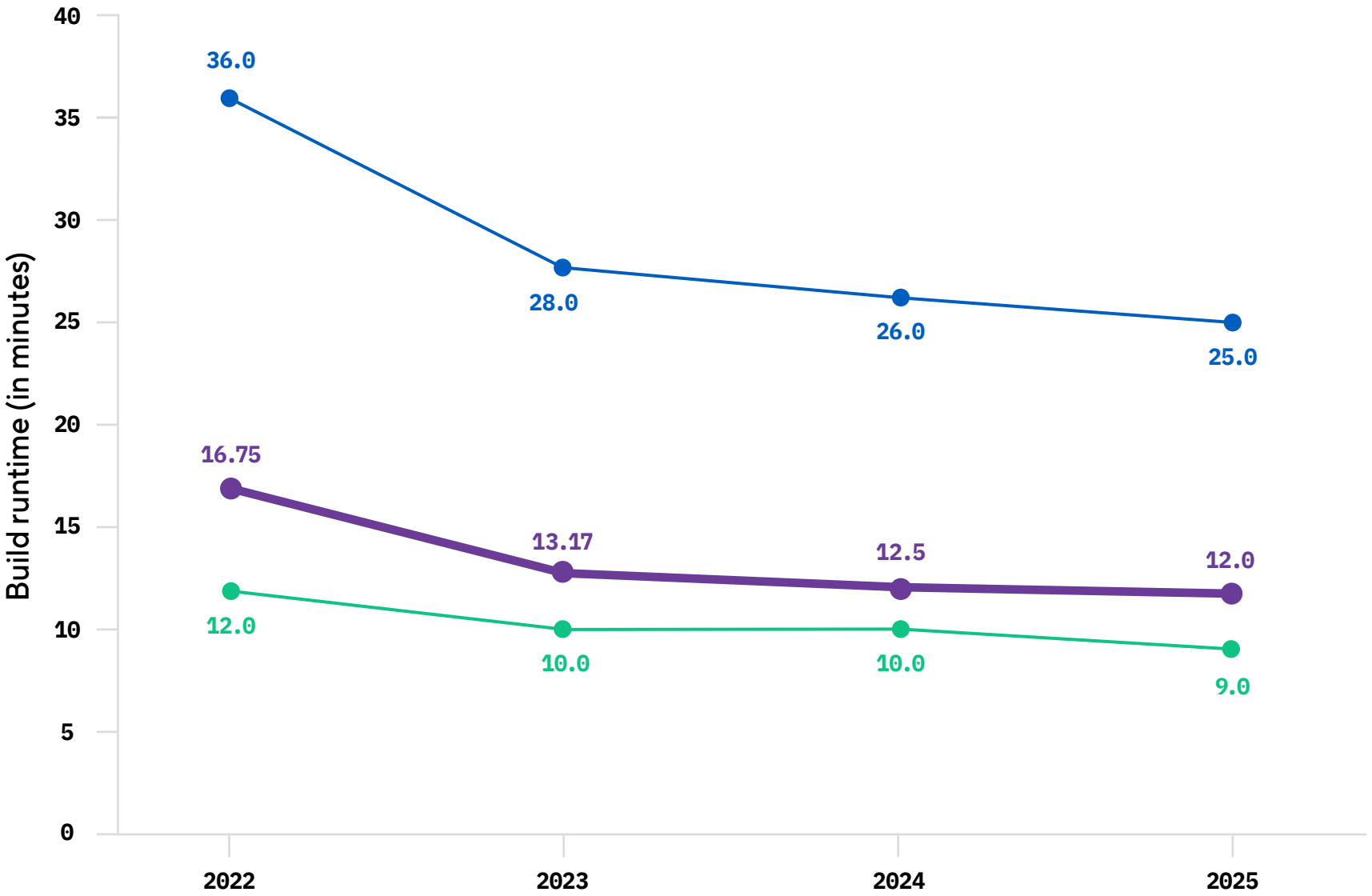
Insight #1

Build times decrease as complexity increases

Our data shows build times dropped 28% across Bitrise users while the number of workflow steps increased by 23%.

By leveraging optimization tools like caching and the latest hardware, our data shows it's possible to expand CI checks like linting automation, testing, and security scanning without sacrificing speed in the CI feedback loop.

Build runtimes have steadily decreased over time



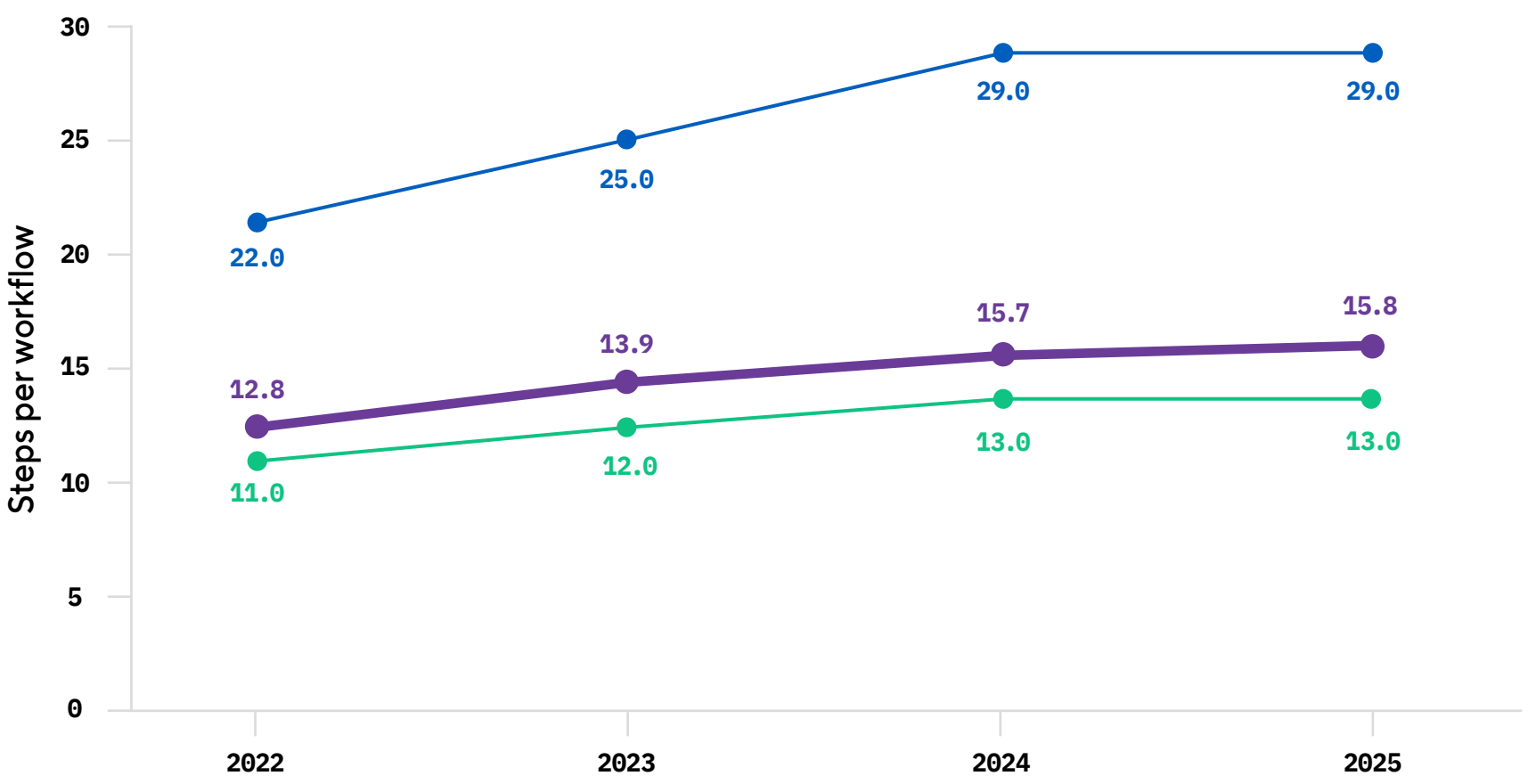
Slowest 10%
↓ 30.5%

Mean runtime
↓ 28.3%

Median runtime
↓ 25.0%

Across every cohort, build runtime decreased significantly. The slowest 10% reduced build times by 30.5%, while the median build time decreased by 25%.

Workflow complexity is ramping up



Most complex 10%
↑ 31.8%

Mean steps per workflow
↑ 23.4%

Median steps per workflow
↑ 18.8%

Across Bitrise users, average build times have declined by 28%, even though average steps per workflow increased by 23%, meaning more work is being performed, which could include additional quality assurance and security checks.

