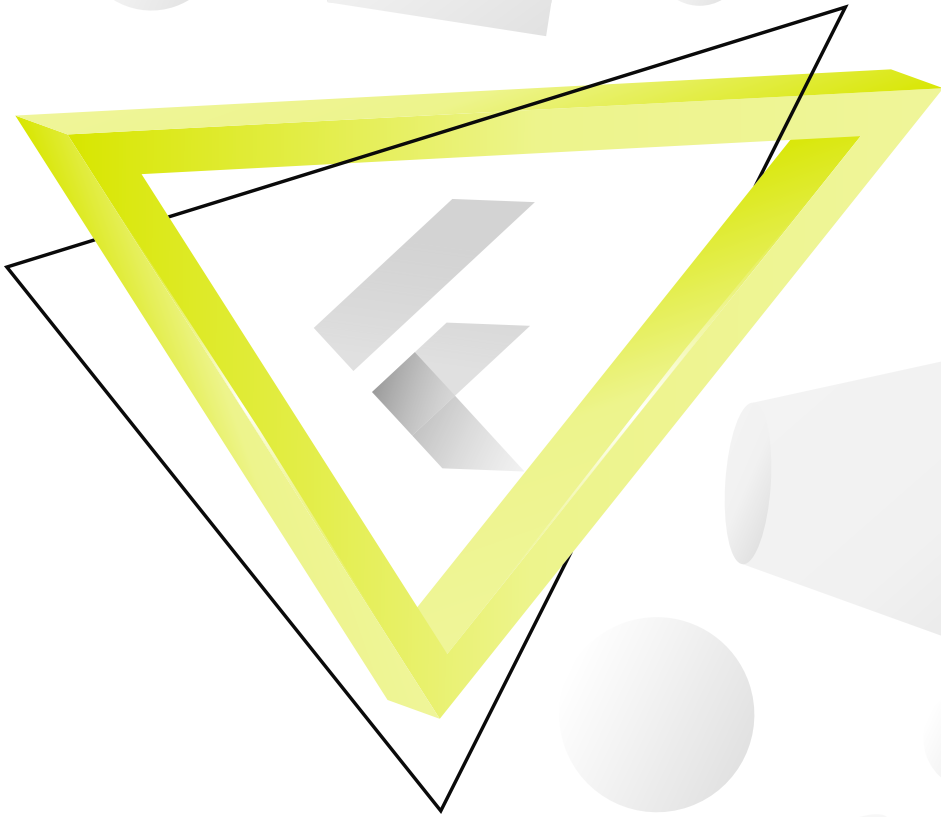
LeanCode

Flutter

CTO REPORT 2024



SURVEY RESULTS. INTERVIEWS. INSIGHTS.

Table of contents

1. CTOs and Tech Leads 10-14
2. The road to Flutter 15-24
3. Feedback after using Flutter 25-43
4. The future of Flutter 44-51

Quick links to our interviews:

1. The Future of Flutter - interview with Craig Labenz.
2. Why do we need Flutter for Web? - with Kevin Moore.
3. We are all Flutter Contributors - with Ander Dobo.
4. What are the frictions in the Flutter Universe? - with Eric Seidel.
5. The evolution of Flutter - with Martin Jeret.
6. Is Flutter Enterprise ready? - with Mateusz Wojtczak.
7. Can Dart be truly full stack? - with Viktor Lidholt.

Foreword from the Flutter Team.

Craig Labenz

Developer
Relations Engineer
at Google

When LeanCode first told me of their intentions to run an independent Flutter survey, I felt tremendous gratitude around our community's generosity and collaborative spirit. From the hosts of HumpDay Q&A to our countless open source contributors to now LeanCode's survey, Google has long known that we could never accomplish all of our goals without our community. And so first and foremost, I want to thank the team at LeanCode for conceiving, designing, and executing this survey all on their own. The Flutter team was certainly happy to assist them spread the word, but the survey's true value came from its independence. Of course, this is an attribute my colleagues within Google's walls are incapable of achieving, so initiatives like this are uniquely helpful.

I went into the results with an open mind, and as you'll see below, came out heartened - Flutter's road map is well aligned with what this survey suggests our developers want out of the framework. Of course, finite staff means not all items on the community's collective wish list are currently receiving Google's attention, but the Flutter team has already been hard at work on many of these topics for months.

One interesting result was Kotlin's strong position as Flutter developers' second favorite language. This is unsurprising for multiple reasons, including Kotlin's obvious strengths and position within the Android ecosystem; but it suggests that Dart was wise to take inspiration from Kotlin when pursuing new language features. Null-safety, records, and pattern matching were all differentiators for Kotlin that the Dart core language team worked very hard to incorporate. It's clear to me that this work delivered direct value to our ecosystem.

Another interesting result was that most new Flutter developers arrive not from other cross-platform frameworks, but from native development. When teams want to expand the reach of their established apps by adding more build targets, cross-platform frameworks make it possible to avoid multiplying their size, budget, and number of headaches. Flutter believes deeply that every great iOS app should be available on Android, and every great Android app should be available on iOS. Users around the world should all enjoy the best software experiences possible; as both developers and users.

Relatedly, 64.4% of teams distributing their software as web apps tells me that the colossal task of porting a once mobile-first framework to the web has succeeded. Admittedly, Flutter's early Web days were humble - but the tripling of positive sentiment between the previous 3 years and next 2 years is a great signal. Three out of the four most anticipated upcoming features (SEO, hot reload in JavaScript, and WasmGC) being web-related also suggests serious demand for high quality, uncompromising web apps written in Flutter. And that's great news, because the Flutter team has been working on all three for most of 2024!

The survey's Net Promoter Score of 73 and the fact that 95.7% of respondents would choose Flutter again reflect a strong vote of confidence and matches data we've seen elsewhere that suggests professional experience with Flutter more than triples a developer's probability of recommending the framework. This motivates us to keep innovating and supporting our developers to achieve even greater heights.

Despite the positive feedback, the survey also sheds light on some key areas where we can do better. Concerns about the discontinuation of product support, limitations with Web implementation, and challenges with hiring skilled Flutter developers stand out. The Flutter team is committed to working with the developer community to address these issues on an ongoing basis.

Flutter's future depends on continued alignment between our goals and your wishes, so please know that responding to surveys is its own form of contribution. Flutter's quarterly survey is our primary source of community sentiment, but LeanCode's Flutter CTO survey added a fresh vantage point of where and how Flutter can improve. I would like to thank LeanCode for their diligent work in conducting this survey and to all of you CTOs and developers who participated. Now, let's get back to work!

The evidence we have all been waiting for.

Łukasz Kosman

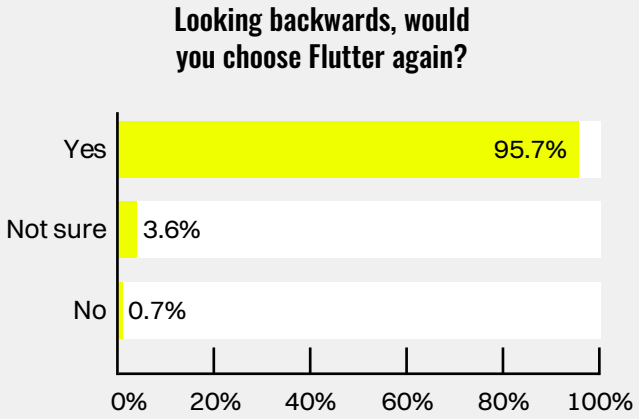
Co-founder & CEO
at LeanCode

Flutter launched in 2018 with a bold promise: code once, run anywhere. This cross-platform vision was alluring, yet met with skepticism.
Five and a half years later, the Flutter community has matured significantly. To understand the current state of affairs, we conducted a comprehensive survey, engaging nearly 300 CTOs, CIOs, and Tech Leads.

This report, based on real-world data, provides crucial insights to help you make informed decisions about incorporating Flutter into your technology stack.

What's the verdict?

Is Flutter a good choice? With an NPS score of 73, the vast majority of respondents would enthusiastically recommend Flutter to their peers. Moreover, a staggering 95.5% of respondents would choose Flutter again when looking back on their experiences.



The speed advantage is significant.

Flutter's advantages when it comes to development speed are undeniable. Over 56.4% of respondents reported a more than 50% increase in development speed compared to native approaches, with over 80% citing a minimum 20% improvement. These time and cost savings are simply too significant to ignore.

Is Flutter the ultimate solution?

Does it mean that you should close your eyes and choose Flutter no matter what? Absolutely not. This universe still has its points of friction, which are clearly highlighted by our respondents. Make sure you undertake your evaluation carefully.

Don't just take our word for it: listen to the experts.

While Flutter boasts numerous strengths, it's important to acknowledge the existing challenges. We've addressed these concerns directly through interviews with leading voices in the Flutter community.

- **Craig Labenz:** Tackling the persistent "Google will kill Flutter" narrative.
- **Kevin Moore:** Examining Flutter's readiness for building web applications.
- **Eric Seidel:** Exploring the friction points within the Flutter ecosystem.
- **Ander Dobo:** Delving into the package ecosystem and contributions to the framework.

A community-driven effort.

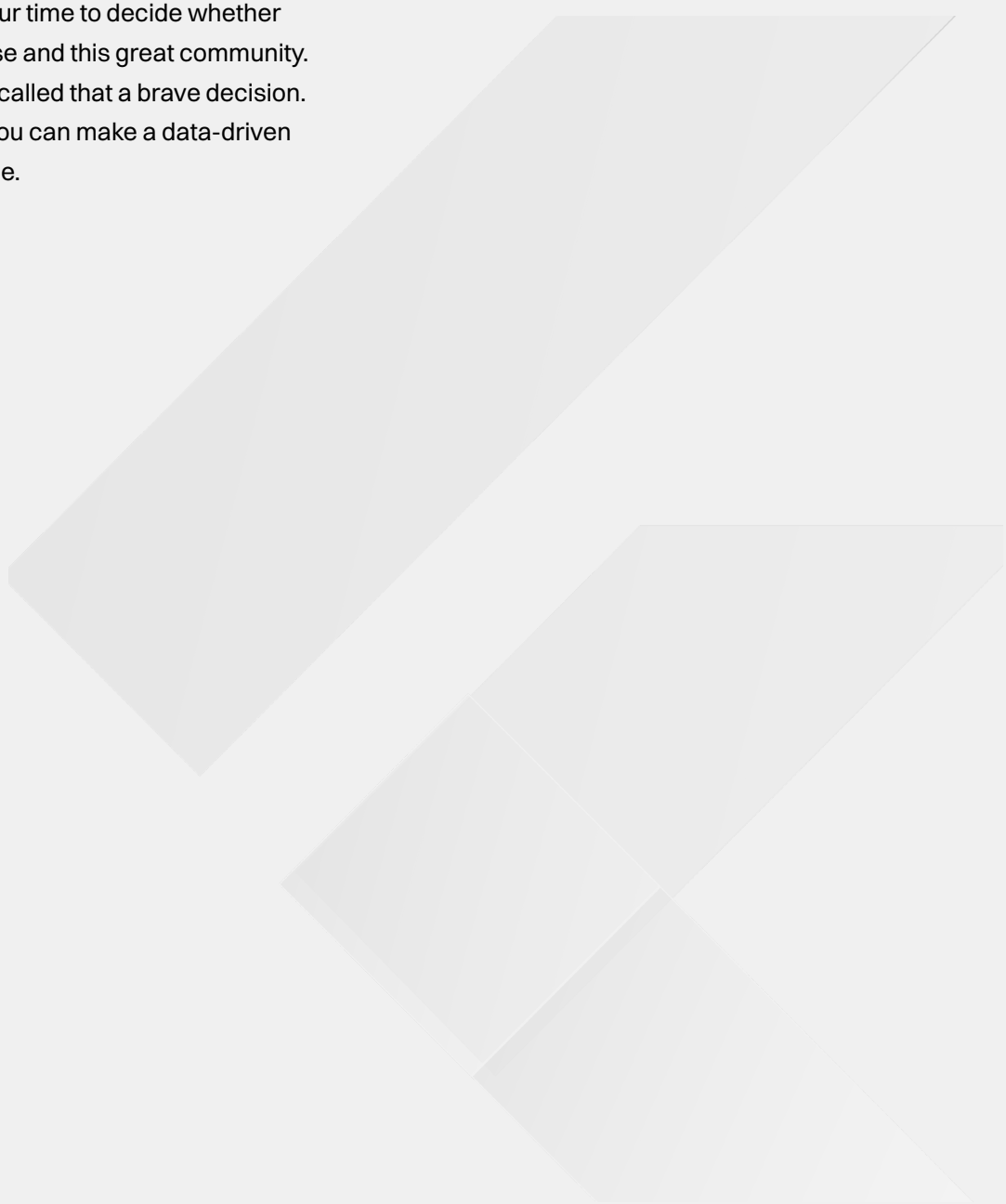
This report is a testament to the strength and collaborative spirit of the Flutter community. This initiative was completely independent, with no financial support or input from Google. We are incredibly grateful to the partners who believed in our vision and supported this project: Widgetbook, Codemagic, Talsec, Shorebird, Serverpod, Devangels, Fluttercon, Flutter&Friends, Full Stack Flutter, Flutter Warsaw, Flutter Vienna, Flutter Hamburg, Flutter Berlin and Flutter Stockholm.

Thank you to our participants.

This report would not exist without the dedication of our participants. You, our valued respondents, generously devoted an average of 19 minutes of your valuable time to share your insights. On behalf of the entire LeanCode team, thank you for your contribution.

Now it's your turn!

Dear Reader, now it's your time to decide whether to join the Flutter universe and this great community. Years ago, I would have called that a brave decision. Today, with our report, you can make a data-driven decision. You're welcome.



Flutter is hugely popular and widely adopted.

Jakub Wojtczak

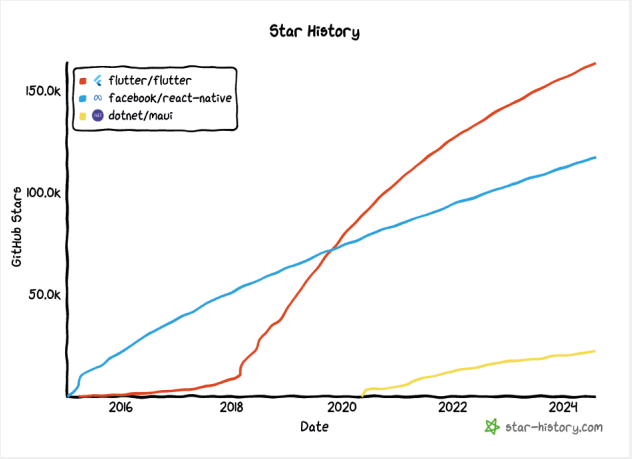
CTO at LeanCode

Flutter is becoming a default cross-platform technology.

Is Flutter still just a niche technology? That was the question echoing over the last couple of years among decision-makers who were scared to choose the stack, which can be exotic and cause maintenance problems. While it is clear that building two almost identical apps separately for both platforms makes no sense, it wasn't clear which cross-platform framework to choose.

The entry level for new technology such as Flutter was also very high, with React Native being a strong candidate to solve the cross-platform challenge.

Yet, over time, due to developers' great experience (stateful hot reload, excellent tooling, stable package, etc.), experts tend to gravitate towards Flutter, which is reflected by a renowned vanity metric: stars on GitHub.



The better developer experience is also reflected in the Flutter CTO Report, where 98% of CTOs gave positive feedback on Flutter, and 37.4% disliked React Native.

Which of the following technologies have you used in the past, and did you like or dislike them?

	Didn't use	Disliked	Liked
Flutter	0.4%	0.7%	98.9%
React Native	43.1%	37.4%	19.6%

Enterprise companies using Flutter.

This coincided with the arrival of great case studies from experienced teams working on large-scale apps, which ignited the belief that everything is possible with Flutter. The abundant evidence coming from teams like Credit Agricole, BMW, Virgin, Tide, Philips, Tencent, Betterment, Toyota, and iRobot was really eye-opening. It proved that not only are large-scale apps achievable with Flutter but also that they are performant and bulletproof when it comes to security and penetration tests.

Companies are increasingly betting on Flutter.

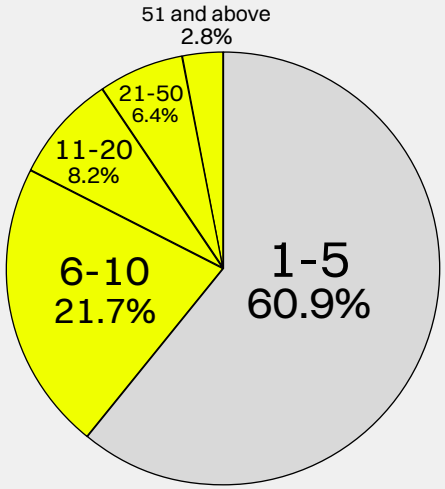
Since Google I/O 2024, we have witnessed even more examples of companies using Flutter. LG confirmed its plans to shift the Flutter-based applications to tens of millions of users, Universal Studios presented their new systems built with Flutter to visitors of their amusement parks, and GEICO selected Flutter as the framework for optimizing and innovating digital experiences across multiple channels.

How does Google use Flutter?

