Exploring Mobile Application Development Tools

Monika Kohli, Harmeet Kaur

Abstract: - With the advent of Smartphone, constant up gradations in developing Smartphone's through apps are growing at a rapid speed. Wide variety of platform, operating system and tools market is growing. In this paper, we have discussed different tools which are available for native and cross-platform mobile application development. Long-term and viable solution will prevail the market and comparing different tools and techniques in app development help the developer to choose as per the requirement. Cross-platform apps development is also escalating but Native app development is a better contender so far. But Developers are migrating to cross platform application development tools in order to reduce the cost of development and reach out to maximum users across several platforms.

Keywords- Smartphone's, developing, different, Cross-platform, Developers, tools, platforms

I. INTRODUCTION

Mobile devices bring sea change in information and communication technology. The handheld device not only make people around the world connected but also encourage them to get benefitted with this technology using different type of apps. It has the prospective to attract every person from any sphere may be health care, social media, information, entertainment etc. Mobile device is the most economical device which one can buy to have internet access. Another advantage is that it is a handy tool and the launch of services by private operator provides facilities to not only urban but to rural part of the country too. There are number of applications provided which can be beneficial to individual and to the society as a whole. Number of platforms with different tools encourages developers to take up new challenges and engage with different app ideas. For Example, iOS uses Xcode, Dashcode, Android uses Eclipse, Android Studio, Windows 8 uses Visual Studio Express or Visual Studio, Blackberry uses Eclipse, Cascades etc. Various programming environments and development tools are available for native app, web app and hybrid app.

II. APPROACHES TOWARDS MOBILE APP DEVELOPMENT

No. of mobile apps with user friendly interface are increasing day by day making life effortless. Different approaches can be followed but suffer from some limitations as well. Some of these approaches are discussed briefly:

Cross-compiler: It separates the build environment and the target environment. Developer uses platform independent API to build the mobile app which is the transforms into platform specific native app by cross-compiler. It give a native app like experience to the user in terms of performance and abilities to use camera, sensors etc. But it is not easy to write cross-compiler.

Virtual Machine: In this technique, framework provides both APIs and runtime environment to run the application. The main advantage is that it is easier to maintain and is more flexible than cross-compiler technique with a limitation that it run a bit slower.

Web-technology standard: It involves using web technologies like HTML, JavaScript, CSS including embedded SQL databases, local storage, animations, web sockets etc. to build the app .In the hybrid model, Most use a mix of HTML, CSS, and JavaScript, plus some native "wrapper" code for accessing hardware features like GPS, a camera, or an accelerometer. This approach is cheap and covers wide range of platforms. It might be good for business applications but not suitable for interactive and visually rich applications like games.

Native apps (Android apps- written in Java, Apple iOS apps- written in Objective-C) are platform specific i.e. they can run on the specific device they are coded for. As each platform is different, developing a native application which requires a platform specific programming language and software development kit typically means having to learn a development language and maintain a code base for each platform and then one can launch them from the Play store. A key idea behind native development is to deliver the optimal user experience or to get the most efficient and direct access to device hardware.

Android app for example, use the Eclipse development environment and java programming language. To run the same application on apple iPhone, iPad or iPod, we need to develop an application for those devices and it requires the use of Xcode, a specific integrated development environment (IDE) and programming the application in Objective-C. But there is a competing need to deliver application functionality across different device vendors that may be Web app or hybrid app. The increase in the smartphone market gives solution to port the existing applications that have become mature, stable and secure with fixes over time to new operating systems. It needs a common application development framework. The application development on this common platform will then be able to use on any other device to which this framework can be ported. It thus reduces the drawback which native apps may suffer. The benefit of cross platform tool is the reduction in coding and programming required by the programmer. Developing cross-platform mobile web applications with a single codebase is one strategy that allows organizations to stay ahead of the mobile curve. Number of tools is available which further reduce the knowledge of APIs. It further reduce the development time and maintenance cost.

Revised Version Manuscript Received on September 07, 2015.

Monika Kohli, Assistant Professor, Government College for Girls Ludhiana, Bharat Nagar, Sita Nagar, Ludhiana (Punjab). India.

Harmeet Kaur, Assistant Professor, Government College for Girls Ludhiana, Bharat Nagar, Sita Nagar, Ludhiana (Punjab). India.