Build caching consistently delivers speed gains across complex builds

Build caching is the process of storing and reusing outputs from previous builds so that unchanged parts of your codebase don't need to be rebuilt. This can drastically reduce build times, especially for large projects, because you're not recompiling or reprocessing the same code over and over.

Our data shows that teams that adopt build caching see a 25% reduction in median build time, and up to 30% on the slowest builds.

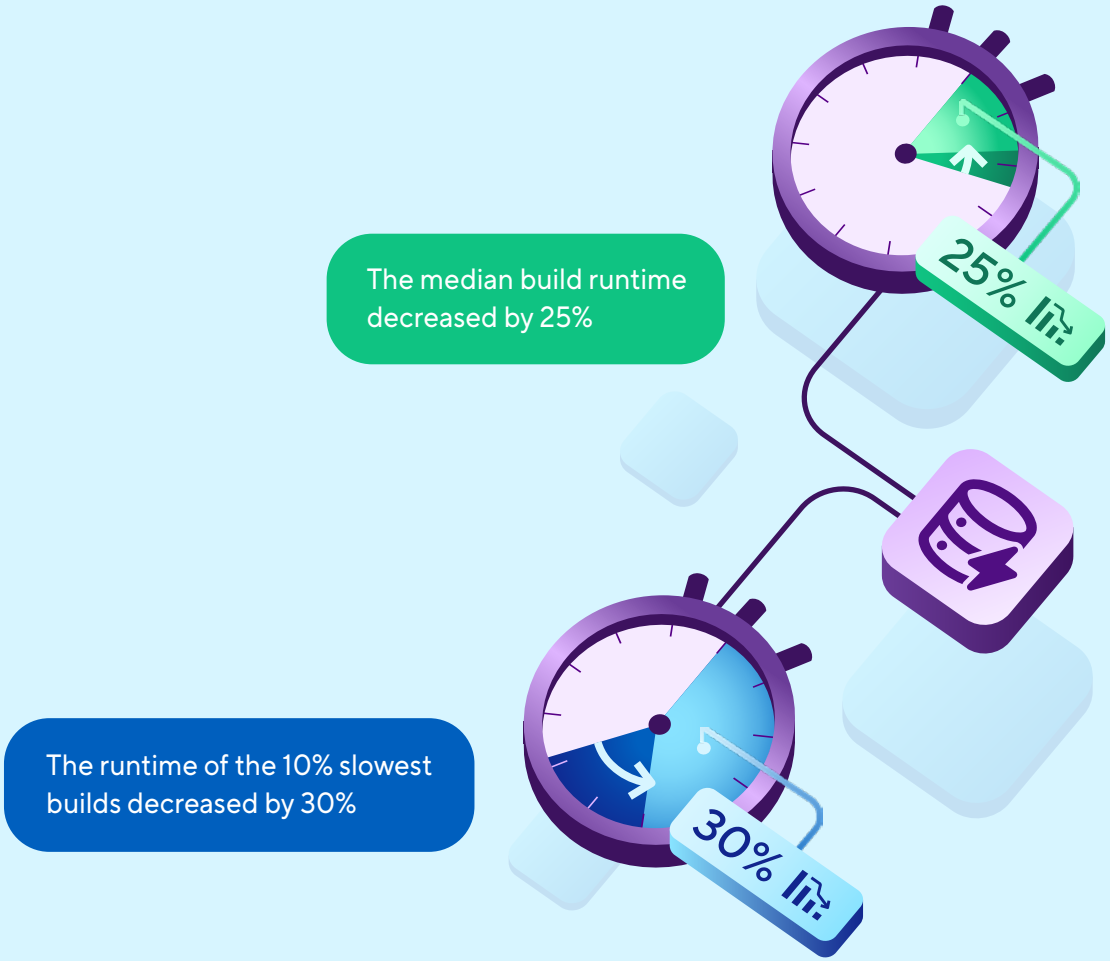
At scale, those small gains translate into:

→ Hours of waiting time saved every day

→ Lower compute costs

→ More time for developers to focus on higher-value work

Impact of build caching on build time (2025)



Bitrise top tip

Enable build caching now

Bitrise Build Cache supports Xcode (new), Gradle, and Bazel; turn it on, measure cache hit rate, and optimize the most frequently run workflows first.

Customer insight

"The biggest win has been the boost in build speed. With Bitrise, our build times have dropped from over 30 minutes to as little as 5 in some cases. Now, I can kick off a build, grab a coffee, and come back, and it's done. The whole process is much more straightforward and predictable. I can focus on one thing at a time, which is every developer's dream."

Josh Walker
Principal Software Engineer
@BuzzFeed

BuzzFeed

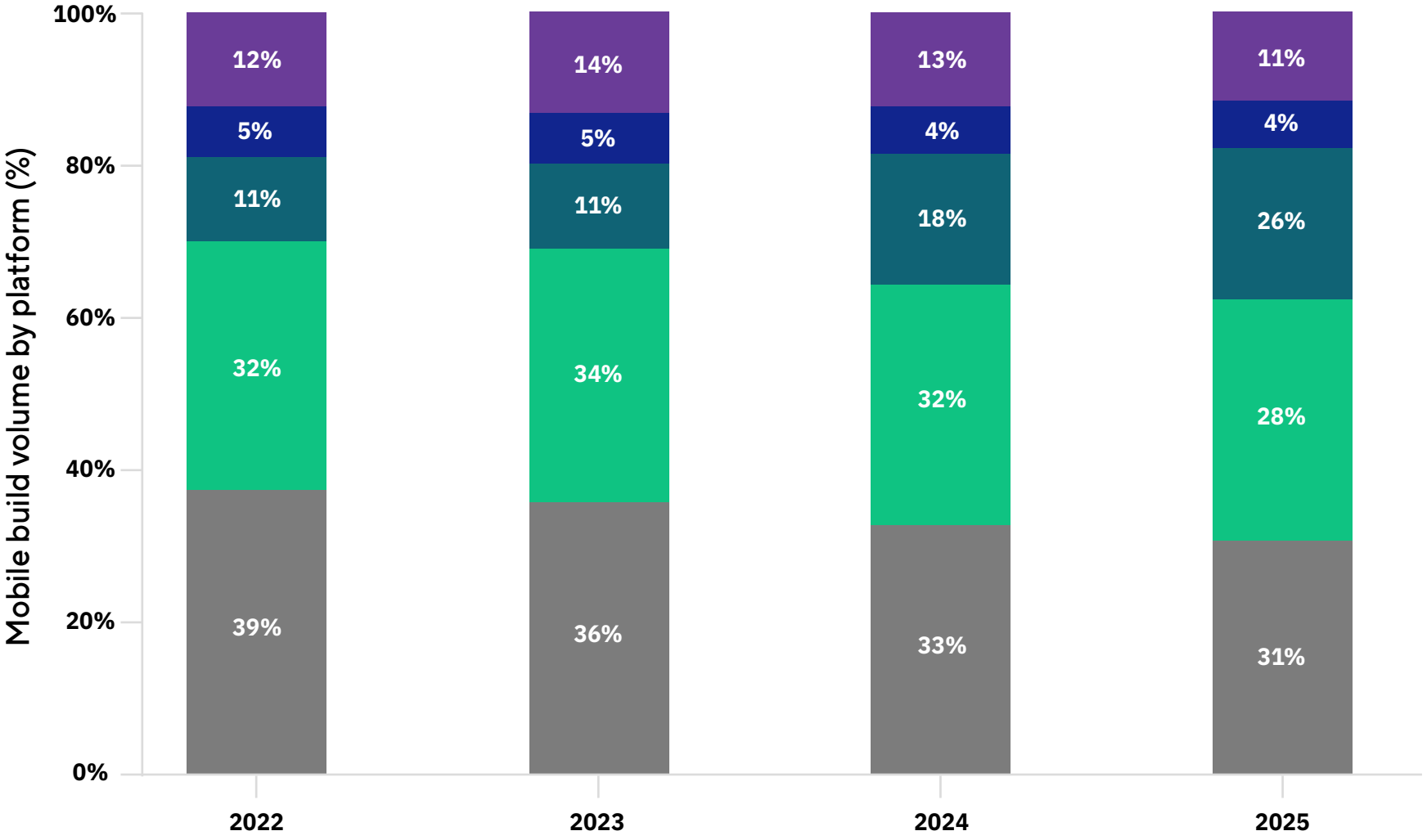
Insight #2

Cross-platform adoption is gaining traction

Cross-platform frameworks are going mainstream, reflecting the increasing pressure to deliver fast, consistent, high-quality experiences across devices.

