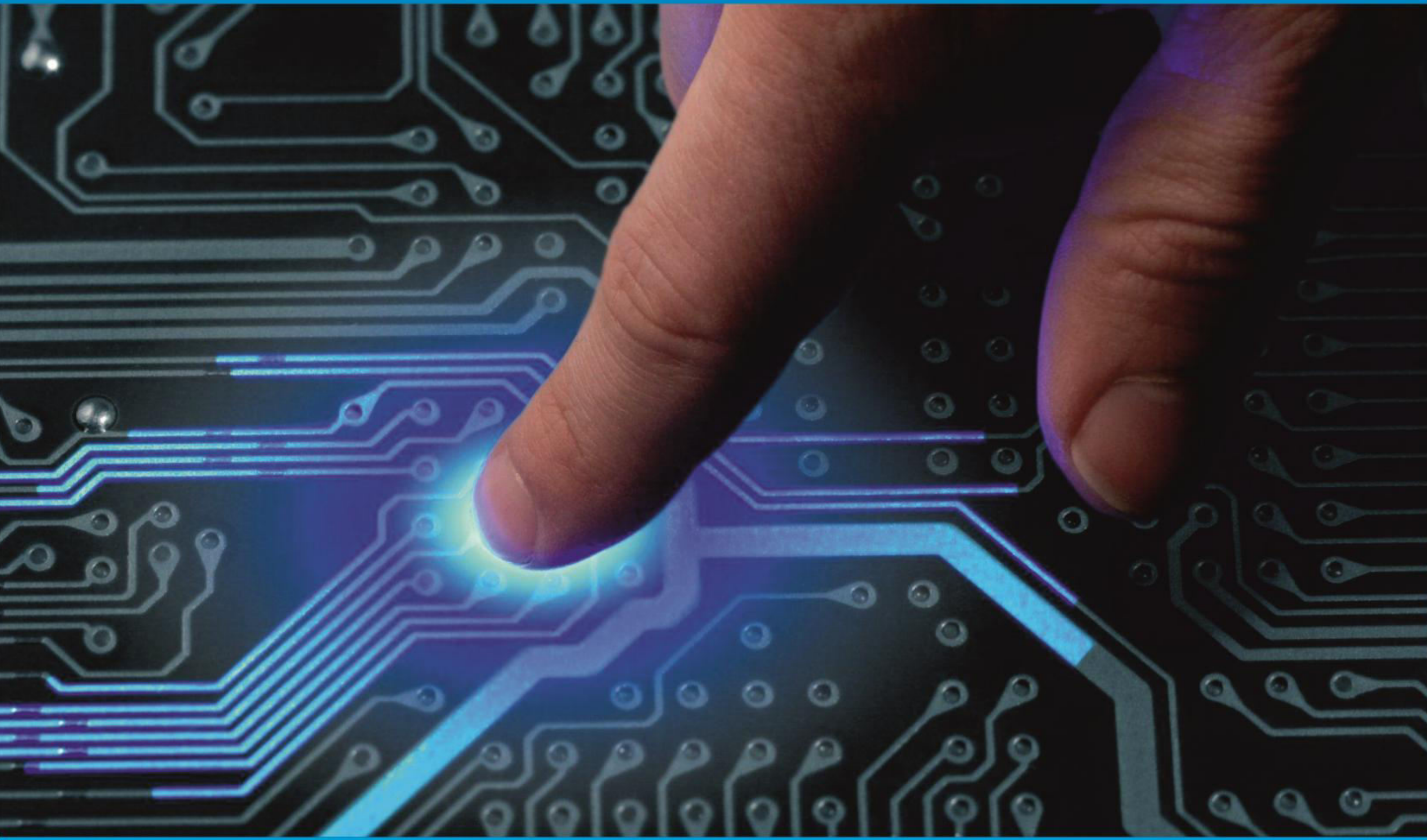
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 5, May 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

| DOI: 10.15680/IJIRCCE.2021.0905155|

A Study on React Native

Aarti Kothawade, Rama Bansode

P.G. Student, Department Master of Computer Application, Modern College of Engineering, Shivaji Nagar, Pune, India

Associate Professor, Department Master of Computer Application, Modern College of Engineering, Shivaji Nagar, Pune, India

ABSTRACT: React Native is created by Facebook which is an open-source mobile application framework. It is used in development of mobile applications for Android and iOS by enabling developers to use React alongside native platform capabilities. Using React Native we can develop natively rendering mobile applications for iOS and Android. Supported by React, by using Facebook's JavaScript library for building user interface, but it targets mobile platforms instead of targeting browser. Web developers can now develop mobile applications with comfort of a JavaScript library. Plus, because most of the codes you write are often shared between platforms, React Native makes it easy to simultaneously develop for both Android and iOS.

