

Comparative study of Xamarin and PhoneGap

Ravi Chitiken¹, Harshali Patil²

¹Student, MET Institute of Computer Science, Maharashtra, India

²Associate Professor, MET Institute of Computer Science, Maharashtra, India

Abstract - In the emerging world, smartphone industries are continuously growing, also coming up with new technology and developing useful, complex applications. As there are many mobile platforms like iOS, Android, and Windows Phone, etc which differ in SDK tools and development methodologies that make application much complex and cost-effective. So, the need to get better alternatives by focusing on one SDK tool that can run/work on different platforms as native applications and the best alternate is cross-platform mobile development. This paper describes the comparison of two cross-platform mobile development technologies.

Key Words: Cross-platform mobile development, Xamarin, PhoneGap, iOS, Android, XML Layouts, StoryBoards

1. INTRODUCTION

Nowadays, software developers have to develop applications that can support & run on different platforms like iOS, Android, and Windows Phone. Focusing on the same, we need to check user experience across different operating systems irrespective of platform or device specifications. The problem here is writing separate code for different platforms and by different teams, which becomes expensive for such development. So, the solution is using cross-platform mobile development tools and frameworks, that not only reduces development time but also cost. By this solution developers can make use of single codeshare to run on different platforms. As, there are many cross-platform development tools and frameworks, depending on the development approach the developers can choose the technology to work [1]. This paper describes in brief about the comparison of PhoneGap and Xamarin, according to criteria like efficiency, development cost, use of programming language, performance, security, tools, and framework support.

2. LITERATURE REVIEW

Over the last decade, the rapid evolution of new frameworks and tools led trends in mobile software development which differ in the platform and operating system. The available platforms and SDKs are used in cross-platform mobile application development based on the requirements. According to the current research, it is known that the cross-platform framework and tools are progressively used for fast and easy development of mobile applications by developers and many companies. From the past studies on cross-

platform mobile development, it is found that active research on a large number of topics is done that includes the performance of cross-platform frameworks and tools, development approaches, technical comparisons, study on user experience and analysis of cross-platform applications. This paper presents the relatable work like such previous studies and researches where it compares two cross-platform frameworks Xamarin and PhoneGap based on criteria. Further, each framework is described in detail.



Fig -1: Mobile App Development - Infographic with Statistics [2].

3. WHAT ARE CROSS-PLATFORM APPS?

Cross-platform applications are built using a single source code that can run on many platforms and this code gets converted to native apps, this helps developers instead of creating applications for each platform separately [3]. This process has many benefits like performance similar to the native apps, cost-effective and faster development [4].

4. WHAT IS XAMARIN?

Xamarin is a tool for mobile app development provided by Microsoft. The applications developed using this tool support platforms of Android, iOS, and Windows Phone or a combination of platforms [5]. The software required are Visual Studio or Xamarin Studio to build applications. Developers can write applications using C# programming language and create user interface with XAML code [6].

5. WHAT IS PHONEGAP?

Adobe PhoneGap is a mobile development framework for building applications using HTML5, CSS, and JavaScript. PhoneGap is a free and open-source framework that targets

many platforms using one codebase [7]. The applications developed using PhoneGap are not native but similar and run in an embedded web browser. PhoneGap apps make easy access to native functions using provided libraries and API bindings to access the device sensors, data, and network status [8].

6. XAMARIN ARCHITECTURE

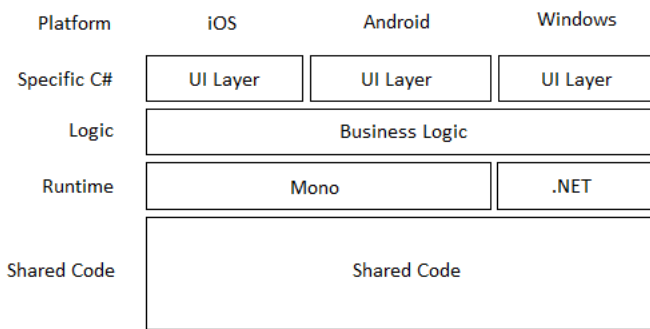


Fig -2: Architecture of Xamarin [9].

The diagram depicts the architecture of a cross-platform Xamarin application. Xamarin allows developers to create native UI for each platform and business logic code in C# language that can be shared across multiple platforms. Xamarin makes 80% code sharable [5]. Shared code allows us to refer many different projects. The source code compiled can be of respective projects and that consists of compiler directives which can serve as platform-specific within a shared codebase [10]. Mono is an open-source version of the .NET Framework and it follows the .NET European Computer Manufacturer's Association (ECMA) standards. As Xamarin resides on Mono and Mono resides on OS kernel, this structure makes code to invoke native API bindings [11].

