SUSAN GOSSELIN
CORPORATE WRITER

CONTENTS:

PROCODERS BLOG

REACT NATIVE VS. FLUTTER

DATE: AUGUST 12, 2020

VERSION: #2

SUSAN GOSSELIN
CORPORATE WRITER

<H1> FLUTTER VS REACT NATIVE—WHICH ONE IS BETTER FOR YOUR NEXT PROJECT?<H1/>

<H2>BUILDING A CROSS-PLATFORM MOBILE APP? WE'LL DISCUSS THE DIFFERENCE BETWEEN REACT NATIVE AND FLUTTER, THEIR PROS AND CONS, AND HOW TO PUT THEIR STRENGTHS TO WORK FOR BUSINESS. <H2/>

In the tech world, it's all too easy to jump on the latest trends, especially when new frameworks roll out with new pre-coded modules and promises of lightning-fast speed. Clients come to ProCoders feeling a little overwhelmed by their choices.

Many are struggling with a decision—how to compare Flutter and React Native as the framework for their next project. Both are cross-platform mobile frameworks with tremendous functionality, robust user communities, and an enormous amount of pre-programmed modules that can supercharge your development speed. But they both have their own advantages and disadvantages, and understanding the differences between Flutter and React Native can save a substantial amount of development headaches.

Read on, for all the important information a company will need to make a decision in the Flutter vs. React Native debate.

<H2>WHAT IS FLUTTER?<H2/>

The Flutter framework was developed by Google in 2017. It is designed to be a cross-platform app development framework that can be expressed on any mobile device, or tie into your existing web applications. It is considered one of the leading open source frameworks available today, and considering that it's created and supported by Google, it's likely to only increase in popularity.

Its advantages are many, including a complete development ecosystem, hot reload functionality, and the fact that it's opensource and free to use. Yet, even with all this, it has disadvantages, including its large size, and reliance on native tools and technologies when you develop in the framework.

Even so, Flutter remains a favored choice, especially of clients who have large scale platforms, such as Alibaba, Hookle, Topline, OfflinePal, Hamilton, etc. The community of developers using it has grown through 2020, and with its speedy rendering and other benefits, many say it will overtake React Native's popularity in the future.

PROCODERS BLOG

REACT NATIVE VS. FLUTTER

DATE: AUGUST 12, 2020
VERSION: #2

SUSAN GOSSELIN
CORPORATE WRITER

<H2> WHAT IS REACT NATIVE?<H2/>

Developed by Facebook in 2015, React Native is a well-used, well-loved cross-platform app development technology that at present, is one of the most popular frameworks in the world. It's achieved widespread adoption because it allows developers to use React and JavaScript side by side with many other native platform abilities to build mobile apps.

This highly reliable, open-source framework offers easy implementation of Native UI components, hot reloading, platform specific code and one of the largest UI support communities in the world. Developers love the ease of its dozens of plug-and-play program modules.

With that being said, there are some significant disadvantages of React Native. Specifically, it has a more complex UI and the navigation isn't always that seamless. However, it is still a favorite of many important online companies—like Facebook, Skype, Artsy, Vogue, Bloomberg, and Tesla, etc. —who have built their entire online systems on the backbone of React Native. This is a trend we don't expect to slow down anytime soon.

<H2>FLUTTER VS. REACT NATIVE: A CLOSER LOOK<H2/>

What to choose? Flutter or React Native? First, let's be clear. At ProCoders, it's not uncommon for us to recommend either framework, as they each have their advantages and disadvantages depending on the project parameters. But to understand which one of these options is the right one, it's best to study the features, and not the trends. Let's dig in.

<H2> PROGRAMMING LANGUAGE<H2/>

<H3>FLUTTER: DART<H3/>

Flutter uses the Dark programming language, which was first introduced by Google in 2011. Many say that programming done in Dart is a little faster to run than React Native. And its syntax is elegant, overall. But it uses a number of object-oriented concepts that have to be learned. Long story short, there simply aren't as many developers who know the Dart language as React Native. So, when we compare Flutter to React Native on ease of programming language use, React Native still gets the game point here.