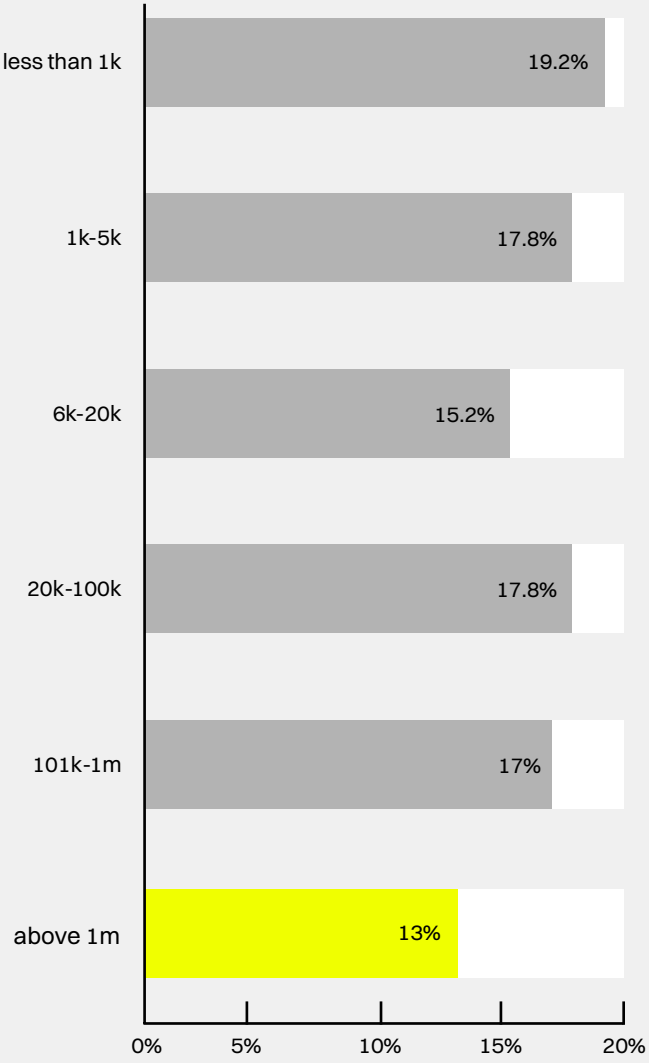
What is more, Google itself is also making significant investments in building the Flutter teams internally, which ship new products. This portfolio of internal applications includes Google Pay, Google Earth, Google Ads, Google Classroom, YouTube Create, Google Cloud, Google One, Family Link, Crowd Source, Google Analytics, and many others.

Our survey also reflects this trend. From the Flutter CTO Report, we can see that size doesn't matter. While Flutter remains powerful for PoC and MVPs, it is also widely adopted by larger teams, as almost 40% of Flutter teams have 6 or more Flutter Devs. What is more, according to the Flutter CTO Report, 13% of Flutter apps have an audience of 1M users and above.

What is the size of your mobile development team?



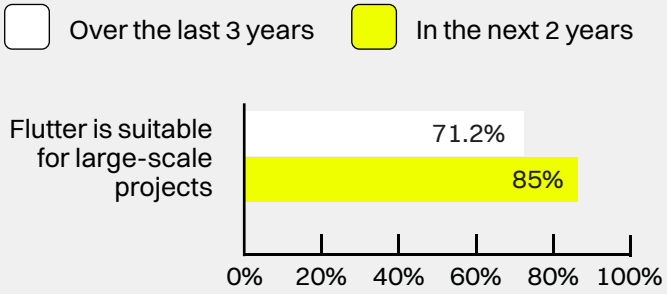
How many downloads does your most popular app have?



Not surprisingly, Flutter is also becoming a default solution for large-scale apps. In our report, a dominant 85% consider Flutter suitable for large-scale projects, in comparison to 71.2% for the past 3 years.

Please indicate the extent to which you agree with the following statements over the last 3 years/ in the next 2 years.

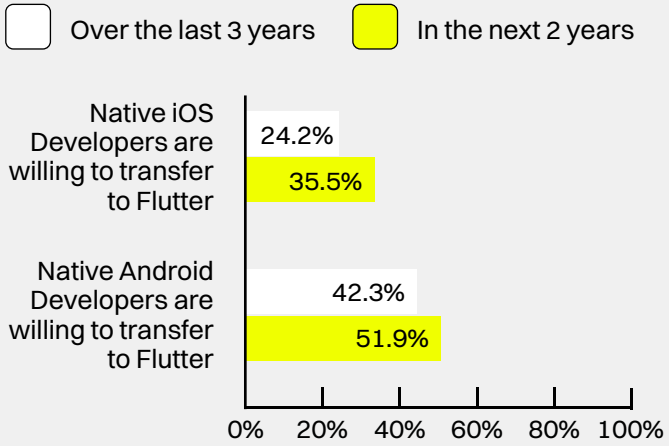
Showing only “somewhat agree” and “strongly agree” responses.



Last but not least, there is no longer a problem with the shortage of developers on the market. Over the years, while Flutter was becoming more and more popular, we could observe that a substantial share of experts gained sufficient expertise to lead complex projects in Flutter. Therefore, new developers are more willing to migrate from both native Android (an increase from 42.3% to 51.9%) and iOS (an increase from 24.2% to 35.5%) ecosystems.

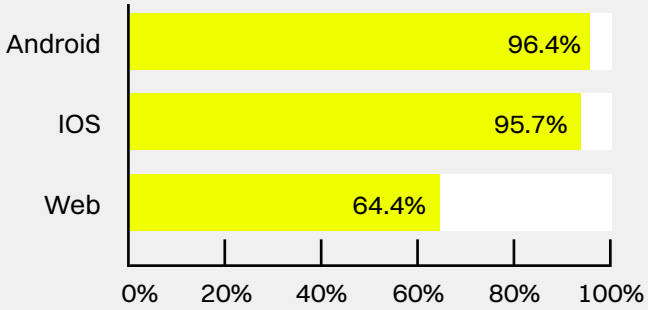
Please indicate the extent to which you agree with the following statements over the last 3 years/ in the next 2 years.

Showing only “somewhat agree” and “strongly agree” responses.



What else is worth noting and what the Report clearly shows is that Flutter’s capabilities are not limited to mobile applications. We can see that 64.4% of projects are using Flutter for Web implementation, which proves that the recent announcements regarding WASM stability are definitely a step in the right direction.

What do you use Flutter for?



All in all, we see massive evidence supporting the claim that Flutter is a widely adopted, cross-platform framework, which is becoming a default solution for achieving the goal of targeting different interfaces from one codebase.

CTOs and Tech Leads have spoken

We have gathered qualified responses from almost 300 CTOs, CIOs, Tech Leads, and other Tech Executives from around 70 countries.

In this section, you will learn the following:

1. What career paths have our respondents taken?
2. What is the geographical coverage of the survey?
3. What industries are represented?
4. What is their experience with other technologies?
5. What is their opinion on other cross-platform frameworks?

We have collected exactly 300 submissions from nearly 70 countries. The purpose of this report is to create curated data with a focus on the high seniority of participants. Therefore, any submissions made by regular developers or junior developers were eliminated from the final dataset.

As a result, the total number of qualified submissions for the 1st edition of our Flutter CTO Survey was 281.

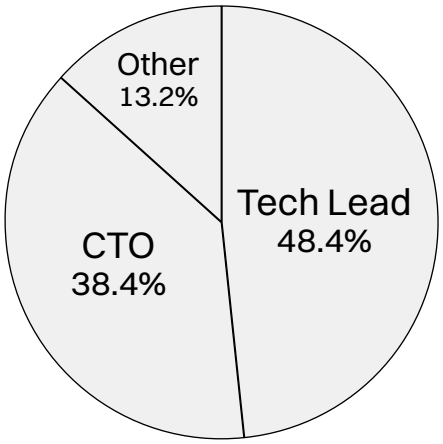
Our respondents answered 52 questions, taking an average of 19 minutes to complete the survey. We are beyond grateful for their effort and the insights they provided.

Meet our respondents.

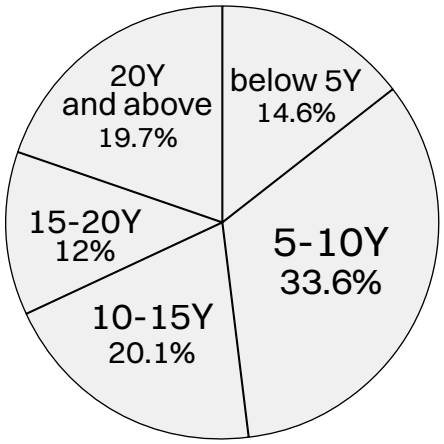
85% of respondents have more than 5Y of experience, and more than 50% have been in the IT business for more than 10Y. 40% of our respondents work in mobile teams with more than 5 developers, and 22.8% work in organizations with more than 200 people.

This clearly shows that although Flutter was initially embraced by smaller teams, the framework became increasingly popular among enterprise teams, who build large-scale applications using Flutter in their mobile development teams.

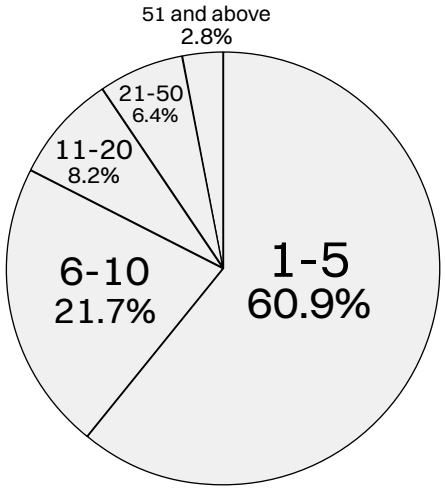
What is your current role?



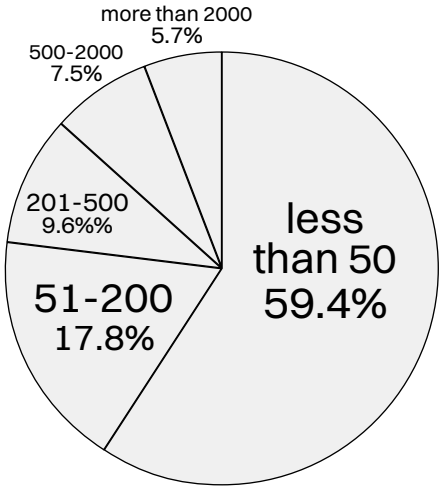
What's your total experience in years as an IT professional?



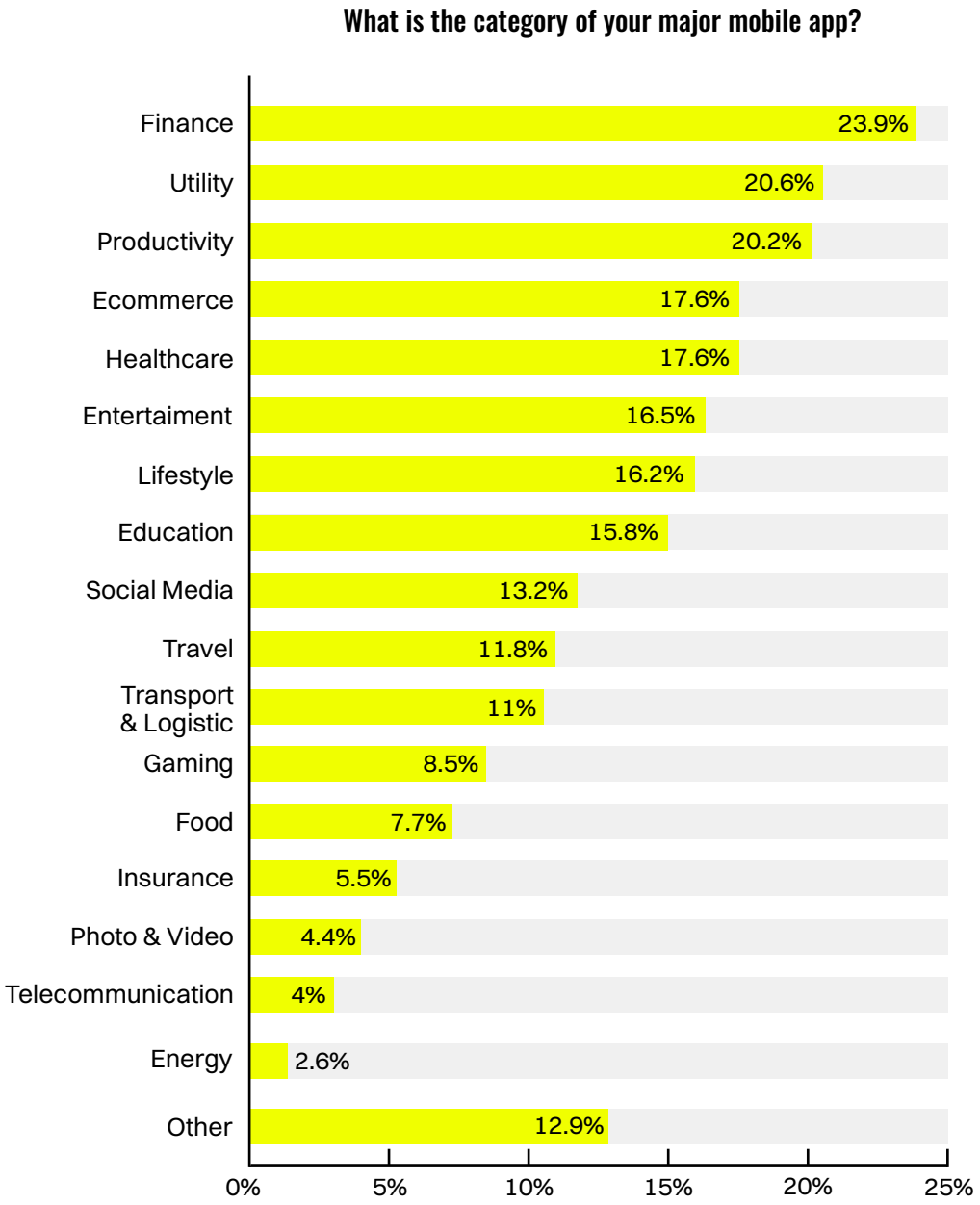
What is the size of your mobile development team?



What is the size of your current company in terms of the headcount?

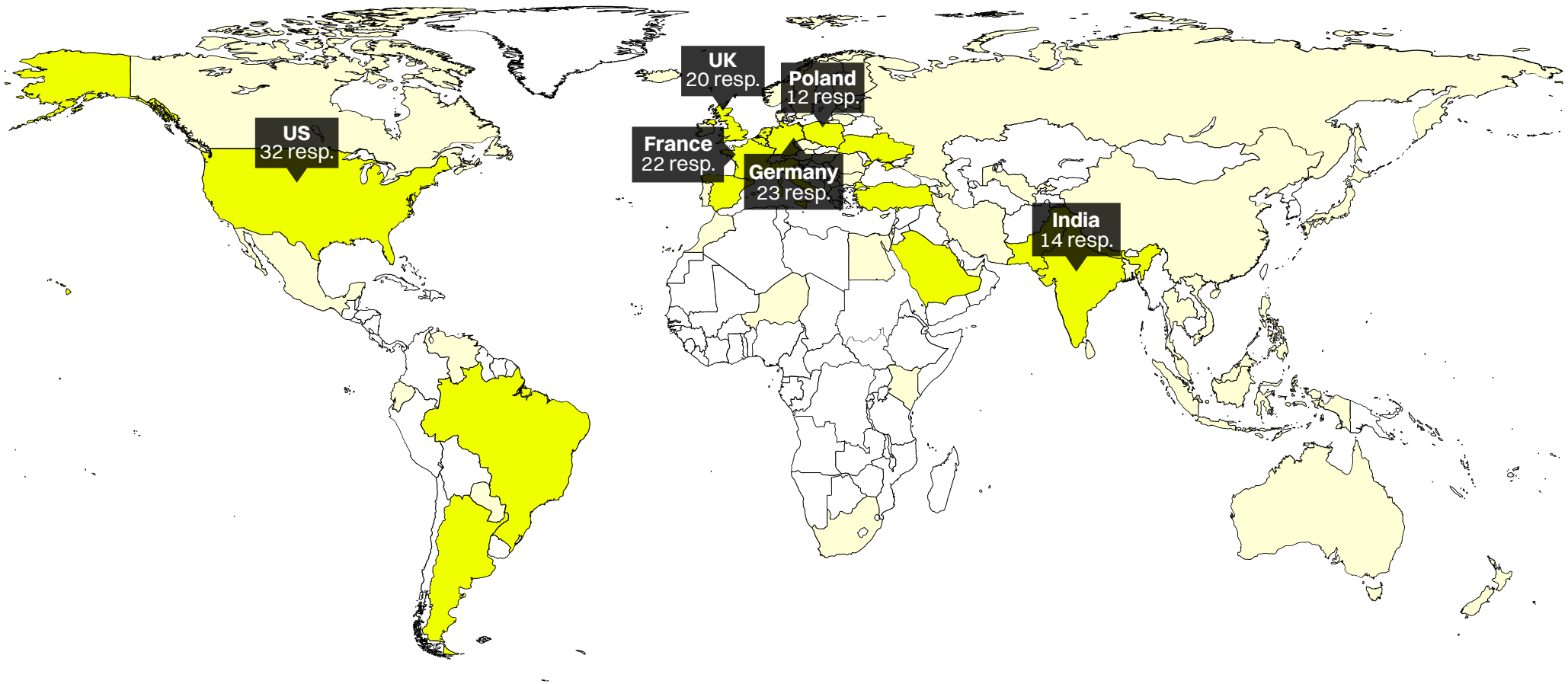


The Flutter CTO Report covers insights from all industries.