7. PHONEGAP ARCHITECTURE

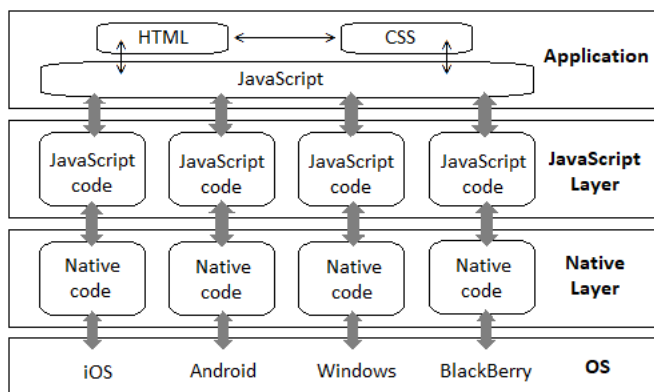


Fig -3: Architecture of PhoneGap [12].

The diagram depicts the architecture of PhoneGap with four

layers. Operating system as the base layer of the PhoneGap framework. Developers building native apps would work on this layer and it consists of APIs by the provider of OS. On top of the OS layer is the native layer, to obtain native functionality to the browser, PhoneGap built a link within JavaScript and native code. This layer consists of WebKit View to run developed applications. JavaScript layer consists of APIs that can be used by developers in the JavaScript code to build PhoneGap apps. For function implementation using these APIs, this layer returns to the implementation, and if not then it calls one such native function in the below layer. The app layer consists of application code, which includes HTML5, CSS, and JavaScript [13].

8. COMPARISON CRITERIA

8.1 Developer Experience

Developer Experience has a major effect in the development of an application. Which includes how smooth the development process is? The maintenance of it, throughout its service also important factor. Selection of a technology to work on matters the most, less support, and less development experience this can lead to problems in an application, and also it is very difficult to find developers to work on with such technology [15].

Xamarin

Xamarin uses the Model-View-ViewModel (MVVM) architectural pattern to achieve cross-platform development. This technique focuses on the separate development of the front-end from the back-end, so that the view does not matter on any specific platform model [14].

PhoneGap

Phonegap uses the Model View Controller (MVC) architectural pattern and phonegap.js library for better UI development. Phonegap also uses Ionic which provides Angular directives that make development easy, where a CSS framework is available instead of working from scratch with HTML.

8.2 Code Reusability

Reusable code is the best point of cross-platform development, where one codebase can work with multiple platforms and it also saves a lot of time. Any developer can work with a single codebase for different platforms at the same time [16].

Xamarin

Xamarin makes reusable code through shared projects and portable class libraries (PCL). The MVVM pattern allows

code separation without concerns from UI code to business logic code [14].

PhoneGap

As PhoneGap follows MVC pattern the view and controllers run on different devices that make most of the code reusable and here sharing of code is not required.

8.3 UI Development

UI development is building websites, web applications, mobile applications, and other software. User Interface plays a major role in software development because UI design aims to achieve user goals [17].

Xamarin

Xamarin allows building native UI's using XML code in Android and StoryBoards in iOS. Visual Studio is used for both platforms to work on visual design. This benefits in using design paradigms like Android's Material design and iOS's Human Interface Guidelines for better UI development. In Visual Studio for Android UI design it is done in XML layout screen and for iOS UI design with editing of StoryBoards in XCode [14].

PhoneGap

Phonegap UI design is similar to an HTML5 application and its codes are checked by running in a web browser to get required UI design and it takes less time but the UI created is not great as in Xamarin. So the use of the Ionic framework achieved the UI suitable for the platform. Ionic made the UI development easy but need to check designs at runtime.

8.4 Compile Time

The compile-time of an application is the amount of time it takes to compile an application that can have a substantial effect on the development of software. Considering the compile time for release build for PhoneGap project build on Mac and Xamarin project from Visual Studio paired with a Mac. The compile time for the Xamarin project takes 70 sec on iOS and 40 sec on Android respectively, whereas PhoneGap project takes 18 sec on iOS and 25 sec on Android respectively. Mac specification is sufficient for the PhoneGap project as it takes less compile time than the Xamarin project while Xamarin needs high-end machines to reduce the compile time [14].

8.5 User Experience Comparison

As earlier mentioned, the PhoneGap uses HTML5 and CSS framework for visual design whereas Xamarin uses XML layouts. PhoneGap uses Ionic framework to achieve iOS and

Android like visual design. However, some platform related adjustments need to be done to look like native. The point of this comparison is how PhoneGap can accomplish a native user experience with minor changes but Xamarin already provides a native experience [14].