It basically reduces the overall effort needed to develop mobile apps for each platform as in the case of native apps.
Web apps are built using Web technologies such as HTML, CSS, and JavaScript and are accessed via mobile browsers. Fig[1] shows the general view of a cross -platform app development. They suffer a limitation that they lack access to some features which are device specific only limited set are available with HTML5 [1].

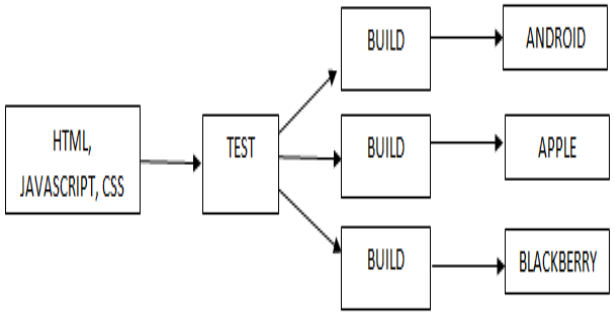


Fig [1]: Cross-platform app development work flow

Hybrid apps such as Apace Cordova[2](earlier Phone Gap)use native component that provides access to device features. Both, Web and hybrid apps, look and behave like Web sites, because the browser engine is responsible for rendering the UI. Titanium [3] uses a separate run-time environment are, to build up the UI with native components. On each target platform, an interpreter at runtime interprets the source code of the app written in a scripting language.
Many Cross-platform development tools like Rho Mobile's Rhodes, Mo Sync, Appcelerator, and Phone Gap, can be used to create native applications on various brands of Smartphone's. Many other tools such as Corona, Widget Pad, Sencha, Titanium, TotalCross are also available but they do not have the versatility of supporting a wide range of mobile OS's. Both native and cross-platform apps has advantages and disadvantages, briefly given in Table[1]. Isabelle Dalmasso et.al in[4] discusses various criterion factors on behalf of which you can decide to opt among different type of apps.

Table[1]: Native and cross-platform app (advantages and advantages)

	Advantages	Disadvantages
Native Apps	1. Rich user Interface and experience as direct access to device OS-specific features and hardware. 2. Provides good security. 3. Full technical support with high quality user experience.	1. It limits the users as it is device specific. 2. Difficult to maintain different codes. 3. Development time increases and increases the probability of errors. 4. Both initial development cost and maintenance cost increases.
Cross-platform Apps	1. It increases the range of app users as it is not device specific and can run on multiple devices. 2. Development cost is less.	1. Its User Interface is not as good as Native apps. 2. Security is a major concern in cross-platform apps.

III. LITERATURE REVIEW

This work has been motivated by some previous work in which need and importance of mobile app is discussed with number of tools. Mobile is a handy tool and the launch of services by private operator provides facilities to not only urban but to rural part of the country too. There are number of applications provided which can be beneficial to individual and to the society as a whole. Number of platforms with different tools encourages developers to take up new challenges and engage with different app ideas. The use of mobile technologies for the development of mobile applications (m-apps) for real-time projects involved in e-Governance[5][6]. Morten et.al in [7] provides comparison and conclude that Android and Windows Mobile provide a better solution than Java ME. [8] discusses the essential to apply software engineering processes in order to development secure, high-quality mobile applications. New touch screen interfaces guidelines are suggested in [9] which enable blind people that will be beneficial for both blind and sighted people. Component based web development [10] gives a way so that components can be reused and can migrate to new platform rapidly and easily. [4][11] compared and discussed cross platform tools and concluded that each one has pros and cons regarding different features.

IV. NATIVE APPS

4.1 Android Mobile application development tools.

4.1(a) J2ME IDEs: Software development for mobile devices using simulation environments from development kits such as Java Micro Edition (JME). These simulators, however, lack of complete picture of the real-world behavior of mobile computing devices. Moreover, such learning technique only emphasizes more on the programming solutions needed to develop mobile applications but less on the importance of having a systematic sequence of process for the mobile application development project.
4.1(b) *Android SDK*: The Android Software Development Kit (Android SDK) contains the necessary tools to create, compile and package Android applications. Android debug bridge (adb): The Android SDK contains the Android debug bridge (adb), which is a tool that allows you to connect to a virtual or real Android device, for the purpose of managing the device or debugging your application. The Android Developer Tools (ADT) are based on the Eclipse IDE. ADT is a set of components (plug-ins), which extend the Eclipse IDE with Android development capabilities. Google also supports an IDE called Android Studio for creating Android applications. This IDE is based on the IntelliJ IDE.