<H3>REACT NATIVE: JAVASCRIPT<H3/>

React Native uses JavaScript, a nearly universally used programming language that nearly every developer and website designer knows well. Its programming community and online education resources are extensive, so if there's something you don't know, there's almost always a tutorial on that. And React Native's pre-built modules can save you a lot of time and money, too. When our developers are asked will Flutter replace React Native, they usually say no, for this reason. React Native has a large, enthusiastic and well-supported user base. And that's a trend not likely to change soon.

PROCODERS BLOG

REACT NATIVE VS. FLUTTER

DATE: AUGUST 12, 2020
VERSION: #2

SUSAN GOSSELIN
CORPORATE WRITER

<H2>ARCHITECTURE<H2/>

<H3> FLUTTER: SKIA<H3/>

Flutter uses Skia, a 2D graphics rendering library, and the Dart language VM in a platform-specific shell. Because it uses the Dart language, Flutter compiles the source code ahead of time to native code. This point is critical. Flutter's use of native code throughout most of its service means that it is fast, and elegant, integrating seamlessly with iOS or Android. There's no awkward programming bridges needed to make things work. And the end result can be an app that is faster and cleaner than competitiors. To compare React Native and Flutter on this point alone, Flutter is the clear winner for the elegance of its design.

<H3>REACT NATIVE: FLUX<H3/>

Because React Native is based on JavaScript, it relies on the JS runtime environment architecture, which most of us know as JavaScript bridge. It also uses the Flux architecture from Facebook. Because the JavaScript bridge is needed, this inherently makes app design go a little slower that Dart, and that's a key difference between Flutter and React Native.

<H2>INSTALLATION<H2/>

<H3>FLUTTER: BINARY DOWNLOAD FROM SOURCE<H3/>

To install Flutter, you must download the binary for the specific platform you choose from GitHub. If you're installing for MacOS, there's even an additional step here—an additional flutter.zip file that has to be downloaded and added as a PATH variable. This is a significant extra step. So, if you're looking for the differences between Flutter and React Native, this is an important one. Comparatively speaking, Flutter is trickier to install, and will take significantly longer.

<H3>REACT NATIVE NPM<H3/>

When it comes to installation, React Native is easy to love. Installing this framework is seamless using the node package manager. For developers having JavaScript knowledge, installation Is an easy process. It's important to note that both Flutter and React Native lack one-line installation, and a native package manager for OS. Yet, Flutter installation calls for one additional step. So in the comparison between React Native vs. Flutter, we'll choose React Native every time.

SUSAN GOSSELIN
CORPORATE WRITER

<H2> UI AND DEVELOPMENT API<H2/>

<H3>FLUTTER<H3/>

Flutter is a framework that relies primarily on its own customized development language. And that creates a stark difference in The UI development experience between Flutter or React Native. Flutter functions using proprietary widgets that are 100 percent customized. And this may very well be Flutter's key advantage overreact Native, because the widgets are automatically consistent with Material Design for Google and Cupertino for Apple. No third-party apps or custom components required. When people ask the question will Flutter replace React Native, this is one of the biggest reasons why.

<H3>REACT NATIVE<H3/>

React Native taps into native Android and iOS components, and it does it with the help a third party or custom components. The result can often be a patchwork affect in the code. However, it's important not to make too much of this, because React Native's JavaScript development language does offer its own set of advantages. However, in a comparison of which is better, Flutter or React Native, Flutter definitely wins the day.

<H2>DEVELOPMENT TIME<H2/>

<H3>FLUTTER<H3/>

Which framework is fastest to develop on? Which is better Flutter or React Native? Between the two, Flutter has a real disadvantage here. The Dart language is not as widely used or understood by developers. While it does have a hot reload feature, it lacks support of many text editors and IDEs, which adds more time to the development process.

<H3>REACT NATIVE<H3/>

The difference between Flutter and React Native is JavaScript. The framework's reliance on JavaScript makes development easy and accessible for more web developers and web designers. Even better, developers are free to use any IDE or text editor of their choice. React Native is more mature and has more developer support available for when It's needed. When it comes to development speed, there really is no comparison. React Native is clearly superior.

PROCODERS BLOG

REACT NATIVE VS. FLUTTER

DATE: AUGUST 12, 2020
VERSION: #2

SUSAN GOSSELIN
CORPORATE WRITER

<H2>CODE REUSABILITY<H2/>

<H3>FLUTTER<H3/>

In any comparison between React Native and Flutter, the issue of code reusability inevitably gets mentioned. Here, Flutter comes out as a clear winner. In Dart, Flutter's code base is a lot more reusable. Developers can simply define a single UI widget tree and then reuse the defined logic. That means not a lot of differentiation has to be done. Thanks to Google Flutter has an elegant interface and those details play out in benefits like this one.