KEYWORDS: React Native, Why to use React Native, Importance of React Native

I. INTRODUCTION

React Native supports both iOS and Android. The overwhelming majority of the codes we write are going to be cross-platform. Many companies like Facebook, Palantir, and TaskRabbit are using it in their applications. The working principles of React Native are almost like React. It runs during a background process (which interprets the JavaScript written by the developers) directly on the end-device and communicates with the native platform via a serialization, asynchronous and batched Bridge. React components wrap existing native code and interact with native APIs via React's declarative UI and JavaScript. This permits native app development for whole new teams of developers, and may let existing native teams work much faster. React Native doesn't use HTML or CSS. Instead, messages from the JavaScript thread are used to manipulate native views. React Native also allows developers to write down native code in languages such as Java for Android and Objective-C or Swift for iOS which make it even more flexible.

So you would like to be ready to build mobile apps for both Android and iOS. What do you have to learn? The individual native languages for every app which is Java for Android and Swift/ Objective-C for iOS? Actually No. Native Android and iOS development are very different and may be expensive- first, the language itself is sort of different and second, all the underlying API's are different- the way of using the GPS is different, the thanks to create animation is different, the way you create network calls is different. React Native may be a JavaScript framework used for developing a true, native mobile application for iOS and Android. It uses only JavaScript to create cross platform mobile applications. It's like React, which uses native component instead of using web components as building blocks. React Native is predicted on React, which is a JavaScript library of Facebook for developing the interface. It targets the mobile platform instead of the browser.

II. HISTORY

In 2012 Mark Zuckerberg (CEO of Facebook) said, "The biggest mistake we made as an organization was betting too much on HTML against native." He also promised that Facebook will try to deliver a much better mobile experience.

In Facebook, Jorden Walke found a new way to get UI elements for iOS from a background JavaScript thread. They decided to organise an inside Hackathon to perfect this prototype soon be able to build native apps with this technology. After months of development, Facebook released the first version of React JavaScript Configuration in 2015. During a technical talk, Christopher Chedeau explained that Facebook is using React Native in production of Group App and their Ads Manager App.

III. WHY TO USE REACT NATIVE

- First we will see what is React Native
  React Native framework developed by Facebook for creating native-style apps for iOS and Android under one common language, JavaScript. Initially, Facebook developed React Native to support iOS. However with its recent support of the Android OS, the library can now render mobile UIs for both platforms.

- Now we will see why to use React Native
  Whenever there's an update for apps written in Swift/ Objective-C or Java, the entire app must be recompiled and a replacement version has got to be distributed to the App Store again. All this will take a couple of weeks depending on the App Store review process.
  To avoid this hassle, React Native apps add a special way; a native app is in a position to locate specific JavaScript code, which is later downloaded and compiled when the app is launched on an actual device.
  By this, updating the app is often done instantly without having to submit a replacement version to the App Store again and again.

IV. IMPORTANCE OF REACT NATIVE

- With React Native Framework, you will render UI for both iOS and Android platforms.
- It is an open source framework, which might be compatible with other platforms like Windows or other OS within the near future.
- Since React Native components have the counterpart rights, you will reuse these components for building both Android and iOS apps.
- React Native Development is relatively simple, quick and efficient.
- React Native can be a good choice for those developers who have enough experience in development using JavaScript, as there is nothing to learn Android specific Java or iOS's Swift.
- React Native may be a UI focused, which makes the app load quickly and provides a smoother feel.
- React Native framework has gained popularity in recent years and one of the reason is that the framework helps with the needs of both the platforms at one go and ends the confusion of selecting either iOS or Android development for mobile developers. Hence, big companies have shifted their eyeballs towards it.
- Moreover, using the React Native Framework fills the gaps between that specialize in a good marketplace and making the profit. The number of iOS users is less as compared to Android users and yet able to generate more profit, so many industries get into difficult situation whether they should focus on making more profit by developing iOS apps or increase user strength by developing Android apps only.