The fact that financial and healthcare-related applications are mentioned in the top 5 industries using Flutter is a clear indicator that Flutter is a reliable framework that allows users to build secure applications for the most demanding environments.

Respondents come from nearly 70 countries on 6 continents.



In which country is the HQ of your company?

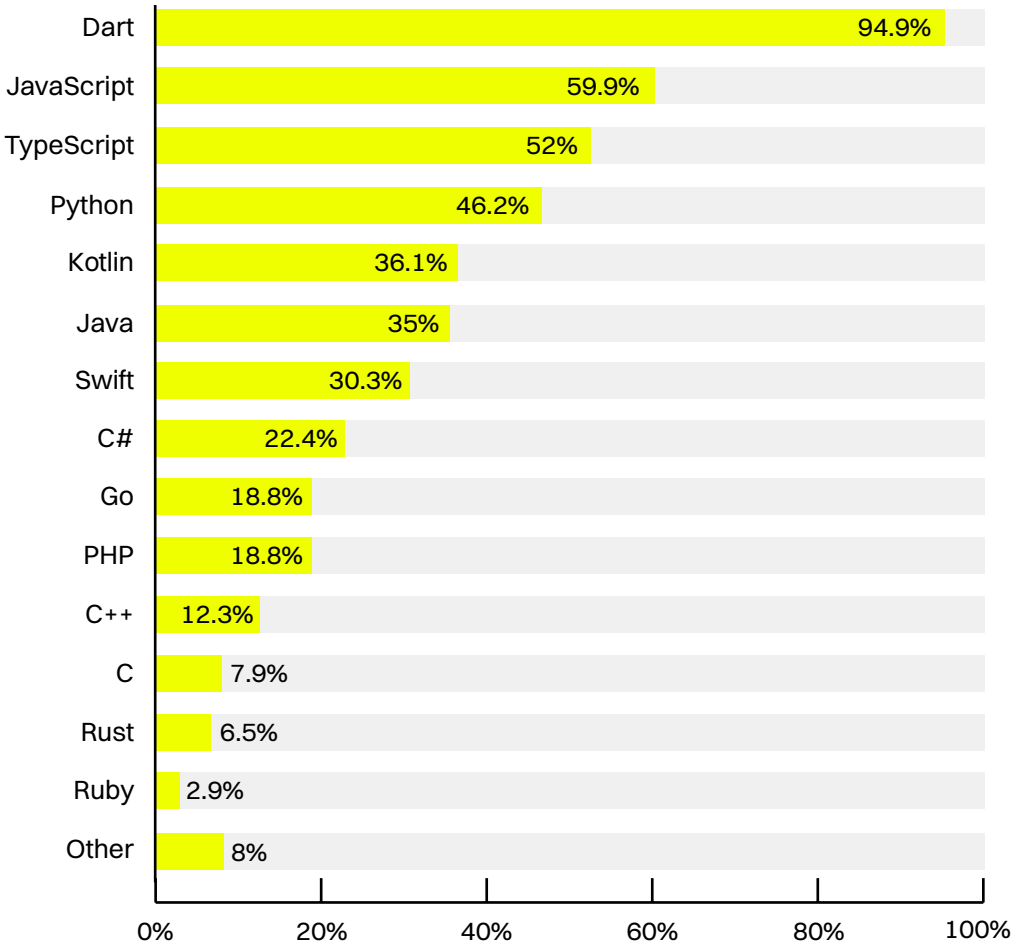
5 respondents and above

1-4 respondents

no respondents

Voices from CTOs and Tech Leads experienced in different programming languages and mobile development frameworks.

What programming languages are used in your company?



Which of the following technologies have you used in the past, and did you like or dislike them?

	Didn't use	Disliked	Liked
Flutter	0.4%	0.7%	98.9%
Ionic Capacitor	79%	18.9%	2.1%
Jetpack Compose	64.7%	7.1%	28.5%
Kotlin	32.4%	12.1%	55.5%
Kotlin Multiplatform	67.7%	12.8%	19.6%
PWA	56.2%	21.7%	22.1%
React Native	43.1%	37.4%	19.6%
Swift	35.6%	15.7%	48.8%
SwiftUI	52.7%	8.2%	39.1%
Xamarin/.NET MAUI	65.5%	28.8%	5.7%
NativeScript	82.9%	13.9%	3.2%
Apache Cordova	68.7%	28.5%	2.8%

The road to Flutter

How companies are adopting Flutter.

In this section, you will learn the following:

1. What are the typical target platforms for Flutter apps?
2. What is the level of Flutter adoption?
3. What were the key arguments for starting with Flutter?
4. What were the strongest arguments against Flutter?
5. What were the steps companies took before starting with Flutter?

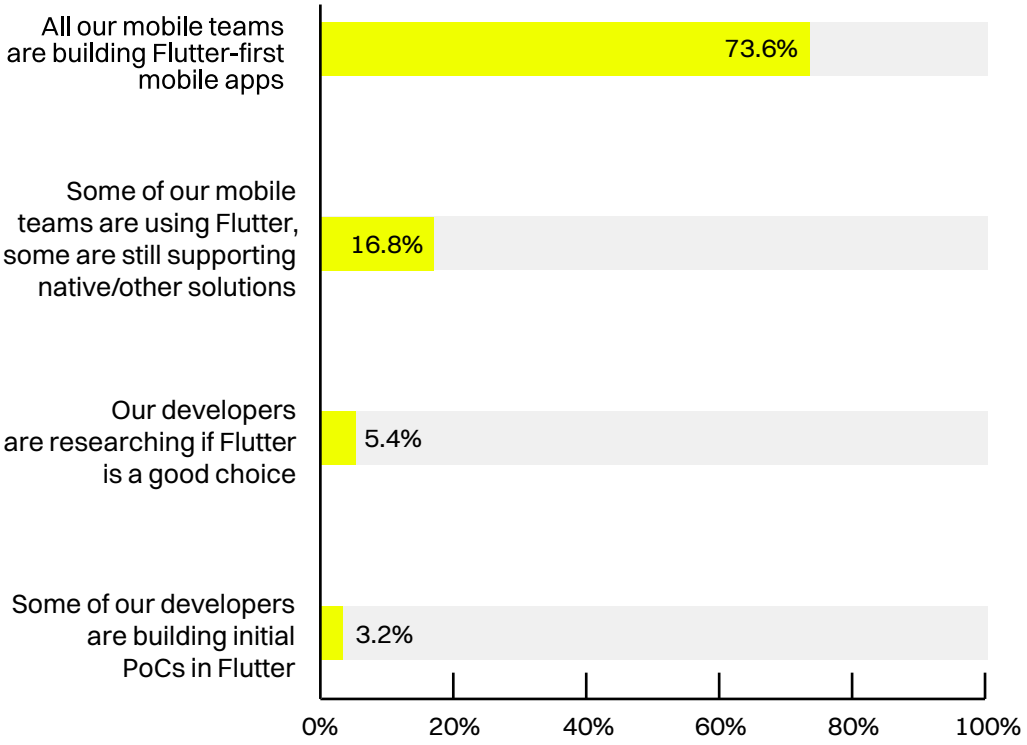
For companies still considering Flutter, the biggest risk is a non-technical one. They are afraid that Google will abandon Flutter. Anticipating those voices, we have invited Craig Labenz for an interview to ask the eternal question: Is Google going to kill off Flutter? He made his opinion very clear, using three powerful arguments. You can find the link to that interview in this section.

For those already using Flutter, it has proven itself to be a truly cross-platform framework.
Only 30% of respondents use Flutter solely for iOS or Android apps, with the 3rd most popular target platform being the web, as indicated by a meaningful 64.4% of them.

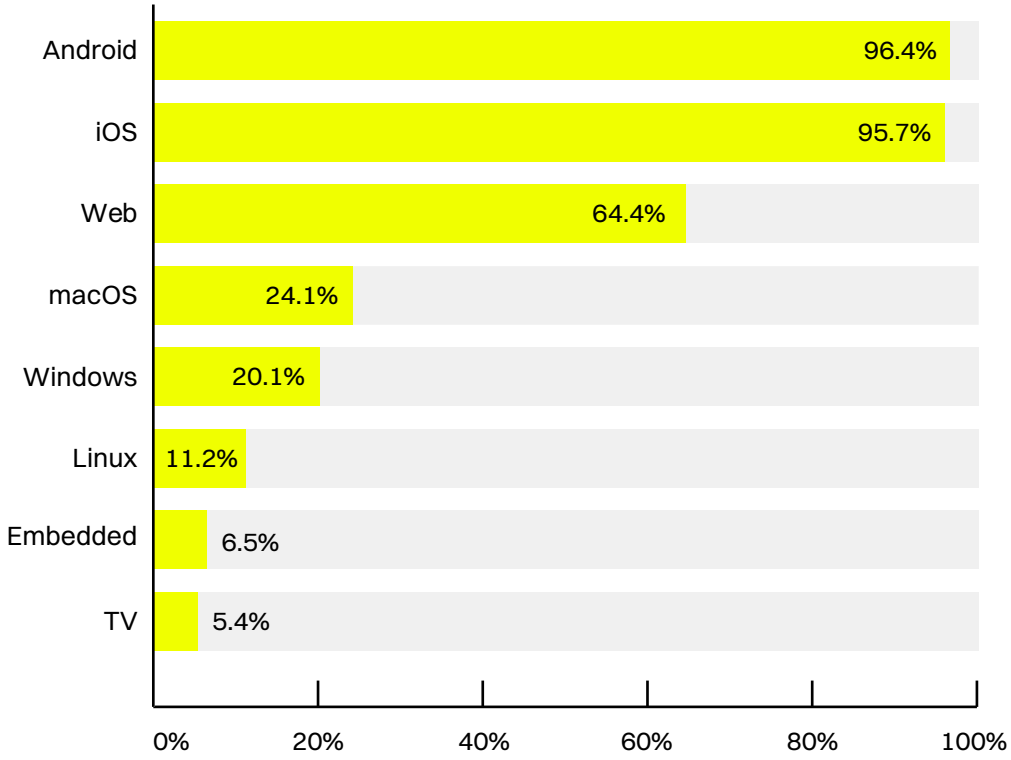
The majority of tech teams are migrating to Flutter for their entire product portfolio.

Flutter has proven to be a truly cross-platform framework, with only 30% of companies using it just for iOS and Android.

How would you describe the level of Flutter adoption in your company?

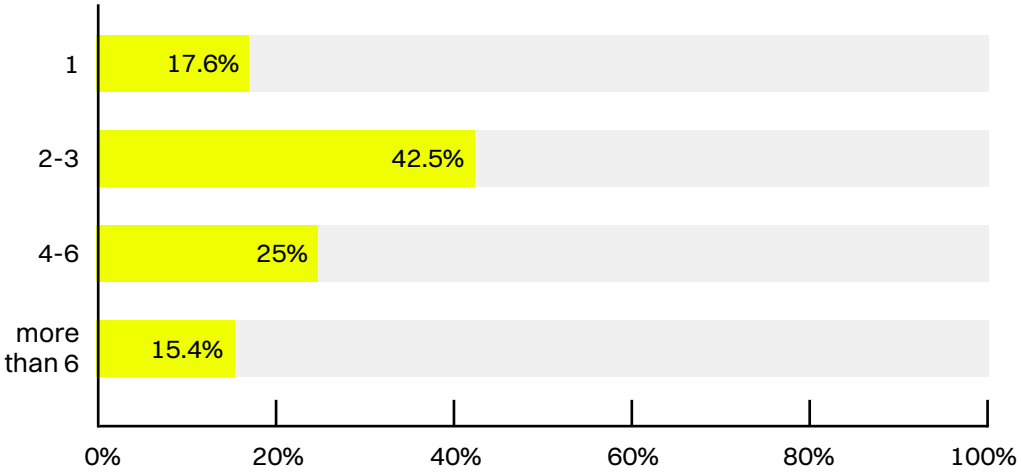


What do you use Flutter for?

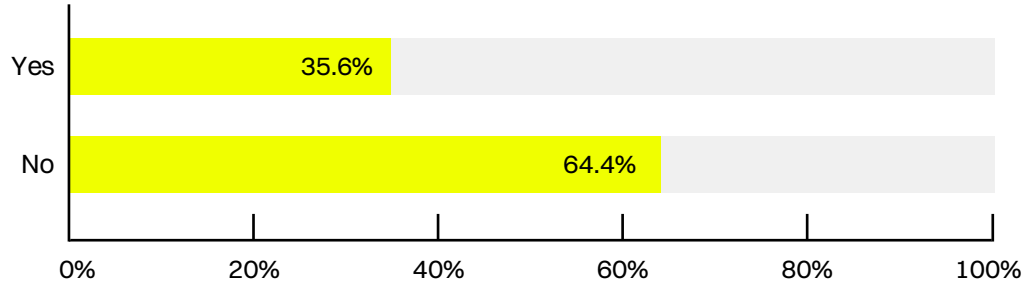


Flutter makes it easy to build multiple applications. More than 80% of respondents build more than one application.

What is the number of Flutter apps that you manage?

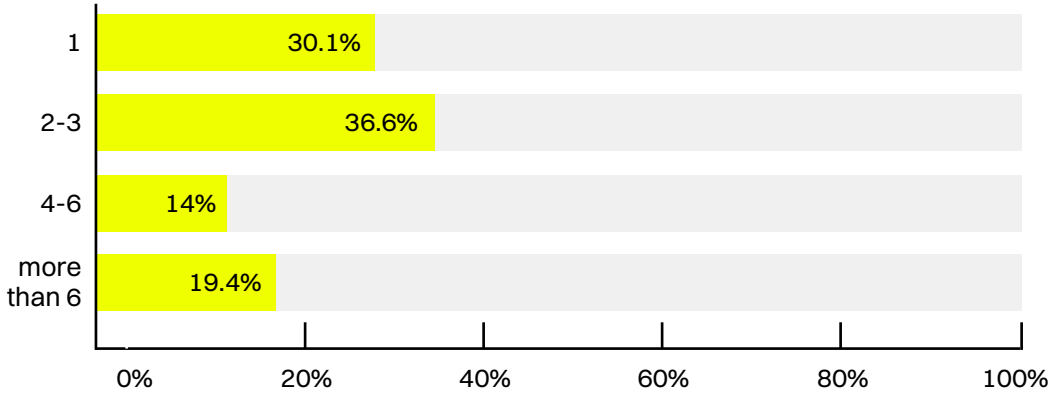


Do you manage any non-Flutter mobile apps?



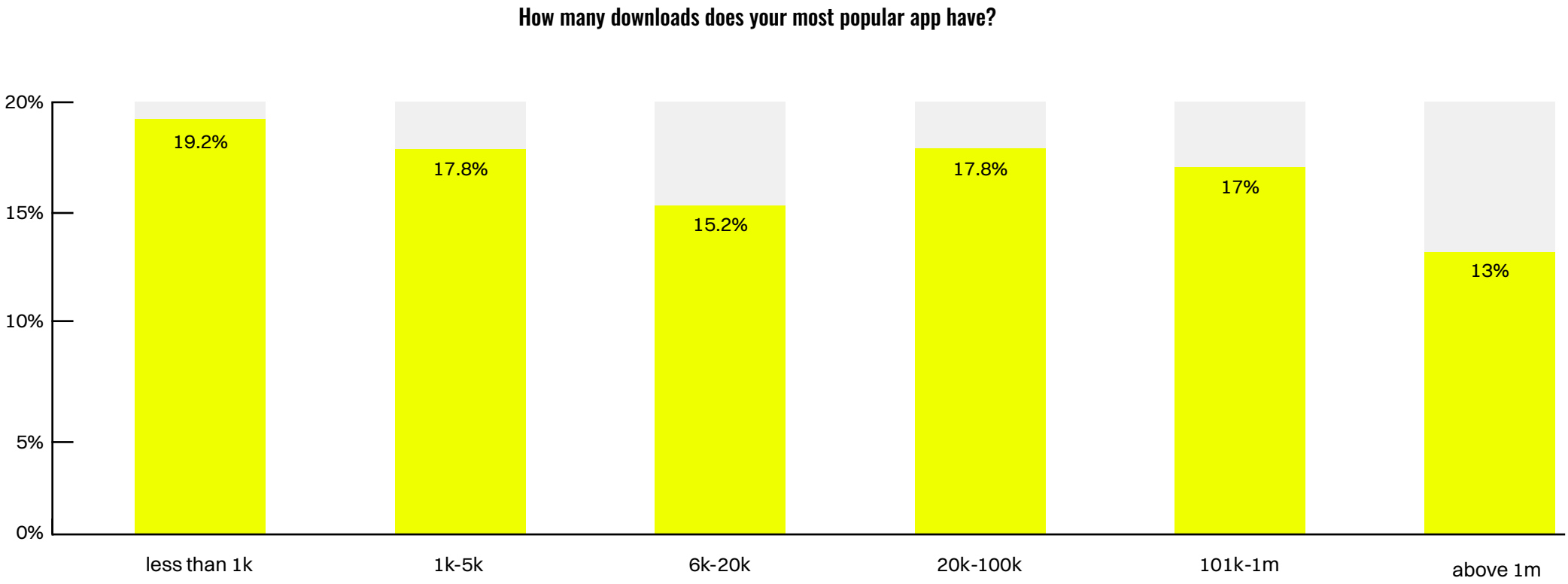
Flutter also enables teams to build new products quickly, as more than 40% of respondents declare they managed 4 or more Flutter apps.

What is the number of non-Flutter mobile apps under your management?



The impact of Flutter apps is growing. 36 responses came from teams serving a user base larger than 1M.