<H3>REACT NATIVE<H3/>

In matters of re usability, there is a clear difference between React Native and Flutter. And in this case, using React Native does add to your development speed. Why? Because React Native isn't always compatible. In general, React Native will allow you to write a code once and apply it to every platform. However, developers have to be careful, because compatibility on every type of mobile app platform isn't always guaranteed. Developers will have to look into what platform they're running on and load a different set of components that work. That means modifications have to be made on occasion, adding to your development time. Code reusability is an area that in the future we hope React Native will address.

<H2>QUALITY ASSURANCE<H2/>

<H3>FLUTTER<H3/>

Like other programs developed by Google, Flutter offers many integrated quality-control testing features. Want to test an individual widget? You can. Want to run Integration tests? No problem. All these capabilities come on board with Flutter's native framework. And even better, Flutter also provides lots of detailed testing documentation. In this respect, we believe Flutter's framework is more in keeping with the trends in 2020 and is generally more elegant.

<H3>REACT NATIVE<H3/>

As a JavaScript-based framework, React Native doesn't have any official integrated testing features, and that's a significant drawback. It only comes with a few unit tests. That said, developers can generally fill in this gap by using third party frameworks like Jest and Detox. It's an add-on, but not that big of a problem to integrate with these additional resources. So in a quality assurance comparison between React Native and Flutter, it's really a bit of a draw. Both work equally well in their own way.