V. REACT NATIVE: THE PROGRAMMING LANGUAGE

React Native is predicted on React.js, which is written in JavaScript. This is often a great advantage when it involves the very fact that JavaScript is such a popular language that it's very hard to seek out a programmer who didn't work on JavaScript at least once in their career.

What is the disadvantage of using JavaScript though? Well... it's JavaScript. While Java and Swift/ Objective-C are strongly-typed, compiled languages, JavaScript is interpreted and sometimes called an untyped language. This means that your variables are often anything at any time and a compiler isn't getting to assist you, so if you are not extra careful while writing your apps, the ground is paved for JavaScript horrors to happen next. When true native apps have more control of the variables so the logic of the app is more predictable, JavaScript relies on the programmer's experience, Lint tools and automatic tests.

Does this mean that JavaScript programs have more errors? No. But it does mean that you simply might spend tons longer checking out that app-breaking error which your compiler would have gotten for you in Java or Swift.

Still, it's a definite fact that JavaScript remains one among the best widely-used programming language within the world, and therefore the maturity of the JavaScript community hints that finding a React Native developer for a project should usually be silly/ extremely easy.

VI. REASONS TO USE REACT NATIVE WHILE BUILDING MOBILE APPLICATIONS

1. React: React may be a popular framework that's used to build web applications. It popularized the utilization of a virtual DOM, which greatly increased render performance for web apps. By using an equivalent principle as React, React Native maintains a virtual representation of the view hierarchy. When

Changes are made, rather than doing expensive operations on the view hierarchy, the virtual hierarchy is compared using an efficient doffing algorithm, then appropriate changes are made to the particular view hierarchy. This enables a developer to ditch one among the foremost dreaded things about building UI managing state. A React component always returns what it should appear as if from its render method. Everything else is taken care of. Another principle of React that's fascinating is encapsulation of components. With iOS, there are source files and Interface Builder files that work together to create the interface. Android has something similar with its XML interface files. Web apps have traditionally used three different types of files: HTML, CSS and JS. React throws away that thinking and encapsulates each single component into one file. All styling, logic and UI are defined within one file. It's a special way of thinking, but I have found that it's largely positive. Combines this with React's uni-directional data flow and it's very easy to know what's happening together with your components. As data comes in, the component renders itself and the UI is updated. Anytime the state of the component changes, the component will automatically render again and therefore the UI is usually a particular representation of the state. It's wonderful.

2. Quick Feedback Loop: In the earlier days of iOS development, it was quick and straightforward it had been to create an app. There have been hardly any APIs, Xcode seemed fast, there have been only a few devices to support and compile times were fast because apps were small. Mobile apps still grow in complexity and therefore the tools haven't scaled well. Xcode may be a beast that's slow and cumbersome. Compile times, especially with Swift, have gotten slower. It's easy to urge distracted when expecting an app to compile. I have worked on apps that take over quarter-hour to compile from scratch. While there are processes in situ to mitigate this, like only compile files that have changed, it still feels slow and it's easy to leap on Twitter while expecting a compile. With React Native, compile times are gone. This seems like a particularly freeing experience once you get within the groove. You will add file watchers to automatically reload the app once you save a file. Meaning, whatever you have got in your files is automatically updated and you will see your results instantly. While this not appears to be an enormous deal, I have found it to be incredibly helpful. I have been ready to maintain incredible focus because I am never waiting to ascertain my changes applied to the app. They are always instantly there. It's incredible. From a knowledge perspective, if a developer waits 30 seconds for compiling 20 times per day, that's 10 minutes per day; to not mention the context switching that happens when waiting. That's 3,250 minutes a year that would be saves for one developer assuming around 260 work days a year. Over 50 hours.

3. Cross Platform: This is obvious, but it really must be mentioned. React is such an excellent abstraction around, to build a UI that's often adapted to multiple platforms. At the moment, meaning iOS and Android. However, there are adaptions to Windows Universal App Platform and macOS also. The time, cost and communication saving by supporting one codebase against multiple is straightforward to know. Facebook uses React Native in their Ad Manager app, and that they have 85% code sharing between the iOS and Android platforms. Any company, larger or small, can enjoy cutting their app development costs in half, or in other words, doubling the dimensions of their mobile team with an equivalent number of developers.

4. Community: The community surrounding React Native is very positive. Another part of this is less known which dependency management is. Managing dependencies on iOS is a nightmare using either Carthage or Cocoapods. Historically Apple has not done a great job with dependency management. NPM and Yarn are great dependency managers that are actually fast. It becomes trivial to add new dependencies with these that otherwise would require adding a new pod, downloading everything, recompiling and running the app.

