Evaluating Usability of Cross-platform Smartphone Applications

Gebremariam Mesfin
Addis Ababa University
Addis Ababa, Ethiopia
mesfin.assres@gmail.com

Gheorghita Ghinea
Brunel University
London, UK
george.ghinea@brunel.ac.uk

Dida Midekso
Addis Ababa University
Addis Ababa, Ethiopia
mideksod@yahoo.com

Tor-Morten Grønli
Norwegian School of IT
Oslo, Norway
tmg@nith.no

***Abstract* – The computing power of smartphones is increasing as time goes. However, the proliferation of multiple different types of operating platforms affected interoperable smartphone applications development. Thus, the cross-platform development tools are coined.**

Literature showed that smartphone applications developed with the native platforms have better user experience than the cross-platform counterparts. However, comparative evaluation of usability of cross-platform applications on the deployment platforms is not studied yet.

In this work, we evaluated usability of a crossword puzzle developed with PhoneGap on Android, Windows Phone, and BlackBerry. The evaluation was conducted focusing on the developer's adaptation effort to native platforms and the end users.

Thus, we observed that usability of the cross-platform crossword puzzle is unaffected on the respective native platforms and the SDK require only minimal configuration effort. In addition, we observed the prospect of HTML5 and related web technologies to enhance usability in composing Web APIs for smartphone applications.

Keywords: Usability; PhoneGap; Android; Windows Phone; BlackBerry; Cross-platform applications

I. Introduction

It is well known that smartphones are playing a very important role in people's life. They are used in education, healthcare, business, etc.

The smartphones may be described as kinds of mobile phones with increased capabilities such as touch screen, intelligence and alertness, though there is no agreed definition of a smartphone in the literature. However, the specific question that needs to be answered for a specific application is how smart the smartphone is?

For the purpose of this study, we consider the smartphone features as described in [1] and hence we define smartphones as mobile phones that are capable of accessing the Internet and are equipped with mobile operating systems such as Apple's iOS, Google's Android, Microsoft's Windows Phone, and BlackBerry and software can be installed on.

On top of the operating system, smartphones are equipped with software development kits (SDKs) that enhance the characteristics of smartphone application software and configurations such as reusability, and interoperability.

Smartphone applications can broadly be categorized into native and cross-platform based on the software development environments they are produced from.

The native applications belong to one category of smartphone applications that are written and developed for specific operating system. Jobe [2] describes native applications to have unhindered access to device hardware and support all user interface and interactions available in the respective mobile operating environment.

Cross-platform applications, on the other hand, can be dedicated mobile web applications, generic mobile web applications (also called mobile websites), and hybrid applications [2]. They are implemented based on a web browser, and the fundamental web technologies are HTML5, JavaScript, and Cascading Style Sheet (CSS).

Dedicated mobile web applications - are designed to mimic the native applications of the host operating system but they execute on a web browser.

Generic mobile web applications - correspond to mobile versions of websites.

Hybrid applications - are a combination of both mobile web and native applications developed with standard web languages, but typically have access to the native device APIs and hardware and they are typically distributed through 'App stores'.

In terms of productivity and time to market, cross-platform smartphone applications are preferred to native ones. However, cross-platform smartphone applications are challenged by the limitation in user experience when deployed on native platforms. In this work, we evaluate the usability of such applications on their respective deployment operating platforms.

This paper is structured as follows. Smartphone application development and cross-platform development tools are discussed in sections II and III respectively. Section IV provides related work in brief. Concrete comparison of usability of cross-platform applications is presented in section V.

In Section VI we discuss our findings and eventually, we draw our conclusion in Section VII.

II. Smartphone Application Development

In this section we consider the native and cross-platform application development in some depth. In addition, we consider the Sun Microsystems's Java ME separately as it contains such set of runtime environments and APIs developed for a wide range of resource constrained devices including the Smartphone.

Java ME

Java ME is designed to use smaller virtual machines and APIs to create content for the severe restrictions on computational power, battery life, memory, and network bandwidth of the devices. It is a platform, a collection of technologies and specifications, that are designed for high-end and low-end consumer devices and smartcards through its CDC, CLDC, and Java card APIs configurations respectively.

