

Building Enterprise Apps with Flutter



**VERY
GOOD
VENTURES**

Building Enterprise Apps with Flutter

An App Building Approach on the Rise

Application development has become a mission-critical objective for a huge cross-section of organizations. Across industries and around the world, organizations have realized that the success of their mobile applications is directly tied to their ability to connect with their audiences. If your business falls into this category, you need to do what so many other companies have done and find a way to build scalable, flexible apps capable of reaching a large audience.

The general way to reach that wide segment of users, while keeping development costs low and not overtaxing your teams by forcing parallel development streams, is to use a multi-platform framework. Your choice of such a framework can determine whether your app development efforts reach their potential.

Organizations embracing this multi-platform model are increasingly using Flutter as their platform of choice. This flexible framework from Google is relatively recent, but its thriving community, simplicity of use, and ability to create powerful user experiences have already made it popular.

You can learn from the approach practiced by both internal development teams and dedicated developers at Very Good Ventures: putting Flutter to work, studying the ways experts use the platform, and discovering how it may fit into your own development workflows.



Why Are Dev Teams Choosing Flutter?

Before developers select a framework for their application building efforts, they have to settle on their overall methodology. Choosing multi-platform over native development is driven by the simple desire to have a single, unified development team for both Android and iOS versions of apps, as well as web applications.

Today's multi-platform frameworks are powerful enough that developers don't have to worry about inadequate performance or a lack of quality for their apps. When choosing which of these multi-platform ecosystems to work within, developers are increasingly turning to Flutter.

Director of Product and UX for Flutter & Dart Tim Sneath has announced that over 1,000 new Flutter apps are published every day, with the package ecosystem consisting of more than 25,000 packages.¹ This is a level of mainstream adoption that shows developers trust Flutter and shows that many of the most pressing issues have already been anticipated and resolved.

The Flutter ecosystem now hosts a thriving network of users. One of those 25,000 packages, developed by a fellow Flutter developer, may save your team considerable manual work. Best practices conceived by your team may, in turn, become useful packages for future creators.

¹<https://medium.com/flutter/announcing-flutter-3-3-at-flutter-vikings-6f213e068793>



The companies building apps with Flutter include some of the biggest names in their respective industries, such as:²

- **Toyota:** The automaker is using Flutter development for its in-vehicle technology, embracing the Flutter toolset for fast iteration and to include advanced features in the displays, such as smooth-touch mechanics.³
- **Betterment:** By moving an existing native app to Flutter, Betterment has provided an exciting use case for the platform's add-to-app capability. The app, which allows customers to invest from mobile devices, is core to Betterment's business.⁴
- **Nubank:** In the three years since switching mobile development to Flutter, Nubank has seamlessly integrated this framework into its development workflows, improving its practices as well as its codebase.⁵
- **BMW:** By using Flutter to develop its My BMW App, BMW enabled drivers to interact with their cars' tech systems from a consistent interface, across both Android and iOS. The company's dev framework automatically builds, tests, and deploys 96 variants daily.⁶
- **Alibaba:** Alibaba needed massive scalability and a consistent, user-friendly experience across platforms while building out the largest used-goods marketplace in the China market. Flutter's ability to push high performance across both Android and iOS has proven pivotal.⁷



¹<https://verygood.ventures/blog/top-companies-using-flutter>

²<https://flutter.dev/showcase/toyota>

³<https://verygood.ventures/success-stories/betterment>

⁴<https://flutter.dev/showcase/nubank>

⁵<https://flutter.dev/showcase/bmw>

⁶<https://flutter.dev/showcase/alibaba-group>

These examples show off some of the specific reasons why development teams are uniting around Flutter, but there are also a few general, overarching concepts that have helped the framework stake its place. These include:

- **Cost-effectiveness through smart resource use:** The value of having one development team instead of separate Android and iOS teams cannot be overstated. Rather than putting in twice the work to create parallel codebases that must be reconciled, companies are concentrating their resources on unified, efficient, and effective development teams.
- **Development team efficiency and quality of life:** Development efficiency isn't just about maintaining a single codebase. There's also the dev interface itself. Flutter's user-friendly feature set enables fast iteration, easy bug fixes, and the ability to onboard and reassign team members quickly. Furthermore, teams can create easily reusable code to boost their future efficiency.
- **Organizational alignment with overall objectives:** The development team can be a more seamless part of the whole organization when their workflows are simplified. Design, strategy, product, and operations teams find it easy to collaborate with developers who have a single codebase. The efficiency of Flutter extends to the whole organization, even in terms of learning from customers and responding to their feedback.

It's easy to see how such a development process can be a major point of differentiation for a company. Businesses are delivering app updates on a faster and more responsive cadence, all without running into issues such as over-complex codebases and excessive tech debt.