VII. EXAMPLE OF REACT NATIVE CODE

A simple Hello World Program in React Native:

```
import React from ‘React’;
import { AppRegistry, Text } from ‘react-native’;

const HelloWorldApp = () => <Text>Hello World!</Text>;
export default HelloWorldApp;
```

```
// Comment following line if using Create React Native App
AppRegistry.registerComponent( ‘HelloWorld’, () => HelloWorldApp );

//Using following line of code, we can reuse code for component which is already developed import HelloWorldApp
            from ‘./HelloWorldApp’;
```

VIII. ADVANTAGES AND DISADVANTAGES

Advantages:

1. Known for optimal performance
   Undeniably, React Native can be a real asset when it involves improving the performance through native and modules. The React Native gets connected to the native components for both Android as well as iOS and generates a code to the APIs upfront and freely. Now the performance enhances thanks to the very fact that it makes use of a special thread from UI and therefore the native APIs. You can also use other alternatives like WebView, but it will decline the performance level.

2. Can Reuse the Codes and Pre-Developed Components
   One of the most important benefits that you simply can gain from React Native is that the advantage of code reusability. The developers feel blessed and thankful to Facebook as they don’t need to develop a separate mobile app for every platform. In fact, you integrate 90% of the native framework for reusing the codes for both the OS. It is due to this unique feature that the developers tend to save lots of your time and also curtail the value of app development. What’s more pleasing to notice that you simply also can reuse the online application codes for creating a mobile app, if they are written in React. React Native also accelerates the speed of development thanks to the presence of the pre-development components available within the open-source library. It simply means the codes are already written beforehand and you are just required to implement it consistent with the app requirements. You are absolutely liberal to use it.

3. Large Community of Developers
   The fact that React Native is an open source JavaScript platform where every developer is liberal to contribute to the framework and it’s easily accessible to all or any. So, you will take full advantages of the community-driven technology. This means that if any developer gets stuck somewhere while developing an app, he can take the assistance of the community members who would guide him. The support of an outsized community is additionally useful because it allows you to share your experiences and portfolios in order that you will choose better coding. There is one platform GitHub React Community, which inspires the developers to share their experiences whenever they learning something new about React Native. They also receive the feedback and reviews on an equivalent establishing better communication with the community members. React Native has been introduced by Facebook. So, you did also receive extended support from social media giant. Indeed, Facebook has created a specific platform to assist out the developers initiating the long interactions where you will share your issues and best practices.

4. Advantage of Live and Hot Reloading
   The React Native is understood for its support for Live and Hot Reloading. Don’t get confused as both are different features. If we mention Live Reloading then it’s a tool that helps in compiling and reading the file where the developer had made the changes. It also offers a replacement file to the stimulator, automatically reading the app from the start. The Hot Reloading is predicated on the recent Module Replacement (HMR) and made its inception after the initial reloading process. Although there’s not much of a different in function and features, but the most aspect comes into play during saving the changes. The HMR inter-mediator will put the updated files into the precise place albeit the app remains running. The key plus point of Hot Reloading is that it allows the modification of the ASCII text file and intrinsically, you don’t need the app recompilation for viewing the codes. Thus, if you have got a few of open windows including that of code and therefore the app screen, you did be ready to view the result instantly after making changes within the code. In fact, Hot Reloading also plays an important role in reducing the waiting time for changes.

5. React Native is a Cost Effective Solution
   We have already discussed within the earlier paragraph how the code reusability in React Native helps to scale back the value of the app development. The developers don’t get to use separate codes for both the platforms as both OS are often coded with single programming language. Now, this makes your project cost cheaper as

you don't need to employ an outsized team and different developers for getting the work done. This is another advantage of React Native as a little team size is sufficient to hold out the task.

6. Offers Simple User Interface
The React Native also offers a simplified mobile interface, that's why it's more referred to as an open-source JavaScript library instead of being a framework. If you have got developed a perfect app, it's important to urge the sequence right, and React Native is simply the optimum choice for getting it. It offers a more responsive UI designs and reduces the loading time also.

7. Support for Third-Party Plug-in
The React Native also renders its support for the third-party plug-in offering a few of options that include native modules and JavaScript modules. This is often because it doesn't have some components within the main framework. For instance, if you are implementing any Map in your app, React Native allows you to do so by connecting the plug-in with a native or third-party module.