According to the description given in [3] and the context given in our study, the CDC represents the smartphone domain. The CLDC and Java card APIs are thus beyond the scope of this review. Isakow *et al.* [3] noted that the CDC targets larger devices with more capabilities like the smartphones and newer CDC applications are written like the Java SE systems but with a subset of APIs available in Java SE.

In the CDC configuration, a device stack is situated on top of the smartphone hardware and operating system but beneath the smartphone applications (See Fig 1). The stack contains the configuration information, device profile and personal profile layers.

The configuration layer is a Java runtime environment for a family of devices consisting of a JVM to execute Java byte code, the native code that serves as an interface to the underlying system, and a set of core Java runtime classes.

An example of device profile is the Foundation Profile (FP) that adds additional Java SE classes to the CDC configuration which helps as a foundation for building other profiles. The Personal Profile (PP) provides Java ME specification for the devices that need a high degree of Internet connectivity and web fidelity.

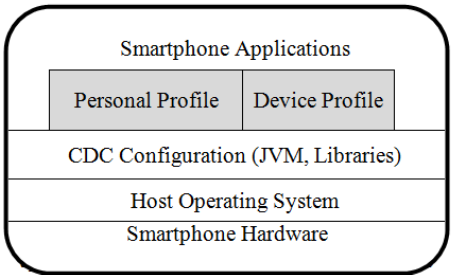


Fig 1. A Java ME Software Layer for a Smartphone

Similar software layering and configuration approaches are also provided by other smartphone application platforms such as the Windows Phone, Android, and iOS, and the cross-platform Smartphone application development environments [4].

Native Applications

The native applications are written and developed for specific operating system such as Windows Phone, Android, BlackBerry, iOS and Firefox OS. In the following paragraphs, we provide brief information on these operating systems and their corresponding integrated development environments [2].

Windows Phone

Windows Phone is one of the operating systems for smartphones. In the latest versions of Windows Phone, smartphone applications are written in managed code by frameworks that support multiple languages such as c# from the Microsoft.NET environment.

Windows Phone is primarily built with the Windows Phone SDK. Where Silverlight is an add-on for powerful, engaging interfaces and XNA for 2D or 3D games and development is done on Visual Studio.

Programs created for Windows Phone are packaged into XAP files, which is the Silverlight application package [4].

Android

Android is one of the leading operating systems for smartphones. It is based on the Linux kernel and developed as an open source system platform.

In addition to the operating system, Android provides development environment to write managed code with Google's Java libraries, and the Dalvik Virtual Machine for the smartphone applications to run on. The development environment enables to use 2D and 3D graphic libraries, a customized SQL engine for persistent storage, and 3G, 4G and WLAN network capabilities [4]. Eclipse and IntelliJ IDEA are software development tools for Android.

iOS

iOS is an operating system for many Apple's devices including iPhone and its applications are written in an object-oriented programming language called Objective-C - which is an extension of the C language, and using a library called Cocoa Touch.

Development for iOS requires a computer or a VMware running Mac OS. Xcode is the most commonly used integrated development environment to write iOS applications. It includes an editor, analysis tool, iOS simulator, and the SDK [4, 5].

Firefox OS

Firefox OS corresponds to a new approach for smartphone operating systems based on web technologies, namely, HTML5, JavaScript and Web APIs.

Grønli *et al.* [4] described this new approach in such a way that it brings open Web APIs to communicate directly with the smartphone hardware and provides a direct link to the web-based application marketplace.

In general, the native development environments are good at exploiting each device's capabilities. However, they lack cross-platform compatibility.

III. CROSS-PLATFORM DEVELOPMENT TOOLS

The smartphone operating systems are so rich in libraries and built-in features. However, they still face the heat of the market to match customer's high expectations because their basic architecture and support of programming languages is different.

Literature such as in [6] describe that the proliferation of a fragmented smartphone market with multiple operating platforms makes the development of native mobile applications a challenging and costly endeavor. To alleviate this situation, the literature and industry envisions cross-platform development approaches.

The essence of cross-platform environments is a subset of the software development environments aiming at building platform independent applications. Cross-platform application development environments work based on the general principle of “write once, and run everywhere".

