



Study on Cross –Platform Mobile App Development With Xamarin

Harish Nair, Vineet Yadav

Student, Department of Master of Computer Application,
ASM's Institute of Management and Computer Studies, Mumbai, Maharashtra, India

ABSTRACT

Mobile phones have started a revolution in the lives of people in the over the past few years. The ability to store, access, communicate and compute content via mobile phones has redefined connectivity and communications because of this Mobile Application Developers of this generation want their apps to be available on all the platforms. To address this issue, cross platform tool have been developed and was found to solve device functionality features, platform and device coverage problems. Choosing one single Cross platform tool for development was difficult, it was suggested that, Titanium and Xamarin were picked as a preliminary starting point. But there is android application development software where developers can develop the android application using C# and .NET Framework in Xamarin Studio. Xamarin is a Cross platform mobile app development platform for building native apps on iOS, Android, and Windows apps from a common C#/.NET codebase.

In this paper we are focusing on native application development approach and cross platform application development approach. Developing mobile applications inside Visual Studio provides a number of benefits such as:- Creating cross-platform projects for Android, Windows and iOS applications, Working with a familiar IDE, while taking many advantages of Xamarin. iOS bindings of all Apple's API Although developing applications using Xamarin saves time and resources, there are some inherent disadvantages.

Keywords: *Xamarin, Cross-Platform, Visual Studio, Android, iOS*

I. INTRODUCTION:

Mobile phones have become necessity of life nowadays. In recent times, the world has seen a significant growth in terms of mobile application and the market has created a vast and complicated set of alternatives for developers because of an increasingly important role in everyone's life. The mobile software technology consists of a wide list of plat-form technologies, and architectural choices. Hence, it makes mobile application development easier and cost-effective. Mo-bile application developers must address both business and technical issues in deciding which platforms to support based on feasibility and cost concerns. There are numerous number of platforms for which developers can build applications. Applications developed for a particular Operating System or platforms are not compatible on different OS's. Indeed, they force developers to rebuild the same applications for them. In recent times the Cross-platform mobile development tools are gaining wide spread popularity in the world due to their ability to compile the application source code for multiple supported OS's. To ease the efforts of developers, cross-platform mobile development tools have been developed with the purpose to help the developers to write the application source code once and run it on multiple different platforms.

Cross-platform technologies are those techniques which combine the source code of an application with a runtime environment and then, this environment interprets the app's code at runtime and thereby executes the app. The runtime environment has to be specific for each mobile platform, while the app's source code is platform-independent.

II. NOMENCLATURE

1. NATIVE DEVELOPMENT APPROACH VS CROSS PLATFORM DEVELOPMENT APPROACH

For the most part, mobile devices run one of two operating systems: iOS and Android. iOS is developed and supported by Apple and is used on only on their own iPhone and iPad. In terms of Apple, they control both the hardware and the software. Android is an Operating System and is developed completely by Google and supported by Google also. It is often considered a more open platform compared to Apple. In fact, android is an open source operating system, which means that anyone can use their code to run a device. Google have started to manufacture and sell a few devices of its own, but Android normally runs on devices built by other companies. There isn't any overlap between the apps of each of these devices, that is, native iPhone apps won't run on Android phones and vice versa.

The majority of the apps on your mobile device are native apps except Web Apps which are primarily written in JavaScript, whereas native apps are written in languages that the platform accepts. Java is primary language used to write native Android apps, and C# is used for Windows Phone platform apps. A native application is an application program developed specifically for use on a particular platform or device which is to be developed. Because native apps are written for a specific platform, they can interact with and take advantage of operating system features and other software that is typically installed on that platform.

The main reason for using native app's development platform is because it has significant advantages over the alternatives.

- They offer the fastest, most reliable and most responsive experience to users. This is unlikely to change in favor of web apps.
- It is rather easy to tap into the wide variety of functionalities offered by the device including the camera, microphone compass, Accelerometer and swipe gestures, etc. The Most popular apps available in the Google play store like Pokemon Go, Twitter, and Waze, are fully native.

The Main disadvantage in terms of using native app's development approach is having to develop Apps separately for each platform you want to cover for the project. This is still true if you want to stick to the

native SDKs provided by Apple and Google for the specific platform, but in the last few years, several alternatives have become available which benefits cross-platform development without sacrificing the user experience or access to native APIs.

According to the some survey conducted on mobile application development time, it takes an average of 18 weeks to build a standard native mobile app; 10weeksfortheback-end and 8 for the front end. Keeping in mind that not all apps have a back end and some may use a back-end as a service to reduce development time and complexity. The actual time frame might vary widely from this average time, but this is still a good reference if you're new to the world of app development.

Hybrid platforms on the other hand are applications that execute on the native user's browser, such as UI Web View in iOS and Web View in Android (not Safari or Chrome) developed using HTML, CSS and JavaScript, and then wrapped in a native application using platforms like Cordova, Phone Gap. Cross-platform approaches allow developers to implement an app as a single code base that can be executed on more than one platform.

Over the last few years, several alternatives have become available to which the benefits cross-platform development without sacrificing the user experience or access to native APIs of which Xamarin and React Native are both worth look.

2. XAMARIN

Made by Microsoft, Xamarin is a platform that lets developers build one app that work on multiple platforms in C#. Xamarin. Forms is now freely available to all users of Visual Studio. Microsoft also provide free tools to build, test, distribute, and learn from your apps. Xamarin is a more like complete IDE than Phone Gap and Titanium, even offering a test environment where you can test your app on thousands of virtual devices before launching (this is crucial for cross- platform apps). Xamarin also offers a few prebuilt apps you can use to get a quick start.