8. Offers More Stable Apps
The React Native offers one great feature which is simplifying the method of knowledge binding. This is often thanks to this aspect that your apps become more stable and also the extent of reliability increases. The child element doesn't have any influence on the parent data. The React Native only allows the updating of the permitted components and if you have got to form any modification, then you are required to vary and apply the updates.

9. Modular Architecture
The Modular Programming helps to segregate the program functions into various different free and interchangeable blocks referred to as modules. Now this is often software technique that creates the event more flexible and establishes better co-ordination with one another to urge the updates. React Native is benefited to get this intuitive modular architecture, which helps React Native developers to simply upgrade the apps at quick pace. You will also reuse the modules same as codes for web and mobile APIs.

10. Providing Libraries
React Native gives you the advantage of getting the simplest list of ready-made solutions and libraries to simplify the task of mobile app development.

Disadvantages:

1. React Native is Still New
React Native is new as compared to other Android and iOS programming languages, it's still in its improvement stage and this will have a native impact on the apps. One of the areas where it lacks consistency is regarding the discharge of the frequent updates, which sometimes are confusing for the developers. This is often because whenever a replacement update is released, the developers need to make tons of changes and make regular upgrades. Popular app Airbnb faced tons of problems in upgrading. Secondly, the developers also are sometimes required to write down the additional native code for components not compatible with the React Native aside from writing just the codes. At times, the React Native components turn poorer as compared to the native components. It's very difficult to integrate the long and large list with React Native.

2. Learning the Rope is a Tough Ask
React Native can be a tough rope to learn, especially if you are a fresher in the app development field. This is thanks to the presence of JSX within the JavaScript syntax extension where the HTML element gets combined with JavaScript. A mean learner won't find it that easier as compared to other cross-platform apps.

3. It Lacks the Security Robustness
We know that React Native is a JavaScript library and open source framework, which creates a niche within the security robustness. But, sometimes, you would like to use some extra security specifically if you are creating banking and financial apps where data is very confidential. So, therein cases, experts advice to not choose React Native.

4. Take More Time to Initialize
   The problem with React Native is that it takes tons of your time for initializing the runtime even for the hi-tech gadgets and devices before it are often rendered initially. This is again due to the JavaScript thread which takes time to initialize.

5. Managing Memory
   React Native isn't fit use for the computation of the intensive apps and again, the responsibility for this drawback falls on JavaScript.

IX. CONCLUSION

The key thing about React Native is that it's still in development and that we are yet to ascertain its full potential. Within the future, it's going to be more powerful and efficient and permit for even more use cases, except for now it cannot fully replace native mobile development. However, its write once, use everywhere paradigm are often an incredible time and money saver if used on the proper projects.

In the end, is React Native worth it? The solution is: It depends on your project.

- Do you have to make an iOS-only or Android-only app? Go native.
- Do you have a little team with limited time and resources, and wish to form an app for both platforms? Go React Native.
- Do you have to make a highly complex app which utilizes an outsized portion of platform-specific code? Go native.
- Do you want to require advantage of fast build time, and features like hot reloading and live reloading? Go React Native.
- Do you decide to maintain the app over an extended period of your time, without worrying of Facebook quitting React Native? Go native.
- Do your developers have a robust React/ Web development background? Go React Native.
- Does your app got to support new mobile OS features as soon as they are released? Go native.
- Is your app getting to look and behave an equivalent on both platforms? Go React Native.

REFERENCES

1. Daniel sson, W. Froberg, A.& Berglund, E. (2016). React Native Application Development- A comparison between native Android and React Native , (pp. 1 -70)
2. Beyshir, A. (2016). Cross-platform development with React Native.(pp. 1-32),
3. https://uu.diva-portal.org/smash/get/diva2:971240/FULLTEXT01.pdf
4. William Danielsson, React Native application development – A comparison between native Android and React Native, https://www.diva-portal.org/smash/get/diva2:998793/FULLTEXT02
5. Wenhao Wu, React Native vs Flutter, cross-platform mobile application frameworks,
6. https://www.theseus.fi/bitstream/handle/10024/146232/thesis.pdf?sequence=1
7. Nitin Rajpal, Going Native with React Native,
8. https://www.coforgetech.com/resource-library/white-papers/going-native-react-native

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING