

The Business Value of Flutter

How switching to a single codebase reduces the cost, effort,
and time to bring beautiful digital experiences to market.



The Business Value of Flutter

How switching to a single codebase reduces the cost, effort, and time to bring beautiful digital experiences to market.

Software Drives Delightful Customer Experiences

When you glance around an airport, café, or train station and see almost everyone staring down at their phones, it's hard to believe that it has been less than 20 years since smartphones were introduced into the world. Phones are globally ubiquitous, and organizations of all types offer mobile applications as a key touchpoint in their overall customer experience.

For most organizations, mobile applications are a critical part of an overall customer experience. For others, mobile applications are one touchpoint among many others, including web, desktop applications, and even embedded systems, such as smart homes and smart cars.

But for almost all businesses and organizations today, software applications are a necessary element of how they do business and a critical part of their overall customer experience.

Top brands are choosing Flutter

What does it mean when the top performers in many industries are moving to develop with Flutter? From BMW to Alibaba, early adopters are seeing the value Flutter brings to their business.





To reach your customers in most markets, you need to offer Android and iOS apps, and also a Web browser-based experience.

Building and managing multiple digital experiences for multiple touchpoints and platforms is costly and time consuming. The CTO of a smart home product company described a typical development experience thus:

“We had been struggling with the problem of maintaining multiple teams and multiple platforms – iOS and Android were two separate teams. Our standard procedure is to kick off an app in iOS, and when that is 70% done, we fire up an Android team. This approach results in higher cost, more delays, and the core functionality such as Bluetooth has its quirks on each platform. We wanted a way to get out of this hole.”

This experience is not uncommon – we have come to accept that in most instances we need to offer customers an Android and iOS application, as well as a web app, and perhaps even customer experiences for custom hardware such as smart connected devices.

Enter Flutter

Flutter is an open source framework from Google, which allows you to deploy beautiful Android, iOS, web, desktop and embedded applications from a single codebase. It meets the promise of “write once, run everywhere.” Flutter applications are

built in Dart, an open source language with similar syntax to Java and C#, making it easy to learn for application developers. Flutter compiles to native machine code for each target platform which means the **applications run fast, look great and feel like a native application to users.**

See <https://flutter.dev/> for more details.



Does it really write once and run everywhere?

The promise of platform-agnostic software is almost as old as the technology industry, but the idea of “write once, run everywhere” was popularized by Sun Microsystems in their 1995 marketing campaign for Java. Fast forward to the mobile era, and developers with experience of React Native or Xamarin may be skeptical of cross-platform application development frameworks due to the performance hit. However, because Flutter apps are compiled into native code, it performs better than those that are run in a JavaScript engine. In fact, most Flutter applications provide smooth performance that is indistinguishable from native applications. While many software frameworks have developer communities, Flutter’s developer community is one of the most popular, active, and thriving.