In the smartphone application development, Dalmasso *et al*. [7] described the general architecture of the cross platform mobile application development tools as shown in Fig 2.

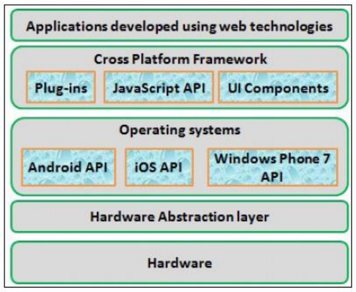


Fig 2. General architecture of cross platform mobile application development tools

However, as pointed out in [6], the diverse hardware and software platforms inevitably make portability a hassle for mobile-application developers. Portability primarily depends on runtime support and the feasibility of achieving identical look-and-feel and functionality across platforms.

There are several attempts of implementations of cross-platform smartphone application development environments. For example Java ME supports cross-platform development through configurations and profiles. Gavalas and Economou [8] describe a configuration as the minimum Java VM features and library set for devices with similar processing and memory limitations, user interface requirements, and connection capabilities, while a profile comprises libraries specialized in the unique characteristics of a particular device class.

Grønli *et al.* [4] investigated the strengths and weaknesses of the mobile application development ecosystem and pointed out that the developer support has been improved the performance of developer tools through provision of higher level abstraction of performance-critical third party libraries.

However, according to Grønli, cross-platform development environments like the Firefox OS are being challenged by the different implementations, immature platform support, variety of devices, and variety of browsers while the platform specific ones like Windows Phone, iPhone, and Android are benefiting from being tightly integrated with their respective operating system.

The work by Grønli *et al.* [4] showed that there is better integration between the development environment and deployment devices on the platform specific ones than that of the cross-platform environment. This indicates that the cross-platform application development still is in its early stages.

Literature such as in [6,7, 8] showed that cross-platform development tools are flourishing aiming at addressing user experience, stability of framework, ease of updating, cost of development for multiple platforms, and the time to market of an application. When realized, the interests of many developers would be satisfied in terms of releasing applications for major mobile platforms such as iOS and Android and provide a consistent user experience across the platforms with minimal or no change to the original code.

PhoneGap, Rhomobile, JQuery Mobile, and Xamarin are some of the cross-platform mobile application development tools available. We provide a quick overview of these tools as follows.

PhoneGap

PhoneGap is an open source cross-platform smartphone application development tool developed by Adobe System Inc under Apache license. It provides a decent toolbox for building native mobile applications using only HTML5, JavaScript and CSS [7], [9].

PhoneGap is quite popular among users mainly because of its flexibility, straightforward architecture and ease of use. Its architecture is mainly composed of Web application, PhoneGap, and the operating system along with native APIs (See Fig 3).

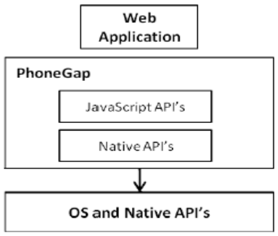


Fig 3. Interfacing Layers of the PhoneGap Architecture

Palmieri *et al.* [9] explained that PhoneGap is a "wrapper" that allows developers to enclose applications written in known programming languages into native applications. That is, applications developed using PhoneGap are neither purely web-based and nor purely native and thus some layout rendering is done via web-view instead of the native language of the operating system, and there is lack of support of HTML in some functions.

PhoneGap does not provide its own IDE to develop applications, but developers have to write the source code with an IDE and port their code into other IDEs such as the Eclipse for Android and XCode for iOS.

Thus far, PhoneGap permits the creation of applications for Windows Phone, Android, iOS, Bada, BlackBerry, Symbian, and WebOS operating systems.

RhoMobile

RhoMobile is another cross-platform mobile application development tool developed by Motorola solutions Inc. It is used to build application for iOS, Android, BlackBerry, Windows Phone, and Symbian [9].

RhoMobile is an open source Ruby-based mobile development environment used to develop enterprise applications and data on a single source code across the different operating systems listed above.

The RhoMobile suite provides an IDE called RhoStudio. Alternatively, it offers the possibility to write applications with any other IDE which supports HTML, HTML5, CSS, JavaScript and Ruby such as Eclipse, Visual Studio, Netbeans, and IntelliJ. RhoMobile uses the Model View Controller (MVC) pattern to develop mobile applications.