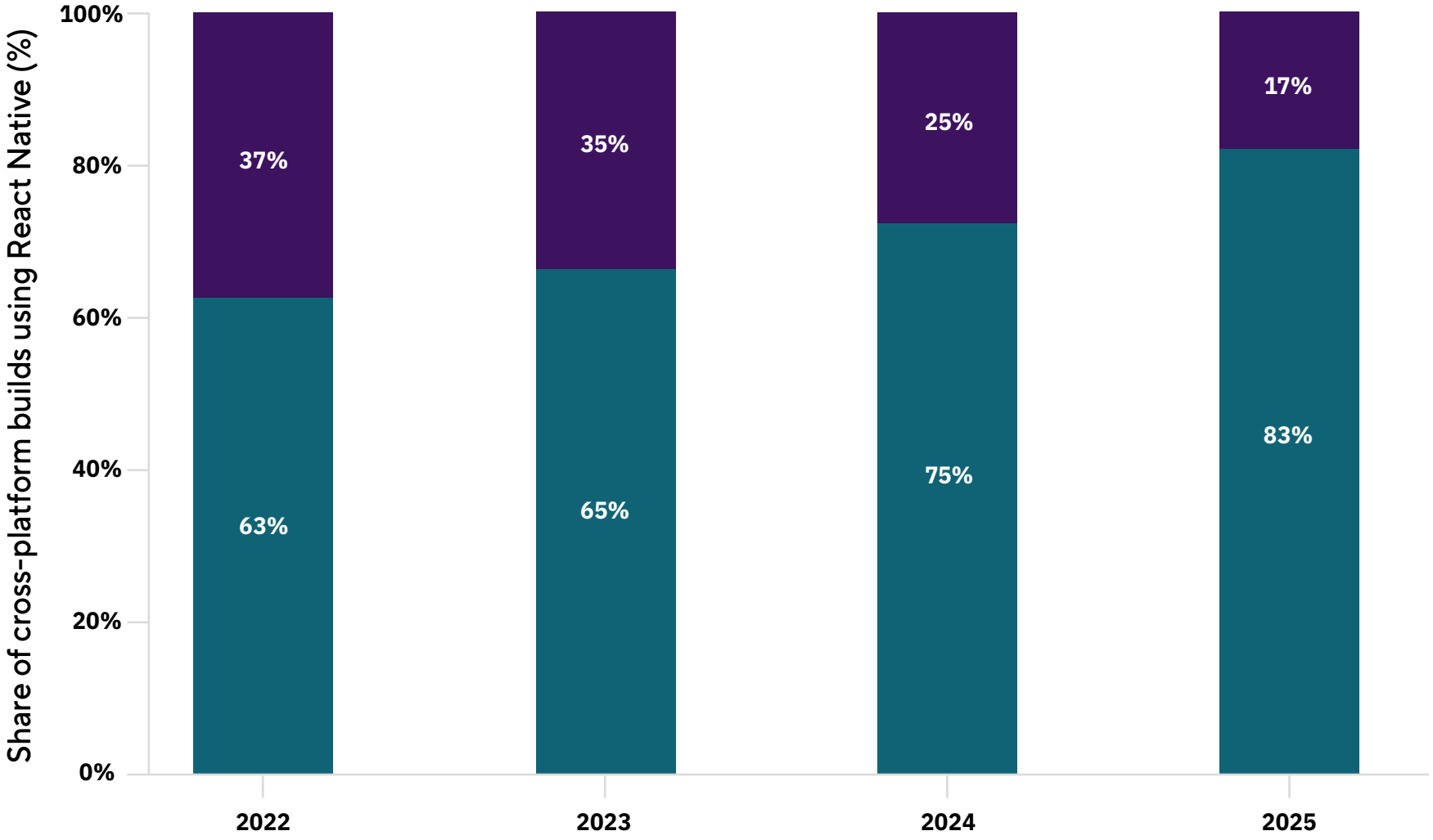
To be future-ready, your CI/CD should be framework-agnostic, ideally able to handle native iOS, Android, React Native, Flutter, and even Kotlin Multiplatform.

30% of builds are now cross-platform



The use of cross-platform frameworks increased to 30% across our customer base in 2025, with React Native accounting for 26% and Flutter 4%.

React Native leads cross-platform growth



Within the cross-platform segment, React Native accounted for 63% of cross-platform builds in 2022 and 83% in 2025.

Bitrise top tip

Accelerate cross-platform builds

Use Bitrise Pipelines to split CI workflows for cross-platform apps into smaller, parallel build and test execution workflows (like concurrent execution of shared tests, iOS-specific tests and builds, and Android-specific tests and builds). Then bring them back together for artifact deployment. This speeds CI feedback loops and increases developer velocity.

Customer insight

"We're seeing more demand for shared business logic between mobile and web. Technologies like Kotlin JS and WebAssembly are emerging to meet that need, and consistency is going to continue to be really vital. That's where Bitrise comes into play again. It's already ahead of the curve. That level of flexibility and readiness is a huge advantage."

Gary Butcher
Chief Technology Officer
@Apadmi

APADMI

Insight #3

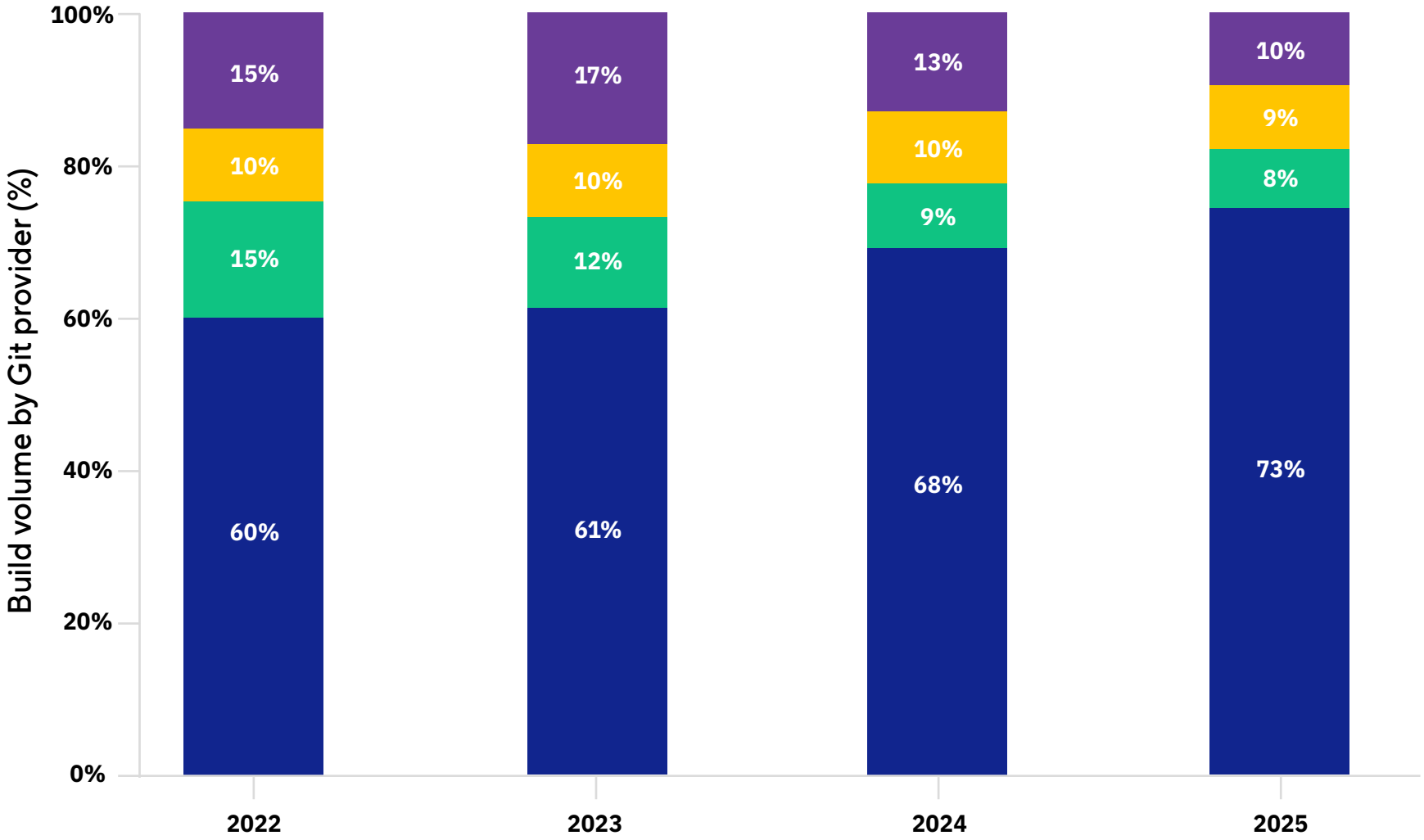
The version control ecosystem is consolidating

Our data suggests that GitHub is strengthening its position as the leader in source code management, with 73% of Bitrise builds originating from its platform triggers.

