International Journal of Science, Mathematics and Technology Learning
ISSN: 2327-7971 (Print) ISSN: 2327-915X (Online)
Volume 31 No. 2, 2023

Cross-Platform Compatibility to Analyze Behavior of Hybrid Mobile Application Framework

Dr. Rupesh Sendre[1]

[1]Deptartment of International Institute of Professional Studies

Devi Ahilya University, Indore, Madhya Pradesh, India

Abstract

One of the most intriguing challenges in mobile application development is its fragmentation with respect to mobile platforms (e.g., Android, Apple IOS, and Windows Phone). Mobile is an integral part of daily life. With time, customers are expecting good and very versatile applications in less time. There are different number of mobile apps investigated, in this research our aim at identifying, analyzing and understanding the features and distinctions of publicly available hybrid mobile apps within their real-life context. One of the main focuses of the comparison is to provide an overview on the availability of application programming interfaces, programming languages, supported mobile operating systems, licenses and integrated development environments. Furthermore, it also presents some critical points such as the factor of extensibility in tools and the effects that they may bring on market share. The comparison is aimed at supporting developers to make the right choice with respect to their needs or constraints.

Keywords

Mobile application, Hybrid platform development tools, Web-based OS, Native API, Smartphone – Android, IOS, BlackBerry.

---

1. Introduction

Hybrid mobile application development approach is a mixture of Native and Web-based approach. Native applications are specifically and technically designed to run on a device's

[1]Corresponding Author, email: rupesh.sendre@iips.edu.in

Acceptance: 10July2023, Publication: 18July2023

operating system. They are coded with a specific programming language and they are fast, reliable and robust but are attached to a mobile platform. Developers have to replicate them using the appropriate programming languages in order to target various mobile platforms. Portability across time, i.e. language and OS revisions has always been a big aspect of portability HTML, XML and JavaScript versions favors portability as these languages are common in web browsers. At this juncture alternative development of mobile applications influences the developers. One of the alternative approaches is hybrid application development. The native part of the application uses the operating system API's to produce an embedded HTML rendering engine that assists as a bridge between the browser and the device API's. This bridge authorizes the hybrid app to manage the features of modern devices.

1.1. PhoneGap: PhoneGap is a free and open-source framework that allows you to create mobile apps using standardized web API's for the platforms. To develop for iPhone, we need a Mac OS X computer. PhoneGap Lib is a static library that enables users to include PhoneGap in their iPhone application projects. We can also create new PhoneGap-based iPhone application projects through an Xcode project template. Xcode is Apple's development environment for Mac OS X and iPhone that includes the iPhone SDK capabilities of the framework. In case of android a developer needs to install the Android SDK and Eclipse plus the Android Development Tools development plug-in for the Eclipse. ADT extends the capabilities of Eclipse to let you build Android projects and APK's in order to distribute applications [12].

1.2. Sencha Touch: Sencha touch is the best environment for cross-platform development based on HTML5 and CSS3. It enables developers to build powerful applications that work on iOS, Android, BlackBerry, Windows Phone and more. The Sencha Touch API is pure JavaScript. Sencha Touch apps can not only be accessed via browsers, but can also be deployed as hybrid apps using native wrappers. Sencha Touch is not dependent on jQuery, so is compatible with both the iPhone and Android. It uses XML and HTML to create interface design and procedural code for creating a UI object. The latest version of it supports Apache Cordova APIs for camera capture, connection, events, geolocation, media, notification, splash screen and storage. These are few features available to native apps that are essential to app developers [3].

1.3. IWebKit: the IWebKit is a framework focused on being fast, lightweight and specifically for developing web applications and websites for Apple's devices. It can easily be integrated into iPhone application developed in Objective-C and applications developed using Rhodes and PhoneGap frameworks. Developers familiar with HTML and CSS framework can easily use iWebKit. The iWebKit framework includes a comprehensive set of style sheets, icons, JavaScript and some basic template to add to your application.

