Software Engineering Challenges in Multi Platform Mobile Application Development

Hammoudeh S. Alamri, Balsam A. Mustafa

1Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Pahang, Malaysia. Ha.amri@gmail.com, balsam@ump.edu.my

Mobile application development for smart phones has been increasing in last years as mobile phones have become an essential part of our contemporary life. As with any new phenomena, mobile application development has its own set of challenges, which attracted interested researchers in this new field. Nowadays, mobile applications become more complex, moving beyond inexpensive entertaining applications to more business critical uses, thus, it becomes essential to apply software engineering process to assure the development of secure, high-quality mobile applications. This paper highlights important issues in mobile application development using native development kits and cross-platform frameworks; it also discusses challenges in the software engineering activities for mobile applications development.

Keywords: Software Engineering, Mobile Application, Cross-Platform, Smart Phone, Mobile Operating Systems.

1.0 INTRODUCTION

Mobile application development for smart phones has been increased in latest years; with widespread of smart phones especially Android OS from many phone manufacturers and iOS on iPhone, iPad and iPod devices from Apple. This encouraged others to enter this new market and develop their OS’s for handset and tablets devices like Window phone, Ubuntu, Tizen and Firefox OS. Each platform of these operating systems has its source development tools (SDK) which are used to develop “Native Application”. As a result of this diversity, cross-platforms have been appeared to help developers to develop their applications and reuse the code across different platforms. According to Hartmann et al., these cross-platforms have different types which can be summarized as: Library: its small toolkits provide some functions like user interface (UI) widget or 3D graphics libraries, Framework: A set of libraries, software components and architecture guidelines, Platform: A set of frameworks, tools and services that not only allow developers to build a complete mobile application but also to configure, package and distribute it to apps’ stores or the cloud, Product/Service: Offers a specific functionality or service ready to be used and integrated into a mobile solution. Research in this area has explained that, platforms and libraries can be categorized into: Cross-compiler: in this approach cross-platform compiler converts the source code to native binaries, Virtual Machine: in this approach there is an virtual machine or interpreter which executes the code at runtime, Hybrid: hybrid application is developed using web technologies and gets executed inside native container on the mobile device and Web Based Platforms: is a web application which is designed to execute in the web browser of the mobile device where the Web Based and Hybrid libraries are of more popular use. The main disadvantages in all these approaches can be summarized as (1) poor performance, (2) poor native user interface, (3) inferior or limited device hardware accessibility. Since mobile devices have special hardware sensors, limited power consumption issues, and need to consider special security issues, thus, mobile applications development requires special effort and strategy for the development life cycle. Many developers and application publishers currently use cross-platforms for their advantages especially those related to cost and maintenance. However, several studies and statistics report that the better performance are the native applications. Because the native application is stable,
secure, and more capable for accessing mobile resources than cross-platforms this makes it the choice for most game, social, and communications applications developers.

Some practitioners have noted the importance of native applications as Mark Zuckerberg – Facebook CEO – (2012) stated: “The biggest mistake that we made, as a company, is betting too much on HTML5 as opposed to native... because it just wasn’t there” 14.

Previous research in this area pointed out the problems of mobile application development as mainly in providing non-functional requirements such as, performance (efficient use of device resources, responsiveness, scalability), reliability (robustness, connectivity, stability), quality (usability, installability), and security.

For this study, the authors have conducted critical review for many previous researches in the area of mobile application development, and several problems have been figured out of these studies. Based on practical experience in the industry of mobile devices and its application, the authors tried to compare challenges presented in previous researches with others faced practically from different case studies to conclude solutions and highlights important issues that need to be considered in software engineering for multi-platform mobile application development.

The contribution of this study is that it tries to convert the attention of software engineering researchers to investigate important issues and challenges in software engineering for multi-platform mobile application development. Moreover, it helps software engineers of mobile application development to be aware of some challenges that may face them during the application development life cycle and try to resolve problems before they occur to improve the performance of mobile applications.

This paper is organized as follows; Section 2 discusses the differences between mobile devices, Section 3 discusses the different categories of cross-platform approaches and native applications, Section 4 highlights main issues and challenges in the software engineering process of mobile application development, Section 5 presents related work in the area, and finally the paper is concluded and future work is suggested in Section 6.

2.0 MOBILE PLATFORMS AND DEVICES

To understand the challenges in multi-platform development, we have to discuss the differences found in mobile devices:

- Mobile devices: each application should possess certain requirements in order to be compatible with its device, which may be defined as specific requirements for the application. For example, the iOS platform operates devices like iPhon, iPad and mini iPad, each of these devices has different screen resolution measured by Pixel Per Inch (PPI) which ranges from 163 ppi to 326 ppi, but all of these devices use virtual keyboard. Android platform devices also have difference in screen resolutions and some devices may have hard keyboard. In addition to software differences, and hardware differences there are other difficulties related to the availability of each device hardware sensors like (GPS, touch, light sensors, Accelerometers/Gyroscopes, NFC … etc.) which are necessary for platform devices 4.