Applications developed with RhoMobile are compiled into Java bytecode to be executed on BlackBerry, or compiled into Ruby bytecode to be executed on all other operating systems that are running on real or virtual devices.

The RhoMobile architecture is composed of Controller, HTML templates, source adapter, RhoStudio (or any other editor), Ruby executor, device capability such as the APIs for camera and GPS, object relational mapper (ORM) called Rhom which provides an interface to interact with the SQLite database, a RhoSync client, and RhoSync server.

jQuery Mobile

jQuery Mobile is a mobile application development framework that enables and supports touch events and design elements for a wide variety of tablets and smartphones in order to make them look and function like native applications[2].

The jQuery Mobile framework is a JavaScript library developed and maintained by a project called jQuery. It is compatible with the major mobile operating systems and desktop browsers, and provides a means to customize aspects of the user interface and CSS in order to imitate the user interface of the host operating system.

IV. RELATED WORK

As discussed in the previous sections, application development for the smartphone platforms is not straightforward with respect to portability and hence the cross-platform approaches are put in place.

However, the native application development environments themselves are excellent in some respects such as higher quality of user experience, and exploiting device capabilities.

The work in [7] describes the mobile application development approaches as native, mobile web and cross-platform and provides a comparison of these approaches, and pointed out that the quality of user experience of cross-platform applications is not as good as the native applications.

The cross-platform approaches, on the other hand, strive to provide suitable generality in order to develop applications for several smartphone platforms and also they have to enable the developer to take the advantages of the specific smartphone's capabilities.

Heitkotter *et al.* [10] compiled a set of criteria to assess cross-platform development approaches. Based on these criteria and considering the native approach as a reference, they evaluated the mobile web applications; applications developed with PhoneGap as a hybrid approach; or Titanium Mobile as a self-contained approach. For the purpose of this study, we consider the look and feel, GUI design, and ease of development criteria.

Thus, Heitkotter *et al.* pointed out the following features of the mobile web, the hybrid and self-contained application development approaches:

Look and feel – in the case of the mobile web, the usage of native UI elements from within the browser is not possible and thus, the design and layout of applications depend on CSS - for example, CSS3 facilitates simple and fast development of user interfaces. Moreover, using a manifest file in HTML5, a website can request to save data in the local storage for later resume during interruption. PhoneGap as a hybrid approach is better implemented than the mobile web such as providing events for relevant changes in the application's status, but the challenges with user interface remains unchanged. The Titanium as a self-contained approach, on the other hand, instead of HTML5 and CSS3, it interprets JavaScript code by implementing native user interface elements and thus it requires far less time and effort than mobile web or the hybrid.

GUI design – in the mobile web, most tools for web user interface design offer WYSIWYG editors having special settings such as display size and resolution for developing smartphone applications. In this regard, as the web application can rapidly be reloaded on the target device without having to recompile it, GUI design is comparably fast. Similarly, PhoneGap as a hybrid approach implements GUI using standard browser and WYSIWYG editors. The Titanium as a self-contained approach, on the other hand, requires programming to implement GUI using JavaScript API and thus is cumbersome.

Ease of development – in the mobile web, with the help of the high quality of documentation and as the concepts used in HTML, CSS and JavaScript are intuitive; the ease of development is higher than with any of the other frameworks. Similarly, PhoneGap as a hybrid approach is easy because it provides clearly structured and easy documentation with code examples. Titanium provides code examples, but it requires considerable knowledge of the framework.

In general, except with the look and feel the authors found out that there is a high resemblance between the applications generated with cross-platform tools based on hybrid approach, and native applications.

In a similar context, Hang *et al.* [11] explained that advances in web technologies such as HTML5 has empowered the end users and made it possible for them to compose their own web applications. The authors presented a service composition approach for end users called HyperMash, which allows creating their own composite services by combining RESTful services with SOAP-based services at runtime.

Hang et al. adds that this approach provides a full set of RESTful interface features to make it easier for the end users to compose RESTful services.

Literature on the evaluation of cross-platform frameworks for mobile applications such as in [12] recommended the cross-platform solutions. However, the cross-platform approaches have still limitations in quality requirements including usability or native user experience.

Wargo [13] also explained that the PhoneGap framework provides access to device-specific features and applications and leaves it up to the developers to customize their applications. That is, web developers can use the capabilities provided by HTML5, CSS3, and JavaScript to enhance the user interface of their PhoneGap applications. For example, the jQuery Mobile is normally used to enhance the user interface of a PhoneGap application.

Marino *et al.* [14] also described that the JavaScript libraries can be used to handle AJAX request such as the SuperAgent. SuperAgent is a light-weight AJAX API crafted for flexibility, readability, and a low learning curve.

Holzinger et al. [15] described that the expectation of end users on the usability of the mobile applications is increasing. Hence, the user interfaces of mobile applications need to be well-designed and the design needs to be verified with proper testing.

However, testing mobile user interfaces require testing on different screen resolutions, and multiple platforms. In some platforms such as Android, different hardware providers often add their own user interface customization that can introduce slight changes in the behavior of the platform. For example, Samsung Android devices, and HTC can specifically change the behavior of certain buttons, and fields and this is a particular problem for the home screen app. This in turn requires either to write tests for each hardware types or not to consider these differences at all. The problem is further complicated for all platforms by the different versions of each of the platform [15].

However, the above mentioned literatures did not show the extent of usability of the cross-platform applications in general and on the specific deployment platforms in terms of the five usability attributes – ease of learning, ease of use, ease of remembering, number of errors, and level of delight in use [22].

V. Comparison of Usability of Cross-platform Application

In this study we employ literature review to frame the research setting, and usability technique to evaluate usability of the smartphone application developed with PhoneGap as a cross-platform development tool.

PhoneGap is chosen because it is the most widespread and for technical convenience of the researchers to work with the Eclipse IDE on the Windows platform, and Visual Studio.

Usability is defined by the ISO [16] on its guidance on usability as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. Based on this definition, Carol Barnum [17] underscored the need to consider the target group of users (not all users) for a particular product; the specific design goals of a product are identical with the goals of its users; and users use the application in a certain environment (context) and it is essential that the application is designed to be used under those terms.

In this research, a crossword puzzle is developed with draggable alphabets, target drop slots on a table, and list of clues as shown on Fig 4. The puzzle was initially developed in [21] with HTML5, JavaScript, and CSS and has been modified to fit into our context of use.

In this study, usability evaluation is conducted from the viewpoint of both the developer who does the development and adaptation to the native platforms and the end users who actually play the game.

The developer viewpoint is framed under the developer-tools usability theme [20] that is the ease to use of the tool to develop applications for multiple platforms. Specifically, for the interest of time, we just considered the efforts required by the developer to adapt to the Android, Windows Phone, and BlackBerry native platforms in terms of lines of code as shown on Table 1.

Table 1 Developer Adaptation Effort of Cross platform

Feature	Mobile Web	Cross platform App		
		Android	**Windows Phone**	**BlackBerry**
Text	Same view as on the desktop browser	Bigger font size as compared to the mobile web version	Smaller font size as compared to the Android version	Same as the Android version
Image	Nearly same view as on the desktop browser	Same view as the mobile web version	Images disappeared both from alphabet pallet and the puzzle board. The' build action' file property of all images is converted from resource to content and app rebuilt.	Same view as the Android
Hover	Hovering needs long touch as opposed to the point-and-hold on desktop browser	Long touching an image does not respond. Fairly large number of JavaScript lines of code and plug-ins applied & single touch selects an object.	Single touch selects an object. This has been achieved without any adaptation efforts related to this feature	No lines of code added or removed from the source code of the Android version.
Drag	Dragging needs selecting the "Drag" command from a popup menu, pointing and then swiping	Drag-able object could not be dragged. Fairly large number of JavaScript lines of code and plug-ins applied & swiping into the target drags the object.	Swiping into the target drags the object. No adaptation effort needed.	Same as the Android
Drop	Releasing the touch at a destination (drop point) drops the object.	No object to drop. However, clicking on the drop point highlights. Fairly large number of JavaScript lines of code and plug-ins applied & single touch on the target drops the object.	Single touch on the target drops the object. No adaptation effort needed.	Same as the Android
Button	Look & feel of buttons is similar to that of the desktop browser. However, the performance is slower.	Similar look & feel, and performance to that of mobile web version.	Similar look & feel; and performance is better compared to the Android version.	Same as the Android
Navigation	Swiping across all sides on the screen	Swiping works across all sides on the screen	Swiping works across all sides on the screen and performance is better compared to the Android version.	Same as the Android
Platform configuration	No! But setting the browser.	The cordova.js, cordova.jar, and config.xml files plugged in; the android manifest tweaked, and more.	Cordova for Windows Phone (PhoneGap Custom and Starter), Google's JavaScript plug-in, and other custom plug-ins.	The PhoneGap-blackberry, and JavaScript plug-ins
SDK used	Eclipse	Eclipse	Visual Studio 2010 Express for Windows Phone	Eclipse

