



Migrating from Native to Flutter the Very Good Ventures Way

Visit us at verygood.ventures



Migrating from Native to Flutter the Very Good Ventures Way

Very Good Ventures saw Flutter's potential early, building the first large-scale commercial app with it in 2017, and we've never looked back. Our experience with dozens of enterprises has allowed us to build migration paths that are both low risk and high impact.

In this paper, we will discuss migrating from native development to Flutter and how Very Good Ventures can help ensure your migration success.

What is Flutter?

Flutter is a multi-platform engine that targets every device with a screen.

It aims to provide a write-once development environment where the same code runs on iPhones, Android phones, Windows, macOS, and even Linux. Even better, it compiles to native machine code on each of the platforms it supports. That means it's fast.

Flutter can also be used to develop web applications, using the **exact same code**.

The code once, run anywhere approach sounds like a dream. Is it realistic? And will it work for your project? We think so, and we'll tell you why.

Comparing Flutter

Flutter isn't the only multi-platform framework out there. Most developers looking at Flutter will have also heard of React Native. Both compile to native code for multiple platforms, but VGV prefers Flutter for a few important reasons:

- ✓ Flutter can be used to create mobile, web, and desktop applications.
- ✓ Thanks to its extensive cross-platform widget catalog, Flutter enables a great UI to be built quickly.
- ✓ React Native apps require a bridge between the app and native code, which can carry a performance penalty.



Application Development is Expensive

Software development is a specialized skill, and specialists are expensive—there is no getting around that. The costs add up quickly, because an application that needs to run on multiple platforms needs multiple teams of developers. All the platforms may be able to connect to a single backend, but each requires its own frontend interface.

The problem is that every frontend platform requires different skills, different programming languages, and a different approach. Your iOS expert has built an entirely different skill set than your Android expert, and it is unlikely that either of them will be proficient with both. This means that development teams are creating **silos of specialized knowledge** that can't be shared across target platforms!

The issue compounds when you add in desktop and web platforms. Developing and maintaining a single application with a single codebase can be challenging enough. Half-a-dozen different platform-specific implementations and codebases, each targeting their own operating system or environment, produce a larger and more complex landscape.

This has led to some difficult problems.

- 1 Application development is slowed down because new features need to be implemented once per platform, by different teams.
- 2 Application development is more expensive, because each platform needs its own team of developers.

These challenges are why so many popular applications such as the podcast app Overcast, Notepad++, and Omnifocus are not available on every platform.

But it doesn't **need** to be that way.

Visit us at verygood.ventures

Go Deeper:

Multi-Platform

What does it mean when Flutter is described as a multi-platform engine? It means that your team isn't coding the same element three or four times—once per target platform. Each screen, button, input field, knob, and widget **only needs to be programmed once**, and it's available on web, Android, iOS, and desktop. Your team can focus on building the product, instead of building for specific platforms.

Flutter supports the following platforms:

- ✓ Mobile
- ✓ iOS
- ✓ Android
- ✓ Desktop
- ✓ Windows
- ✓ MacOS
- ✓ Linux
- ✓ Web
- ✓ Embedded devices

When is the Right Time to Switch to Flutter?

As applications grow and development becomes divergent across platforms, it frequently seems like the time to switch was yesterday.

Moving to Flutter involves more than just shifting programming languages. It also involves adapting styles and GUI elements to a new widget toolkit. This process involves both frontend developers and the design team, and it can honestly take a few code sprints to get the core elements in place. Managers should expect a few development cycles before they see any results from the shift to Flutter.

Many developers and managers know the pain of pivoting to a new programming language. Flutter has a feature to address this crux: add-to-app functionality. We will cover add-to-app in the next section.

There are a few signs that indicate a change is needed in your team's approach to application development. Are features taking longer and longer to deploy? Are your applications getting out-of-sync across platforms? In addition, when it is time to make a pivot in direction, such as towards a mobile-first approach or towards modernizing your stack, it is prime time for migrating to Flutter.