1.4. Titanium: Titanium is mainly used for native application development for mobile environment. It consists of an SDK that provides the necessary tools, compilers, and APIs for building for the target platform and a visual environment for managing development. It utilized web technologies that are both trendy and powerful, including AJAX, HTML5, CSS3 and jQuery. Titanium is available for Mac, Linux and Windows. Developing for the Android requires the Android SDK and can be done using Mac, Windows or Linux. The Titanium framework comes with a platform-independent API that can make applications feature-rich because it can access advanced features such as touch screens, cameras, GPS, navigation, contacts, storage and much more. Titanium also supports augmented reality features like Screenshot, Shake and Record video apart from Cordova and Cordova-based tools such as PhoneGap, is the fact that you don't have any HTML and CSS files unless you want to create an application that uses both native and HTML based user interface. Titanium ship with a very useful mobile toolset that help you emulate (or simulate) your application on the actual platform, rather than browser. When your app is run on the device, it doesn't get wrapped into Web View, but is interpreted by a JavaScript engine instead (JavaScript Core in iOS or Rhino in Android) [4].

1.5. Rhodes: It supports cross-platform web application development in HTML, CSS, JavaScript and Ruby. Its tool can be used across Mac, Windows and Linux. The user interface of app is created using HTML and CSS. It requires Apple SDK for iPhone or iPad with Mac OS for development of app. For android Mac, Windows or Linux can be used. The Android native development environment is required, but no need of Eclipse IDE. Blackberry is java based with windows to run its tools and no need for eclipse. Rhodes support windows mobile 6 but not 7. MS Visual studio not used with Rhodes. Device capabilities supported by Rhodes in different platform are geolocation, contacts, camera, date\time picker, audio\video capture, Bluetooth, SMS, Landscape orientation and native maps.

1.6. RhoSync: Mobile user can access information even in offline mode on device due to synchronization servers. It is a sync server framework that provides web services to Rhodes based app running on smartphones. It is data stores that stores information as an object attributes values and works as a middle tier between web services and mobile app. In RhoSync ruby support query based information retrieval, data submission, creation, deletion, updating and user authentication [11].

1.7. jQTouch: It supports HTML pages that look like a native iPhone app. It can influence cross platform such as PhoneGap and Rhodes. It is a source code lib that includes JavaScript and CSS. Creating a new app is jQTouch is being simple, but the modification is difficult.

1.8. Adobe Air: It uses the same technology to build web application for different platforms but this framework is not suitable for an app that requires high computation resources. Irrespective of operating and browser get access to the services of the same site, app portability ad rich user interaction. JSON used for data transfer between web server and mobile app. It provides a rich internet application with desktop software and network capabilities and complete control over app by the user. It is a runtime program with no specific language and heavyweight solution. It supports flash, flex, JavaScript, AJAX and HTML.

1.9. Cordova: Development in Cordova is similar to the development needed to build a web page as HTML, CSS and JS all combine to create a web view what is wrapped in Cordova.

1.10. Xamarin: Xamarin has a C# shared codebase using this programmer can write a native Android, iOS and Windows apps with the help of native user interfaces and share program or code to multiple platform focused on C#. Xamarin works in a similar way to Cordova [10].

1.11. DragonRad: DrogonRad is a cross-platform mobile application development platform by Seregon Solutions Inc. and distributed under a commercial license. It helps programmers to program, manage and deploy mobile applications once and use them across Android, iOS, Windows and BlackBerry Mobile. The tool focuses on database driven mobile enterprise applications with easy and wide range of databases support. It provides the D&D environment which help developers to save programming time and to create logics. DragonRad has their own client architecture; it is required to setup server and database based on the needs of programmers with all prerequisites of server and database like Tomcat, MySql etc. DragonRad is commercial tool with the support to its own language D&D, the possibilities of extension in terms of adding plugins and other support to the framework are quite limited. DragonRad facilitates the integration and synchronization of database system with native functions of above defined mobile OS's, such as Contacts, Calendar, Geolocation, Menu and Storage [3].

1.12. MoSync: MoSync is an open source solution developed by a Swedish company targeted to mobile market. It provides the full fledge Eclipse-based IDE and the use of standard C/C++. It provides well documented API's both in C/C++ and web-based. The idea involved to support multiple mobile OS's is different from other tools and also in very isolated way from other mobile operating code. Applications in MoSync are built to target a device profile by using GNU Compiler Collection (GCC) and pipe-tool. After writing the application, pipe-tool is used to compile the resources present in the application. GCC uses it to produce MoSync intermediate language which then fed in to pipe-tool. Then, pipe-tool plays a role as the bridge between MoSync applications to target device profile. The profile database helps the application in ensuring that it has adapted correctly to the device. Runtimes are libraries which are bound to provide support related to all like regarding graphics, audio, communications, input, uniform interface to low level system API's and other device features. MoSync is completely open source and based on the Eclipse for IDE, so it provides to all extendibility in the same way as Eclipse does, i.e. plug-ins or adding external library [1].

2. Material

Traditional software engineering approaches and methods used in the development of desktop applications may not be directly applicable to a mobile environment. Therefore, it is critical to develop and implement suitable processes for the development of mobile applications as there are a number of key issues and challenges that are basically different from traditional enterprise applications. Since mobile software development is still unwieldy, a methodology geared towards supporting the development of such mobile applications is still inadequate. There is still a lack of understanding of some real issues and challenges faced during the development of mobile apps and best practices that can adopt to address those challenges. There are several major issues when looking at the end-to-end process of developing a mobile application, from business discovery and development to support and marketing. The client and website options for distribution were easier for developers, and the process for users to install downloaded apps on their own was a barrier to widespread adoption. Some challenges in the market of mobile development are the short basis of practicality and innovation by all but a few of the largest brands with the budgets to support it. The authors provide a survey of the top development and distribution options consisting of mainly handset and OS vendors including the iPhone, Android, BlackBerry, and Windows Mobile. Further introducing emerging cross-platform solutions covering both proprietary and open-source frameworks, emphasizing building native applications. And identifying techniques for using HTML to create a native look and feel for web applications and services [4].

Authors, identified issues related to mobile application development based on development processes, tools, designing user interface, portability of applications, quality, and security. Main challenges observed in mobile software development project particularly, when building universal user interfaces, reusing software across various mobile platforms, searching context-aware mobile apps, and stabilizing agility and ambiguity in the requirements. However, it should be noted that the traditional software development process may not be directly implemented to mobile device software, including user interface, divergent mobile platforms, and the novelty of a truly mobile computing platform. To provide a student technology support system for university students with first duty was to provide a mobile application second duty was to ensure that we could deliver dynamic content to students quickly and with real-time alerts. Hence, the authors began looking at cross-platform mobile development frameworks like the Mobile Web Framework. The challenge quickly became to build a strong and web-based mobile application [2], [6].

To assist the student or scholar in the teaching-learning process the authors proposed a model of a mobile e-learning application that can run on cross-platform, that is developed on Android and BlackBerry platforms. The application program is created using PhoneGap which uses HTML5, CSS3, and JavaScript based on Hybrid Mobile Application. On the server side, the source code is based on PHP so that the application program can run with the help of the system. The application has a different interface from the website version. The interface is designed to be suitable for a mobile which is has limited screen space. The application program can run on cross-platform, those are Blackberry and Android mobiles [7].

A strategy and approaches used to develop and deliver existing Web and Desktop applications as smartphone apps. In this proposal, authors discuss a variant of the Hybrid development model that utilizes program code translators to translate existing Web or Desktop applications for the target smartphone devices [9]. Various security challenges due to the abundance of mobile software applications in recent years and conversed about the potential risks that smartphone devices are exposed to due to the lack of application development principles and best practices [16]. Creating the same specific apps for each platform demands greater budgeting, time, and hardware investment. To reduce these problems, mobile development frameworks for multi-platforms have been invented, making it possible to create multi-platform apps using the same source code. Then, a new problem arises in front of the programmer the plurality and diversity of these frameworks. This research author discussed that indicates which multi-platform application development framework is most suitable for a given project, implemented as a knowledge-based recommendation system that considered the critical factors of a given project [17].

Based on the material the above-mentioned issues may pose the following challenges:

- Creating universal user interfaces: screen size and resolution.
- Designing context-aware mobile applications: managing the complication of providing applications across various mobile platforms; requirements upfront essential, fixed notion of the context where the changes are nothing, small or expected.
- Enabling software reuse across mobile platforms: Designing context-aware applications (OS platforms, hardware makers, delivery methods, computing platforms).

These issues are imperative and should be considered during the early development process in order to mitigate the impact of poor choices for the successful development of mobile applications. Hence, very few have been investigated and highlighted the best practices implicated in the process of mobile application development. Large and complex software development projects have adopted Agile development practices, such as Scrum, test-driven development and moving away from a process-intensive approach [5].

3. Proposed Work

The objective of proposed research study is to perform the performance analysis of different hybrid mobile application development frameworks. There can be various factors which quality to be a benchmark in evaluating or selecting mobile applications framework cross platform development approaches but we have marked to some of the potential criteria as: Look up and experience, Mobile operating system platform support, Utilization of device capabilities, Regular development on time, Performance, Learning outcomes of device, Development accelerators, Long-term feasibility and developer community group, Packaging and distribution, Platform cost, and licensing model and maintenance etc. [10].

When it comes to choosing cross-platform technologies for mobile application development, many critical factors contribute to the decision making process like: Target audience (geography, language, etc.), Type of business (B2E, B2C), Feature list and its impact on devices (device hardware dependant/independent), Code partition (device and/or server/cloud code), and Time to market, etc. So, by funneling all such decisive factors into architectural mould, we can view the mobile applications in one or many of following perspectives: UI rich application (e.g. photo shop, gallery apps), Applications with intensive computation implementation on device (e.g. games), Information rendering applications (e.g. e-Book reader, sales dashboard apps), Applications with intensive hardware utilization (e.g. torch, map, compass apps), and Applications productively operable in offline mode, etc.

Xamarin Mono touch is designed in such a way that code share or reusability of code happens at the application layer. This means GUI implementation will be repeated for platform specific UI implementation. So in order to reap maximum benefits from Xamarin it is advisable to build the mobile applications which require intensive computation (deep application logic) on the device. It is suitable for applications that can benefit from offline mode operations and also the applications that rely heavily on the hardware capabilities of the device. Phone-Gap is more suited for application which acts as information renderer by pulling the information from the cloud and also for the applications where native look and feel of the application is of least priority. Dashboard intensive applications in sales force sector or applications that monitor an enterprise KPI get an inherent technological benefit from HTML5 and some of the open source of JavaScript frameworks. By deciding on Phone-Gap for such applications development life cycle is greatly reduced and hence favors go-to-market timeline of the application. When we have to choose an intermediate path by giving significant consideration to look and feel, where UI is not a top priority, then the enterprise business applications that requires inherent analytics and features like MAM and MDM can be built using the Titanium development platform. A rich set of specific APIs of Titanium development platform provides close to native look experience [10].

4. Discussions

From table-1, it is easy to understand that Phone-Gap offers more compatibility for development on different mobile OS's, which is optimal to gain the maximum profit for both developers and business model owners because one built application can be registered at all respective business models, whereas IDE's hosted in personal computer is the same for each platform.

Table-1: Comparison on mobile platforms compatibility and development environments OS support

Tool Name	Mobile OS Support	OS Support
Rhodes	Android, BlackBerry, iOS, Symbian, Windows Mobile, Windows Phone	Linux, Mac, Windows
Phone-Gap	Android, BlackBerry, iOS, Symbian, Web OS, Windows Phone	Linux, Mac, Windows
DragonRad	Android, BlackBerry, iOS, Symbian, Windows Mobile	Linux, Mac, Windows
MoSync	Android, BlackBerry, iOS, Symbian, Windows Mobile	Linux, Mac, Windows

.

Table-2, describes several characteristics of the tools presented in the paper, such as supported programming languages, accessibility to native APIs, IDE and plug-in extendibility. Starting from the programming languages, Rhodes, PhoneGap and MoSync are the only tolls that have support for web programming languages, such as HTML, HTML5, JavaScript and CSS. Differently to other tools, PhoneGap provides support for CSS3, whereas MoSync for C and C++ instead DragonRad has support for its own language D & D.

Since Rhodes, PhoneGao and MoSync are web oriented development tools, the access to native APIs is done through JavaScript APIs. In addition, MoSync also offers the accessibility through C/C++ APIs. The second side, DragonRad have their own APIs to interface the application layer with the operating system layer. With respect to available IDEs, Rhodes has two useful ways to develop applications; in fact it provides some different solutions as RhoStudio IDE to develop in locate and RhoHub IDE to develop via remote connection. Furthermore, it provides the 184 possibility to use alternative IDE's such as Eclipse, Visual Studio, Netbeans, IntelliJ, Textemate, etc. MoSync offers an Eclipse based IDE, whereas DregonRad offers its own IDE solution. Only, PhoneGap is the platform that has kinds of approaches among tools, indeed it offers an extension that could be applicable on all native IDE's. Some examples of native, IDE's are Xcode for iOS, Eclipse for Android and Visual Studio for Microsoft, etc. This kind of result is good but limits the flexibility for programmers to use the IDE of their choice. A careful reader would argue, what are the advantages of this approach if developers have to create applications on different IDE's [11].