The preferred solution for any application is native design. This is where Xamarin and its unique feature come in. Xamarin Studio (IDE) enables code completion in C#. It provides the **advantages of native User Interface, so that it** can access to specific-device features, and most importantly, native

performance. Code sharing across platforms is a bridge with Xamarin, helping you reduce that development cycle. Xamarin is the only IDE that allows for native Android, iOS and Windows app development within Microsoft Visual Studio. Xamarin provides add-ons to Microsoft Visual Studio that allows programmers to build Android, iOS, and Windows applications within the IDE using code completion and IntelliSense.

3. XAMRIN IN VISUAL STUDIO

Xamarin with Visual Studio is introduced recently, Enlightened IDE with many features for creating iOS, Mac and Android applications. It contains a rich set of tools such as editors, debuggers, native platform integration with Android, Windows and iOS and also a consolidated source control and many for such exciting new and enhanced features. The primary area of concern for Xamarin's is on compilation technologies and also the basic sets of .NET libraries:

- Xamarin. Mac, which has extended from the Mono Mac project.
- Xamarin. iOS, which has extended from Mono Touch.

Xamarin platform includes libraries in it, Mono Droid. Collectively. These libraries consist of .NET versions of the native Mac, Windows, and Android APIs. Developers using these libraries can write programs in C# to target the native APIs of these three platforms, but also (as a bonus) with access to the .NET Framework class library. Application Developers can use Visual Studio to build Xamarin applications, targeting corresponding iOS or Android as well as all the various Windows platforms.

4. XAMARIN NATIVE APPLICATIONS

4.1 Shared app logic:

The way apps functionalities are shared across multiple platforms makes Xamarin a must-use cross-platform development tool. The application logic behind the User Interface layer (UI Layer), like input verification, web service calls, database interactions, and back end enterprise integration are wrapped inside only once in C#. Developers can share as much as 75% of the code across the different-different operating systems. This saves time required to re code and leaves less room for bugs to creep in shared app logic.

4.2 API integration:

Xamarin binds the same Application Programming Interfaces and User Interface controls that are used to create iOS, Android and Windows applications in their respective platform specific environment. For Windows development, Xamarin with Microsoft Visual Studio offers Windows Phone and Windows 8 applications. Code can be contributed between iOS, Android and Windows using Portable Class Libraries (PCL) and proper application architecture. With Xamarin, support is always around the corner. Their uncommon or rare binding technology enables them to give support for new features soon after they are introduced in the device's operating system.

4.3 Xamarin Component Store:

Peoples in Xamarin clearly hop in association and distributing. This is a proof to their ongoing efforts on the Xamarin Component Store. Programmers can choose from a host of free or paid components, which include User Interface (UI) controls, cross-platform libraries, and third-party web services to applications with just a few lines of code. The Component Store is built right into Xamarin Studio and Xamarin's Visual Studio extensions.

Xamarin is one of the best cross-platform mobile development frameworks in the true sense. Xamarin isn't like traditional HTML, JavaScript mobile development frameworks. It allows you to use almost 100% native libraries of both Android & IOS.

Advantages:

- Using Xamarin, a single platform could code for three separate platforms as of now.
- Xamarin allows its users to make use of the knowledge of C# framework and apply it on iPhone and Android. It can use existing c#/.net Code in iOS and Android.
- Excellent Support.
- Xamarin & .NET are very well architected. C# is a fantastic language.
- Shared code. If we want to share the actual logic/link code, Xamarin allows us to do just that.
- Focus on the product & the frameworks rather than the language. We can use C# everywhere and focus learning on the IOS & Android SDKs.
- It's easier for programmers to move from one project to another. We won't be stuck with Android developers not being able to do anything while the iOS developers are too busy.

Disadvantages:

- Xamarin doesn't come cheap
- There's no deny of it that Xamarin can do wonders, but it isn't an inexpensive proposition. Microsoft has a number of extremely good products in their bags and even programmers dedicated to Xamarin are experts, so there is 100% coverage to native APIs, and this doesn't come in low prices.
- It is not a "Write once run anywhere" platform. It basically allows programmers to code the applications in a single language C#, which allows code sharing a cross various platforms.
 - It doesn't matter if you have knowledge of iOS/Android, you can develop application using Xamarin. Recruitment may be harder. Mobile development is tough.
 - If you have prior experience as in Java, you may have developed an attachment for it, thus it might not be an easy task to switch over to Xamarin quickly and developing on Java or Android Studio would be better than Xamarin

IV. CONCLUSION

At the end of the day, it depends on the application, project requirements as well as time frames, If you are quite certain to target only Android platform, then using the native Java, Android SDK would be a better option. To develop cross platform application using Xamarin Studio it needs libraries to be downloaded, but it provides more functionality for smooth developing it reduces the time to develop the application or web application for different platforms. Xamarin provides platform to develop native applications with the help of C# language.

ACKNOWLEDGMENT

We would like to thank all the people who guided us throughout the whole period of time and encouraged us for successfully completing the paper. We would like to express our sincere gratitude to our Head of department Prof Reeta Singh, and Mentors Prof Sheeba James and Prof Priya Ped-nekar for the continuous support, guidance

REFERENCES

1. <https://docs.microsoft.com/en-us/visualstudio/cross-platform/visual-studio-and-xamarin>
2. <https://forums.xamarin.com/>
3. <https://msdn.microsoft.com/en-in/library/mt488768.aspx>
4. <https://docs.microsoft.com/en-us/xamarin/cross-platform/macios/native-references>
5. <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/>
6. <https://code.tutsplus.com/tutorials/getting-started-with-xamarinforms-basics--cms-21564>