PROCODERS BLOG

REACT NATIVE VS. FLUTTER

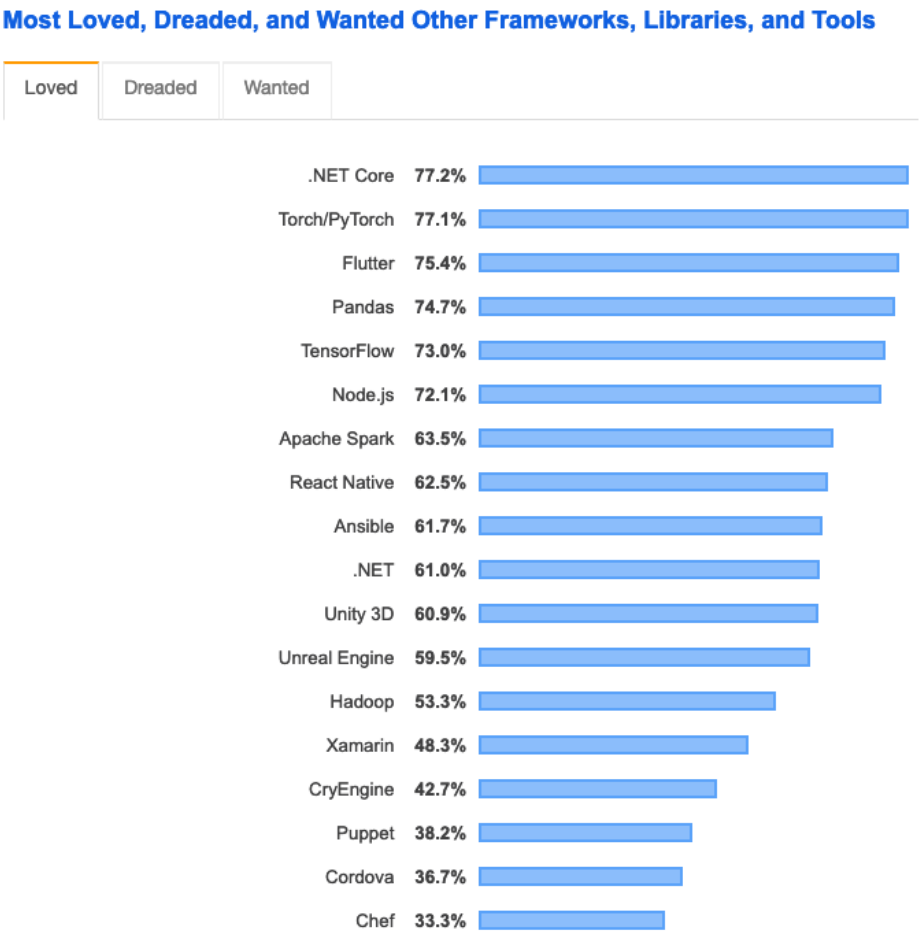
DATE: AUGUST 12, 2020
VERSION: #2

SUSAN GOSSELIN
CORPORATE WRITER

<H2>ECOSYSTEM AND COMMUNITY SUPPORT<H2/>

<H3>FLUTTER<H3/>

If you judged a framework based on its ecosystem alone, Flutter would be the clear loser in a comparison between React Native and Flutter. But just because Flutter doesn't have the traction React Native does in 2020, doesn't mean it won't have it in the future. The community of developers using Flutter is growing fast. According to Stack Overflow's Developer's Survey of 2019, in the most loved/wanted frameworks category, 75.4 percent of people showed interest in the Flutter framework. When asked about React Native, 62.5 percent of developers showed interest. With the continued investment from Google Flutter will be a framework trend that will continue to grow.



<H3>REACT NATIVE<H3/>

On the market since 2015, React Native gained popularity for the ease of its JavaScript programming. Now, it has a developer ecosystem of more than 10,000, an enormous online database of tutorials, and React Native developer conferences, held all over the world. It's not just a framework, it's a tribe. That's a considerable difference between React Native and Flutter. So, the advantage goes to React Native in this comparison. For now.

SUSAN GOSSELIN
CORPORATE WRITER

<H2>DEVOPS SUPPORT<H2/>

<H3>FLUTTER<H3/>

When evaluating DevOps in Flutter or React Native, most people would agree that React Native has the advantage. However, what Flutter lacks in community support it gains in CI and CD official documentation. Flutter has an entire section on this documentation including several links making the addition of CI/CD for flutter apps an extremely easy process. While this is no substitution for deep developer community support, it is a good start. In the future, Flutter usage is likely to grow.

<H3>REACT NATIVE<H3/>

React Native's huge developer community is well supported by Facebook. It has enormous third-party libraries, help guides, third-party support and more. React Native is the clear winner in Dev/Ops support, hands down. And developers enjoy using it. For those who wonder—will Flutter replace React Native? The simplest answer is maybe. But the framework will have to equal React Native's global developer support community first.

<H2>PERFORMANCE<H2/>

<H3>FLUTTER<H3/3>

In a Flutter vs. React Native performance comparison, Flutter comes out on top. But this answer is not as clear cut as it appears. For the record, Flutter runs faster than React Native. It does not have the JavaScript bridges for initiating interactions that are the hallmarks of React Native. Because of this, the speed of development and running time gets expedited drastically. Flutter has also set its animation standard at 60 FPS—which gives proof to its quick-running performance. And the fact that it is compiled directly into the native ARM code for both Android an iOS surely doesn't hurt Flutter performance and speed, either. But even though it is faster, it is also a substantially bigger program with more native components. So, it's a point that must be weighed before you proceed. Fortunately, Flutter provides a wealth of documentation on how to reduce file size.

<H3>REACT NATIVE<H3/>

Again, the Flutter vs. React Native performance comparison will always come out in favor of Flutter, due to the elegance and power of its native design. It's one of the key reasons many developers are now considering it a future-forward trend to develop in Flutter. The JavaScript bridges to its back code and plug-and-play modules do slow down the running of React Native. But it still doesn't make React Native apps slow. Both are excellent programs in this regard, or they wouldn't be a tool we use for developing our client's websites, at all.

PROCODERS BLOG

REACT NATIVE VS. FLUTTER

DATE: AUGUST 12, 2020
VERSION: #2

SUSAN GOSSELIN
CORPORATE WRITER

<H2>RELEASE<H2/>

<H3>FLUTTER<H3/>

To compare Flutter to React Native on Release protocols, Flutter wins every time. Why? Because Flutter has a seamless release process that will make it easy to get an app into the App Store or Google Play. Flutter has official documentation for both fast Lane deployment and releasing to iOS Android. And recently Bitrise announced out-of-the-box CI support for Flutter. Better yet, Flutter apps can be deployed from the command line itself.

<H3>REACT NATIVE<H3/>

In React Native's defense, its release process is a fairly standard one. The only problem is, it's entirely manual. Any automatic deployment that is done requires third-party tools. That makes the release process considerably more labor intensive, and that slows down development speed. In a React Native vs. Flutter comparison on this point, React Native has a long way to go to catch up.