Features of Android:

Compilation: With Android 4.4, Google introduced the Android RunTime (ART) which is a optional runtime for Android 4.4. It uses as default runtime for all Android versions after 4.4. ART uses Ahead Of Time compilation. During the deployment process of an application on an Android device, the application code is translated into machine code which results in 30%(approx.) larger compile code, but allows faster execution from the beginning of the application.
Battery Usage: the compilation is only done once i.e. during the first start of the application which saves the battery life.
Garbage Collection: The garbage collection in ART has been optimized to reduce times in which the application freezes.

The dex2oat tool takes the .dex file created by the Android tool change and compiles that into an Executable and Linkable Format (ELF file). This file contains the dex code, compiled native code and meta-data. Keeping the .dex code allows that existing tools still work.
Security:
Permission during installation: The check of the permission is only performed during installation; permissions cannot be denied or granted after the installation.
Services and libraries for Android application programmers: Google Play also offers an update service. If a programmer uploads a new version of his application to Google Play, this service notifies existing users that an update is available and allows them to install the update.

4.1(c) *Google MIT Inventor*

MIT App Inventor / MIT App Inventor 2
App Inventor for Android is an open-source provided by Google and maintained by MIT(Massachusetts Institute of Technology). It has a graphical interface which assist beginner to create applications.
Characteristics of in MIT App Inventor 2:

a) Visual blocks-based programming language, with Interface designer.
b) .apk installer packaging option is available to install on the android enabled device directly. Emulator is also available.
c) Limited debugging tools built into IDE but easy to learn for beginners
d) Integrated Development Environment: Web-based interface designer, with connection to Java web-start program for blocks programming.
e) Android devices are used Cross-platform deployment.
f) No deployment tool costs.

4.2 Apple iOS

SDK (software Development kit) for developing apps for Apple iOs devices is built into Applet's programming environment. Apple iOs devices like iPhone, iPad, iPod-Objective-C and XCode are the developer's tool and are exclusively available from Apple App Store. No separate installation of an SDK or plugin is required. Mac operating system is needed to develop iOS applications. Mac Mini is considered to be more sufficient to create iPhone and iPad apps.
Mark et.al discussed in[12] the comparison of iOS and Android. Apple presents a stable, exclusive platform for app developers with clearly specified tools, defining both their potential and boundaries. This makes it much convenient for the iOS developer to ensue with the task ahead.
[13] Discusses various reasons of selecting Apple's iOS mobile devices as the target device. Number of trends can be identified by the various m-health apps in Apple's App Store.

V. CROSS-PLATFORM APPS

Cross platform apps can run on different mobile device with different operating system. It is a technique of writing a single codebase for apps that will eventually be used on different operating systems

Although Native apps provide fast reliable and most responsive to the user, can tap wider flexibility of device, has the access to the core device capabilities yet the stumbling block is that they are platform specific and can be expensive. Moreover, the type of app one need develop also to be taken into account e.g. Chat app. One need to develop it for every platform or other way is to develop it using cross platform tool. A Comparative Analysis in [14] investigates the effectiveness of the Cross-platform Development Approaches effectiveness in practice.

5.1 Phone Gap:

Phone Gap is used to build applications for Apple iOS, Android, Windows Phone, blackberry , palm. It eliminates the need of compiling mobile apps for multiple platforms. is a standards-based, open-source development framework for building cross-platform mobile apps with HTML, CSS and JavaScript for iOS, Android™ and Windows® Phone 8. It is easy to use and allows the developer to work with device hardware features such as accelerometer, GPS/location, camera, sound. The PhoneGap Developer App helps to develop app locally and then see the changes instantly on your mobile device with our cross-platform app. JavaScript API provides access to hardware features. The phoneGap engine provides native functionality and user can have web view of the app. PhoneGap is an open source distribution of Cordova. PhoneGap was donated to the Apache Software Foundation (ASF) under the name Apache Cordova. Through the ASF, future PhoneGap development will ensure open stewardship of the project. It will as remain free and open source under the Apache License, Version 2.0. The coding environment for PhoneGap app can be any text editor such as vim, notepad as HTML, CSS & JavaScript is used. The application is developed directly in the device OS's preferred IDE or IDE plugins. Plugins give access to device and platform functionality that is ordinarily unavailable to web-based apps. Utilities such as CLI, plugman utilities are used to test these plugins. All the main Cordova API features are implemented as plugins. CLI(Command-Line Interface) utility can be used to apply plugins. plugman utility to whether the plugin installs correctly for each platform. Accelerometer, multi-touch-gestures, scanning of barcodes, can be added using plugins. PhoneGap Build compiles an app for different platforms in the cloud which eliminates the need of installing the platform SDKs. PhoneGap apps are compiled and then chose the required platform.The major advantage is that the performance is better when measured in terms of memory, CPU usage and power consumption. PhoneGap with Sencha Touch 2.0 work significantly well when available memory is not an issue and better UI is desired as discussed in[4][15].