Flutter as a framework is also attractive to teams addressing a large user base. 30% of submissions were from apps with more than 100k users, and 13% were from apps with more than 1 million users.

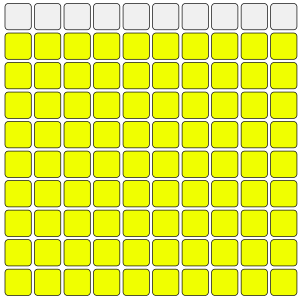


Code reusability and developer experience are the key arguments in favor of using Flutter.

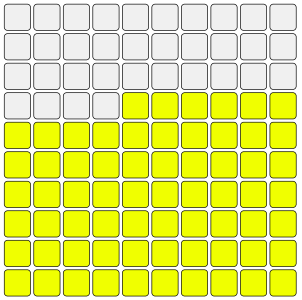
The idea of building multiple applications from one codebase is Flutter's biggest value proposition. This translates into lower development costs and/or shorter delivery times, though respondents mention these side effects of the core value less often.

From a technical perspective, developers find the experience, high performance, and ease of integrating native SDKs to be very attractive value drivers.

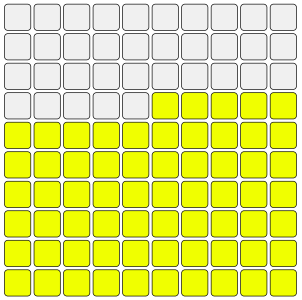
What were the most decisive factors in favor of choosing Flutter?



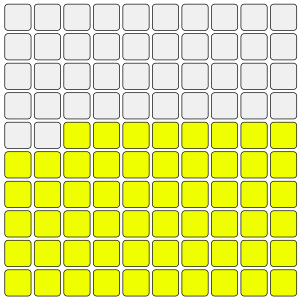
89.7%
Code reusability between iOS and Android



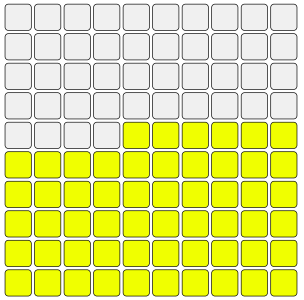
66.2%
Developer experience



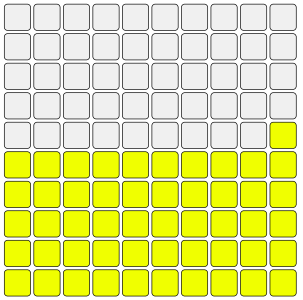
64.8%
Lower cost of development



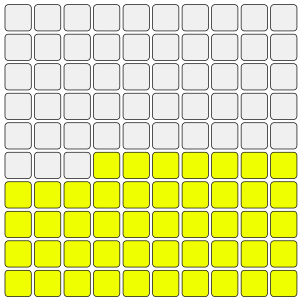
57.7%
Quicker time-to-market



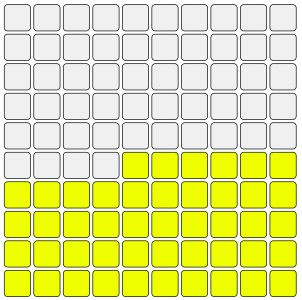
55.5%
Unified UX/UI across platforms



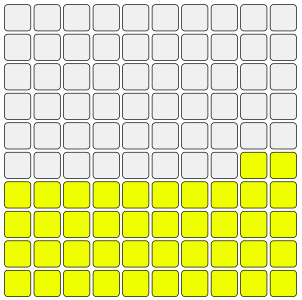
50.5%
High performance



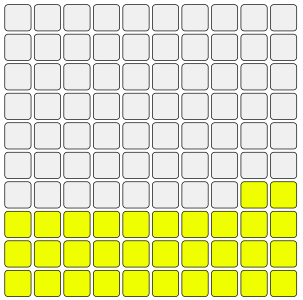
47.7%
Lower cost of maintenance



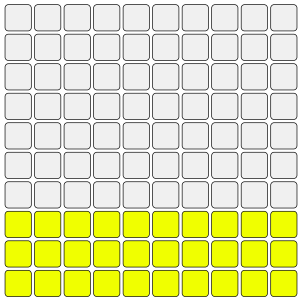
45.6%
Possibility of deploying to the web



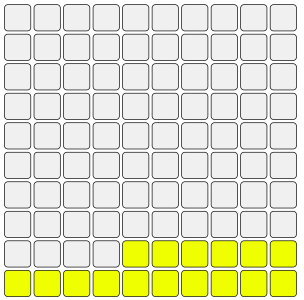
42.3%
Community support



31.7%
Easy integration with native SDKs



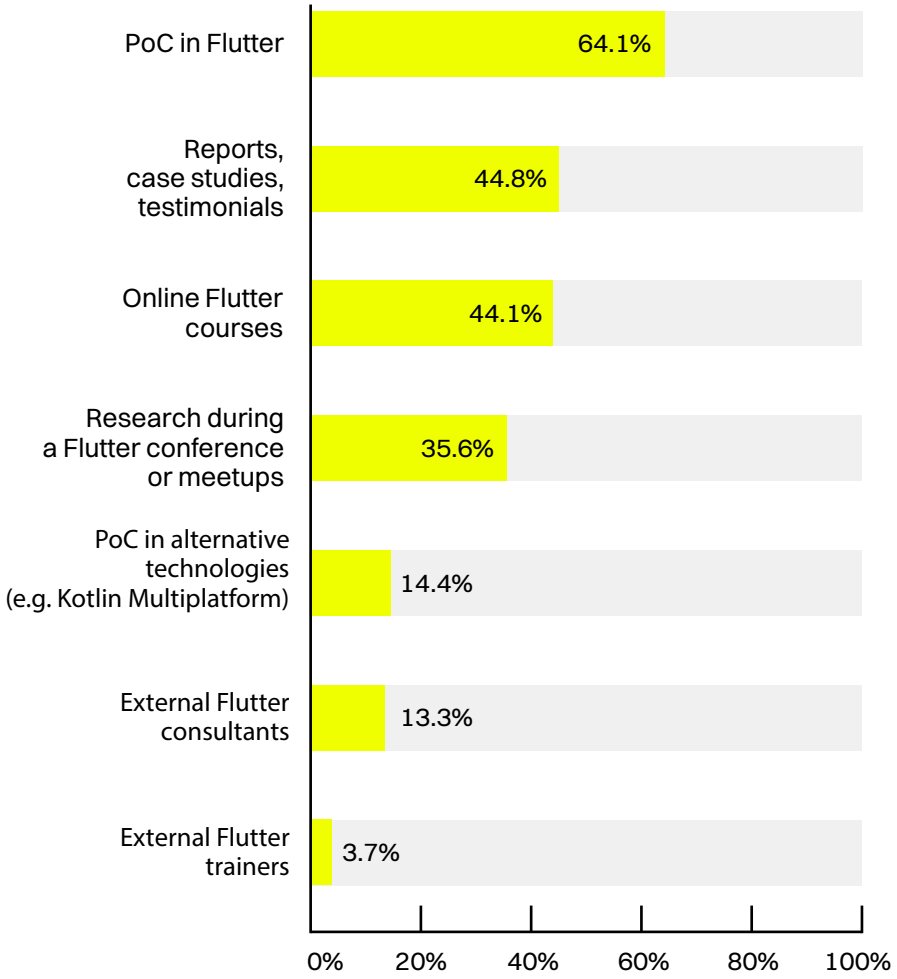
29.9%
Possibility of deploying to the desktop



16%
Growing number of Flutter developers for hire

Almost 65% of teams build their proof-of-concept apps first before starting regular development with Flutter.

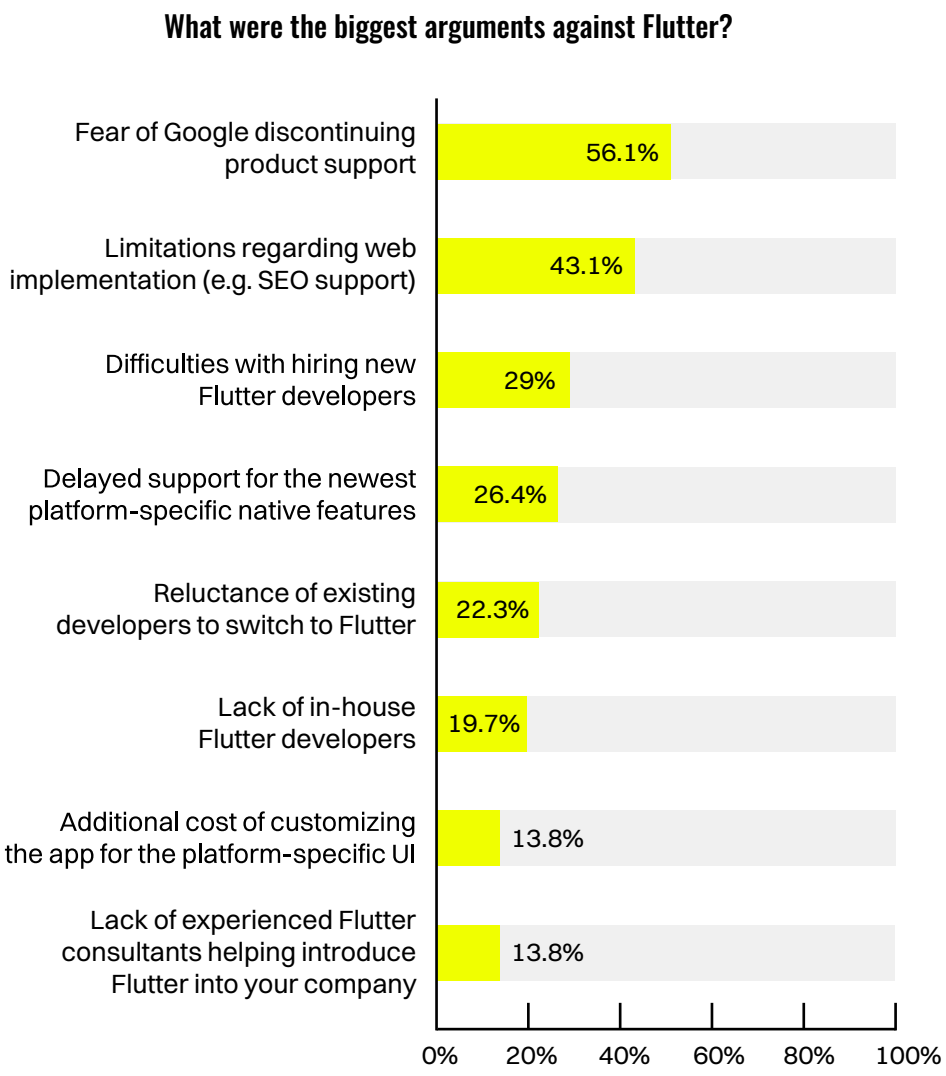
What resources have you used for the decision-making process?



There are various ways in which developers can start using Flutter in their projects. Since choosing the proper framework is an important decision, it is not surprising that companies are willing to invest time and resources in building PoC applications, taking courses, or hiring external consultants.

At LeanCode, we call this service Staff Augmentation 2.0. We provide the opportunity to form a hybrid team with in-house developers and senior Flutter consultants from LeanCode. This hands-on experience encourages knowledge transfer and helps our clients build their Flutter capabilities in the shortest possible time.

The fear of Google discontinuing support was mentioned by 56% of respondents.



This eternal question is well addressed by Craig Labenz in the interview accompanying this report, "The Future of Flutter". The second argument, about SEO support, was addressed by Kevin Moore in the interview, "Is Flutter for Web Production Ready?". We strongly encourage you to listen to their opinions on both subjects.

While the first argument is fairly easy to dismiss, the second one is technical. No announcements have been made at the time of this report's release indicating significant improvements in this area. Thirdly, 29% of respondents mentioned having difficulties hiring Flutter developers. The challenge of hiring good developers appears to be valid regardless of the technology discussed.

INTERVIEW

THE FUTURE OF FLUTTER

Craig Labenz

Developer Relations Engineer at Google

Łukasz Kosman

CEO at LeanCode

Find the answers to questions like:

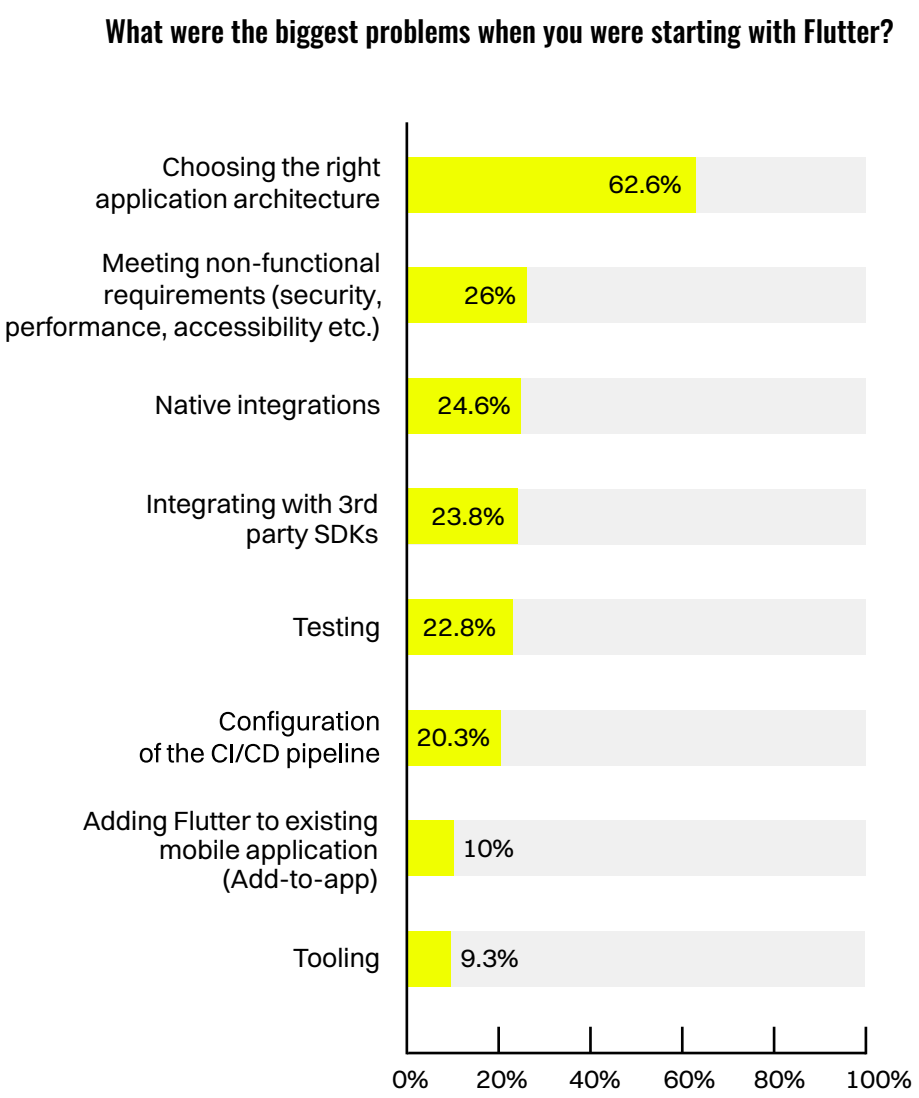
- ☑ Would you recommend Flutter to your best friend who is starting a business?
- ☑ Is Flutter Enterprise ready?
- ☑ What are the objections against Flutter you hear the most often?
- ☑ Will Google abandon Flutter?
- ☑ What is now the main advantage of Flutter compared to other competitive cross-platform frameworks?

Take a look inside.

"Will Google kill Flutter is a timeless question. I don't think it will ever go away. What does Flutter mean to Google? (...) By a solid margin, Flutter saves Google more headcount than even the teams of Dart and Flutter. And they aren't small. Flutter doesn't cost Google money to build because of Google's own internal usage of Flutter. Beyond that, we know that Developers who use Flutter tend to route more money to Google than Developers who don't".

View on YouTube

Choosing the right Flutter Architecture is by far the biggest problem for the teams starting with Flutter.



Since Flutter is still a very young framework regarding the technology lifecycle, it lacks default, well-established solutions in some areas. Take state management, for example. With options like Bloc, Riverpod, Provider, and others, it is easy to get confused and paralyzed by analysis.

Other obstacles mentioned in the survey are typical for researching new technologies, including questions about non-functional requirements (26%), native integrations (24.6%), and integrating with third-party SDKs (23.8%). These are areas where Flutter excels and do not appear on the list of challenges Flutter faces today.

WEBINAR

HOW TO CHOOSE THE PROPER ARCHITECTURE FOR A FLUTTER APP?

HOSTED BY:

Mateusz Wojtczak

Head of Mobile at LeanCode, GDE for Dart and Flutter

* "Flutter CTO Report 2024 by LeanCode"

Choosing the right application architecture is a problem for 62.6% of respondents*.

In our comprehensive webinar, we will give an overview of an architecture for large-scale Flutter application:

- ☑ **Flutter Architecture Overview** (project structure, application logic & state management, navigation, dependency injection, multi-module architecture).
- ☑ **Efficient communication with the backend.**
- ☑ **Cache & offline mode.**
- ☑ **Testing** (unit & widget testing, golden tests, E2E UI testing with Patrol).
- ☑ **Bonus topics** (flavoring, localization & internationalization, design system, etc.).