<H2>DOCUMENTATION<H2/>

<H3>FLUTTER<H3/>

A key difference between Flutter and React Native is its documentation. With Dart Flutter has invested a great deal in its native documentation, and it shows. It's easy to love their documentation guides, which makes it easy to transition from another framework, or optimize a current project. With Flutter, expect many helpful graphics and videos that are available directly within the framework.

<H3>REACT NATIVE<H3/>

With the global community as large as React Native's, the framework must have extensive documentation. And it does. But scratch below the surface and React Native's onboard documentation is spotty and poorly done. In fact, to get the kind of documentation most projects need, developers often end up going to third-party sources. It's a significant weakness of the framework. So, to compare Flutter and React Native on documentation, Flutter truly will win every time.

PROCODERS BLOG

REACT NATIVE VS. FLUTTER

DATE: AUGUST 12, 2020
VERSION: #2

SUSAN GOSSELIN
CORPORATE WRITER

<H2>COMPARISON BETWEEN REACT NATIVE AND FLUTTER<H2/>

In this ProCoder's review of the difference between Flutter and React Native, the comparison has been made point by point, category by category. But in an overall comparison between React Native and Flutter, which framework wins? The answer is—It depends.

Flutter wins every time on framework design and performance metrics. On usability, though? It's a bit of a toss-up. For developers comfortable using Dart, then Flutter provides a clean, fast-running, elegant and native option for your app development. Developers preferring the ease of JavaScript and the tremendous amount of plug-ins and widgets and modules it offers, then React Native is the answer, even if it does run a little slower.

And if a developer community and wealth of third-party supports is important to a client's development process, then React Native gets the recommendation.

Don't forget, any of the decision points discussed here could also be thrown out the window by a project's hardware requirements. Before starting, take a hard look at hardware-specific APIs as they may make or break the app. How well those APIs can be accommodated by Flutter or React Native will depend completely on your project.

<H2>THE FUTURE OF FLUTTER AND REACT NATIVE<H2/>

2020 has seen a lot of improvements in both React Native and Flutter, largely due to improvements made to the frameworks in 2019.

For instance, Flutter released its version 1.12 in 2019, which included features such as:

- Android X Support—so developers working in Flutter could update their designed Android apps without compromising on backward compatibility features.
- OpenType Rich Typography Features—including tabular and old-style numbers, stylistic sets, and more.
- Flutter for Web—which expands the framework to Chromebooks, Windows and Mac, and makes designing desktop apps with Flutter a reality.
- Dart 2.7
- Beta Web Support
- Support for iOS 13 Dark Mode, and Mac OS Desktop
- And much more.

As for React Native, they have plenty of news to share, as well. In their upgrade to 16.9, they've added improvements, including:

- Fast Refresh, which is a combination of their hot reloading features
- Fixes for frameworks! CocoaPods support
- Added Window Dimensions Hook
- They've also announced a push to make APIs stable, create a better GitHub Repository, improve documentation, clean up their surface area, and add support tools in the open source community.

SUSAN GOSSELIN
CORPORATE WRITER

<H2>THE FINAL VERDICT<H2/>

So, the final verdict is, when it comes to Flutter vs. React Native, or React Native vs. Flutter, there is no universal “better” framework. The framework to choose will depend completely on the project’s parameters, and the business needs surrounding it.

Speaking only in the most general of terms, Flutter is better for bigger apps and apps that need native programming. React Native is great for apps that can benefit from its plug-and-play modules and large developer community.

There are some specific “fit” issues to consider, as well. For instance, if a project needs 3D, Flutter doesn’t support that. Similarly, Flutter should be avoided if the app design is platform-specific, requires multiple interactions with an OS, or requires little-known native libraries.

React Native also has its particular disadvantages. You should avoid React Native if your app needs to handle less common, or very specific tasks (like calculations) in the background. Also, avoid React Native if a project requires custom communication via Bluetooth, or the plan is to create an app for Android only.

As you can see, there’s a lot to consider when you make a comparison between Flutter and React Native. Our project managers at ProCoders would love to help you sort it all out. Simply contact ProCoders at xxxxxxxxxxxxx! We’ll be happy to set up a free consultation!