Perhaps the simplest and most compelling selling point for Flutter is that it lets organizations build the mission-critical apps they need, quickly, cost-effectively and without sacrificing performance or user experience. The resulting apps are the software solutions that connect them with their audiences, no matter how ambitious their objectives become.



Building Scalable Enterprise Apps with Flutter

“**Scalable**” is the key term in modern application development. Organizations may scale up in several ways. This includes direct efforts such as adding new features, as well as internal changes, like shifting additional developers to work on a project.

The first kind of scalability is challenging to get right — if an application has flawed or brittle code, a new feature could introduce unexpected bugs or inefficiencies to the user experience. By focusing on sound app architecture, as well as extensive testing and the use of predictable, standardized code, companies can achieve this type of scale.

As far as scaling up development efforts, there’s also room for error. Development efforts that have become too complex or are mired in tech debt may slow down when trying to get a new designer or developer up to speed on the project. Businesses that embrace configuration best practices, consistent architecture, and elegant code don’t tend to encounter these problems.

The Very Good Ventures method of building apps uses the Flutter framework to accomplish both types of scalability. By focusing on architecture, testing, configuration, automation, and the use of the rich Flutter ecosystem, this collection of best practices unlocks positive outcomes for businesses’ development teams.





The Importance of Positive App Outcomes

Before taking an in-depth look at the best practices of Flutter application development, it's worth stating the value of applications that are created according to these standards.

Organizations that cannot achieve stable, scalable codebases are setting themselves up for challenges in the future as they attempt to maintain and grow their applications. Those that do follow effective Flutter development practices can end up reaping the rewards, such as:

- **Simple maintenance over time:**
When organizations find themselves spending too much time, money, and effort on maintaining their applications, everything else becomes difficult. On the other hand, if maintenance is simplified by automation, elegant code and consistency in architecture can put their time and effort into value-adding work, not just upkeep.

- **The ability to add new features on demand:**
How fast and easy is it to expand capabilities and add a new feature? In today's app-dominated business climate, this question is key to a company's success. Apps built according to best practices are simple to expand and develop further, helping businesses respond to audience demand.
- **Confidence in the quality of user experience:**
Many applications have become the faces of their respective businesses. Can those organizations be confident that they're creating a consistent, positive user experience backed by rock-solid code that's highly stable? If they're using best practices, including clear documentation, consistency, and a lack of custom code, they can.

Organizations aiming to unlock this kind of reliable, trustworthy app performance can achieve it, provided they're using the right tools and techniques.

Flutter App Development, the Very Good Ventures Way

The app development method used by Very Good Ventures consists of five general areas, each contributing to a solid overall approach to creating scalable enterprise applications that will help companies reach their specific goals. These Flutter practices are applicable to all sorts of use cases, helping development teams overcome their specific challenges.



Architecture

Consistency is the keyword in application architecture. When there's consistency in each feature, developers can get started building out new capabilities quickly and without hesitation — this gives them more time to spend on creating unique features, as opposed to performing repeatable actions.

The Very Good Ventures team uses a layered approach to architecture, which consists of:⁸

- **Data layer:** Used for interacting with APIs.
- **Domain layer:** Applies business rules to one or more data sources from the data layer.
- **Business logic layer:** Part of the application layer, contains feature-specific logic and integrates with the output of the domain layer.
- **Presentation layer:** Renders UI components according to the output of the business logic layer. This layer also handles user interaction.

While a company's exact division of layers may differ, the key concept is to keep responsibilities separate — functions in each layer stay in that layer. This in turn makes for easy-to-understand, "boring" code, perfect for error tracking and bug fixes.

Having a **consistent, well-understood state management solution** is another useful practice for architecture. This, too, is a decision that each development team will make for itself, but the Very Good Ventures team defaults to `flutter_bloc`, due to its familiarity, simplicity, testability, and documentation.⁹

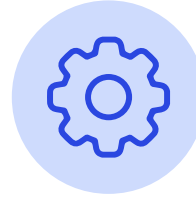


Testing

Tests can help catch bugs and most importantly, ensure that code works as expected, that is according to the specifications set by the team. Tests can also help ensure code continues to work over a long period of time, and even across multiple refactors. Developers hoping to create optimized applications in Flutter should aim for high levels of test coverage. Very Good Ventures aims for a standard of 100% test coverage.¹⁰ While not possible in every case, shooting for such a high level of coverage adds the assurance that each line of code has been tested at least once.

Developers should stick to their threshold of automated test coverage, whatever it may be. Organizations can start at a rate below 100%, for instance 80%, and then either stay consistent or raise their coverage rate with each test until they reach 100%.

Automated testing should not be the only quality control method for your application. **Human quality assurance remains an essential part of development.**



Configuration

In keeping with the theme of standardization and simplicity, the Very Good Ventures team practices and encourages the use of standard Git processes in app development. This means making small, efficient pull requests and using PR templates and peer code review.