For those who have selected GitHub as their git forge, success depends on deep integration with GitHub's ecosystem while supporting mobile development's unique hardware-specific requirements, idiosyncratic deployment processes, and complex dependency management.

Unlike basic webhook integrations, modern mobile teams require deep GitHub integration including fine-grained trigger management, secure access patterns, robust status reporting, and mobile-specific optimizations. Seamless integration is key for providing a superior developer experience.

GitHub's dominance in source code management is growing



Based on Bitrise data, the number of builds triggered from GitHub has increased from 60% to 73% over the past three years. In contrast, Bitbucket's share of total build volume has declined from 15% to 8%, while GitLab's dipped slightly from 10% to 9%.

Bitrise top tip

Gain an edge with integrations

Whichever git forge you select, ensure there are strong integrations available with your CI that are designed to run mobile specific workloads.

Customer insight

"At Elli, we are exploring a move from Azure to GitHub for code hosting, as it offers tighter integration with Bitrise out of the box and opens up even more possibilities for the future."

Ganesh Bala Subramanian
Engineering Team Lead
@Elli, part of the Volkswagen Group

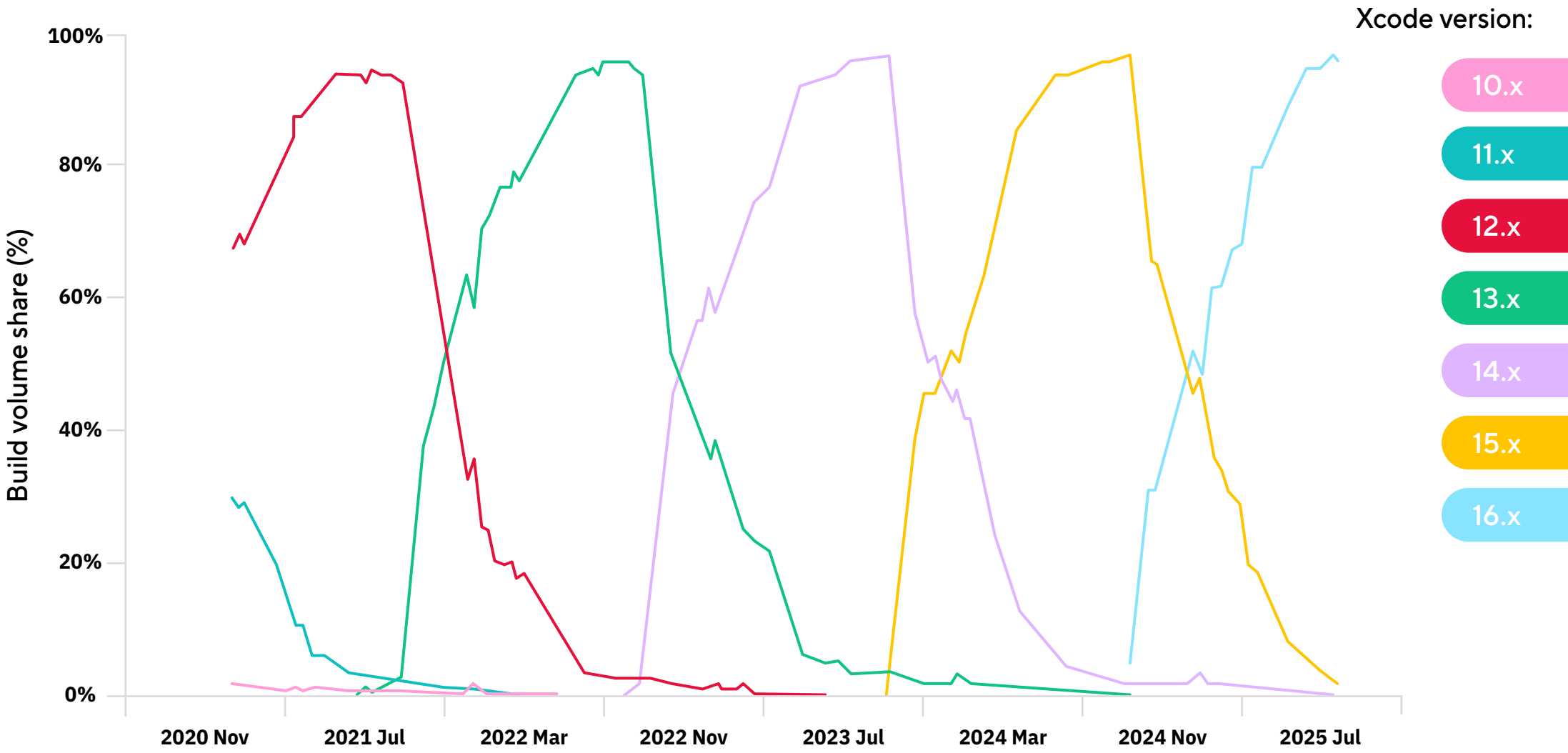
Insight #4

Winning teams move quickly on new Xcode releases

Xcode updates matter because they are tightly coupled to the new iOS versions that end-users install. Teams need to adopt new Xcode releases quickly to access the latest APIs, avoid functionality issues, and test against pre-release betas to catch breaking changes early.

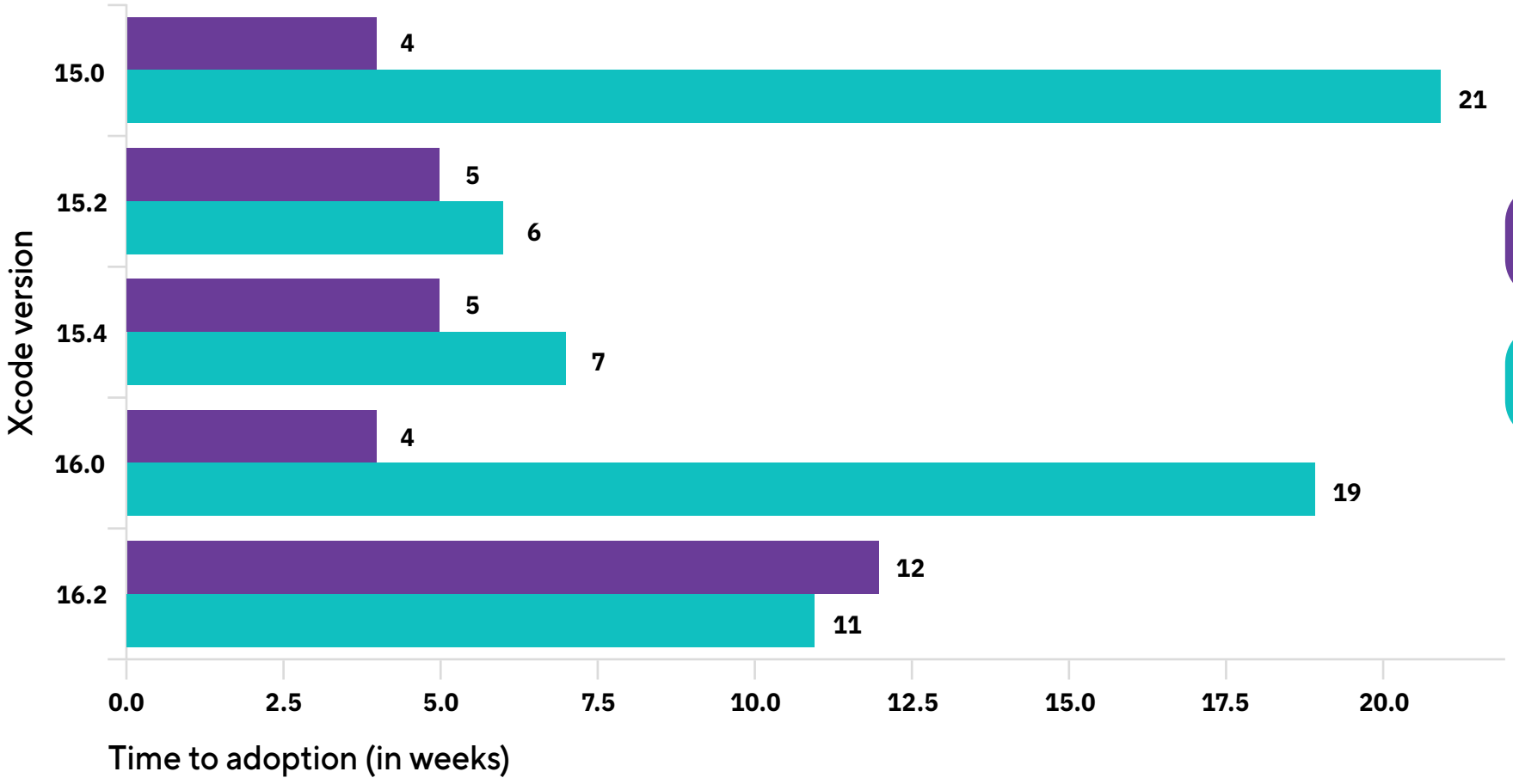
Quick adoption is also a business advantage, as Apple often spotlights apps that showcase new OS features—making efficient delivery a key App Store differentiator.