- Platform guidelines: every platform has its design guidelines, which include a set of recommendations and principles to achieve desired quality. These recommendations pertain to graphical user interface (GUI), user experience (UX), and other available facilities, and standard of the platform SDK. These guidelines differ between platforms depending on the platform standard and device facilities, for example iOS devices have only one hardware button to navigate the applications which take the user to Home screen, view running applications if user click double, or open Siri if user press and hold (Siri is an intelligent personal assistant and knowledge navigator works as an application for Apple Inc.’s iOS) 3, on the other hand, windows phone devices have three buttons as standard - start, back and search buttons - start is used to go to Home screen or speech order if press and hold, back is used to finish current activity and go to the previous one or show running applications if press and hold and finally search is used to search for anything in the mobile or web 5. Another example to explain the differences in user experience (UX) in different platforms, is that, for iOS application, user preferences and settings page is located with other applications settings into one page of settings, while in Android the user preferences and settings page can be found inside the application itself as an internal page, these examples explain how platforms have different using experience (UX) and guideline concepts.

3.0 NATIVE APPLICATION VS. CROSS-PLATFORMS

One suggested solution to the diversity in mobile application platforms is using cross-platform, which keeps the project cost on budget and makes it easy to maintain the product during the project life cycle.

3.1 Cross-platform Approaches:

Cross-platforms as mentioned by Hartmann 1 and Raj7 are categorized into many approaches:
- **Cross-compiler:** Although this approach generates native applications for multi-platforms, but these generated applications are limited, and cannot reuse specific features like camera, local notifications in different platforms and it provide poor user interface.

- **Virtual machine (VM):** This approach looks like a solution to access and reuse more resources in cross-platforms, but the main drawback of VM is that the development depends on the VM Application Programming Interface (API) not on the target platforms which increases the development limitations. The other main problem is the poor performance if compared with other approaches, as this problem is caused by the middleware between the code and the targeted platform.

- **Web based:** This approach is the most poor and limited particularly when the developer wants to access device resources. Web based approach use web tools like Hyper Text Markup Language (HTML5), Cascading Style Sheets (CSS) and JavaScript to make the web site in the internet compatible with different mobile browsers, and this type of applications can be navigated by the user via mobile’s browser.

- **Hybrid:** Hybrid approach is developed using web tools (HTML 5, CSS and JavaScript) but it executed into web browser container and use web browser engine of the device to access the device recourses. The main challenges in this approach are the inferior performance and the poor non-native user interface.

### 3.2 Native Application

Different studies found that applications built by the cross-platforms approach have less performance, and lower quality than the native applications, and all cross-platforms are just trying to achieve native applications performance and quality. However, if any organization tries to build a native application for its business and request to cover all mobile platforms, this will cause cost crisis for the organization, especially if its business need complex application. For example, if they try to cover the main three platforms (Android, iPhone and Windows Phone); they need three different environments and tools for that purpose, because each platform needs different language, compiler, and different experiences to develop the application.

**Why native application?**

Although multi-platform application development using native application may cause high cost, but the highest quality and full performance of native application can overcome the cost problem. As an example, if we review one of the most used applications in Google play and App Store – Facebook Application - which built using native SDKs, we can notice two things:

- Although the Facebook users have become experienced in using a web base version of Facebook, but the native application uses different layout for each platform.
- Facebook native applications can access all needed features on the device like camera, user contacts list, etc. directly.

![Facebook Time Line Activity In a) iPhone Application and b) Android Application](image)

To illustrate that, figure (1) shows the differences between the two applications layout, where (a) shows iPhone application screenshot which include list of options at the bottom of the page and has back button at the top left side, and (b) shows Android application screenshot that does not include back button at the application because standard android device already has back button, and the main menu appears at the top of the page with more options shown in the top left side.

### 4.0 SOFTWARE ENGINEERING FOR MOBILE APPLICATION DEVELOPMENT CHALLENGES

Software engineering for mobile application development has become, as mobile application development become more complex, also Business-to-Business and Business-to-Employee applications are
increasing more and more when it compared with Business-to-Customers applications. Thus, mobile application project now needs its own effort rather than being part of a web or desktop project, which means that software engineering for mobile applications development demands a separate team to develop the mobile application.

The next sections highlight the main issues and challenges in software engineering, and raise questions from different perspectives in the software development life cycle.

4.1 requirements issues and challenges

The requirements for mobile application are considered unique given the differences (mobile users experience, device capability, device limitations, device sensors, etc.) between the mobile application and other application for desktop or web.

