An Analysis of Cross Platform Mobile Application Development

K.Samuel David

PG Student, M.Tech, Computer Science, Invertis University, Bareilly, India

Abstract: Nowadays mobile applications are widely used in almost all the sectors of the world, so it is very much important to evaluate the development process in building information system on the mobile. This paper provides a detail overview on software engineering issues related to the development of mobile application using Cross Platforms mainly Xamarin and Phone- Gap platforms. The review is made to help the software developers to code an application in a single Integrated Development Environment and deploy the application on multiple Mobile Operating System like Android, iPhone and Windows. This is to understand which cross-platform development tool like Xamarin and Phone-Gap is more efficient in developing a mobile application based on their development cost, development technology, and learning curve of the developers.

Keywords: Xamarin Studio; Phone-Gap; Cross Platform; Hybrid Apps; Android; iOS

I. INTRODUCTION

The mobile devices are popular nowadays because of the hardware and software features like the camera, sensor, and touchscreen. The Android, iOS and windows are worth considering as they are well established and evolving operating system in the mobile they have at least 10% market share based on Gartner [3]. They also differ from each other in technology, programming language, implementation and approach, the possession of different technology skillset, has become very challenging for the developer community and enterprises in building applications on mobile to reach wider & diverse audience of target users.

Cross-platform development is a solution to the above-mentioned challenges. The developer to uses unified cross-platform technology that is compatible across the mobile operating systems. This paper outlines our studies, analysis and examination of cross-platform development platforms and tools based on their Graphical Interface, OS platform supported, Learning Curve, Time taken to develop an application and Long Term Maintenance. As these differ in their architecture, implementation approach in the technology skills, it expects the need for establishing criteria to evaluate the cross-platform technologies like Xamarin and Phone-Gap.

A. Xamarin Studios

Xamarin studios is a cross-platform technology developed by Xamarin in San Francisco, which is available on both Windows and Mac operating system. Xamarin grew out of the Mono project in 2011 and allows C# developers to create native executable apps for OSX, iOS (including Apple Watch), Android (including Android Wear), and Windows Phone, Windows Store, and Windows 10 Universal apps [2]. The applications that are developed are both distributed in public and private app stores.

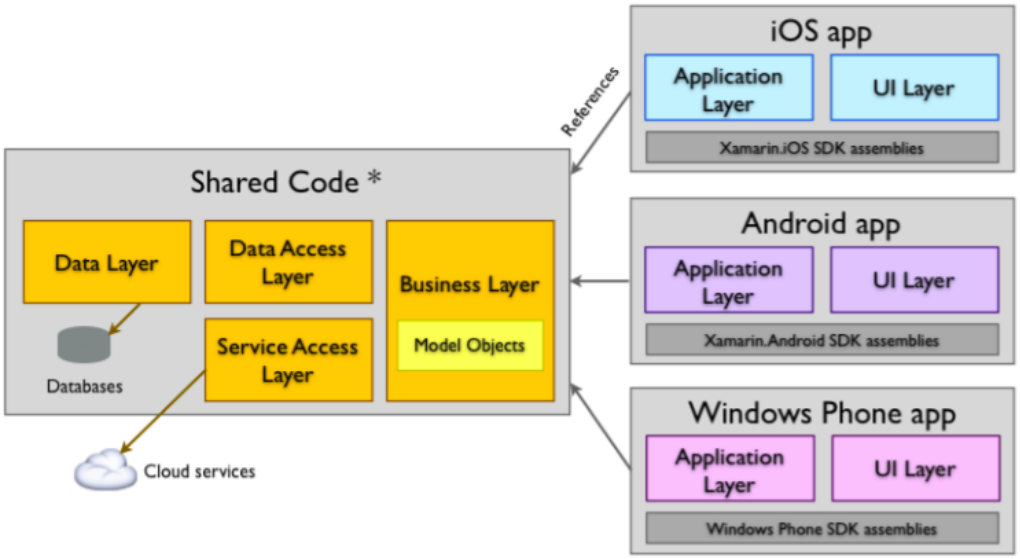

Figure 1: Xamarin Shared Code.

B. ***Phone-Gap***

Phone-Gap is a software development framework that is used to build cross-platform application which was developed by Adobe system. The developers must require knowledge on web development programming languages in order to develop mobile application using Phone-Gap namely Html, CSS, JavaScript. Phone-Gap produces apps for all popular mobile OS platforms such as iOS, Android, BlackBerry, and Windows Mobile [4]. Phone-Gap applications can also be distributed in public and private app stores.

