



Why React Native Code Trumps the iOS Native Application always & every time!



There are two routes that you can go with mobile app development. You can either choose to do it the platform native way or you can use the React Native. It may be argued that the approach - native versus React Native- depends on your particular use case. We, however, hold the opinion that React Native Code is a consistently better app development tool over the native approach on all dimensions.

Why, you ask?

Well, that is what we will be explore in our blog today. We will get to better understand how these app development methods work, how each performs across a variety of parameters, and why, in the end, React Native is the superior option in most scenarios.

How Does Native App Development Work?

An important thing to keep in mind is that if you are developing your app natively, you will have to consider the platform it is being designed for. This means that your approach to designing an app for an iOS device will be different than if you were working on an app for Android devices.

- **Native App Development on iOS Devices**

Swift programming language, which used to be known as Objective-C, is typically used for true native app development for iOS devices. Another option is to use Xcode IDE, which is an integrated development environment.

- **Native App Development on Android Devices**

For Android devices, the most commonly used language is Java which all of us are quite familiar with. Some individuals also use Kotlin or Android Studio IDE.



How Do You Go About Using React Native?

React Native is a framework provided by Facebook that helps build true native mobile applications using the React.js library. It uses native elements with the JSX syntax that helps embed XML with JavaScript. For those that have not used React Native, the coding style is akin to the way you would define and organize HTML components. Similarly, you can style your RN components using StyleSheet just like you would with CSS. The platform gives you access to general components along with platform specific elements which can manifest accordingly on iOS as well as Android devices. Some of the most widely used applications, including Instagram and Uber Eats, incorporate some portion of React Native development.

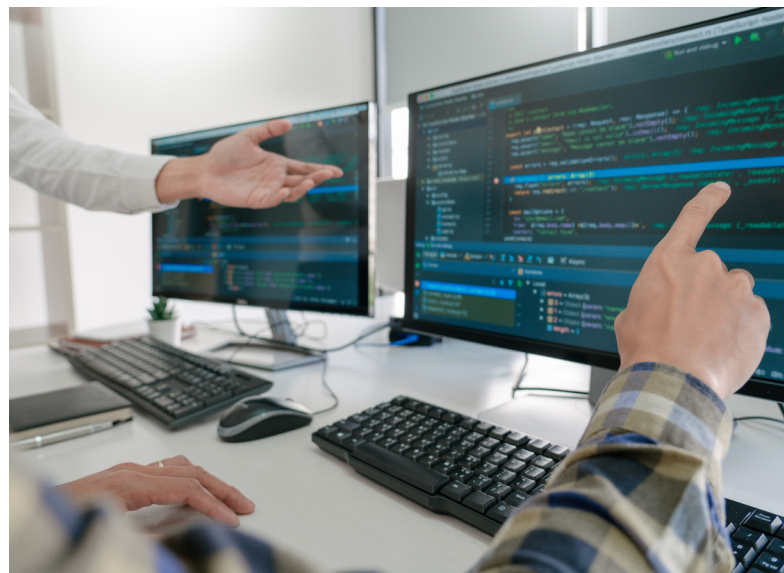
React Native vs. Native iOS App: Let's Judge

To get a better idea of how React Native trumps native app development, let's take an objective look at their potentials across some critical parameters:

Programming Language

As we mentioned before, React Native uses React.js which is written in JavaScript, a language most developers are highly familiar with. Native app development platforms, on the other hand, tend to use strongly typed, compiled languages like Java and Swift.

[Image of code written with React.js]



Cross Platform Deployment

The most obvious disadvantage of using native app development is that it limits your application to only one platform. Your code will either work on Android devices or iOS devices, not both. This means that if you want your app to work on both platforms, you will have to put in extra work to create two entirely different workflows, even if you want the two apps to look exactly the same.

When using React Native, you need not pay mind to whether your app will be used on iOS or Android devices. The code that you write will translate into a functional application for either of the platforms.

[Images of same code manifesting as similar looking apps on iOS and Android devices]



User Interface

In React Native, the source code can be translated into native components that are rendered differently on different platforms. Let's say you have written some code for a button that appears and behaves a certain way on iOS devices since it is native to them. If you are using React Native to write the code, the button will appear and behave the same way on the Android device as well. This allows you to build unique, interactive UIs without having to compromise on certain features and components.

Associated Costs

Most companies do not have the workforce or the budget to hire two different teams to build and look after two different applications. As such, using a solution like React Native proves to be much more cost effective. Even 30% of a shared codebase saves you a lot in terms of time and money.

Development Speed

Even if you are creating one workflow, it will still take you a considerably longer time working with the likes of Swift or Java than you would spend in React Native. With the latter, since you are only creating a single source code, it takes less time.

Access to APIs and 3rd Party Libraries

APIs enable your app to access other mobile platforms like the camera and you can find a lot of UI features in third party libraries.

If you do not have access to React Native, that does not mean it is all doom and gloom. You can still build decent, platform dedicated applications using native tools and technologies. One of the major benefits of building your app natively is that you have a lot of options when it comes to APIs and third party UI elements. Additionally, if you are looking for an easier, less complicated way out to building a unique user interface with very specific animations and graphics, then native languages are better to deal with.

Compared to native tools, React Native gives you access to a smaller berth of options and variations.

Maintenance and Bug Fixes

Maintenance is also a breeze with React Native considering you can make the fixes in one place and expect them to be applied across all platforms.



Which Option Should I Choose?

At the end of the day, even if you do not necessarily need cross platform deployment for your application, the multitude of other advantages of using React Native as opposed to native app development techniques really tips the hat in the former's favor. With a small team of expert developers, you can build intuitive, good looking mobile applications in quick time. Mobile app development is going to be an ever present part of your digital strategy- might as well make it easier and more convenient a process for yourself and your tech team.