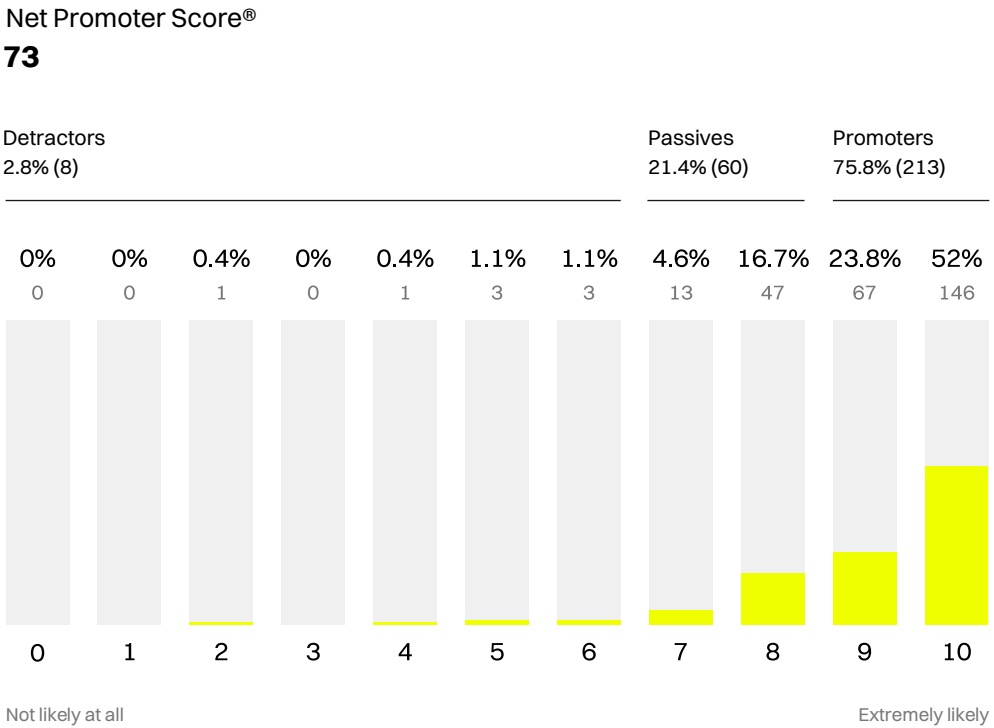
View and sign up

POWERED BY LeanCode

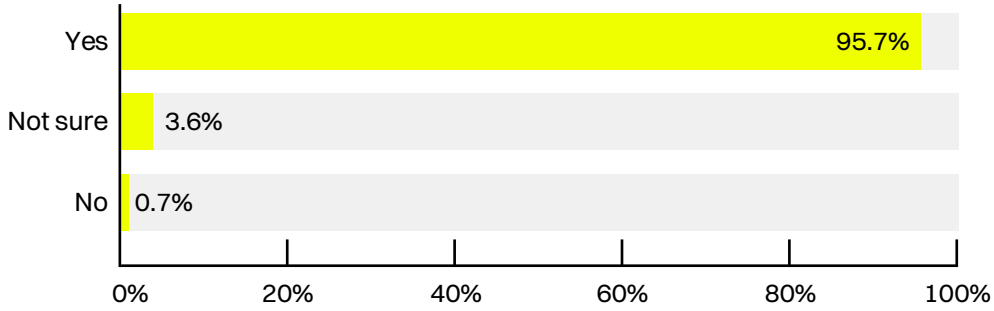
Feedback after using Flutter

95.7% of respondents would choose Flutter again, and 52% are extremely likely (10/10) to recommend it.

On a scale from 0-10 would you recommend using Flutter?



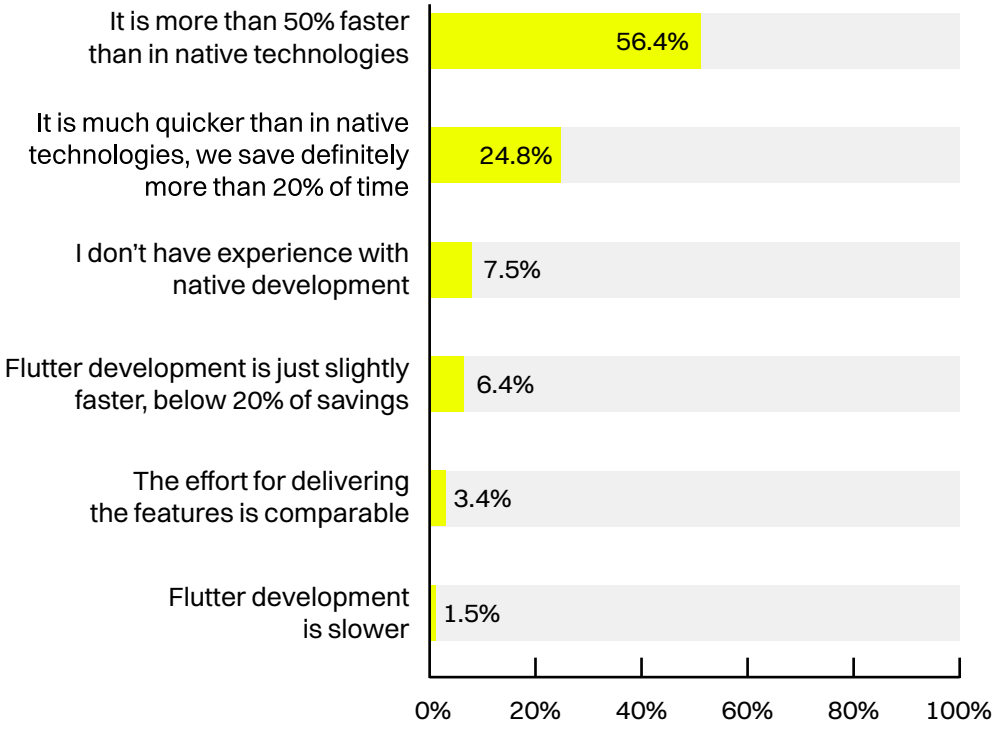
Looking back, would you choose Flutter again?



CTOs, Tech Leads, and developers love Flutter. More than 50% of them are willing to recommend it to their peers. Is it perfect? Certainly not. An NPS of 73 points indicates there are still some points of friction to address. However, if asked whether they would make the decision to use Flutter once again, a staggering 95.7% of CTOs and Tech Leads would say yes, clearly showing that Flutter is the best alternative for them.

According to 56.4% of respondents, Flutter is more than 50% quicker compared to native development.

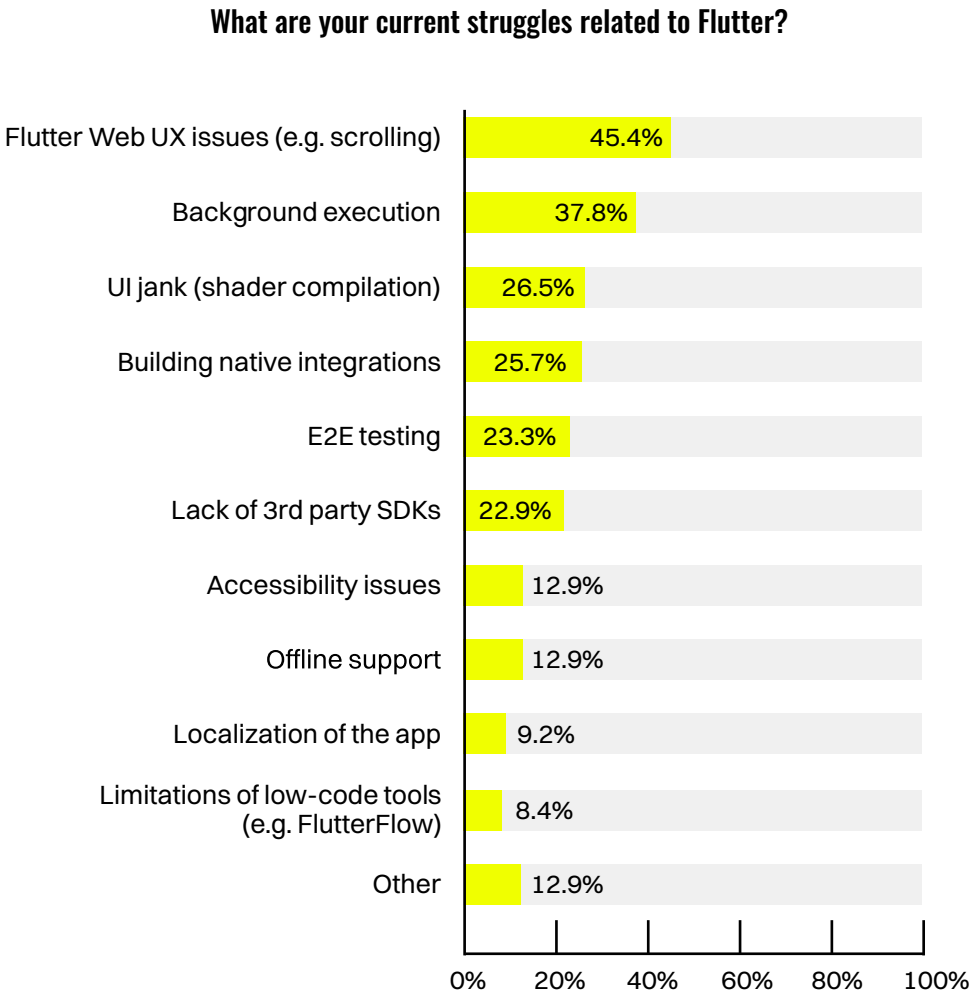
Based on your experience, is Flutter development quicker in comparison to native technologies?



The true power of Flutter comes from the fact that it allows users to build applications for different platforms from one codebase, which improves the performance of the mobile development team. Yet, this value was hard to quantify. In our survey, we attempted to measure the improvement in efficiency based on the high seniority and experience of our respondents.

The majority declared Flutter to be more than 50% quicker than native technologies, and more than 80% said it is at least 20% quicker. These are savings that should not be ignored. This means that your team can finish a year-long backlog within 8 to 10 months, giving you plenty of room for new features.

The Flutter ecosystem still has some points of friction, with web-related user experience being the most frequently mentioned issue.

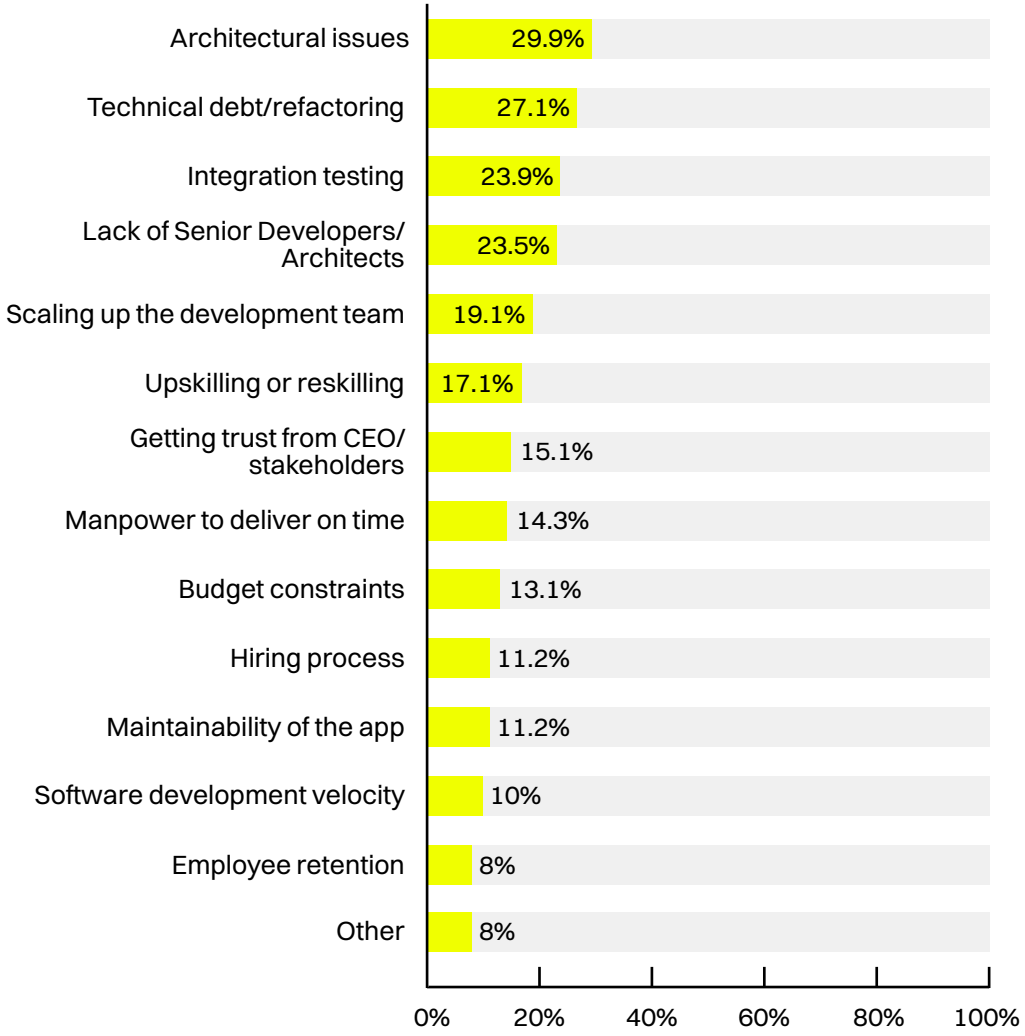


At the top of the list of concerns for teams currently using Flutter are features related to web UX issues, such as scrolling. This concern aligns with the high popularity of the web as a target platform for Flutter-based apps, as shown in the earlier parts of this report.

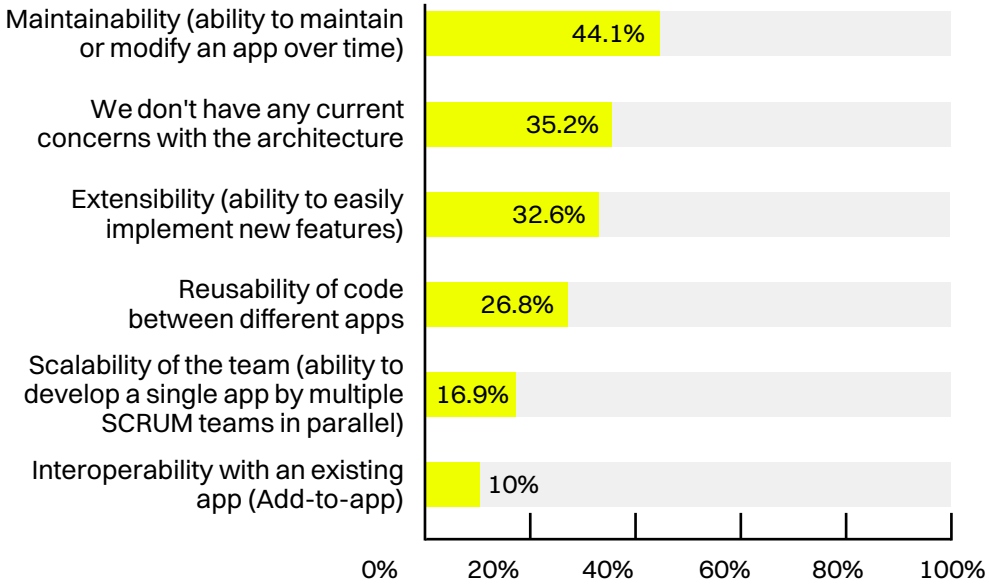
The good news is that these impediments have been addressed by the Flutter team, with the recent announcement that Flutter for Web will be stable for WASM. This is a major improvement that will vastly improve the performance of Flutter for Web applications in browsers supporting WASM.

The biggest challenges are related to architectural issues and initial decisions, which have resulted in technical debt.

What are your biggest challenges at the moment?



What are your biggest concerns when setting up the Flutter architecture?



Interestingly, 26.8% of teams would like to build a reusable architecture to implement their solutions in different applications. It is also notable that 23.9% of respondents find integration testing to be a persistent problem that needs to be addressed. This is the issue we aim to solve with one of our open-source packages called **Patrol, a framework for integrated end-to-end (E2E) UI testing.**

WEBINAR

HOW TO AUTOMATE E2E TESTING IN FLUTTER?

HOSTED BY:

Mateusz Wojtczak

Head of Mobile at LeanCode, GDE for Dart and Flutter

For 23.3% of respondents, E2E Testing for Flutter is challenging.