While finding the right time to make the switch is going to be dependent on each team's requirements and roadmap, the right time to switch is right away. The sooner you make the change, the sooner you'll be developing each feature just once instead of once per platform.

Go Deeper:

Developers are Happier with Flutter

[Developers agree](#) that Flutter has made application development faster and more efficient.

92% agree that Flutter reduces the time to build and publish new applications.

90% agree that Flutter enables faster development on existing applications.

84% of developers agree that Flutter makes their applications more beautiful.

Getting Started with Flutter

Through our years of experience, Very Good Ventures has developed a training program that will get your team started with Flutter quickly. This is how we do it.

Since Flutter applications are written in Dart, learning Dart is an important first step. Very Good Ventures offers a three-day facilitated Flutter training course to teach a consistent foundational knowledge of Flutter, Dart, developer patterns, and best practices.

While we will help get your developers proficient with Dart, many developers prefer to do some self directed learning. The Dart website has [many helpful tutorials](#) that cover both the basics of the language and some of its more significant concepts, such as asynchronous streams and shared packages. It is also entirely possible to skip the Dart-only tutorials and jump straight to [building your first Flutter app](#).

“Developers reported being able to build features faster and Flutter would allow us to control everything from UI to the business logic with one codebase.”



Sam Moore

VP of Architecture at Betterment

In 2020, Very Good Ventures helped [Betterment pilot their first Flutter feature](#). They found that over a period of six months, they were able to start adding Flutter features to their app while training their entire team to be Flutter experts.

Visit us at verygood.ventures

Go Deeper:

Dart's modern features

Flutter is written in Dart, and developers writing for Flutter will appreciate its many modern features that make programming easier, faster, and safer.

Features such as type safety and garbage collection help developers code faster and more safely by catching more errors at compile time and keeping runtime memory consumption lean. Hot reload means developers can immediately see the impact of their changes.

VERY
GOOD

The Very Good Ventures Pilot Program for Flutter Adoption

Very Good Ventures understands that getting started the right way takes a smart plan.

First, the decision makers need to be clearly identified and fully engaged in the migration to Flutter. Likewise, an internal team of two to four engineers should be fully dedicated to the pilot program for its entire duration, though as many engineers as possible should be getting ramped up on Flutter and Dart.

During training, the first round of code is written for evaluation purposes only—it will not be production ready. However, the engineers need access to systems and resources to meet their requirements.

Learning Dart

Flutter applications are written in Dart, which means that developers get to take advantage of all of that language's advanced features. Luckily, most developers find Dart easy to learn.

```
1 //Here's "Hello, World!" in Dart
2
3 void main() {
4   print('Hello, World!');
5 }
6
```

Very Good Ventures Pilot Timeline

Stage 1: Defining evaluation criteria	1 Week
Stage 2: Facilitated Flutter training	1 Week
Stage 3: Applied Flutter training	2 Weeks
Stage 4: Developing a business-specific proof-of-concept	4 Weeks
Stage 5: Building an implementation roadmap	2 Weeks
.....	
Total: 10 Weeks	

Stage 1

Define evaluation criteria

Very Good Ventures will collaborate with you to develop the criteria that matter most to your team and business. Once we have this framework, we can use it to ensure that Flutter fits the bill. Here are some of the criteria we will help you evaluate:

- ✓ Developer experience (tooling, IDE integration, hot reload)
- ✓ Long-term viability
- ✓ Learning resources

► For a good discussion of Flutter and the developer experience, check out [**Nubank's account of their Flutter journey.**](#)

Stage 2

Facilitated Flutter Training

[Optional]

Once you're committed to trying Flutter, your team receives hands-on live [training](#) delivered by VGV's Flutter experts. We've developed a curriculum informed by our years of expertise that teaches everything you need to know to build scalable, cross-platform applications.