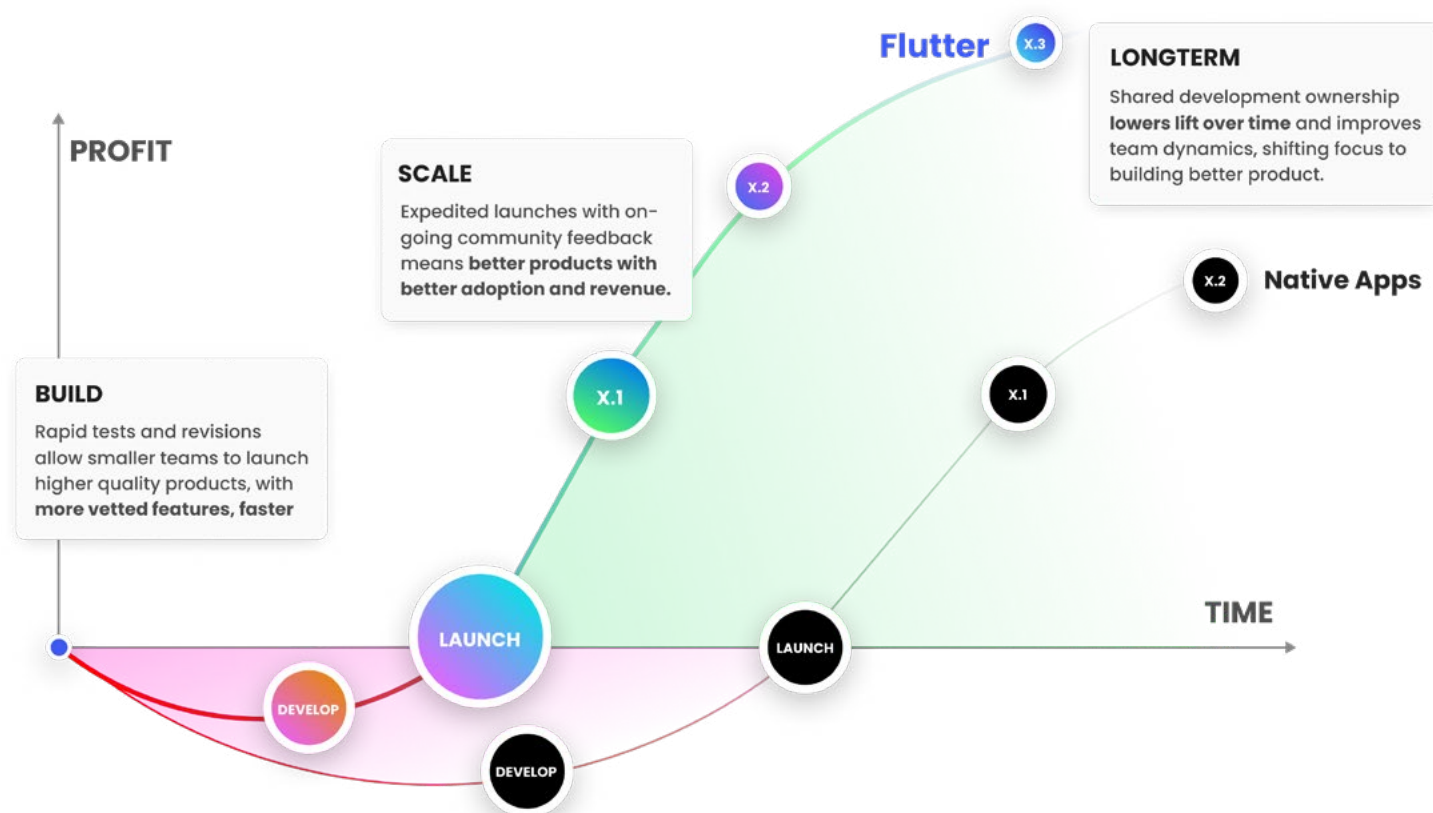


Quantifying the Value of Flutter

What does it look like when a multi-platform product team decides to adopt Flutter?

Every company is unique but based on a series of interviews with teams who have been on the journey to adopt Flutter, we built a model of a typical company making the switch from offering Android, iOS, web, desktop and embedded applications built specifically for each platform, to using Flutter for across the same touchpoints. We will use this model company to illustrate the costs and benefits of this switch.

Looking across the product development process we uncovered several differences across three key phases of a product lifecycle. The phases include the **Build Phase** where we move from concept to prototype to MVP to launch, the **Scale Phase** once a product is launched and is out in the market, and the **Long-Term Impact Phase** which is when post-launch maintenance occurs and the longer-term impact to team and company culture become clear.





62%

Reduction in launch time with Flutter

Compared to working with multiple teams on native apps

1

> Build Phase

Across the companies we interviewed, the **time to launch a product to market with Flutter was reduced to 62%** of the time it took when compared to working with multiple teams on native applications. This is clearly a dramatic reduction in effort and cost during the design and development phase. Of course, the time to launch varies greatly by the complexity of an application, as well as the product development lifecycle within a given industry. For example, launching an in-car application in the automotive industry such as maps or the instrument cluster on a heads-up display can take years, whereas a mobile app for wellness can take months. But overall, it took the Flutter teams we interviewed an average of six weeks to get to MVP, and 15 weeks to get to launch across multiple touchpoints.

How does Flutter deliver these advantages?

First, it requires fewer engineers and designers to build applications in Flutter. Several F500 companies we spoke to maintain a 30-50 person product team to build fully featured products on iOS and Android. **With Flutter, they delivered the same functionality with a team 55% the size**, as they can write once and run everywhere. For many companies who live under the constant pressure to deliver more to the market, these teams have the ability to redeploy the additional team resources to new growth areas.