As for keeping development, staging, and production environments separated and discrete, the team uses build flavors. The flavors functionality allows developers to work on different builds of the application. For example, you may have a development flavor on your device to track the latest changes to your app, whereas a production flavor is the latest stable version of your app (likely the one you've published to the app store).¹¹

Continuous integration and deployment are also part of the picture when it comes to app configuration the Very Good Ventures way. Having a solid CI/CD framework is essential for everything from automated testing to the efficient but low-risk deployment of new features.

¹⁰ <https://verygood.ventures/blog/flutter-testing-resources>⁹ <https://verygood.ventures/blog/why-we-use-flutter-bloc>

¹¹ <https://developer.android.com/studio/build/build-variants>



Automation

The automation of repeatable features is a general development best practice. The more that developers can automate manual functions, the more time they have to spend on more creative, value-adding development work.

One of Very Good Ventures' many open source tools, Very Good CLI, offers templates that help developers (ours and others) to implement widespread automation.¹² Users can create a ready-made skeleton for a:

- **Flutter starter app**
- **Flutter feature**
- **Flutter package**
- **Dart package**
- **Dart command-line interface**

Those skeletons have Very Good Ventures' best practices built in, helping the development process keep moving smoothly with less manual effort, and all the time savings that come along.

The use of GitHub Actions and workflows is another valuable way to automate tasks, saving time and taking away the risk of human error.¹³

Very Good Ventures doesn't just recommend using these actions, **our team has created a variety of them** that any development team can use to enhance their development practices.¹⁴

¹²https://github.com/VeryGoodOpenSource/very_good_cli

¹³<https://docs.github.com/en/actions/using-workflows>

¹⁴<https://verygood.ventures/blog/github-actions-for-flutter-apps>



Ecosystem

Working within an established, popular, and well-documented ecosystem provides its own kind of advantage for developers. When there are numerous packages available, each one represents a potential shortcut to effective application development workflows and best practices.

Despite being a relatively new technology platform, Flutter has **over 25,000 packages in its ecosystem**. This means that no matter what kind of development requirements a team is facing, there is likely a package out there that can provide a speed boost to Flutter-based efforts.

Whether created by our internal teams or the Flutter community at large, the Very Good Ventures development method includes numerous useful and important packages.¹⁵ When assessing a new package for development use, it's important to read the documentation and determine that the tool is right for the job – the best packages come with plenty of information on their ideal usage.



¹⁵ <https://verygood.ventures/blog/top-flutter-dart-packages>

Accelerate Flutter App Development with Very Good Ventures

When it's time for your business to select a partner for Flutter development, Very Good Ventures is the No. 1 partner. This has been true since the beginning of the platform's rise — our team at Very Good Ventures coded the first commercially available Flutter app outside of Google.

Our in-house team members at Very Good Ventures have built an industry-leading store of experience and expertise with the Flutter platform. Three of these employees were on the team that switched an automaker's applications to Flutter, back when the framework was still in alpha form.

In development, pioneers are often the ones who discover mistakes by making them, and then develop efficient and elegant new workflows with what they've learned. This is the role Very Good Ventures has played in the Flutter ecosystem, with our experts taking on every challenge so your development team doesn't have to.

If your team needs assistance with a new Flutter development process, or in establishing best practices for your organization as a whole, you can turn to our team. It's easy to see what types of best practices, tech tools, and philosophical concepts go into the Very Good Ventures methodology because they're reflected in the wide variety of [open source](#) tools and [demo projects](#) we've created.

The overall commitment to creating open source tools and being an active part of the thriving Flutter ecosystem shows confidence in the platform that your organization can take advantage of. This rings true whether you're engaging directly with us at Very Good Ventures, using one of our open source tools — or anything in between.

Whatever your level of engagement, as long as your team follows best practices and reliable development methodologies, Flutter can be the ideal platform for building scalable applications. In an era when apps and their respective organizations are tied more closely than ever before, this may have powerful repercussions for your future.





**VERY
GOOD
VENTURES**

Sources

<https://medium.com/flutter/announcing-flutter-3-3-at-flutter-vikings-6f213e068793> | <https://verygood.ventures/blog/top-companies-using-flutter>
<https://flutter.dev/showcase/toyota> | <https://flutter.dev/showcase/nubank> | <https://verygood.ventures/success-stories/betterment>
<https://flutter.dev/showcase/bmw> | <https://flutter.dev/showcase/alibaba-group> | <https://verygood.ventures/blog/why-we-use-flutter-bloc>
<https://verygood.ventures/blog/flutter-testing-resources> | <https://developer.android.com/studio/build/build-variants>
https://github.com/VeryGoodOpenSource/very_good_cli | <https://docs.github.com/en/actions/using-workflows>
<https://verygood.ventures/blog/github-actions-for-flutter-apps> | <https://verygood.ventures/blog/top-flutter-dart-packages>