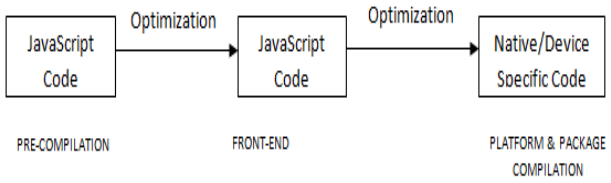
5.2. Rho Mobile

The Rhodes framework support the creation of cross-platform mobile apps. Runtime VM helps to port and run the application on number of devices. The open source programming language, Ruby is used for building the business logic for the mobile app.

It provides Rhom provides database-independent data persistence. This framework provides Rho Sync, Rho Hub, Rho Gallery tools. Rhosync is for synchronization and back-end integration, Rho Hub for app development and deployment and Rho Gallery for hosted cloud management. It uses a cross-compilation approach with different build environment and the target environment. Source code is written using Ruby language and ruby interpreter is used o the device e.g. to run the app on android device, code is cross-compiled into java bytecode to run it natively on the android device.

5.3 Appcelerator Titanium

Appcelerator Titanium's also uses cross-platform approach is based on the cross-compilation technique. It uses JavaScript to implement logic and data, to create the user interface. Titanium engine interprets JavaScript code and creates the user interface. It is carried in three steps: pre-compilation, front-end compilation and platform and package compilation.



Fig[3]: Titanium engine cross-compilation

The pre-compilation takes the app's JavaScript code, optimizes it (reduces whitespace, reduces the size of symbols, etc.) and then creates a dependency hierarchy of all the Titanium APIs used. The front-end compilation step generates the appropriate platform-specific native code, native project (if necessary) and build any specific code that is necessary to compile Titanium for a given platform compiler as given in fig[3]. The platform compiler and packager step effectively compiles the code to native executable using platform specific tools and packages files for running either on the native simulator, native device for testing or for final packaging for distribution.

5.4 Mo Sync

Mo Sync is completely open source and based on the Eclipse for IDE, so it provides to add extendibility in the same way as Eclipse does, i.e plugins or adding external library. It uses pipe-tool and GCC (GNU Compiler Collection) for building the application. Pipe-tool is used to compile the resources present in the application. GCC convert the path of a target device profile passed to it into an intermediate language which is fed into the pipe-tool. Pipe-tool acts as a bridge between MoSync applications to the 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 can be an alternate choice as it does not hve the shortcoming which phone gap suffers. For MoSync, one has to maintain one project for all platforms except iOS which need to use Xcode. Some features are :

1. One project structure for all the platforms

With MoSync, only one project is needed to maintain for all the platforms. Except for iOS.

2. The same JavaScript file

The entire provided functionality for JavaScript is placed in the same file for all of the operating systems. There are no files for plugins.

3. Native UI support

Native UI elements that are more responsive using only JavaScript.

5. JavaScript functionality can be extended using C++ or Java and Objective-C and if you need to code in a native environment to use some specific features of a platform you can always modify the MoSync runtimes and code in Java and Objective-C

6. Built-in support for push notifications

MoSync provides built-in support for push notification with a unified interface on every supported platform.

Some of the basic difference in the cross-platform tools are provided in table [2].