8.6 Developer Tooling

Some of the development tools used for Xamarin and PhoneGap projects are:

Xamarin

- Visual Studio 2019
- ReSharper
- XCode

Visual Studio has several tools comprising source control integration and package management through NuGet packages and the Xamarin Component Store. All these are used in the solution to make reference and manage different versions and packages. Xamarin with Visual Studio provides the development and debugging facilities more sophisticated. Using breakpoints allows us to understand the execution of code during debugging of the application. Debugging applications through virtual and physical devices the experience is great. To run the application in the physical device for Android the device should be paired with the Xamarin installed machine, this helps to compile and deploy the application on the device. But in case to run on an iOS device the need for an OSX machine is required. For iOS development, XCode is used to work with StoryBoards because XCode provides the easy way for editing and then using them within the Xamarin projects. ReSharper provides various tools in Xamarin for XAML code analysis, generating code and other utilities to present quality applications [14].

PhoneGap

- Atom
- Ionic
- OSX Terminal

The development of the PhoneGap project using Atom text editor (open-source tool) that allows us to build great UI design. These PhoneGap tools also allow package management and build integration through Bower, Grunt, and Gulp. As compared to the Xamarin Component Store and NuGet package, their visual interface is lower but they don't need much scripting. Debugging for PhoneGap projects is done with the use of debug tools available in web browsers. These tools provide remote debugging on virtual and physical devices. Atom editor does not have this facility as compared to visual studio.

8.7 Code Organization

Xamarin

Xamarin follows the .NET standard of project structure with a portable class library (PCL) to hold shared code and platform-specific libraries each one for iOS and Android. NuGet and Xamarin Component Store can be used for more packages. PCL consists of all shared logic code which consists of models, view models, data access, common interfaces, and reusable contents that can be shared with other projects. For platform-specific UIs to work on iOS has StoryBoards and View Controllers whereas Android has XML views and activities [14].

PhoneGap

PhoneGap project using Ionic has an Ionic default folder structure that consists of all controllers in one file, all services in one file, and so on. The project structure places the controllers, services, and other files into its path and it organizes by file type. PhoneGap project structure is similar to the Xamarin project structure but there are many differences one such difference is code sharing with different projects makes also the share of javascript files with other projects whereas Xamarin makes only one .dll file to be shared.

8.8 Access to Native Functionality & APIs

Xamarin

Xamarin makes use of C# language and wrapping up native libraries within the .NET framework for cross-platform development of applications. Xamarin apps are native to iOS and Android in case of performance as well as in user experience. The native APIs that are accessible in iOS and Android are also accessible in Xamarin. Presently third-party libraries are not accessible through .NET, but library bindings can be done [14].

PhoneGap

PhoneGap achieves native functionality in the projects using PhoneGap plugins. A repository with a large number of plugins maintained by Apache at plugins.cordova.io and also one can write custom plugins using Java for Android and Objective-C/Swift for iOS respectively. The Ionic also has a huge collection of plugins at ngCordova, within the Angular extensions.

8.9 Long Term Maintenance

As cross-platform applications are a result of the single codebase which reduces development as well as long term maintenance costs. Xamarin and PhoneGap solutions provide

structure and high maintainability through the use of more third-party frameworks [14].

Economic Stability

The technologies like HTML5, CSS, and JavaScript had many changes from over a few years. Since, PhoneGap uses Ionic which provides Angular JS framework that frequently upgrades parts of the framework this leads to difficulty in the selection of proper framework for PhoneGap. In the case of the .NET framework and Xamarin does not need much other framework support as it was mostly dependent on provided NuGet packages and Xamarin Component Store.

Code Maintenance

Different languages take different approaches for code maintenance. C# uses a type system which makes type checking at compile-time and runtime whereas JavaScript at presently does not. Applications that can be stable and maintainable can be obtained by type checking at compile time.

8.10 Performance Metrics

App Size

When it comes to application size the question that comes in mind is about how large the file size is? and how much memory space it would take on the device when installed? In all such cases the PhoneGap application would take less memory space as compared to the Xamarin applications [14].

App Start Time

Once the application is installed and on launch of the application the user expects that the application should not take much time to load and run without any lag. In these terms Xamarin applications have faster start or load time as compared to PhoneGap applications.