Topics covered in VGV Flutter Training:

- | | |
|----------------------|---------------------------------------|
| ✓ App architecture | ✓ Forms |
| ✓ State management | ✓ Testing & automation |
| ✓ Building custom UI | ✓ Internationalization & localization |
| ✓ Animations | ✓ Accessibility |
| ✓ Navigation | ✓ Packages & plugins |
| ✓ Networking | ✓ Backend development |

Stage 3

Applied Flutter Training

[Optional]

This is where we'll go from abstract to specifics. Together, we will build a complete, fully tested app using pair programming so that your team can apply what they've learned.

Stage 4

Business-specific Proof-of-Concept

The best way to validate that Flutter is a solid choice for your team is to put it into practice in your own codebase. Using the skills they have learned so far, we guide your team through building a proof of concept (PoC) entirely in Flutter.

Stage 5

Implementation Roadmap

After building the PoC, we can refer back to evaluation criteria to see if Flutter is the right choice. We help you design a migration roadmap that includes an organization plan, processes, and risk mitigation strategies to guide you on your path to full Flutter migration.

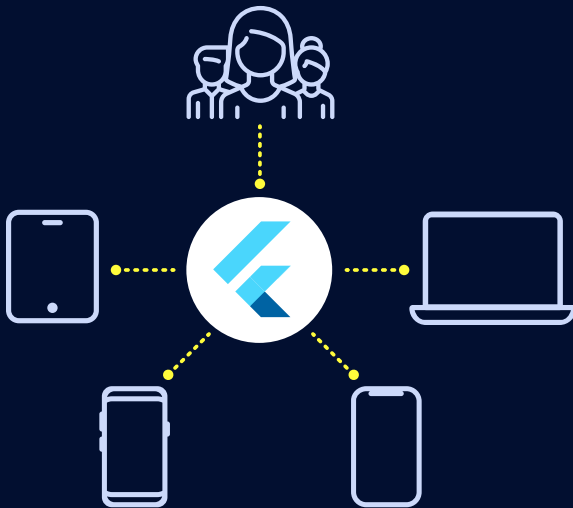
While taking the plunge and embracing a full rewrite in Flutter is always an option, it's not the only one. The move to Flutter can be incremental, thanks to its add-to-app functionality. This allows Flutter to be integrated as a library or SDK, turning portions of an application over to Flutter a piece at a time. In a few steps, you can have functional Flutter code in production. The add-to-app functionality works for Android, iOS, and web applications.

- ▶ Learn more about Flutter's add-to-app functionality in [**Put Flutter to Work.**](#)

The Benefits of Team Transformation

Shifting to Flutter initiates a team transformation. Each specialist can now apply their expertise to a shared codebase, which compiles to native code for every platform your application targets.

The teams duplicate less effort, leading to a more nimble product.



Benefits Begin Paying Off Quickly:

- ✓ Subject matter experts from previously divided teams can now **collaborate on the same code base**.
- ✓ Teams can focus on features instead of platforms, leading to cohesion and **more rapid development**.
- ✓ New features take a fraction of the time, freeing up developers for **more innovation and improvement efforts**.
- ✓ Developers will be **less frustrated and happier** with the development process.

Embracing the Transformation

A good plan is the key to a successful migration, and Very Good Ventures is here to help you learn all you can about the process of moving an application to Flutter. By understanding the requirements, we will help your team reap the benefits of multi-platform application development quickly!

A Flutter transformation affects all teams that are required to manage and grow your apps. As engineering teams and practices are consolidated and optimized around consistent best practices, the way that other teams operate can be radically redefined and optimized as well.

Deciding to make the move to Flutter puts your goals in sight: one development team, one app, and features rolling out so quickly that the sales team can't even keep up!

Ready to help

We are the leading Flutter app consultancy, and have helped companies design, build, and scale successful apps. We have proven that the best development outcomes result from partnership and deep collaboration. Our team works with yours, openly and without barriers.

We have been through the migration process before, and can help you be prepared for both common and unexpected contingencies. Our early adoption and long experience with Flutter isn't just blind zeal—we have seen Flutter bring multi-platform app deployment to many companies.

Interested in learning more about how Very Good Ventures can help you migrate to Flutter? Reach out!

verygood.ventures/contact

Visit us at **verygood.ventures**