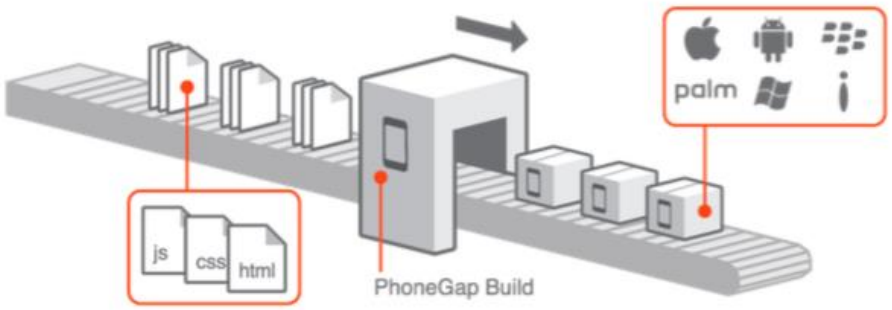

Figure 1: Phone-Gap Build.

II. ANALYSIS OVERVIEW OF THE CROSS-PLATFORMS DEVELOPMENT TOOLS

A. ***Analysis of Xamarin Studio Integared Development Environment***

Xamarin studio is built to develop cross-platform native mobile applications. It consists of a suite of products like Xamarin.iOS, Xamarin.Android, Xamarin.Mac and Xamarin.Forms

- ***Graphical Interface:*** In Xamarin creating and editing the graphical interface of the application it is very similar to the native way in Android Studio or Xcode. This IDE allows to drag and drop elements into the xml file in Xamarin. It also has dedicated flavors of IDEs and interfaces in iOS, Android, Windows and Mac which communicates to the respective native UI APIs. Some functionalities and customizations of the screens and the application are limited.

- ***OS platforms support:*** Xamarin allows you to develop apps for OSX, iOS (including Apple Watch), Android (including Android Wear), Windows Phone, Windows Store, and Windows 10 Universal apps

- ***Learning curve:*** The Developers have to learn the C# programming language. They also have to occur the basic knowledge on basic native application development. The learning part of Xamarin studio is huge. They have three learning curve of studies.

- ***Regular Development time:*** The main objective of cross-platform developments in to reduce the size of code and make it reusable based on the respective native platform. Xamarin also has advance debugging capabilities which helps the developer to debug. The debugging experience in iOS stimulators or Android emulators and physical devices are good. This kind of tooling helps the developer to enforce the development standards.

- ***Long Term Maintenance:*** Xamarin has an ever increasing huge developer base across the globe, presently it is close to half a million [3]. It has more than hundreds of partners in the IT Service industries. Xamarin proactively engages in conferences, rendering technical support and substantive learning material online along with varied SLA support. The C# has better code maintenance and the Xamarin helps in unit testing. This makes the Xamarin studios a better position to deliver long term maintenance.

- ***Performance:*** Applications built using Xamarin will demonstrate the same performance as that of native applications. There will be some reflection overhead incurred as .net runtime assemblies are packaged with the application bundle.

- ***Toolset licensing cost:*** Xamarin studio licensing is based on per developer, per platform and can be paid either on an annual or monthly basis. There are also different subscription levels available depending on if the developer only wants the use Xamarin Studio, if they would like Visual Studio, or if they want more inclusive enterprise support. The applications built using Xamarin studio will bear no license fees/costs.

B. Analysis of Phone-Gap Integared Development Environment

Phone-Gap is a free and open source framework that allows you to create mobile apps using standardized web APIs for the 7-major mobile operating systems [4].

- ***Graphical Interface:*** In Phone-Gap creating and editing the graphical interface of the application it is not like the native way in Android Studio or Xcode. The applications are developed like website development and the design implements are customized through CSS. It is very difficult to build the styles and function perfectly but the benefit is that the objects are not created so the

developer can easily test and develop the app using navigator like Inspect Elements in Google Chrome.

- ***OS platforms support:*** Phone-Gap almost support all the major mobile operating system that are available in the market like Android, iOS, Windows Mobile, Blackberry, Samsung Beta and the web OS.