Aside from the direct cost of employees, smaller product teams work more effectively as their communications overhead is greatly reduced. For example, a team of four people has 6 possible connection points, but

a team of ten has 45, and a team of forty, 780! No wonder large teams spend so much of their time in meetings. Compounding this challenge is the dynamic that iOS, Android, web, and perhaps even an embedded systems team working on novel hardware often compete for resources. Even with the best of intentions and a collaborative spirit, development gets slowed down while teams coordinate on roadmaps, debate release schedules, and prioritize functionality.

Since Flutter is multi-platform, prototyping and testing becomes much faster. Product teams get from idea to application within a day. Particularly for those working with embedded systems such as an in-car experience or smart home systems, rather than needing to work with reference hardware that is hard to acquire, you're able to see what you're developing run on any hardware for faster feedback and iteration loops.

This was also used to their advantage with user testing, where **high fidelity prototypes elicited useful feedback, allowing the team to test and iterate.** According to one engineering leader we spoke to, "There is typically a huge lead time to see what you are designing. Flutter, being multi-platform, allowed us to share a URL that anyone could see in a browser. The ability to show stakeholders high-fidelity prototypes was huge, and it was also really helpful in user testing."



“The ability to show stakeholders high-fidelity prototypes was huge, and it was also really helpful in user testing.”

> *Continued from Build Phase*

A related benefit of Flutter is its hot reload feature, which allows changes made in Flutter to be immediately loaded into the app for the dev team to see the change without recompiling the application. This simplifies iterative development and further reduces the time to build and refine an application.

Flutter’s extensive widget library gives developers access to pre-made functionality, increasing speed to market. Being open source, those working on unusual use cases or novel hardware were pleased to find that the Flutter core team is open to contributions. We heard from several engineering leaders that the Flutter team at Google was helpful, responsive, and streamlined in their approach.

In general, engineering leaders seem to find the standardization that comes through leveraging existing Flutter solutions increased the speed to market. According to one respondent, *“We’ve been forced to accept some standardization and become more open to leaning on standard Flutter solutions which gets us to market faster.”*

Is Flutter mature enough?

One of the concerns about leveraging Flutter which came up in discussion is that the framework is relatively new. Things like AWS library compatibility for Dart can lag compared to building for Android or iOS. For those working with Flutter for embedded systems, you can hit some very specific platform dependencies which can take time and resources to resolve. Flutter is evolving and maturing quickly, with new capabilities and plugins being released all the time, but that does mean that things change.

The way to approach this is to evaluate your use case and make a determination on whether Flutter is an appropriate technology choice at this time, or not. Flutter experts, such as the team at Very Good Ventures, can help you make this determination and work with you to weigh the costs and risks against the benefits.

Many members of the VGV team came from product companies that went through the evaluations and determinations to choose Flutter at firms like BMW which has since adopted Flutter across multiple platforms. Moreover, we have found Google to be very responsive to requests for new capabilities. Some teams may conclude that it is worth pushing the boundaries to remain at the edge, whereas others may opt for a fast follower strategy.



> Scale Phase

There is almost nothing a product manager can do to move the financial needle for their company more than bringing revenue forward by launching a product sooner. We tend to lose sight of this as we are so accustomed to development timelines and launch dates slipping or features in a roadmap being pushed to later iterations. Moreover, product teams are usually held accountable for revenue once a product is launched, but the opportunity cost of launching late is rarely considered. Even for non-revenue generating applications such as companion and support applications that come with connected home, cars, or other products and services, being able to launch sooner generates financial benefits for the company. There is a time value to money, and **our model company showed an ability to launch in 62% of the time compared to non-Flutter alternatives.** While that is an average across all companies, those working in embedded systems (as opposed to mobile apps) showed the greatest increase in speed to market, in one instance launching product into market at 20% of the time it took without Flutter.

The adoption curve of a Flutter product is steeper than conventional alternatives for several reasons:

First, because iOS and Android versions of the product can be launched at once, you are immediately reaching your entire target

market. This is in stark contrast to the more common practice of selecting iOS or Android as the primary platform and launching that first, following on later with the second platform. Not only does that make part of your target market feel like second class customers for being on the “wrong” platform, but it also reduces your reach out of the gate. Our model company, based on the average experience of companies interviewed, **launched with 2.5 touchpoints** as opposed to the benchmark of 1, which is of course increasing the customer reach by 2.5x right at launch. If you also offer web browser access, desktop apps and embedded systems, you are greatly increasing the potential for customer adoption immediately at launch.