While gathering and analyzing requirements, engineers and customers involved should be clear about the definition of the best application scenarios. Requirements engineer should have good experience in mobile devices and mobile operating systems features and the capability to elicit the requirements from customer. Also, the requirements engineer should pay good attention to non-functional requirements especially those related to performance, reliability, quality, and security. On the other hand, customers should fully realize how mobile resources are best utilized for the application, and how the application could be useful for any user. Once the application requirements are gathered and figured out, requirement engineer would make the decision whether the application can be developed using cross-platform and suggest which cross-platform is suitable for the application. This final decision is very important and may be shared with the customer because it affects the application cost, time of development, quality, maintenance and any future addition. For this phase key questions that should be asked by the requirements engineers and/or the customers are: What the end user wants? Why should I use this feature not the other one? How can I utilize mobile resources for this application? How this feature can be used by the user? Why end user may like to use this feature? Can I use cross-platform framework? How many devices this application will support?

4.2 design issues and challenges

In the design phase, the diversity in platform user interface (UI) and user experience (UX) makes it very important to develop prototypes for multi devices and platforms to avoid any poor implementation in any platform design. The designer should also try to answer questions such as: Does this feature work in all devices and OS’s? How much power it may consume, and how can I avoid large power consumption? Can I limit this feature in some devices/OS’s and use alternative ones on other devices/OS’s? Can I design this feature or the whole application for cross-platform? How can I minimize the processing time, data size, etc?

4.3 implementation issues and challenges

As operating systems and devices rapidly change, upgrades become the main concern of the developers, which means they should care about the OS’s changing features, versions differences, new devices that appear with different features, and how to handle any future changes in any feature. However, similar challenges may also appear in web and desktop applications, but the change usually slower than the changes in mobile operating systems and devices. These problems raise important questions about how developers can follow these changes and how these changes can be handled? And what is the application target devices/OS’s? Other challenge may the developer face, debugging the code using virtual mobile device (emulator), emulator has limited capability and poor performance when compared with a real device. Also developer cannot debug the application code in all targeted devices, which means developer has to select the proper devices to debug the application correctly.

In cross-platform implementation developer should care about the selected cross-platform and its compatibility with different platforms and devices, and its performance with each platform or device, sometimes a cross-platform can be used for some of application features to reuse the code with multi platform, however, that will affect the application performance but at the same time can reduce implementation time, cost and makes it easy to maintain the application versions later on.

4.4 testing issues and challenges

Testing becomes difficult to be done with so many targeted devices and operating system diversity. Also emulator never be enough for testing and the mobile behavior changes, for example the connectivity speed changes and power consumption be different with different connection types. Likewise, device performance changes when the application use different hardware sensors like GPS, accelerometer sensor, etc. These differences generate many test cases.

Tester should pay attention to some important questions during test planning and application versions test such as: How many devices should I test the application in? How should I select the devices for test? Should I prepare different test cases for each device/OS?

4.5 maintenance issues and challenges
The rapidly changing in mobile devices, mobile OS’s and SDKs make the maintenance phase very important phase to handle any new changes that may appear after the completion of development.

5.0 RELATED WORK

Previous studies in software engineering for mobile application developments and in cross-platform, discussed software engineering issues for mobile application development. Wasserman [8] discussed and highlighted some software engineering issues for mobile application development, Wasserman found that the characteristics of mobile applications and their operating environments call for a new or different set of research issues. Raj C.P and Tolety [7] discussed different cross-platform approaches advantages and disadvantages. They also discussed how to select the correct cross-platform framework or library for the mobile application depending on the mobile application features. Joorabchi et al [11] conducted a survey and interviews with mobile application developers to understand the main challenges faced by mobile application developers, and they found that the main challenges faced are developing applications across multiple platforms, lack of robust analysis, and testing tools, and the problems of emulators that are slow or miss many features of mobile devices.

6.0 CONCLUSION AND FUTURE WORK

Mobile devices and platforms diversity created different challenges for developers to achieve high quality applications for all known platforms and devices. As a solution for cross platform development, cross-platform libraries and frameworks can be used, but with poor quality and limited accessibility to mobile’s resources. Native application while providing the highest quality application for each device, cannot be reused on other platforms, which increases the cost of its development and maintenance.

This paper reviewed mobile devices and platforms differences, different cross-platforms approaches and highlighted software engineering issues during the development of multi platform application and raised some questions for each phase of SDLC which may help the stakeholders to improve the application quality.

This study of software engineering for cross platform could be extended in future to focus more on practical studies, which include case studies of real application for multi platforms, or conducting surveys to understand the real challenges faced by mobile applications developers, which may help other developers and researcher to figure out these challenges, suggest new software engineering approaches or improve an existing ones for multi platform development of mobile applications.

References:

[12] Yonathan Aklilu Redda, Cross platform Mobile Applications Development