

# Flutter vs React Native



## Which is better for your business?

By Patryk Mierzejewski, iOS & React Native Developer

---

**We know that React Native and Flutter are currently top when it comes to mobile cross-platform development frameworks**, and as we have experts in both, we often get asked which are better for business. Whilst there is no easy answer to this question, we are well equipped to give you a summary of the benefits and disadvantages of each. We hope that you'll find the below article informative and useful.

### THIS ARTICLE IS FOR YOU IF:

- You're looking to develop a new mobile project and are not sure which framework to use
- You want to figure out which solution will offer the fastest time to market
- You want to understand which framework is likely to be most stable and user-friendly
- You ultimately want to know which solution offers the best return on investment

### WHY 10CLOUDS?

- We have been recognised for our achievements
- We have a reputation for being one of the best mobile teams in design communities, such as Dribbble and Behance.
- We're a one-stop-shop for everything you need
- We have experts in front-end and back-end development, design, branding and app-maintenance.
- We create tailor-made applications for both iOS and Android
- We develop apps with platform-specific functionalities and guidelines in mind to provide the best possible experience for both iOS and Android users.

## 1. A QUICK OVERVIEW OF BOTH FRAMEWORKS:

### React Native

React Native is an open-source mobile application framework created by Facebook. It is used to develop applications for Android, iOS. It uses single JavaScript codebase and translates it to native views.

### Flutter

React Native is an open-source mobile application framework created by Facebook. It is used to develop applications for Android, iOS. It uses single JavaScript codebase and translates it to native views.

## 2. TIME TO MARKET

### React Native

React Native is a cross-platform solution, so one codebase can be run on both Android and iOS. This saves a lot of development time - It's safe to assume that writing one codebase for both platforms will take at least 25% less time than writing separate projects for both platforms. Many developers believe that writing React Native apps is easier than writing native applications. When your application is more focused on advanced(native) functionalities such as maps, bluetooth and camera, as opposed to animation heavy UI, then React Native is the right choice for you. It has been around on the market for longer, still has more support from third party libraries than Flutter, and it's way faster to use available solutions than build them from scratch.

## 3. USER EXPERIENCE AND STABILITY

### React Native

In this case single codebase translates into native views. That means the feeling is more native than in Flutter. But still if we want to be native-friendly we usually have to write components for each platform and only share part of it. On the other hand React Native mostly relies on components written by us or on third party libraries. This means that the native feeling might be lost in this process. In case of stability React Native is still officially in beta, the core parts of the framework are under constant improvements, which results in high maintenance costs during new releases.

## 4. RETURN ON INVESTMENT

### Both

The same codebase for both platforms allows you to release app updates simultaneously. This one is hard to achieve when there are separate projects for each. Different availability

### Flutter

Similarly to React Native, Flutter allows for one codebase to be run on both Android and iOS, thus saving roughly the same amount of development time. However, Flutter shines when it comes to custom and rich user interfaces. Because it uses its own drawing engine, we avoid unwanted system defaults that exist on React Native and are more expensive to customize. Also Flutter has multiple UI components built-in, so it makes UI development even faster.

### Flutter

With one codebase for both platforms, you will only need one UI design prepared by designers. The cross-platform application usually follows Material Design principles, with only small adjustments for different platforms (which Flutter usually handles by itself) or makes the whole UI look custom, with elements based on the best solutions taken from iOS and Android. Flutter also has built-in Cupertino style components which simulate iOS feel, but they are a minority. When we talk about stability it's worth mentioning that despite the fact that Flutter is officially a stable release, usually third party libraries that we would like to use, are not. And we have to take into account potential maintenance costs of those external libraries.

of developers, or longer implementation on one platform makes synchronizing difficult. The problem grows when an application update also requires changes in a backend infrastructure.

## React Native

Since it uses JavaScript as a language of choice there is a possibility to perform Over The Air Updates using Code Push. This is crucial when we want to push something to the users really quickly without waiting for reviews etc.

## Flutter

In business React Native is really popular, way more than Flutter, and because of this, developer time is often much more expensive. Using Flutter can therefore be a cost-effective way of getting better developers with the same budget.

### IN SUMMARY...

- Your choice of framework should really be driven by the type of app that you're building and the planned features.
- If your project relies mostly on rich UI and Animations, we recommend Flutter as it will be easier to write, faster to the market and cheaper to maintain.
- If it uses native components, such as maps, camera and bluetooth, we would recommend React Native - currently it still has better maintained libraries to handle those more advanced features, and that would give you significant advantage in time to market.
- If you want complete digital product which includes web and a mobile app with similar features, we would recommend React Native due to the possibility of sharing the code between two platforms.

### LOOKING TO CREATE YOUR OWN PROJECT IN REACT NATIVE OR FLUTTER?

Want to find out more about our work across a range of different frameworks? Why not drop our experts an email on [hello@10clouds.com](mailto:hello@10clouds.com) or visit our website: [www.10clouds.com](http://www.10clouds.com)