	PhoneGap	RhoMobile	Titanium	MoSync
Licenses	OpenSource	OpenSource	OpenSource	GPLv2 ,commercial
Framework/Development Platform/IDE	Native of the mobile OS.(e.g. iOS,Xcode)	RhodStudio,RhoHub and alternate can be Eclipse, Visual Studio, IntelliJ, Netbeans, Textemate	Titanium Studio	Eclipse based IDE
API	It let user's easy access to over 300 APIs and location infomation.	JavaScript API.	Provides API, customizable through a JavaScript API. a large number of API methods are for iPhone only.	It provides well documented API's both in C/C++ and web-based
Plugins	Yes	Yes	Yes	Yes
User Interface	PhoneGap do not have any design tool for good UI output.	UI is provided using HTML, CSS, JavaScript. Quality can be better using template programming.	Titanium uses native UI elements and thus requires quite a lot of API commands.	Native UIs can be created using C++,JavaScript, HTML5.
Programming Languages	HTML, JavaScript, CSS	HTML, CSS, JavaScript, and Ruby	HTML, PHP, JavaScript, Ruby and Python	JavaScript, PHP, Ruby, Python. MoSync now includes Eclipse-based IDE for C/C++ programming. It also added the support with web-based language like HTML, HTML5, CSS and JavaScript
Crossplatorm/Supporting mobile OS	Android, Bada,Palm, Symbian, BlackBerry, Wndows 7, iPhone, iTouch etc.	Android, Windows Mobile, Symbian, iPhone and RIM.	iOS, Android	Windows Mobile, Android, Symbian, Moblin and even a mobile Linux distro

Table [2]: Comparison between cross platform tools

VI. CONCLUSION

While several tools and techniques are available to work, each one is diagnosed with drawbacks. One can use one of these approaches, but has to consider snags and proceed with care. Though cross-platform app development is very demanding today, yet it faces number of challenges. Mobile app development using Google's Android and Apple's iOS is considered a viable solution. Cross-platform apps development is escalating but Native app development is a better contender in terms of user interface, services etc. especially in game developing. As native apps are more reliable, fast, smooth therefore it's been choice for more, an endeavor which significantly require more time and resources. To gain competitive advancement in business world, making apps platform independent, further enhancement in cross-platform tools are required to meet the current challenges. Future scope can be to develop a library which can be shared by two platforms, have platform specific code.

REFERENCES

1. HTML5, 2012. http://www.w3.org/TR/html5/.
2. Apache Cordova, 2012. http://incubator.apache.org/cordova/.
3. Appcelerator, 2012. http://www.appcelerator.com/.
4. Dalmasso, Isabelle, Soumya Kanti Datta, Christian Bonnet, and Navid Nikaein. "Survey, comparison and evaluation of cross platform mobile application development tools." In Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International, pp. 323-328. IEEE, 2013.
5. Kanwalvir Singh, Himanshu Aggarwal, Design of e-Land Record Information System with Google Map Using Mobile Commerce, Journal of Software Engineering and Applications, 2013, pp. 221-228
6. Bhargava, Bharat, Pelin Angin, and Lian Duan. "A Mobile-Cloud Pedestrian Crossing Guide for the Blind." In International Conference on Advances in Computing & Communication. 2011.
7. Grønli, Tor-Morten, Jarle Hansen, and Gheorghita Ghinea. "Android vs Windows Mobile vs Java ME." In Pros 3 Intl Conf on Pervasive Technologies Related to Assistive rd Environments. 2010.
8. Wasserman, Anthony I. "Software engineering issues for mobile application development." In Proceedings of the FSE/SDP workshop on Future of software engineering research, pp. 397-400. ACM, 2010.
9. Kane, Shaun K., Jacob O. Wobbrock, and Richard E. Ladner. "Usable gestures for blind people: understanding preference and performance." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 413-422. ACM, 2011.
10. Pan, Biao, Kun Xiao, and Lei Luo. "Component-based mobile web application of cross-platform." In Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on, pp. 2072-2077. IEEE, 2010.
11. Smutny, P. "Mobile development tools and cross-platform solutions." InCarpathian Control Conference (ICCC), 2012 13th International, pp. 653-656. IEEE, 2012.
12. Goadrich, Mark H., and Michael P. Rogers. "Smart smartphone development: iOS versus Android." In Proceedings of the 42nd ACM technical symposium on Computer science education, pp. 607-612. ACM, 2011.
13. Liu, Chang, Qing Zhu, Kenneth A. Holroyd, and Elizabeth K. Seng. "Status and trends of mobile-health applications for iOS devices: A developer's perspective." Journal of Systems and Software 84, no. 11 (2011): 2022-2033.
14. Xanthopoulos, Spyros, and Stelios Xinogalos. "A comparative analysis of cross-platform development approaches for mobile applications." InProceedings of the 6th Balkan Conference in Informatics, pp. 213-220. ACM, 2013.
15. Heitkötter, Henning, Sebastian Hanschke, and Tim A. Majchrzak. "Evaluating cross-platform development approaches for mobile applications." In Web information systems and technologies, pp. 120-138. Springer Berlin Heidelberg, 2013.