It is worth nothing that PhoneGap, MoSync and Rhodes are the tools that support HTML, CSS and JavaScript languages; since these languages are not native the source code will be the same for all platforms. The only practical approach is required by programmers in using PhoneGap is to program applications on an IDE and perform a simple porting of the source code in other IDE's [12], [15]. Apart from DragonRad, other kind of tools provides the possibility to add plug-ins in the IDE and also give the feasibility to develop plug-ins from grid and then add it to the IDE.

Table-2: Comparison on development features

Tool Name	Language	Accessibility to Native API	IDE	Plug-in Extendibility
Rhodes	HTML, HTML5, JavaScript, CSS	JavaScript	RhoStudio, RhoHub	Yes
PhoneGap	HTML, HTML5, JavaScript, CSS, CSS3	JavaScript	IDE native of mobile OS (Eclipse Xcode)	Yes
DragonRad	D & D	na	DragonRad Designer	No
MoSync	HTML5, JavaScript, CSS, C, C++	JavaScript, C, C++	Based on Eclipse	Yes

The license of tools is second helpful parameter for the comparison on development features. As shown in table-3, the licenses available for these tools are MIT, Apache, General Public License (GNL2) and Commercial. The first two licenses are free and moreover they offer an open-source support. This kind of approach is much important for all developers that would develop applications and want to provide a support to the development platform without having commercial restriction.

On the other hand, commercial licenses could be useful for companies that want to receive the support directly from the manufacturer. Other frameworks like Rhodes are the only framework with MVC support. One of the most important benefits of the MVC design pattern is the possibility to develop applications in a distributed way by means of the model, view and controller with the aim to separate one part from each other; so the advantage is to modify each part independently. Another benefit that MVC offers is the easy migration of legacy programs, because the view is separated from the model and controller. Furthermore it provides an environment that embeds different technologies across different locations and architecture that better support the scalability [13].

MVC framework provides support to native mobile applications, the ability to write real business logic on local native applications. This explains the reason of so many robust enterprise applications written with it [8], [15].

Table-3: Comparison of general features

Tool Name	License	Open Source	Architecture	MVC
Rhodes	MIT, Commercial	Yes	Local Web	Yes
PhoneGap	Apache	Yes	Local Web	No
DragonRad	Commercial	No	Translate	No
MoSync	GPL2, Commercial	Yes	Local Web	No

Rhodes, PhoneGap and MoSync are the tools that provide architecture of interfacing among JavaScript, API's layer and native API's layer. These platforms use a native language of mobile OS to access the hardware and software resources with the purpose to add basic functionalities to the JavaScript Engine and make it easy to understand for the application as the library methods. Based on mobile OS in which the tool is interfacing the user-code will be converted in native-code, such as Objective-C for iOS, Java for Andriod, etc. Some important API's that represent hardware and software functions are listed in Table-3. But in MoSync, the method for interfacing is available both though C/C++ and JavaScript API's. The MoSync has different library named as "WormHole" when the application is build in by using web technologies, wormhole is responsible for interacting with the native applications of the selected mobile operating system. In other hands DragonRad, it provides hardware integration in its own method which is either by using different dependent libraries or by following some of defined wizards [14].

5. Significance of Study

Hybrid applications are easier to develop as well as maintain because you are working with only one platform codebase rather than multiple platform specific codebases. Once you start adding additional platforms with a single line of code, you end up developing a hybrid application. Developers do not need to learn additional languages and frameworks to develop platforms and different versions of the same application, which simplifies the development of hybrid applications. Additionally, as hybrid applications rely on one language for all platforms, hybrid applications are ready for use on both platforms when their development is finished.