Second, because with Flutter you have had greater engineering resources available to you before launch, your products are likely to be more feature rich. Whereas many MVPs are indeed “minimal,” by reallocating the additional engineering resources freed up by Flutter to work on other features, the product at launch is likely to be more capable. This makes the product more attractive, which in turn drives greater adoption. Related to this is also the high performance of Flutter products. Because Flutter code compiles to ARM or Intel machine code as well as JavaScript, its performance on mobile devices is equivalent to native applications. In embedded systems, Flutter really comes into its own.



> *Continued from Scale Phase*

In the words of one engineering leader, *“There isn’t anything else out there that brings consistent user experience [to embedded systems] with the mechanics and interactions people expect from interacting with their smartphones.”* Our model company therefore has a steeper adoption curve than the benchmark company using traditional methods of development.

A third and related reason for the steeper adoption curve is the learning benefits from getting to market sooner and being in market longer. Before a competitor using traditional development approaches even has their initial product in market, our model company is releasing product updates, bug fixes, and new features. The ability to learn from the market and iterate is a tremendous advantage. In our model, the average release frequency moved up to twice a month for companies using Flutter, compared to quarterly or slower updates in their pre-Flutter state. This improvement was particularly dramatic for companies working with embedded systems such as connected home and connected car devices.

A final benefit for Flutter in the scaling phase is lower support costs. While there wasn’t enough data with the model company to quantify this, a strong product induces customers to self-service, and reduces the number of messages to your contact center.

For companion or support apps, encouraging customers to use self-service features is critical for the business model to work. Even for revenue generating apps, self-service lowers the cost to serve and often increases user satisfaction compared to calling or messaging with a support center.

“There isn’t anything else out there that brings consistent user experience [to embedded systems] with the mechanics and interactions people expect from interacting with their smartphones.”



Long-Term Impact on Development Teams

3

> Maintenance Phase

The great wave of digitalization over the past decade has made every company a technology company, at least to some extent. And this in turn has made engineers and designers a scarce resource. Leading internet and consumer tech firms continue to attract and retain vast numbers of technical and product talent making it more challenging for other types of businesses to keep mobile application talent engaged. According to one engineering leader we interviewed, *“It’s difficult to sustain the interest of talented mobile engineers unless you are Netflix or something. It’s especially hard to keep people working on companion applications, which results in a fair amount of churn.”* But despite the challenge, all companies need to build credible technology capabilities and a strong engineering culture.

While Flutter doesn’t necessarily solve the issue of engineer scarcity, it does prove attractive for some of the companies we interviewed, especially those working on touchpoints beyond mobile applications. According to another engineering leader at a F500 company,

“when we decided to bring software development in-house, the small existing team was concerned that we would all become QT developers. Choosing Flutter was a much more attractive option for the team and ...had a positive impact on the team culture.”

Moreover, for businesses driving digitalization of their existing businesses, Flutter provides great benefit by significantly reducing the complexity of technology stacks used, and simplifying the engineering team structure.

According to a 2022 developer survey from Stack Overflow, Dart is used by 6.54% of the 70,000 developers who responded to their survey. This increased to 7.34% when the sample is reduced to those learning to code, which points to continued growth in the future. Within cross-platform mobile frameworks, Flutter and React Native are the two most commonly used, and according to a developer survey conducted in 2022, **Flutter is the most popular mobile framework used by global developers.**

survey.stackoverflow.co/2022/#most-popular-technologies-language-learn



> *Continued from Maintenance Phase*

However, despite its growth, Flutter and Dart are still relatively new to the industry. Several engineering leaders we interviewed had opted to re-train their mobile engineers to use Flutter, and for our model company the **average training time required was 14 weeks** (Estimates ranged from 1 month to 6 months, with a mean 14 and mode of 3.5 months.)

The engineering leader of a consumer electronics firm made the following observation on retraining engineers:

“The people who picked up Flutter on my team tended to be higher quality engineers. Dart’s not anyone’s first language – they have often worked in C#, React Native and so on – so they understand the limitations of all languages. Nine times out of ten, Flutter developers understand the value of state management. They get architecture. It could be the way that Flutter is communicated but the result is higher quality work.”