- ***Learning curve:*** The Learning curve is very much lower comparing other cross-platform application as they are web apps. They should learn web developing programming languages like HTML, CSS, JavaScript and the framework.

- ***Regular Development time:*** Phone-Gap helps to develop apps for seven operating system. So this avoids the repetition of many application and we can also have a small team of developers and this helps in reducing of the development time and cost.

- ***Long Term Maintenance:*** Phone-Gap has large group of developer community. Since the IT companies are showing their keen interest in Phone-Gap and have already launched millions of applications it long term growth can be assured. They are majorly sponsored by Adobe and Nitobi.

- ***Performance:*** Applications built using Xamarin will demonstrate the same performance as that of native applications. There will be some reflection overhead incurred as .net runtime assemblies are packaged with the application bundle.

- ***Toolset licensing cost:*** Phone-Gap is an open source and it is free of cost IDE. It is under the license of Apache 2.0

III. PROS AND CONS OF CROSS-PLATFORM DEVELOPMENT

A. Pros of Phone-Gap cross-platform development

- Even though the Phone-Gap is a server side platform it has large community of developer as there are many experienced developers in JavaScript, HMTL, and CSS.

- We can develop different OS application in same Programming Language. A single standard APIs works across all the device platforms.

- Phone-Gap follow a plugin architecture the APIs can be extended in modular ways enabling the developers to focus on the web-based skills.

- They have lot of free building components that acceleration the speed of developments.

B. Cons of Phone-Gap cross-platform development

- The extended version of any application may be out of date or cannot be deployed to the target platform you need.

- The Hybrid application does the give the user the feel and experience like the native app development.

- It is very difficult to build the large application because of JavaScript's global scoping and library incompatibilities.

C. *Pros of Xamarin cross-platform development*

- The codes that are compiled are platform specific, so it can achieve high performance, use native hardware acceleration, device features and native UI components.
- The language is one of the biggest advantage of developing. Xamarin.Forms that is released in Xamarin 3.0 helps in easy sharing of codes.
- The performance of the app is faster than many other cross-platform IDEs. It also supports full hardware features like camera, accelerometers, and GPS.
- Xamarin coding can be done in Xamarin Studio and Visual Studio. Cross platform becomes even more simple and facile, because of its test cloud, which makes the task truly effortless.
- Xamarin has a component store that accelerators the speed of development process, the developers can either access the component for free or paid.

D. *Cons of Xamarin cross-platform development*

- The cost of Xamarin development tool is not a open source and they are not cheap.
- The Learning curve the developers should know C# and .NET and the frameworks of the native framework and the Xamarin framework.
- Xamarin studio development tool if you are using mac or don't have Visual Studio, its still far away of Xcode or Android Studio. It has improving during this time of course, and is usable but it lacks the features of his native counterparts, and continuous crashes and mysterious file disappearances are routine, not to mention the fear of what will crash now after each system update.
- All these required workarounds are something you cannot plan at the beginning of the projects, nor are problems you can expect if you use standard programming guidelines, and keep adding time, cost and effort.

IV. FUTURE WORK

In this paper, we have only Analysed Xamarin and Phone-gap which is majorly proffered by the development community. We have left out many cross-development platform like Sechna, Appcelerator, and Cocos2d. Many updated plugins that have been updated often so there is lot of possible future works.

Frameworks with many feature and optimized code that are available. I would also like to work on many other feature and try and test them like Cloud Storage, Calendar, Notification, Bluetooth, Map integration and Third party libraries. Performance can also be tested with both lower and higher end mobile devices.

Conflict of Interest: The authors declare that they have no conflict of interest.

Ethical Statement: The authors declare that they have followed ethical responsibilities.

REFERENCES

[1] Gartner (2016). "More Than 50 Percent of Mobile Apps Deployed Will be Hybrid".
[2] Lin, F., & Ye, W. (2009). "Operating system battle in the ecosystem of smartphone industry". In Proc. of the 2009 int. symp. on information engineering and electronic commerce (pp. 617-621). IEEE CS
[3] Adrian Holzer & Jan Ondrus (2008). "Trends in mobile application development". University of Lausanne
[4] Heitktter & Henning (2013)."Evaluating cross-platform development approaches for mobile applications" pp. 120-128
[5] Dzone (2015). "The State of Native vs Web vs Hybrid".