Xcode adoption is fast for high-volume builds, but slow for most customers



Our data shows high-volume builds typically move to a new major Xcode version within about four weeks, but most dev teams don't switch for 19–21 weeks. The long tail of Xcode adoption is likely driven by a combination of factors including project complexity, third-party dependency compatibility issues, and the perceived risk of migrating business-critical applications.

The fastest teams adopt new Xcode within 4 weeks



#1
By build vol.

#1
By dev teams

Our data shows that the newly released Xcode versions become the most used by build volume in ~4 weeks and reach majority adoption by development teams in 19–21 weeks. So, if it takes you more than a month to adopt a new version of Xcode, you're in the laggard group.

Apple is known to spotlight app publishers that use its newest operating system (OS) features, which means an efficient software delivery process is an App Store differentiator.

Bitrise top tip

Stay on top of Xcode updates

A scalable build environment means you can keep your primary workflows on the latest stable macOS/Xcode versions, while testing new Xcode betas in a separate canary workflow. If you don't adopt new Xcode versions quickly, you risk falling behind—both in compatibility and in shipping velocity.

Customer insight

"Before Bitrise we had to wait weeks, even months, to receive access to new releases. Without access, we couldn't build, test, or experiment with new features or widgets on iOS or Android—leaving us constantly playing catch-up."

Josh Walker
Principal Software Engineer
@BuzzFeed

BuzzFeed

Insight #5

Pipelines rule but simplicity reigns

Robust pipelines enable fast, reliable feedback by running tasks in parallel to reduce wall-clock time. Over the past 2.5 years, we've seen multi-workflow pipelines grow from 16% to 42% of total customer builds. Most pipelines are simple, averaging three workflows per run, while the top 10% are highly advanced.

Example quick wins with pipelines include:

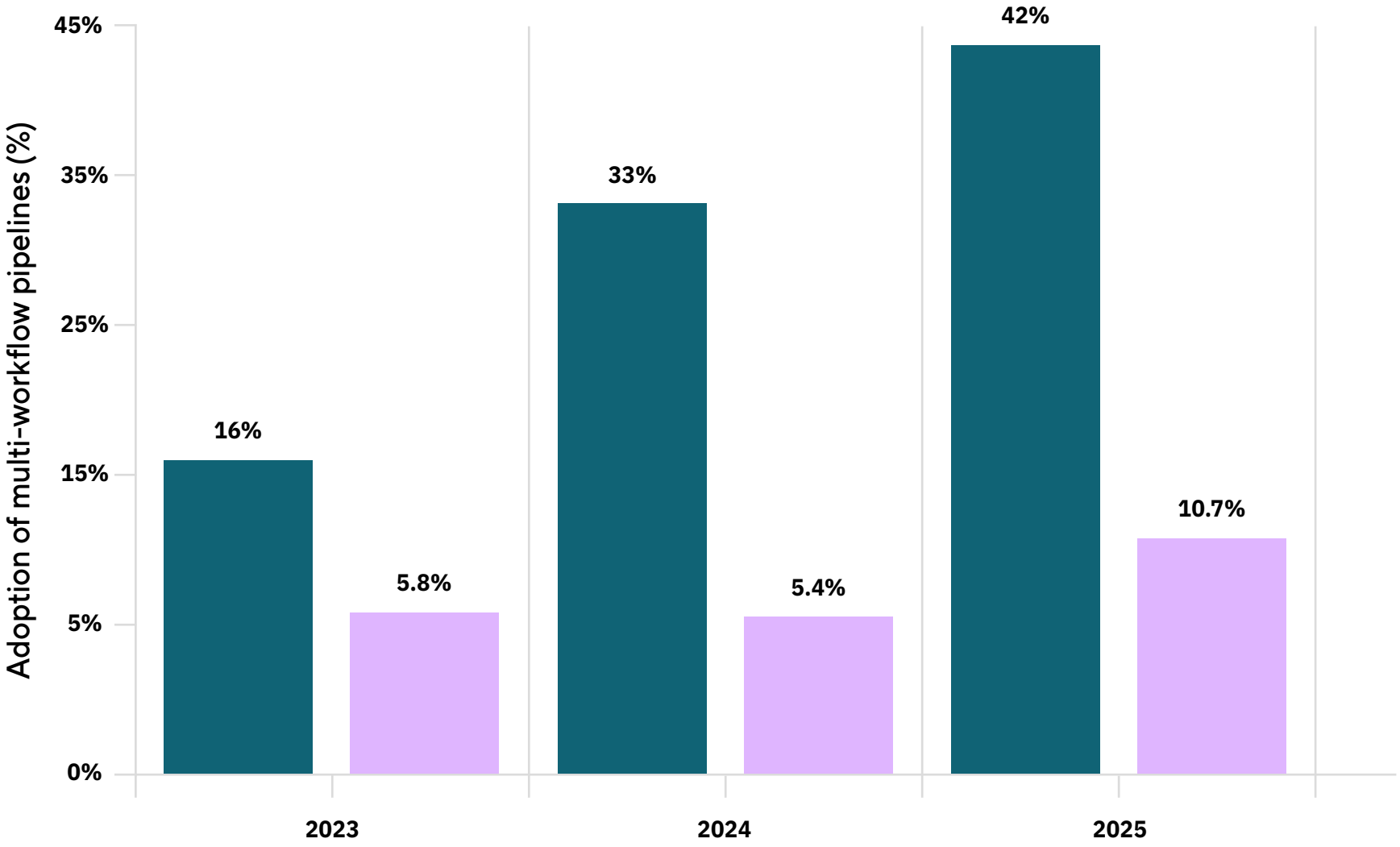
→ Test sharding and parallel execution

→ Parallel iOS and Android builds (cross-platform projects)

