

# Evaluation of cross-platform application development frameworks

Supreetha A. Shetty  
Dept. of ISE

R.V. College of Engineering  
Bangalore, India  
[supreethaas.sit19@rvce.edu.in](mailto:supreethaas.sit19@rvce.edu.in)

Dr. Padmashree T.  
Dept. of ISE

R.V. College of Engineering  
Bangalore, India  
[padmashreet@rvce.edu.in](mailto:padmashreet@rvce.edu.in)

**Abstract**—Software has become an integral part of our daily lives. It is making a profound impact on every aspect of life such as Entertainment, Education, Business and so on. The native application development results in redundant code and consumes more time and resources to develop applications for each platform. Most of the businesses today need their use cases to be implemented across various form factors such as PC, TV, smart phones, wearables and so on.

The cross-platform development approach helps the developers to write code in one platform and be able to run the app in different operating systems whereas in native app development approach, the app is developed exclusively for a single platform.

**Keywords**—Cross Platform framework, Xamarin forms, Single source code.

## I. INTRODUCTION

In this digital era, hundreds of applications are published every day. With rapid advances in technology, applications are no more restricted to Personal Computers or to one particular mobile platform. The focus of developers is to target their applications to various platforms with the least effort and low cost. Hence the individual developers or the organizations are moving towards cross platform mobile development approach.

The paper is organized as follows: Section 2 outlines the previous work, Section 3 briefs about the cross platform frameworks followed by Comparative analysis and Conclusion in Section 4 and 5.

## II. LITERATURE SURVEY

There are several approaches in mobile application development based on the principle “Code once, Run anywhere” such as web-based approach, hybrid approach and cross compilation approach. Though these approaches help to save the cost and time of the developers, each approach has its own limitations when compared to Native approach. The paper [1] by Latif et al., compares several of these approaches available and emphasizes on Model Driven Approach (MDA) to be promising in terms of having performance as native application. However, the code generated by the model is incomplete and needs to be completed manually with the use of native language.

Pinto and Coutinho have compared various mobile development approaches and the tools available for each of the approaches [2]. ERP application was developed using Ionic platform which is a hybrid cross platform tool and was tested on Android and iOS devices.

Jia et al., provides insights to the performance characteristics of native approach for Android and iOS, and various cross-platform tools such as Xamarin, Appcelerator Titanium and Apache Cordova [3]. The performance characteristics are tested by developing a set of apps using different approaches

and comparing them. The observations were made on app’s rendering time, UI response time, memory usage and app size.

Vishal and Kushwah in their paper [4] have compared Xamarin platform to native android development and React native, which is also a popular cross platform framework. Xamarin follows the Model-View-View-Model (MVVM) style of architecture which has the ability to share business logic across platforms. The comparative analysis shows that Xamarin forms outstands React Native in terms of compilation, development environment, architecture style and community assistance.

Xamarin helps in reducing platform-specific code not only by sharing business logic across platforms, even the UI code is shared to some extent. Earlier, Xamarin allowed users to create a project called Portable Class libraries (PCL) which consisted of interfaces for platform specific libraries. Nader et al., developed a testing tool called X-checker [5] and tested the methods in PCL to test the performance of applications developed using Xamarin forms. Few of the bugs were identified and reported to the Xamarin team, which has been addressed by the Xamarin developers. Now the usage of PCL is depreciated and .NET standard 2.0 Library or .NET 5 or later is recommended [6].

## III. NATIVE APPROACHES TO CROSS PLATFORM FRAMEWORKS

### A. Native Android

Native android applications are developed using Java or Kotlin. The native approach results in great performance as it can seamlessly integrate with the device hardware. However, the development and maintenance cost is high as it requires constant updates to match the current version of the operating system.

### B. Flutter

Flutter is one of the most popular cross platform frameworks available today that allows to build iOS, Android, Windows, Linux, Mac and web applications with one code base. It is developed by Google and the programming language used is Dart, which is also developed by Google. The UI components are considered as widgets in flutter and it doesn’t use platform primitives.

### C. React Native

React Native is based on Javascript that allows developers to develop applications across various platforms. It uses the same User interface controls as native platforms which helps in maintaining the native look and feel of the application.

### D. Xamarin forms

Xamarin forms allows developers to develop applications across multiple platforms such as android, windows and iOS using C# for backend and XAML for UI development.

## IV. COMPARATIVE ANALYSIS

The native application development approach though has seamless user experience, the maintenance and development cost is higher compared to cross platform frameworks. Though there are a significant number of cross platform frameworks available today, Xamarin forms, React Native and Flutter stands in top three positions based on their performance. Developers often choose between these three platforms based on their requirements. Table 1 compares these three platforms:

- [4] K. Vishal and A. S. Kushwaha, "Mobile Application Development Research Based on Xamarin Platform," 2018 4th International Conference on Computing Sciences (ICCS), Jalandhar, 2018, pp. 115-118, doi: 10.1109/ICCS.2018.00027.
- [5] N. Boushehrinejadmoradi, V. Ganapathy, S. Nagarakatte and L. Iftode, "Testing Cross-Platform Mobile App Development Frameworks (T)," 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), Lincoln, NE, 2015, pp. 441-451, doi: 10.1109/ASE.2015.21.

Features	Native Android	Flutter	Xamarin forms	React Native
Platform support	Android	Mobile, Desktop and Web,	Android, iOS, Windows ,wearOS, macOS and windows apps	Android and iOS
Language used	Java or Kotlin	Dart	C# and XAML	Javascript and Rect library
Performance	Flawless performance	High performance achieved using native compilers of Dart	Matches the native app development standards , however not suitable for apps with rich user interface	At times lags native performance
Framework	-	Flux	MVVM	Flux
Ownership	Google	Google	Microsoft	Facebook

## V. CONCLUSION

The need for applications to be developed for various form factors has made the developers move from native application development to cross platform development. In this paper, native android development approach and three popular cross platform frameworks - Flutter, React native and Xamarin forms are studied and compared.

## CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

## REFERENCES

- [1] M. Latif, Y. Lakhri, E. H. Nfaoui and N. Es-Sbai, "Cross platform approach for mobile application development: A survey," 2016 International Conference on Information Technology for Organizations Development (IT4OD), Fez, 2016, pp. 1-5, doi: 10.1109/IT4OD.2016.7479278.
- [2] C. M. Pinto and C. Coutinho, "From Native to Cross-platform Hybrid Development," 2018 International Conference on Intelligent Systems (IS), Funchal - Madeira, Portugal, 2018, pp. 669-676, doi: 10.1109/IS.2018.8710545.
- [3] A. Ebone, Y. Tan and X. Jia, "A Performance Evaluation of Cross-Platform Mobile Application Development Approaches," 2018 IEEE/ACM 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft), Gothenburg, 2018, pp. 92-93.