Some companies found that **switching to Flutter changed the culture of their teams**. We have already discussed how having a small, unified team reduces conflict and lowers communication overhead, resulting in greater productivity. Beyond that, one engineering

leader noticed a subtle but profound change to team discussions.

“When we were native only, a lot of conversations were ‘no, because...’ with the blame going to underlying systems. Shifting to Flutter cut a lot of tech debt. The discussions became ‘yes, if...’ It’s a subtle but profound change in how we work. By default, the development mindset has gone to a more positive place.”

Other teams found that adopting Flutter **resulted in the engineering team focusing on the user experience rather than technology** platform. This is a profound culture shift, and one that increases the quality of the customer’s experience with a digital product. Rather than worrying about how a feature is going to be implemented in web, iOS or Android, the designers and engineers focus on what the optimal experience should be. This changes the nature of the conversations within a product team.

Finally, because of the patterns and practices built into the Flutter workflow, many teams found that their culture changed quite dramatically. The level of automation and test coverage increased, and the product team became more unified. Overall, product teams found that code quality increased, and with it, both individual and team pride.



Understanding Adoption Trends

One of the common concerns we heard from engineering leaders was whether there are enough trained Dart programmers for them to hire.

One of the common concerns we heard from engineering leaders was whether there are enough trained Dart programmers for them to hire. In North America and Europe, we heard from respondents who had struggled to find Flutter developers in their region. One CTO had even put future migrations on hold until he was confident that he could hire the talent he requires.

Three mitigation strategies emerged. The first and most common option is to **train existing staff in Dart and Flutter**. One CTO initially lost three engineers who felt that developing expertise in Flutter would be a career cul-de-sac, but within six months he was receiving inbound resumes from developers who had heard his firm was “a Flutter shop.”

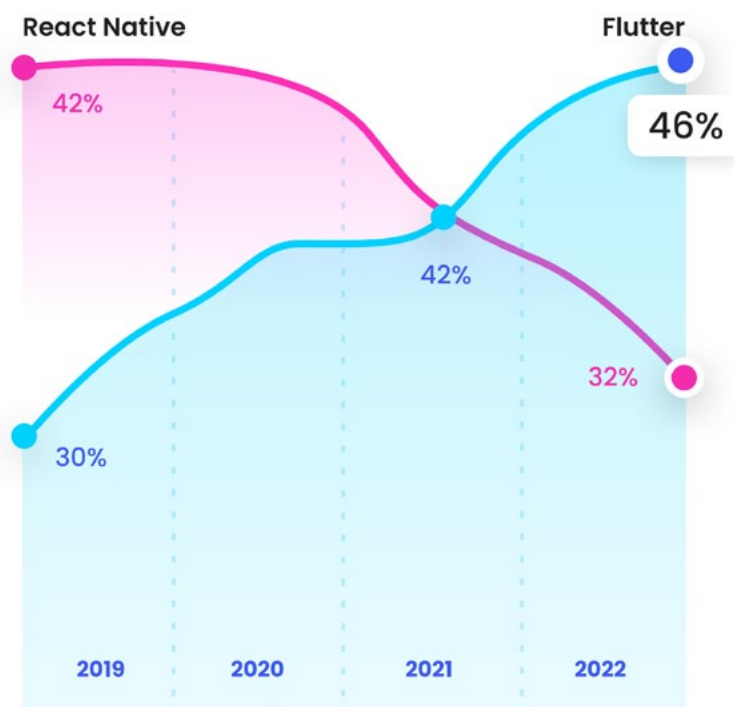
The second strategy is to **widen your aperture and hire nearshore and offshore talent**. A Scandinavian team included engineers in Poland and Ukraine, and a New York based team hired engineers in Mexico.

The third is to **hire consultancies with expertise in Flutter**, such as Very Good Ventures to both train and support your team as it scales.



Are there enough Flutter devs?

As you can see, trends point to more worldwide developer adoption of cross-platform mobile frameworks like Flutter.





Summary of Business Value Drivers

The data in our model company illustrates that there are many direct and indirect business benefits to using Flutter. But broadly speaking, the benefits can be summarized with the following five value drivers:

> Build Time

Because you are building one codebase rather than having multiple teams working in iOS, Android, web, desktop and even embedded systems, the time for a Flutter team to build and launch an application is cut to 62% of the time it takes in a multi-code, multi-team environment.

> Time to Revenue

By launching a revenue generating application a few months earlier to the full addressable user base, you pull revenue forward for your company. Even for non-revenue generating apps such as companion support applications, launching sooner results in generating user value sooner. Getting to market faster generates learning benefits from being in the market longer and sooner than competitors, which allows you to iterate on your application and advance its capabilities before your competitors.

> Team Satisfaction

Most companies in our study reported higher team satisfaction amongst the engineers, designers, and product managers in their mobile product teams from using Flutter. By combining the Android, iOS and web teams into one, the product team had more bandwidth to launch fully featured products and had more time to test and iterate their product. Competition and friction between various teams was reduced, and the teams quickly learned Dart and new skills in Flutter. Moreover, if the move to Flutter is coupled with an overhaul of the product development process, it can result in a common culture and way to get things done, which in turn creates a less chaotic work environment.

> Customer Satisfaction

The engineering team's greater bandwidth, coupled with Flutter's extensive libraries and tools, results in applications that work well and create greater value for users. Moreover, there isn't that typical lag

between launching iOS apps and Android, meaning all customers feel equally valued. Our study consistently found that the customer satisfaction and customer lifetime value scores increased with Flutter.

> Lower Cost

By reducing the time to build an application, and reducing the size of teams required to build, test and maintain a product, Flutter helps lower the cost of engineering. Moreover, the higher-quality Flutter apps tend to result in lower support costs, as customers require less support with applications. The teams we spoke to were able to reduce the number of engineering contractors, while other teams increased their productivity and had their engineering and product teams work on other things. A common fear within legacy teams is that moving to Flutter will reduce the number of engineers and lead to layoffs, but most of the companies we interviewed redeployed their internal engineers on other tasks and workflows to accelerate feature development.



Upshot of Flutter Adoption

Every delightful digital experience sets a new benchmark for us as consumers and resets our expectations for all other digital touchpoints.



In this virtuous cycle, only the very best experiences stand out from the crowd. The past 20 years have been great for software users, but ever more challenging for companies offering digital touchpoints – which is compounded by multiple mobile operating systems, as well as web, desktop and embedded systems. **Flutter gives companies a way to break out from this arms-race and focus anew on the user experience.**

Ultimately that is the great power of Flutter. Flutter has a lot of cost benefits – lower development costs, lower communication overhead, smaller teams etc. – but businesses don't cut their way to revenue growth. **In order to grow, and to be successful in the market, you need to provide great products with delightful customer experiences that meet and exceed the expectations of your customers.** And this is where Flutter truly excels.

“It’s not about making something on one platform and making it work on others. Rather, **defining a user experience and making sure that it is translated across the touchpoints** gives us more power as developers. Flutter is almost a code engine – it renders a screen to look native as a 2D game engine. This speaks to me as it translates the core user experience across multiple touchpoints.”

What You Can Do Now

Train Excellent Application Developers

“iOS developers and Android developers don’t exist,” according to a leader of a F50 mobile product development team. This leader wants his designers and engineers to think of themselves as customer experience designers with a core skill set of building excellent user interfaces. Of course, his team comes to the table with deep iOS, Android, web, desktop app or embedded systems application design experience, but truly great application developers are hungry to learn new skills and approaches. Flutter fits well into the career path of application developers who want to push their craft.

Very Good Ventures has been building Flutter applications since 2017 and has codified best practices in an ebook entitled *Migrating from Native to Flutter, the Very Good Ventures Way*. We provide Dart and Flutter training for in-house development teams, often as part of a Flutter initiative. Get in touch to see how we can partner with you to train your team.

Getting Started with Flutter

Very Good Ventures has been building large scale applications in Flutter since 2017 and has assembled the **best team of Flutter designers and developers in the world**. We partner closely with your in-house product team to design, build and scale successful applications. We have been through the Flutter migration process with many companies across all industries and can help you plan for expected challenges and deal with unexpected contingencies.

Get in touch to see how Very Good Ventures can support your Flutter initiative.



hello@verygood.ventures
🐦 🌐 📷 @VGVentures