→ Static analysis alongside unit tests

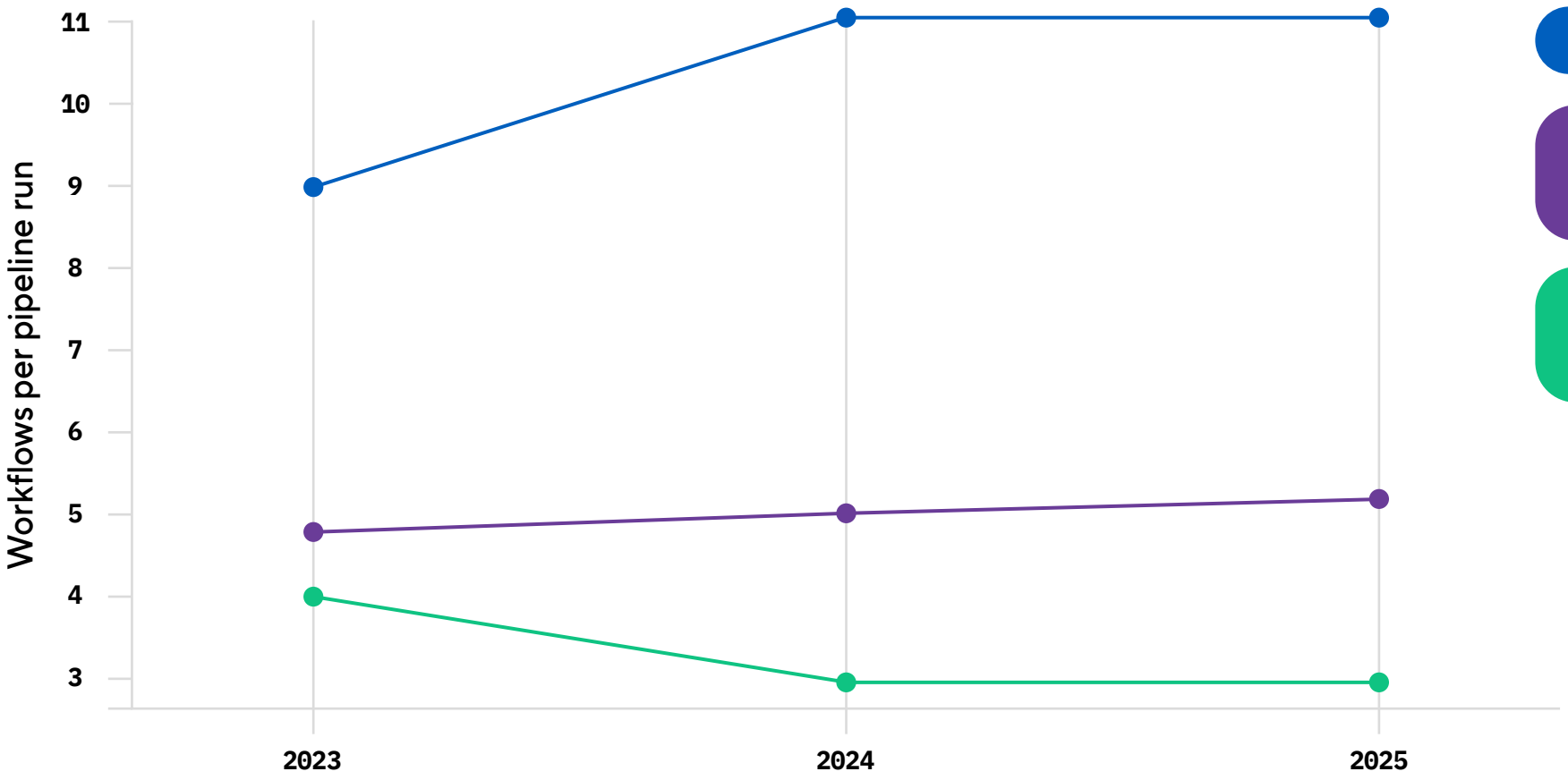
In summary, pipelines are useful for projects of all sizes, and some very large projects implement advanced control mechanisms across dozens of parallel tasks.

Pipeline usage increases significantly



The share of builds executed as multi-workflow pipelines grew from 16% to 42% in under 2.5 years, and the share of active dev teams using pipelines also increased over the same period. Together, these trends indicate broader reach and deeper adoption across teams.

Most teams favor simple pipelines



Most teams keep their pipelines simple. In fact, the median workflows per pipeline dropped from 4 to 3. However, power users (the most complex 10% of runs) increased from 9 workflows to 11, indicating the growing use of advanced parallelization. Multiple, smaller workflows can reduce wall-clock time, though overhead can diminish efficiency gains.

Bitrise top tip

Shard tests to cut build time

Split large test suites into time-balanced shards and run them in parallel across multiple workers. Add a final join step to merge results, coverage, and artifacts. Fail the run if any shard fails and, if you re-run, only run the failed shards for faster results. This keeps feedback fast as your project grows.

Customer insight

"We've significantly improved our release time by using multiple pipelines to run our tests faster. Our release time has been shortened from four days, almost a week, to just one day. That's only been possible thanks to the automation and tooling combination provided by Bitrise."

Francisco José Cantero González
Staff iOS Engineer
@Qonto

Qonto

Insight #6

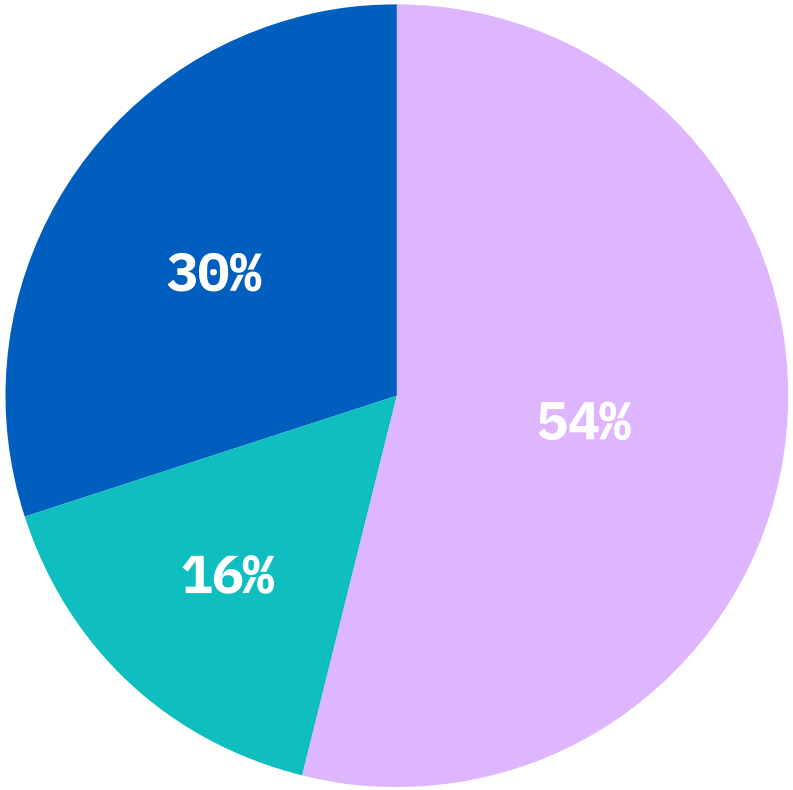
Mature release practices shorten cycle time—even as complexity increases

Our data shows that mobile teams that adopt a structured framework for managing releases, like Bitrise's Release Management tool, consistently ship more frequently, even as application size and complexity increases.

Shipping more frequently matters because it, along with the other DORA metrics, has statistically predictive benefits to organizational performance (like profitability) and non-commercial performance (like employee satisfaction).