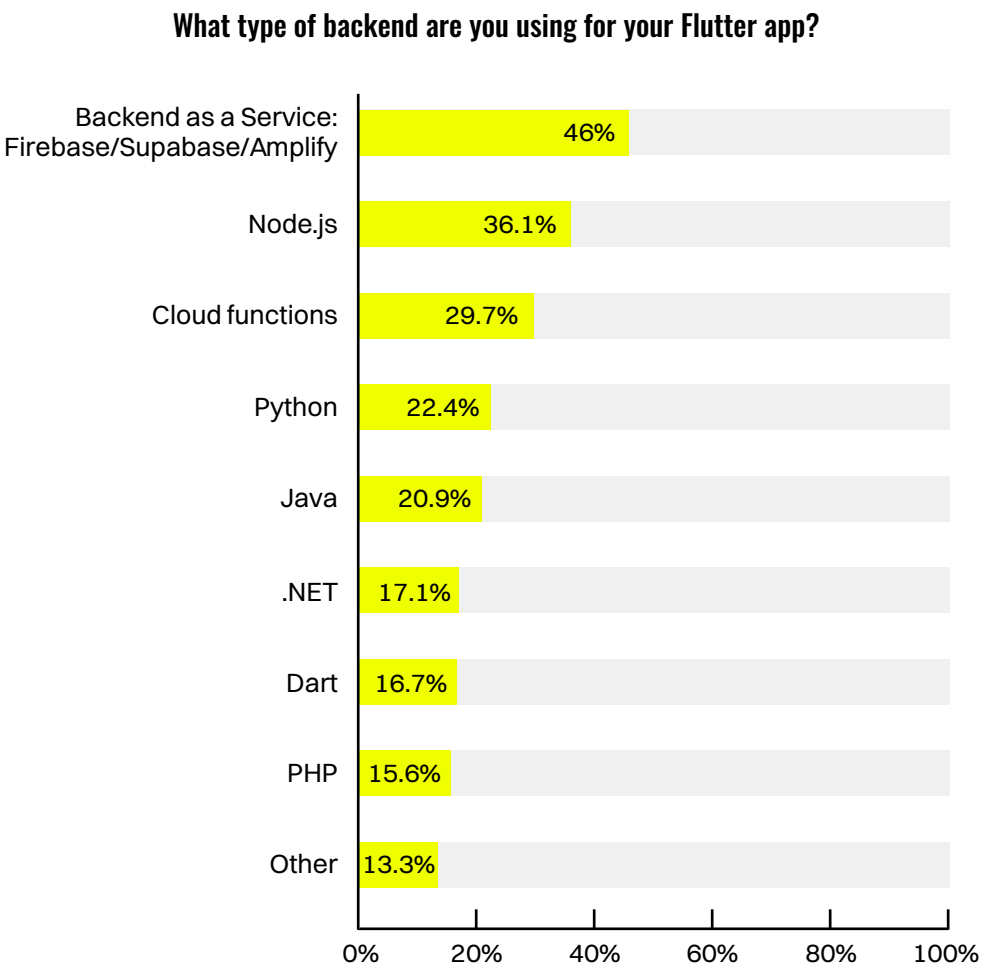
We will cover the following topics:

- ☑ **Understand E2E testing** (why we might need it, what should be tested, and how it can help deliver better products).
- ☑ **Meet Patrol** - a Flutter-first E2E testing framework.
- ☑ **Set up your E2E testing environment properly:** integration with your CI/CD pipeline, device farms, test management & reporting tools.
- ☑ **Best practices and common mistakes of E2E testing.**

View and sign up

POWERED BY LeanCode

Flutter apps use different backend solutions, with Backend as a Service being the most popular approach.



Flutter originated as a UI development framework and has evolved into a truly powerful cross-platform development solution. It focuses on sharing the frontend business logic and the UI layer between different platforms.

Interestingly, development teams make varied choices regarding the backend for their systems. Our report sheds light on the technologies employed to handle the domain logic. It appears that the most popular solutions are known as Backend as a Service (BaaS), with Firebase being the obvious choice for Flutter.

This is fully understandable, as it is a way for Google to monetize their contribution to Flutter development. Additionally, it bodes well to see the high adoption of backend technologies typically associated with large-scale applications, such as Java (20.9%) and .NET (17.1%).

Dart's backend revolution: simplifying server-side development.

Serverpod

Viktor Lidholt
Founder of Serverpod

Founder of Serverpod, formerly worked at the Flutter team at Google. MSc Computer Science. 20+ years of industry experience as a software engineer.

Dart, originally created by Google for building complex, high-performance web applications, has increasingly found its niche primarily in mobile development. Known for its simplicity, strong typing, and native performance, Dart is beloved by programmers for its speed and ease of use. Its seamless integration with Flutter has revolutionized mobile app development, but Dart's capabilities extend far beyond the frontend.

Dart on the server.

Dart's entrance into the backend development brings a host of benefits. Its strong typing and ahead-of-time compilation leads to fast execution and low latency, which are essential for high-performance backend applications. Dart's concurrency model, based on asynchronous programming, makes it highly efficient at handling multiple simultaneous connections, a key requirement for modern server applications.

Moreover, using Dart for both frontend and backend development offers a unified language experience. This reduces the cognitive load on developers, who no longer need to switch contexts between different programming languages. It also streamlines the development process, allowing for a more consistent and maintainable codebase. In addition, developers can share code between frontend and backend.

Introducing Serverpod.

Serverpod is a standout framework designed specifically to leverage Dart's strengths for backend development. As an open-source, scalable app server, Serverpod provides a seamless bridge for Flutter developers to extend their applications into the backend. Its design focuses on making backend development as straightforward and efficient as possible, ensuring that developers can quickly become productive without a steep learning curve.

Serverpod offers a rich set of features that cater to the needs of modern backend development. It simplifies the creation of server endpoints and automates the generation of protocol and client-side code. This automation minimizes the potential for errors and allows developers to focus on writing business logic instead of boilerplate code.

The forthcoming release of Serverpod Cloud this fall is particularly noteworthy. This new service will simplify the deployment and management of Dart servers, allowing developers to deploy Serverpods with minimal effort and infrastructure management.

The competitive landscape.

In the Dart backend ecosystem, Serverpod faces competition from frameworks like Dart Frog and newcomers such as Celest. Dart Frog is appreciated for its simplicity and quick setup, making it suitable for smaller projects or those needing a lightweight solution.
Celest, although new, is gaining traction with its innovative features and active development community.

Serverpod

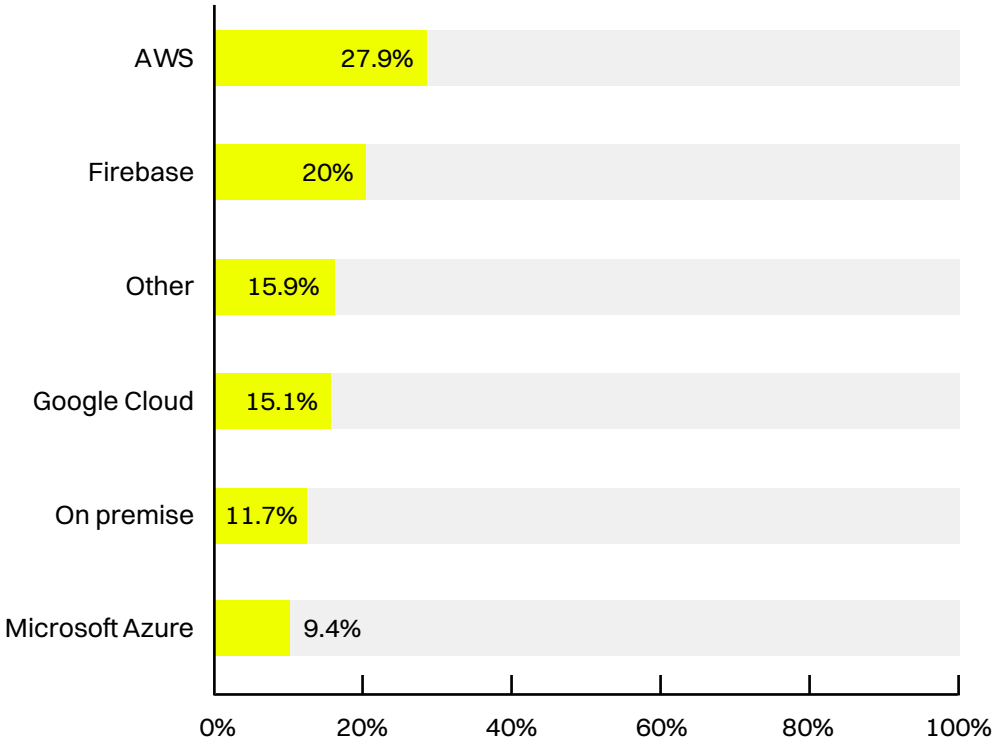
Serverpod is an open-source, scalable app server written in Dart for the Flutter community. It features automatic generation of protocol and client-side code, making remote endpoint calls as easy as local method calls. Additionally, Serverpod includes a revolutionary ORM that uses native Dart types.

Conclusion.

Dart's potential on the backend is being realized through frameworks like Serverpod. By providing a powerful, easy-to-use backend solution tailored for Dart and Flutter developers, Serverpod makes it possible to harness the full capabilities of Dart across the entire development stack.

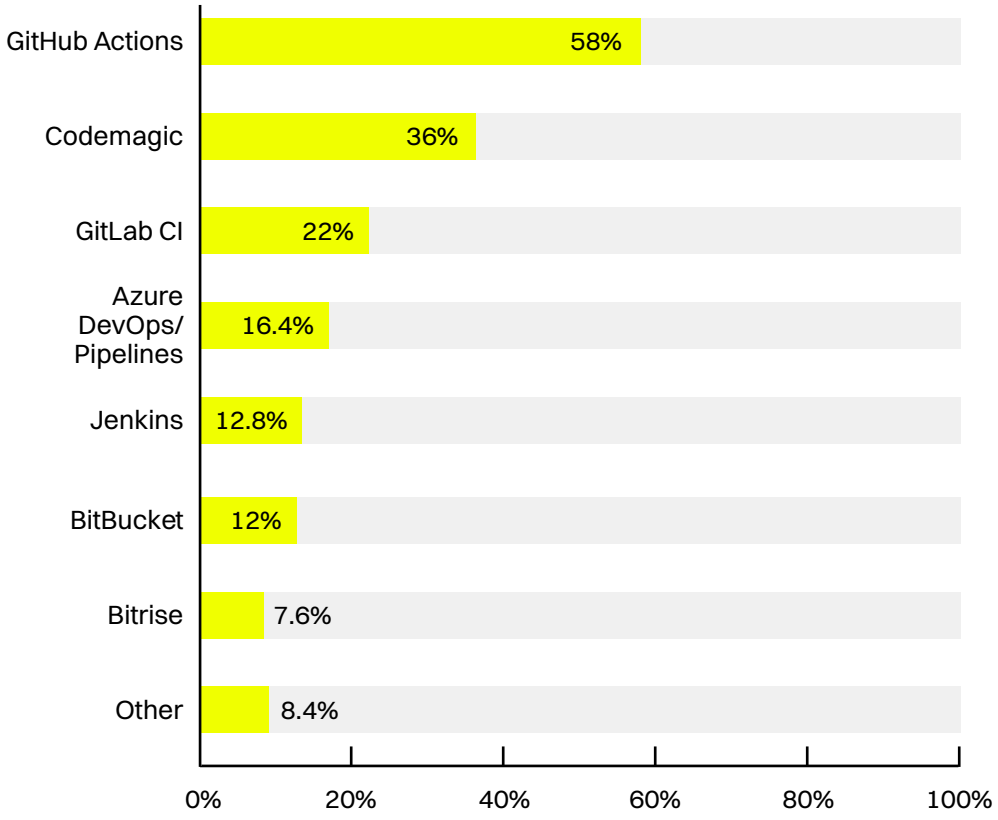
While AWS is the most popular cloud provider, Firebase and GCP also score highly.

Where do you deploy your backend application most often?



GitHub actions are the most popular solution for CI/CD, closely followed by Codemagic.

Which of these CI/CD solutions have you used over the last year?



Embracing Flutter with Codemagic CI/CD.

codemagic

Kevin Suhajda
Senior Solution Engineer
at Codemagic

A Senior Solutions Engineer at Codemagic with over a decade of experience working in the mobile applications space. He helps mobile development teams set up efficient CI/CD pipelines to build, test, and publish their apps to stores.

In the rapidly evolving landscape of mobile app development, choosing the right tools can significantly impact your team's productivity, efficiency, and code quality. For development teams leveraging Flutter, Codemagic offers a premier CI/CD (Continuous Integration/ Continuous Deployment) solution designed to streamline workflows, accelerate build processes, and ensure high-quality app releases.

Continuous integration.

Every time a developer checks in new code, Codemagic starts a new build to ensure the application compiles correctly and no new bugs or issues are introduced. This immediate feedback loop is crucial in maintaining the integrity and quality of your application, preventing potential problems from reaching production. Additionally, Codemagic allows for the integration of static code analysis, unit tests, and comprehensive integration and end-to-end tests into your CI workflows, providing thorough quality coverage and further enhancing app reliability.

Continuous deployment.

Codemagic simplifies the code signing and deployment process, allowing you to easily distribute your app to developers, testers, and stakeholders or directly to the App Store, Google Play, or other app stores. Your app builds can also be distributed using shareable build dashboards, and build notifications can be sent via email or Slack. This flexibility ensures that your app reaches the right hands promptly, facilitating faster feedback and more efficient iterations.

Why Flutter development teams choose Codemagic.

Codemagic has established itself as the go-to CI/CD tool for Flutter development teams, and several factors contribute to its popularity and effectiveness. Launched in collaboration with Google during the official release of Flutter 1.0 in 2018, Codemagic has deep roots in the Flutter ecosystem. This long-standing relationship with Flutter ensures that Codemagic stays ahead in supporting the latest Flutter updates and features. Today, thousands of development teams rely on Codemagic to build, test, and publish their Flutter apps.

Fast build infrastructure.

Codemagic offers a well-maintained and fast-build infrastructure, utilizing cutting-edge hardware such as Apple Silicon M2 machines to reduce build times. If you plan to build for other platforms, Windows and Linux machines are also available.

Timely software updates.

Staying current with the latest development tools is crucial. Codemagic ensures that Xcode versions are made available within days of their release, including beta versions, keeping your development environment up-to-date and compatible with the latest advancements.

Extensive integration capabilities.

Codemagic integrates seamlessly with numerous third-party services, enhancing its utility and flexibility. From real device testing and code testing services to project management solutions and code obfuscation tools, Codemagic provides comprehensive support to cover all aspects of your development workflow.

Support and community.

Codemagic understands the importance of support and community and offers multiple support channels, including a Discord community, in-app chat support, GitHub Discussions, extensive documentation, and a repository of GitHub samples. These resources ensure that you have access to the assistance and knowledge you need to overcome any development challenges.

Transparent and flexible pricing.

Codemagic's pricing model is designed to be both transparent and flexible, catering to teams of all sizes and requirements. Their pay-as-you-go (PAYG) plan allows you to pay for the minutes you use, while their fixed plan offers unlimited build minutes with a specified number of parallel builds, which can be increased as your needs grow. Additional benefits include unlimited apps, unlimited team seats, standard support, and specialized plans for Enterprise teams, which feature SSO login, service level agreements, security compliance, signed NDAs, and priority support.

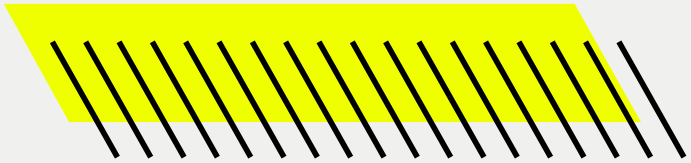
Conclusion.

For CTOs and senior engineers considering Flutter as their development framework, Codemagic offers an unparalleled CI/CD solution that enhances productivity, ensures high code quality, and simplifies deployment processes.
With its expert support, fast-build infrastructure, regular updates, extensive integrations, and flexible pricing, Codemagic stands out as a great choice for Flutter development teams. Embrace Codemagic to unlock the full potential of your Flutter projects and drive your development efforts to new heights.

See Codemagic Documentation

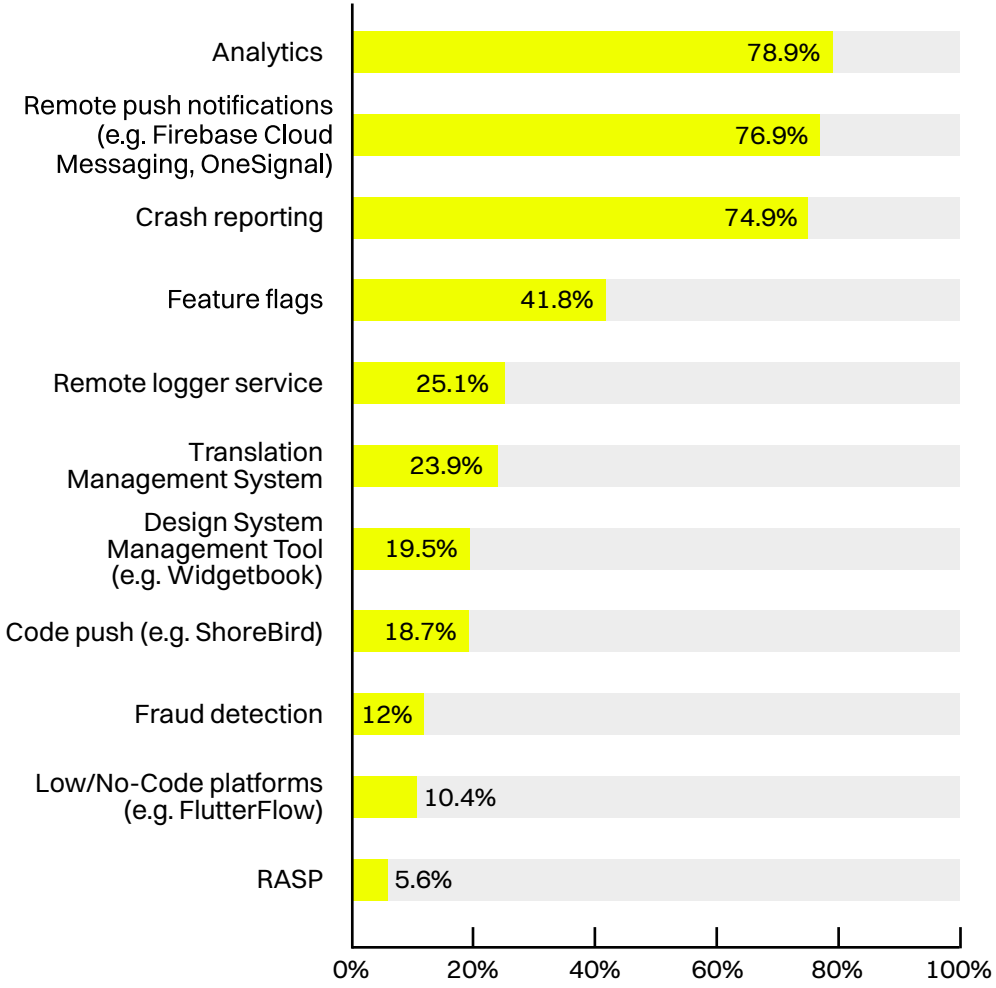
codemagic

Codemagic CI/CD is for mobile dev teams building Flutter apps. Build, test, and deploy your apps using fast Apple M2 machines. Integrate with third-party tools to monitor code quality, run tests, or update your project system. Pay for minutes used or get unlimited build minutes with an annual plan.