6. Conclusion

In this paper, we discussed cross-platform mobile learning development, through the comparison and contrast we have proposed that adopting a hybrid app technique is most suitable for such mobile app developers. From this study, we have learned that there are several frameworks that are feasible for hybrid cross-platform mobile app with each of them having some pros and cons. In order to avoid re-implementation of the same applications for different platform separately, it is vital to compare and make use of the strength of individual framework or the combination of frameworks, so that a better more compatible and more stable cross-platform application can be developed. Studies also show that there are many existing learning systems that are not being explored fully and will probably have a bright future if some more functionality is considered and proper implementation is carried out. Therefore, in our future work we will still continue the studying of the latest cross-platform framework, device features accessible by these frameworks and implement a learning system incorporating new features. Therefore, this study is important for application development in cross-platform mobile learning.

7. Acknowledgements

First, I would like to thank Rahul Singhai, Naresh Dembla and Saurabh Jain for their insightful suggestions and I am extremely grateful for the time they spent working with me on these areas. I would like to thank PG students for providing an amazing experimental environment.

References

1. Andre Charland & Brian LeRoux. Mobile Application Development: Web vs. Native [J]. ACM-Queue, 2011, 9(4) 20-28.
2. Anthony I. Wasserman. Software Engineering Issues for Mobile Application Development [C]// FoSER'10: Proceeding of the FSE/FDP Workshop on Future of Software Engineering research, 2010, 397-400.
3. Adrian Kosmaczewski. Mobile JavaScript Application Development [M]. USA, O'Reilly Media, 2012, 1-168.
4. Sarah Allen; Vidal Graupera & Lee Lundrigan. Pro Smartphone Cross-Platform Development [M]. USA, Springer, 2010, 1-255.
5. Ken Schwaber. Agile Project Management with Scrum [R]// Microsoft Press, 2004, 1-175.

6. Bobby Siegfried. Enhanced Student Technology Support with Cross-Platform Mobile Apps [C]// Proceedings of the 39th Annual ACM SIGUCCS conference on User Services, California, 2011, 31-34.
7. Djoni Haryadi Setiabudi; Lady Joanne Tjahyana & Winsen. Mobile leaning application based on Hybrid mobile application technology running on android smartphone and blackberry [C]// Proceedings of the International conference on ICT for Smart Society, 2013.
8. Daniel Vazquez Sanchez; Erika Hernandez Rubio; Elena Fabiola Ruiz Ledesma & Amilcar Meneses Viveros. Student role functionalities towards learning management systems as open platforms through mobile devices [C]// Proceedings of the IEEE International conference on Electronics, Communications and Computers, IEEE, 2014, 41-46.
9. Parag Gokhale & Sachchidanand Singh. Multi-platform Strategies, Approaches and Challenges for developing Mobile applications [C]// Proceedings of the International conference on CSCITA, 2014.
10. Umesh Narayan Gondhali. An Analysis of Mobile Application Development Approaches [R]. Happiest Minds Technologies, 2014, 1-10.
11. Manuel Palmieri; Inderjeet Singh & Antonio Cicchetti. Comparison of Cross-Platform Mobile Development Tools [C]// Proceedings of the IEEE 16th International conference on Intelligence in Next Generation Network, 2012, 179-186.
12. Adobe PhoneGap Build FAQ. Adobe Systems Incorporated, https://build.phonegap.com/faq, 2013.
13. Chen Xin. Cross-Platform mobile phone Game development environment [C]// Proceedings of the IEEE International conference on industrial and information systems, Haikou, 2009.
14. Damianos Gavalas & Daphne Economou. Development platforms for mobile applications: status and trends [J]. IEEE Software, 2011, 28(1), 77-86.
15. David Jaramillo; Viney Ugave; Robert Smart & Sudeep Pasricha. Secure Cross-Platform Hybrid Mobile Enterprise Voice Agent [J]. IEEE, 2014, 1-6.
16. T. F. Bernardes & M. Y. Miyake. Cross-platform Mobile Development Approaches: A Systematic Review [J]. IEEE Latin America Transactions, 2016, 14(4) 1892-1898.
17. Denisson Santana dos Santos; Hugo Doria Nunes; Hendrik Teixeira Macedo & Alberto Costa Neto. Recommendation System for Cross-Platform Mobile Development

Framework [C]// Proceedings of the XV Brazilian Symposium on Information Systems, Brazil, 2019, 69, 1-8.