More than half of dev teams release every two weeks

Release frequency across Release Management users



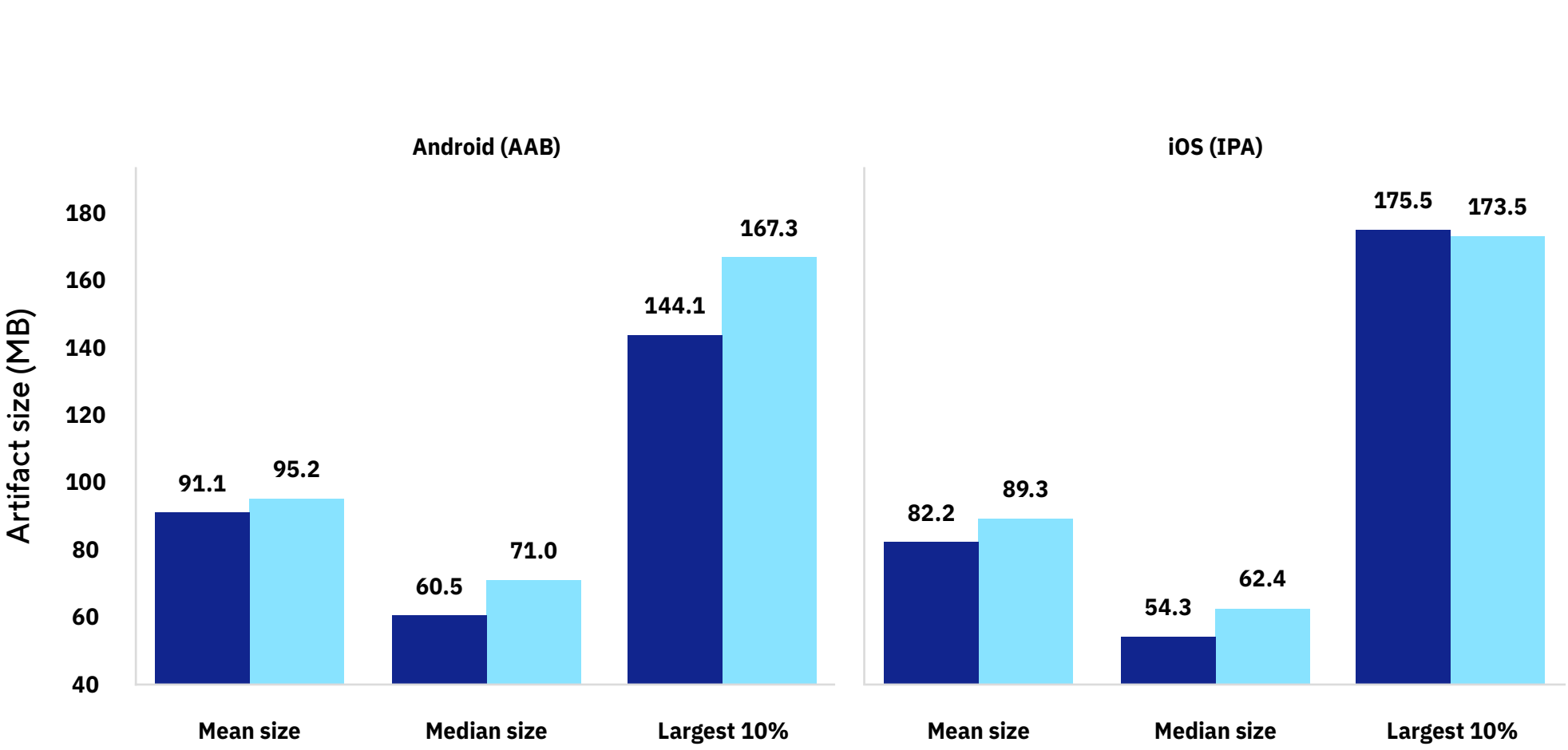
Biweekly or faster
54%

Monthly
16%

Less freq. than monthly
30%

Most teams are shipping faster with Bitrise Release Management: 54% release at least every two weeks. That's twice the cadence of the top 100 apps in the stores.

Artifact size is increasing across platforms



Most dev teams using Release Management still maintain a two-week release cadence, even as their apps grow and pipelines become more complex.

Bitrise top tip

Make release cadence a priority

Set a release schedule and stick to it. If your scope becomes more complex, avoid moving the release, instead adjust the scope to hold the cadence.

Customer insight

"With Bitrise Release Management, we have been able to fully automate the release processes for both our platforms and create a suite of workflows for each team to use in testing and quality assurance (QA)."

Ben Piatt
Mobile Software Engineer
@Maven Clinic

MAVEN

Insight #7

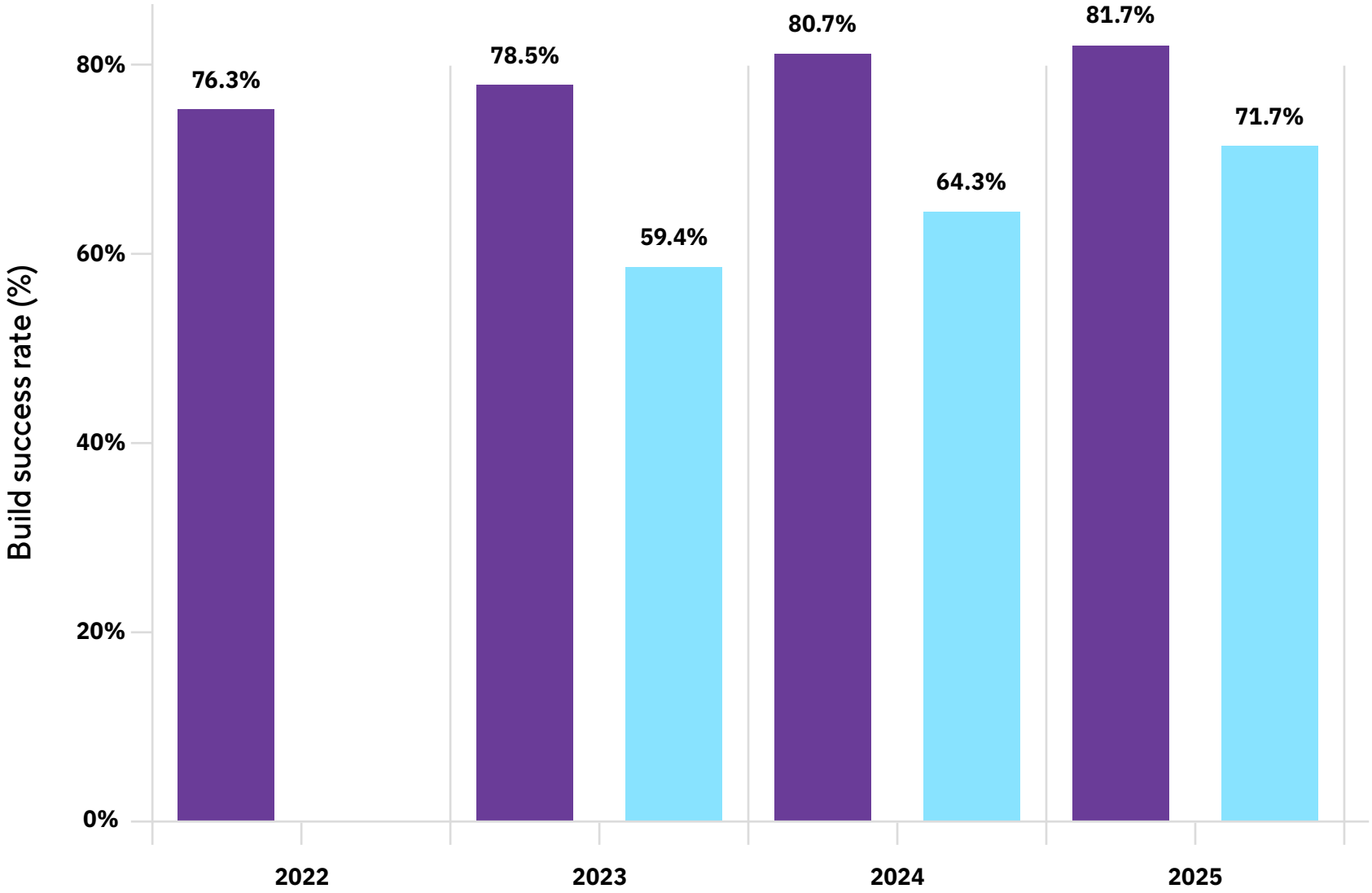
Flakiness is rising, but visibility strengthens reliability

As pipelines grow more complex and teams ship more frequently, monitoring quality and process metrics becomes critical.

A surprising trend we found in our data is that the probability of false positives or flaky tests is steadily on the rise. This matters because flaky tests erode trust in CI, engineers waste time rerunning builds, and they also waste cycles debugging faulty test code. In addition, the need to rerun the tests costs time and money and the overall benefit of CI optimizations are negated by the increasingly higher number of reruns required.