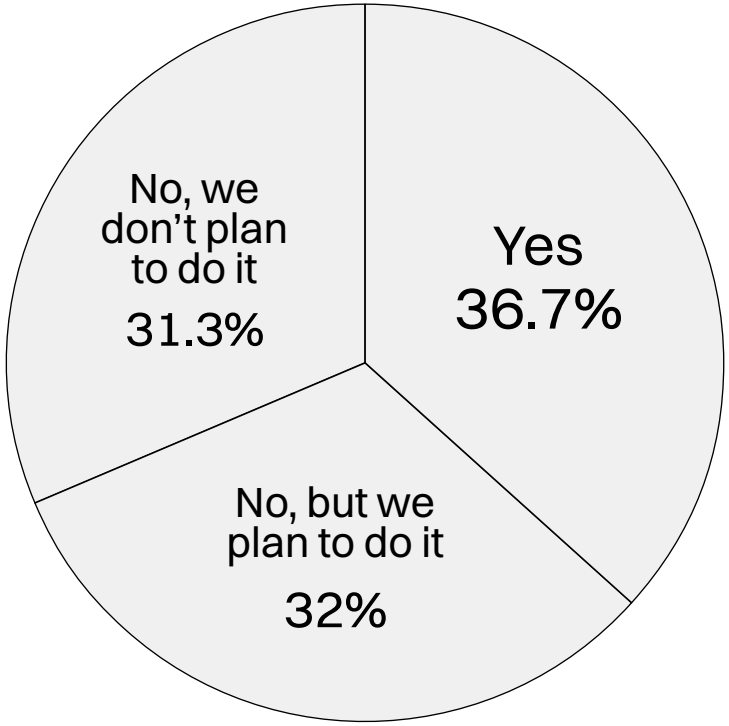


Two of the most integrated services are related to analytics and push notifications, with 78.9% and 76.9% of votes, respectively.

Over the past year, which of the following development tools or services have you used?



Have you built your own Design System in Flutter (e.g. using Widgetbook)?



Case study: LeanCode's enhanced UI development with Widgetbook.

Lucas Josefiak
Co-founder and CEO
at Widgetbook

About LeanCode.

LeanCode, a Warsaw-based software house, specializes in mobile and web application development with a strong focus on Flutter.

The challenge.

LeanCode aimed to set up a robust design system and component library. This was crucial for maintaining consistency and efficiency in their development process. Furthermore, they needed an effective way to share these components with a broader audience, including quality assurance teams and designers, for feedback. Initially, they managed this by deploying Widgetbook themselves, but as the number of branches increased, this approach became cumbersome. LeanCode required a more streamlined solution to make their UI components easily accessible to everyone involved.

The solution.

LeanCode adopted Widgetbook to address their challenges. Widgetbook's open-source package enables developers to catalog widgets in a central library and test them across various devices and themes. Widgetbook Cloud extends this by providing seamless hosting and management of multiple Widgetbook instances, greatly simplifying the workflow. Its automatic detection of visual bugs in pull requests allows LeanCode to swiftly review all UI changes in their Flutter apps. Additionally, Widgetbook Cloud supports LeanCode to keep its Figma files synchronized with the codebase. This integration significantly enhanced LeanCode's quality assurance process by making component previews easily accessible to all team members.

New workflow.

With Widgetbook Cloud, LeanCode introduced a more efficient workflow:

- **Integration:** Each new project includes a Widgetbook setup with a dedicated instance of their design system.
- **Live Documentation:** Clients now have access to live documentation of all design system components without additional maintenance or cost.
- **Quality Assurance:** Strict client requirements for quality and component testing are met through Widgetbook Cloud. LeanCode deploys to the Cloud on every design system change, requiring a designer's check as part of their pull request approval policy.

Key features utilized.

LeanCode leverages several Widgetbook features:

- **Environment Testing:** Addons for checking components against different locales, themes, and accessibility settings.
- **Branch Deployment:** Deployment of main and feature branches, all accessible simultaneously in the Cloud.
- **Review Feature:** Enhanced cooperation with QA participants through the review feature.
- **Golden Test Approach:** Ongoing evolution of the review process with a focus on golden test-like methods.

Value derived.

Widgetbook Cloud has helped to streamline LeanCode's UI development and review processes. The platform lets LeanCode's developers focus on their tasks and quickly identify shortcomings. The integration provides easy access for all stakeholders, enhancing collaboration and efficiency.

LeanCode's senior staff quantified their time savings of several weeks of development time.

At Widgetbook, we are grateful for having such a positive impact on LeanCode's exceptional mobile development and for this excellent cooperation.

Review all UI changes of your Flutter app with Widgetbook. Our open-source package allows you to catalog widgets in one design system and test them quickly in all use cases. Widgetbook Cloud automatically catches visual bugs in your pull request and syncs your Flutter widgets with Figma.

WEBINAR

HOW TO BUILD THE DESIGN SYSTEM FOR A SCALABLE FLUTTER APP?

HOSTED BY:

Albert Wolszon

Head of UI
Flutter Guild
at LeanCode

68.7% of respondents have implemented the Design System or plan to do so*.

In our comprehensive webinar, we will cover:

- ☑ **Design System Development** (component library, tokens, colors, typography, material library, responsiveness, RTL approach, accessibility, motion design, golden tests and Widgetbook).
- ☑ **Cooperation with the Design team** (rules of working with designers, Figma as a source of truth, proper setup of the Figma components).
- ☑ **Cooperation with the Development team** (division of responsibilities, organization of work, enforcing DS rules (lints), versioning).

View and sign up

POWERED BY LeanCode

* "Flutter CTO Report 2024 by LeanCode"

INTERVIEW

WHAT ARE THE FRICTIONS IN THE FLUTTER UNIVERSE?

Eric Seidel

Founder and CEO at Shorebird

Łukasz Kosman

CEO at LeanCode

Find the answers to questions like:

- ☑ What are the current frictions in the Flutter Universe?
- ☑ What does Flutter need to get to suit businesses better?
- ☑ What are the objections against Flutter you hear the most often?
- ☑ What frictions should the Flutter team solve themselves and where are the opportunities for the partners in the ecosystem to provide external solutions?
- ☑ What are the current differences that make Flutter a really cutting-edge technology?
- ☑ What do you think the future of Flutter is? What are the hopes and expectations for this technology?
- ☑ How can Shorebird be helpful for clients whose mobile application is the primary channel for offering their value?

Take a look inside.

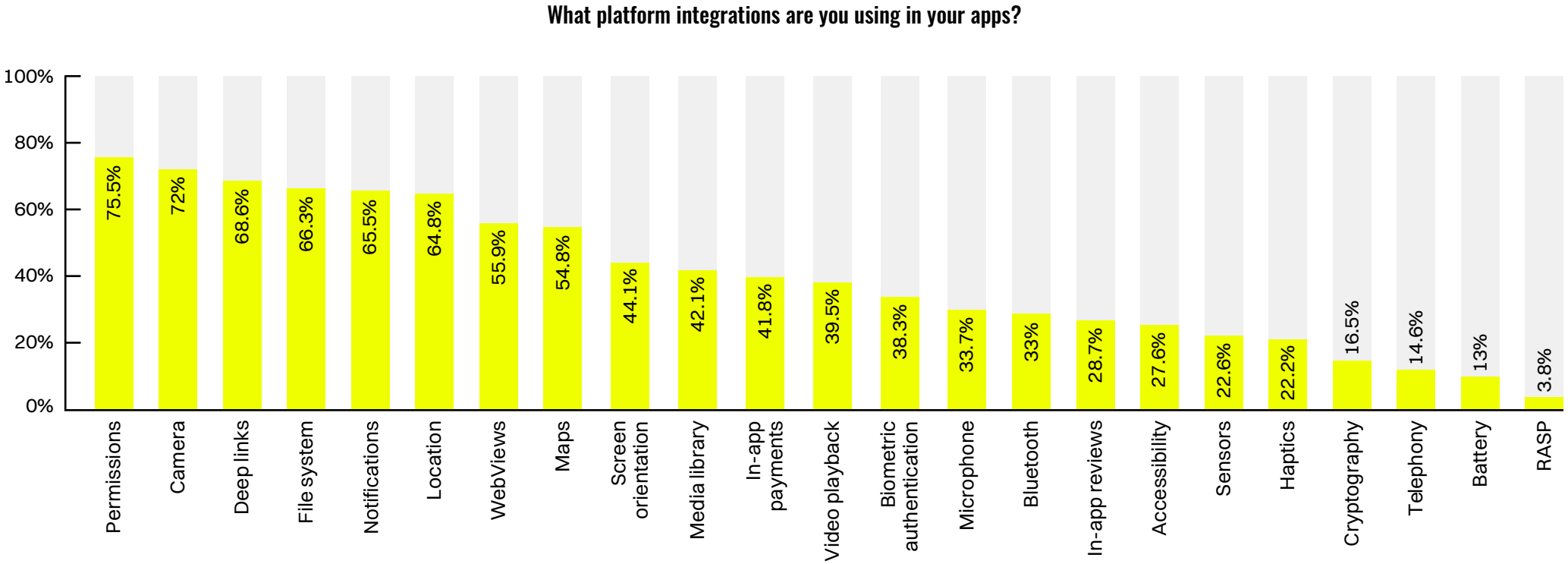
"One of the things that has gone on in the ten years since we started Flutter is that Flutter has grown inside the Google ecosystem to solve Google problems. And that's not bad. (...) but it means that there are a lot of sort of little missing pieces that make it difficult for random businesses to be successful in adopting Flutter. And that is somewhat of the impetus for starting Shorebird".

View on YouTube

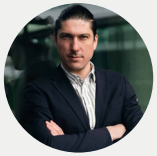
Applications built with Flutter are rich in terms of platform-specific integrations, with permissions, camera, and deep links being among the most notable choices.

Flutter applications are known for their rich integration of powerful mobile features. More than 65% of apps built with Flutter utilize features such as permissions, camera, deep links, notifications, and location services.

This clearly shows how easy it is to integrate components in a Flutter app. It also highlights which components are most crucial from a business perspective when deciding to build a mobile application.



Flutter App Security Trends.



Sergiy Yakymchuk
CEO at Talsec

Flutter has gained significant traction within FinTech, underscoring the crucial need for robust security measures. The platform's popularity attracts attention from both app developers and cybercriminals. Flutter's strong security posture, evidenced by fewer reported vulnerabilities and CVEs, makes it a solid choice for developing sensitive apps.

Flutter Built-in Security.

Flutter is more resilient to decompilation than native apps. Its binary packaging offers better protection of code and hardcoded data, although the number of Flutter-specific reverse engineering tools is increasing rapidly, continually broadening the potential threat landscape (e.g., reFlutter, flutter-spy, blutter).

Common Vulnerabilities.

Despite its advantages, Flutter apps are not immune to common vulnerabilities:

- **Privileged Access Issues:** Rooting and jailbreak concerns remain prevalent.
- **Dynamic Attacks:** Techniques such as hooking frameworks (e.g., Frida, Xposed) pose significant risks.
- **App Cloning and Repackaging:** Unauthorized duplication of apps is a persistent threat.
- **TLS Pinning Bypass:** Critical for defending against man-in-the-middle attacks.
- **Session Hijacking and App Impersonation:** Compromise user sessions and mimic legitimate apps.
- **Malware:** Leveraging app permissions (accessibility misuse, screen sharing, keyloggers, SMS OTP interceptors, etc.) for malicious activities.

Developer Awareness.

Flutter developers must stay informed about security threats and evolving attack vectors across all supported platforms. This demands n-depth expertise and continuous learning, making app security a specialized area within software development.

Essential Security Hardening Measures.

In the financial sector, regulators mandate the adoption of a range of security techniques, which can be categorized into three primary areas:

1. **Runtime Application Self-Protection (RASP):** Implement client-side measures to monitor and react to integrity and environment compromises.
2. **API Protection:** Safeguard against app impersonation using tools like Firebase App Check, attestation services, or API protection SDKs such as AppiCrypt.
3. **Anti-Malware:** Detects and mitigates risks posed by malicious apps on client devices.

Basic controls can often be implemented using freemium or community-supported tools. However, advanced enterprise-grade protection typically requires custom development or commercial security solutions.

Advances in Mobile Security Tools for Flutter.

The proliferation of app-to-API end-to-end protection solutions (such as App Attestation, AppiCrypt, and AppCheck) is effectively countering the escalating threats from mobile-oriented API abuse. These threats encompass App impersonation techniques such as botnets, password enumeration scripts, data scraping, promotional abuse, fake registrations, and phishing campaigns.

However, due to Flutter's compiled nature, Static Application Security Testing (SAST) tools have not yet reached the level of sophistication seen in native applications. This presents a challenge in maintaining security parity with other platforms. Conversely, the advent of Software Bill of Materials (SBOM) analysis has simplified the examination of third-party dependencies, thus enhancing the thoroughness and effectiveness of security assessments.

Overall, while there are still areas needing improvement, the strides made in mobile security tools for Flutter demonstrate significant potential in safeguarding against complex and evolving threats.

Budget Considerations.

App issuers should allocate 20% to 25% of development and maintenance budgets to security features. It's crucial to recognize that due to the dynamic nature of attack vectors and operating system updates, ongoing maintenance costs for security features may be significantly higher than initially estimated.

Conclusion.

Proactive security measures are not just beneficial but essential for app protection in today's dynamic threat environment. Ensuring comprehensive security requires both strategic investment and dedicated expertise.

talsec
Evidence of App Safety

Talsec is a community-driven mobile app security company. The core community product is freeRASP. The company is focusing on providing solutions for the Flutter community.

The Future of Flutter

How will Flutter meet the challenges of tomorrow?

In this section, you will learn the following:

1. How has sentiment toward Flutter changed over the last three years?
2. What are the most anticipated features according to the Flutter community?
3. How are development teams using AI?

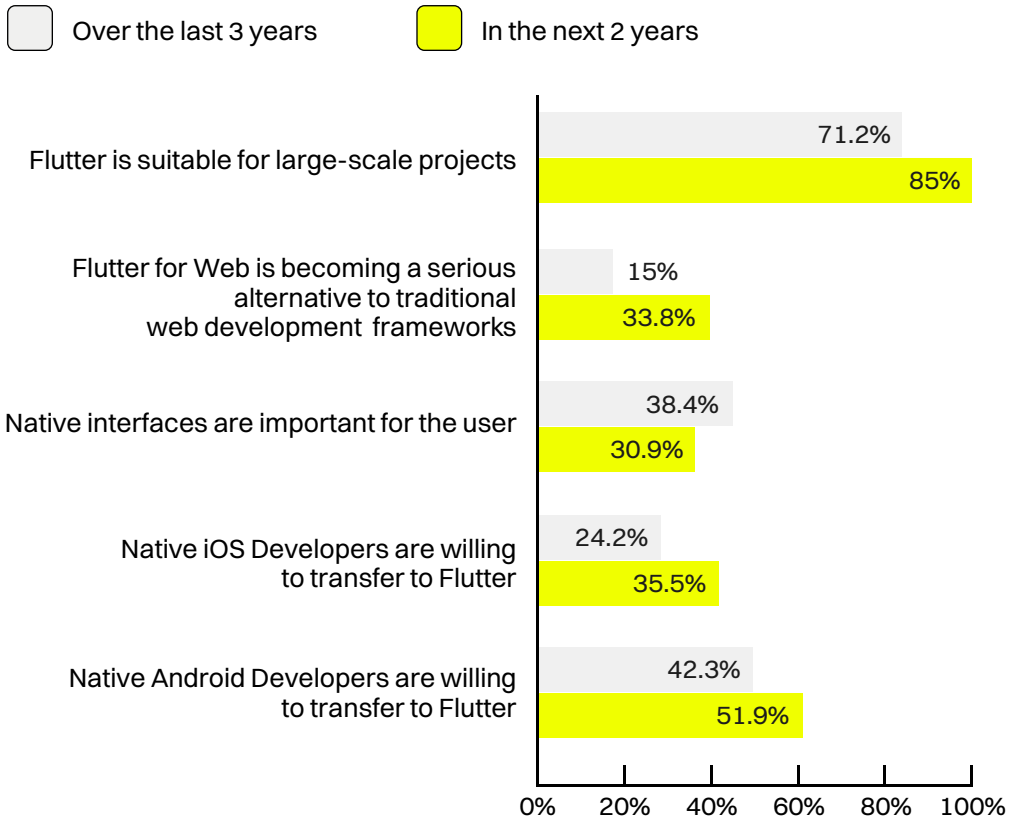
The future looks bright for Flutter. The improvements to the framework expected this year seem well aligned with the community's expectations. Flutter has matured and is increasingly becoming the default choice for large-scale applications. Teams using Flutter are also leveraging the advancements in AI, such as LLM (Large Language Models), within their development tools.