9. CONCLUSION

Finally, in the growing field of technology in mobile app development an alternative solution for native apps is cross-platform. However, both Xamarin and PhoneGap have different approaches in building cross-platform mobile applications each one with constraints. But, it depends on selecting which technology to work based upon the requirements of the project. This paper describes the comparative study of Xamarin and PhoneGap with differences in their technologies which will help beginners to choose appropriate tool for app development. On selection of the cross-platform technology the mobile application development process becomes viable.

REFERENCES

- [1] ResearchGate. (n.d.). (PDF) Comparison of Popular Cross-Platform Mobile Application Development Tools. [online] Available at: https://www.researchgate.net/publication/282816272_Comparison_of_Popular_Cross-Platform_Mobile_Application_Development_Tools [Accessed 27 Apr. 2020].
- [2] Saiyed, J. (2019). Mobile App Development Trends 2019 – Infographic with Statistics. [online] Magneto IT Solutions. Available at: <https://magnetoitsolutions.com/infographic/mobile-app-development-trends> [Accessed 27 Apr. 2020].
- [3] Cygnet Infotech. (n.d.). Comparing PhoneGap vs Titanium vs Xamarin. [online] Available at: <https://www.cygnet-infotech.com/blog/phonegap-or-titanium-or-xamarin-which-cross-platform-should-you-choose> [Accessed 27 Apr. 2020].
- [4] Rishabh Software. (2019). Advantages & Disadvantages Of Cross Platform Mobile Development. [online] Available at: <https://www.rishabhsoft.com/blog/pros-cons-cross-platform-mobile-app-development> [Accessed 27 Apr. 2020].
- [5] profexorgeek (n.d.). What is Xamarin? - Xamarin. [online] docs.microsoft.com. Available at: <https://docs.microsoft.com/en-in/xamarin/get-started/what-is-xamarin> [Accessed 27 Apr. 2020].
- [6] sivaraj, arvindpdmn (2018). Xamarin. [online] Devopedia. Available at: <https://devopedia.org/xamarin> [Accessed 27 Apr. 2020].
- [7] Sanjay Ratnottar (2019). What is PhoneGap & Why You Should Use It For App Development. [online] Medium. Available at: <https://medium.com/swlh/what-is-phonegap-why-you-should-use-it-for-app-development-5ac5e19eb19a> [Accessed 28 Apr. 2020].
- [8] phonegap.com. (n.d.). PhoneGap Explained Visually. [online] Available at: <https://phonegap.com/blog/2012/05/02/phonegap-explained-visually> [Accessed 28 Apr. 2020].
- [9] Xamarin Community Forums. (n.d.). Xamarin architecture diagram (updated). [online] Available at: <https://forums.xamarin.com/discussion/170673/xamarin-architecture-diagram-updated> [Accessed 28 Apr. 2020].
- [10] davidortinau (n.d.). Architecture - Xamarin. [online] docs.microsoft.com. Available at: <https://docs.microsoft.com/en-us/xamarin/android/internals/architecture> [Accessed 28 Apr. 2020].
- [11] developer.rhino3d.com. (n.d.). What are Mono & Xamarin? [online] Available at: <https://developer.rhino3d.com/guides/rhinocommon/what-are-mono-and-xamarin/> [Accessed 28 Apr. 2020].
- [12] Frank Gielen (2012). The Phonegap Architecture. [online] Available at: <https://www.slideshare.net/Inceptum1/case-study-phone-gap> [Accessed 29 Apr. 2020].
- [13] Yumpu.com (n.d.). case-study-on-phonegap-apache-cordova-software-architecture. [online] yumpu.com. Available at: <https://www.yumpu.com/en/document/view/51773453/case-study-on-phonegap-apache-cordova-software-architecture> [Accessed 29 Apr. 2020].
- [14] (n.d.). Magenic | Custom Business and Technology Solutions. [online] Available at: https://magenic.com/media/2353/wp_mgnc-xamarin-vs-phonegap.pdf [Accessed 29 Apr. 2020].
- [15] LogRocket Blog. (2020). Comparing developer experience for cross-platform frameworks. [online] Available at: <https://blog.logrocket.com/comparing-developer-experience-for-cross-platform-frameworks/> [Accessed 29 Apr. 2020].
- [16] www.netguru.com. (n.d.). Cross-Platform Mobile Apps Development – Pros and Cons. [online] Available at: <https://www.netguru.com/blog/cross-platform-mobile-apps-development> [Accessed 29 Apr. 2020].
- [17] www.apterainc.com. (2019). Native vs. Cross-Platform Mobile App Development | Aptera. [online] Available at: <https://www.apterainc.com/blog/native-vs-cross-platform-mobile-app-development/> [Accessed 29 Apr. 2020].