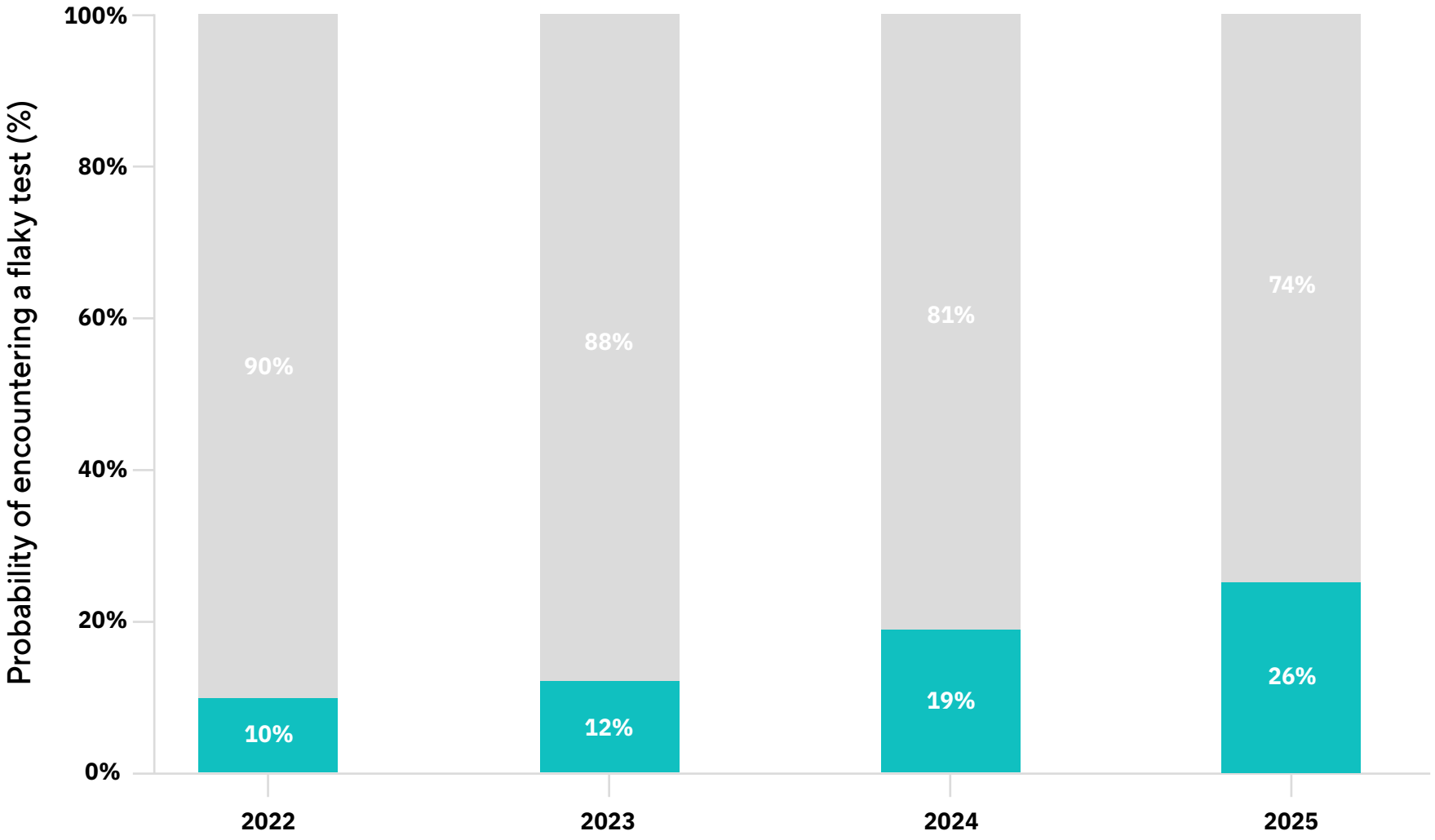
However, teams using monitoring and observability tools see measurable improvements in reliability and experience fewer wasted runs. In essence, visibility into failure patterns, test flakiness, and pipeline performance enables targeted remediation.

Overall CI/CD process reliability is improving



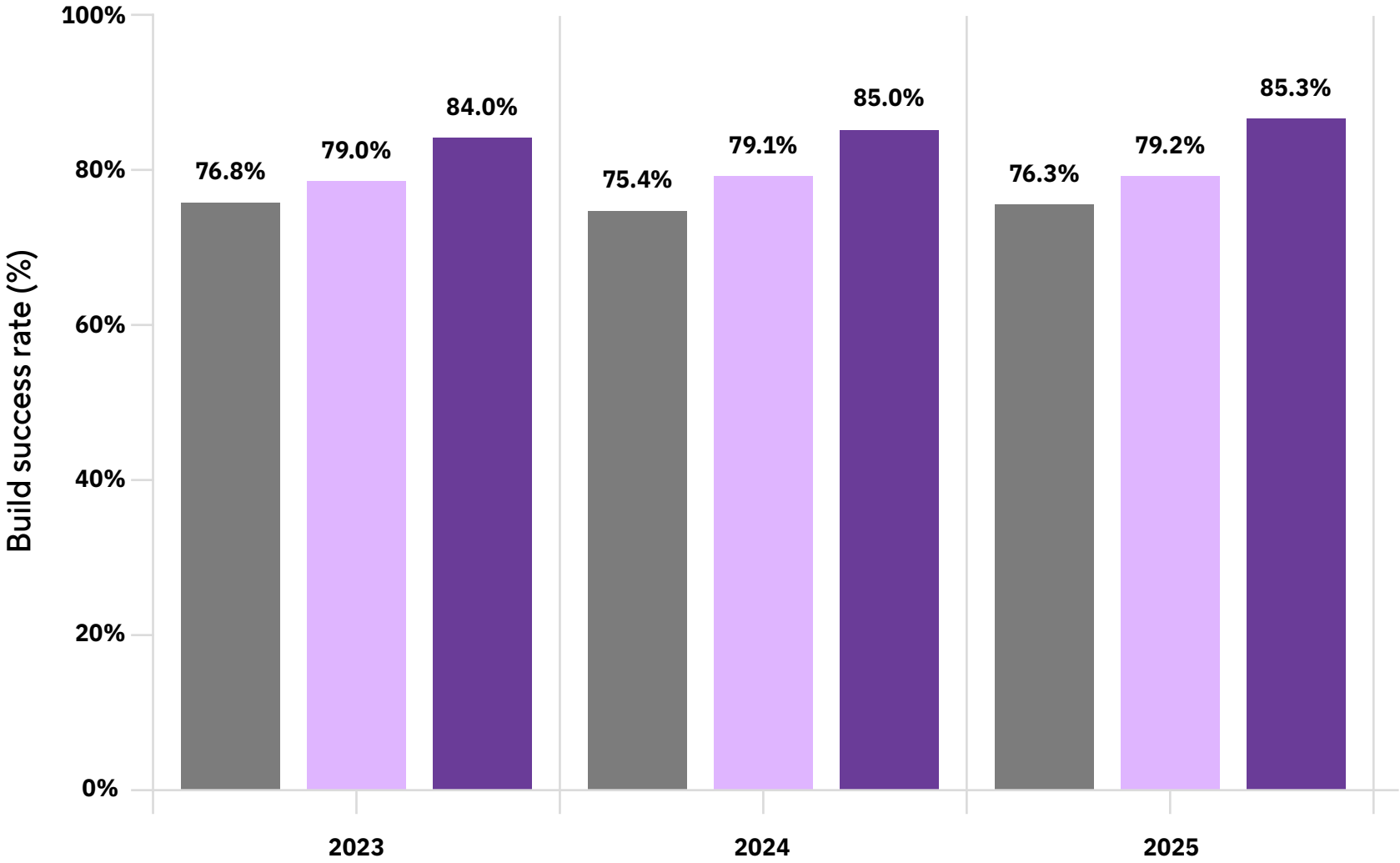
Build success rate is improving on the workflow and pipeline level. The data shows a clear, positive trend in overall CI/CD health, indicating that teams are getting better at managing core build and dependency issues.

The likelihood of flaky tests is increasing



As testing becomes more comprehensive, flakiness remains a persistent challenge. By 2025, 26% of our customers can expect some level of flakiness when running a workflow, an increase from 10% in 2022.

Better observability leads to higher build success



Dev teams who invest in visibility (Bitrise Insights) achieve higher success rates and fewer flaky test reruns (~25% less).

Bitrise top tip

Catch issues early to ship with confidence

Treat CI like production by:

- → Rolling out CI observability across your org
- → Defining performance and reliability SLOs for CI
- → Setting alerts on build performance regressions
- → Tracking and reviewing success, failure, and flakiness on a per test-case basis

This helps you spot problems early, keeping delivery performance (not to mention developer happiness) consistently high.

Customer insight

"With Bitrise Insights, all the data is at our fingertips. I can just go in, find the test, get the exact failure percentage, and then work on improving that. It's transformed our approach to testing."

Josh Walker
Principal Software Engineer
@BuzzFeed

BuzzFeed

Reference resources

Source

Bitrise study, Mobile Development Patterns on Bitrise: Trends & Benchmarks, 2022–2025 H1 (all statistics and figures above are drawn from this report).

Ecosystem context references

Kotlin Multiplatform Stable (November 2023)

React Native New Architecture (2024)

Flutter 3.22 (May 2024)

Android 15 (October 2024)

Xcode 16 series (2024-2025)

About Bitrise

Founded in 2014, Bitrise is the leading mobile DevOps platform that empowers over 8,500 brands worldwide, including Shopify, TripAdvisor, BuzzFeed, and more. Bitrise provides a full-stack, vertically-integrated mobile DevOps platform that unites the tools, processes, and testing frameworks engineering teams need to build best-in-class mobile experiences.

Over 400,000 developers use Bitrise's platform and its products: Bitrise CI, Build Cache, Release Management, and Insights. The company is backed by leading investors, including Insight Partners, Open Ocean, Fiedler Capital, and Y Combinator, among others.

Want to learn more?

Got questions about how to leverage these insights for your team?
Book a demo with a Bitrise mobile expert.

bitrise

Visit bitrise.io for more information.