Google is actively encouraging developers to adopt their Gemini model by releasing the Flutter SDK. A recommended example of this SDK in action is an open-source package developed by LeanCode for automatic translation and localization of Flutter applications, utilizing Gemini and Vertex AI.

Flutter is a mature framework ready for enterprise, large-scale applications.

Please indicate the extent to which you agree with the following statements over the last 3 years/in the next 2 years.

Showing only “somewhat agree” and “strongly agree” responses.



In our report, we aimed to capture the sentiments of our respondents by asking them their opinions on several important topics surrounding the Flutter framework. This perspective spans the last three years and is compared with their expectations for the next two years.

The percentage of CTOs convinced that Flutter is suitable for large-scale applications increased from 71.2% to 85%, indicating a significant improvement in overall trust. This finding is consistent with other insights from the report, where more respondents now consider Flutter a serious alternative to web development frameworks, showing an increase from 15% to 33.8%.

INTERVIEW

IS FLUTTER ENTERPRISE READY?

Mateusz Wojtczak

Head of Mobile at LeanCode, GDE for Dart and Flutter

Łukasz Kosman

CEO at LeanCode

Find the answers to questions like:

- ☑ How has the Flutter framework developed over the last six years?
- ☑ How is Flutter encouraging developers to add new packages and new features to its ecosystem?
- ☑ Why are big companies transferring their apps to Flutter and redesigning their user experience?
- ☑ Is Flutter Enterprise ready?
- ☑ What are the most often-heard objections to Flutter?
- ☑ Is Flutter bringing productivity into the development of customer-facing apps?

Take a look inside.

"Flutter has been maturing a lot: the Flutter framework, and the state of the ecosystem packages. And I think that one of the signs of it is when you see how many products are being built and commercial products that are helping extend and bridge the gaps where the Flutter framework was not able to do before. I think that throughout these six years, we can see how more mature is also the community, the conversations within the community, also on GitHub, and around the issues".

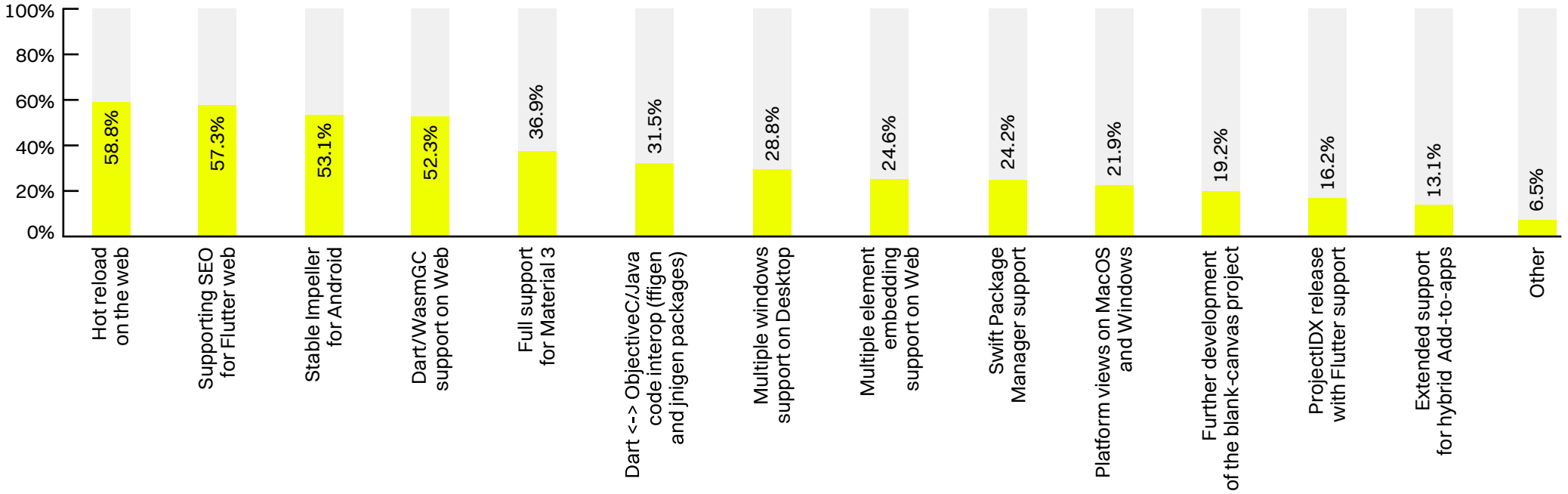
View on YouTube

Within the most expected features, 3 out of 5 are related to Flutter for Web.

The current roadmap of Flutter shows different ways the framework can develop as Google intended. We wanted to examine those paths and ask the community what improvements they wish to see.

Not surprisingly, the majority of votes were cast for web-related improvements. They range from features related to the developers' experience, like hot-reload on the web, which gained votes from 58.8% of respondents, to WASM support for Web with 52.3%, which was recently announced as being production-ready.

Which of these recent and upcoming new features are you most excited about?



INTERVIEW

WHY DO WE NEED FLUTTER FOR WEB?

Kevin Moore

Product Manager at Google

Łukasz Kosman

CEO at LeanCode

Find the answers to questions like:

- ☑ Is Flutter for Web production ready?
- ☑ Why are you betting on developing Flutter for Web when Flutter is known mainly for delivering mobile applications?
- ☑ What are the best use cases for Flutter for Web?
- ☑ Is Flutter for Web suitable for hybrid cases where most activity happens in the app, but some content still needs to be searchable online?
- ☑ Is the scenario of making embeddings in the existing web platforms also good from the perspective of starting with Flutter for Web for a mature product?

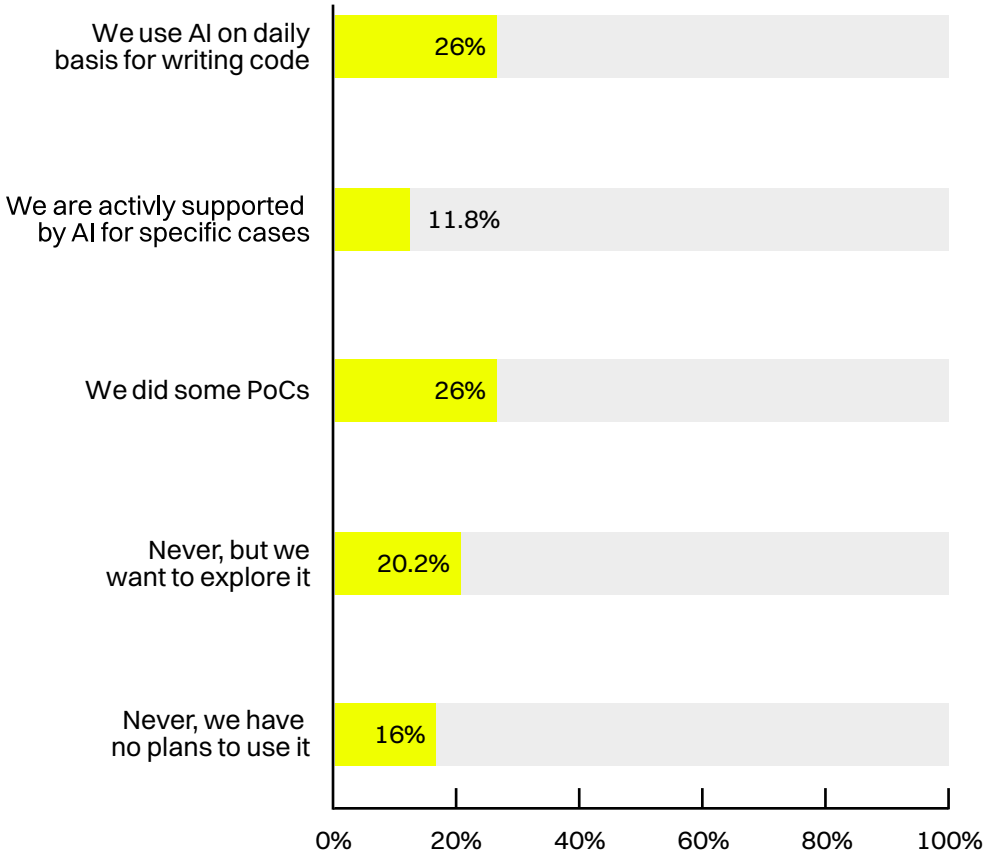
Take a look inside.

"I personally love the web. I think back to a kind of previous memory in my career, the idea that I have an office suite just in my browser. I don't think about installing my mail app or my word app or my whatever app and then installing service packet but just go to URL. I think in terms of my daily productivity, maybe aside from dev tools or maybe a couple of utilities, everything I do productivity-wise, is done on the web. I think even more and more experiences will be just web-based".

View on YouTube

AI is entering the Flutter Universe.

Are you experimenting with AI code generators while building Flutter apps?



Almost 64% of respondents have used AI tools in their development process. Among them, 26% apply Generative AI models to generate code, and 11.8% receive active support in their specific areas. AI not only supports processes related to code creation but also aids in testing and bug fixes. Developers' experiences are rapidly enhanced by various AI applications, significantly boosting productivity.

INTERVIEW

WE ARE ALL FLUTTER CONTRIBUTORS

Ander Dobo

Product Manager at Google

Łukasz Kosman

CEO at LeanCode

Find the answers to questions like:

- ☑ What role does the package ecosystem play in the Flutter framework and its development?
- ☑ What gives Flutter that unique community and many open-source contributors?
- ☑ How does the Flutter team encourage others to contribute to Flutter development?
- ☑ What problems and frictions should be addressed by the Flutter team, and how can other contributors engage to make Flutter better?
- ☑ How would AI change the way we build apps with Flutter?

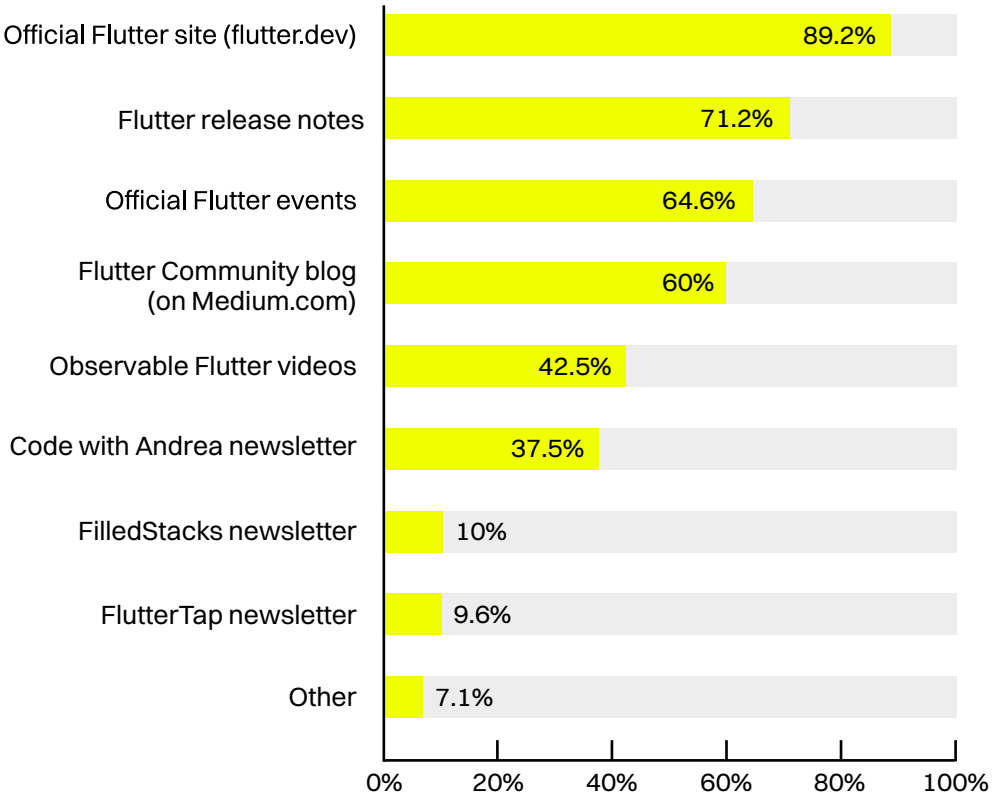
Take a look inside.

"Flutter, in every sense, from the language, the framework, and the ecosystem, is all open source. (...) I try to work with people in the ecosystem, and it is how you make it easier and more seamless for people to collaborate. Because it's not necessarily about everybody building their own package. It's how we can collaborate on building out different packages, whether that be cross platforms, whether that be features, you name it. So that's an important part of the ecosystem".

View on YouTube

Flutter leaders are using both official and community-driven initiatives to keep up with the news.

Which blogs, magazines, or newsletters do you read to keep up with Flutter news?



What clearly differentiated Flutter from alternative cross-platform frameworks, even in its early days, was its excellent documentation and strong community support. This is underscored by respondents who cited official documentation as their primary knowledge source. Notably, unofficial newsletters from ‘Code with Andrea’, ‘FilledStacks’, and ‘FlutterTap’ are also valuable.
For the latest Flutter news, you can subscribe to our newsletter at leancode.co.

INTERVIEW

THE EVOLUTION OF FLUTTER

Martin Jeret

CEO at Codemagic

Łukasz Kosman

CEO at LeanCode

Find the answers to questions like:

- ☑ How has Flutter's development and its community changed, taking into account its entire timeline?
- ☑ Has Flutter usage been growing when it comes to the business side of it?
- ☑ What is the perfect setup, and what are the requirements for using Codemagic CI/CD?
- ☑ Have you observed that Flutter is no longer used only for iOS and Android apps but also for desktop and web apps?
- ☑ What are currently the major obstacles and challenges for Flutter that could be overcome in the future?

Take a look inside.

"Technically there is very little that you can't do with Flutter. It's a very, very good tool to create beautiful applications that run on any screen. But the branding is still a problem. So, if you try to convince somebody who has been building Android apps for half of his life to now start to learn Dart, no matter how natively you compile it or how good those developer tools are, it is very hard to convince them to try something new. The same goes for trying to convince iOS developers to adopt Flutter. It's going to be even harder to do so".

View on YouTube

INTERVIEW

CAN DART BE TRULY FULL STACK?

Viktor Lidholt

Founder of Serverpod

Łukasz Kosman

CEO at LeanCode

Find the answers to questions like:

- ☑ How did it turn out that the founder of Serverpod had previously worked for the Flutter team?
- ☑ What made the founder of Serverpod believe in Dart as a backbone for the Flutter apps and led to the development of the backend for it?
- ☑ What is the business mode for Serverpod, a company that acquired funds from a pre-seed venture at a Danish venture capital firm?
- ☑ To whom is the Serverpod product dedicated with its specific set of features?
- ☑ Are bigger teams experimenting with a Serverpod product and engaging it to build some microservices or other?

Take a look inside.

"I instantly fell in love with Dart. It's a great language and has a bigger potential than just being a language for Flutter. Because it's a great language, it has many modern features like type safety, null safety, and really nice closures. The way you handle asynchronous programs is also very intuitive and easy to work with compared to working with threads in Java, which can be very, very complex. Also, I think more Flutter developers should look into the backend side of Dart because it's really pleasant to work with".

View on YouTube

Would you like to discover more?

Learn about building large-scale apps with Flutter.

Top Flutter Development Company

LeanCode is a software house from Warsaw, Poland. We are a leading provider of mobile apps for scaleups and enterprises and an official Flutter consultancy firm.

Who we are.

At LeanCode, our team of 70+ developers, designers, product owners, scrum masters, and QA engineers is dedicated to crafting exceptional mobile and web applications. Leveraging the power of Flutter, .NET, React, and other advanced technologies, we transform ideas into great apps.

Our expertise.

The majority of our clients represent the Banking and Fintech industries. However, our expertise doesn't stop there. We also develop innovative solutions for Marketplaces, Logistics, SportTech, MedTech, and beyond. Our clients span the globe, including the USA, UK, Germany, and Australia.

Awards and achievements.

We were part of the Deloitte Fast 50 contest for CEE's fastest-growing IT companies in 2021, and we made the EMEA Fast 500 list in 2023. Additionally, we were featured in the Financial Times 1000 Ranking for Europe's Fastest-Growing Companies in 2023 and 2024, and we received accolades in the Forbes Diamonds 2023 and 2024 Rankings.

Community and innovation.

At LeanCode, we believe in giving back to the community and fostering knowledge-sharing. In 2020, we co-founded Flutter Europe, and we're the proud organizers of Flutter Warsaw, Europe's largest Flutter meetup, boasting over 1,900 members worldwide.

We build our own frameworks (e.g., Patrol–an open-source UI Testing framework) and solutions, contribute to open-source projects, have our own custom Internship Program, and give lectures at several universities in Warsaw, Poland.

Our services include:

- **Flutter App Development**
- **Flutter Mobile Applications**
- **Design Sprints for Mobile Applications**
- **Implementation of AI Solutions**
- **Automated UI Testing in Flutter**
- **Flutter Staff Augmentation**

Discover more about our core technologies, services, and successful projects on our website:

leancode.co

Got a project in mind? We're excited to hear about your vision and explore how we can bring it to life.

Let's talk about your plans for using Flutter!

Contact us

LeanCode