The end user viewpoint, on the other hand, is part of the classical usability evaluation as described in Jacob Nielsen [18]. Jacob Nielsen described in his book entitled Usability Engineering usability as a set of attributes of a user interface; namely; learnability, efficiency, memorability, errors and satisfaction.

The descriptions of usability presented above are full of subjectivity and evaluation with these attributes is highly biased. Thus, ten general principles of user interface design are coined as heuristics and a description of each is provided [17, 18].

In a similar context, a ten point usability measurement tool called system usability scale is presented in [19]. Jeff Sauro [19] describes this tool to be technology independent and has since been tested on hardware, consumer software, websites, cell-phones, IVRs and even the yellow-pages.

As the system usability scale tool has become an industry standard [19], we used it to evaluate the usability of the crossword puzzle on Android, Windows Phone, and BlackBerry devices. The system usability scale is a five point Lickert scale with the ten questions as shown on Table 2 below.

Table 2. Questions in the System's Usability Scale Tool

No	**Questions**
1	I think that I would like to use this system frequently.
2	I found the system unnecessarily complex.
3	I thought the system was easy to use.
4	I think that I would need the support of a technical person to be able to use this system.
5	I found the various functions in this system were well integrated.
6	I thought there was too much inconsistency in this system.
7	I would imagine that most people would learn to use this system very quickly.
8	I found the system very cumbersome to use.
9	I felt very confident using the system.
10	I needed to learn a lot of things before I could get going with this system.

Nine users participated in the evaluation of the crossword puzzle in three groups and played on each of the three operating platforms. Each user was briefly introduced about the puzzle, asked to play, and respond to the questions. Accordingly, a summary of their responses on the usability of cross-platform crossword puzzle on the three different operating platforms is shown in Table 3.

Table 3. Summary of Responses on Usability the Crossword Puzzle

Question numbers	Response	Number of respondents for		
		Android	Windows Phone	BlackBerry
1,5	Very high	2	3	1
1,5	High	1	-	2
3,7,9	High	3	3	3
2,8,	Low	3	3	3
4,6,10	Low	3	2	3
4,6,10	Very low	-	1	-

The Android version of the puzzle has been tested on a real Samsung Galaxy Ace device. However, the usability evaluation result shown on the table above is based on the tests on the respective device emulators.

VI. Discussion

As described earlier in the previous section, our discussion of results bases on both the developer and the end user viewpoints.

With the developer viewpoint, we considered the efforts required by the developer [20] to adapt to the Android, Windows Phone, and BlackBerry platforms taking certain features of the crossword puzzle into account, namely, the text, image, hovering, dragging, dropping, button click, navigation, platform configuration, and the SDK used.

Initially the puzzle was developed with HTML5, CSS3, and JavaScript but targeting the desktop browser. The same source code was also accessed as a mobile web with only limited usage difference such as hovering needs long touch in mobile web as opposed to the point-and-hold on the desktop browser.

However, the same source code was deployed on the Android platform as a cross-platform app with PhoneGap and the hovering, dragging, and dropping features were lost completely.

The source code has been debugged by applying fairly large number of JavaScript lines of code and plug-ins to make it compatible with the Android platform and the puzzle's behavior was preserved as shown in Fig 4.

The same source code that correctly run on the Android platform with its PhoneGap plug-in has been deployed on the Windows Phone and BlackBerry platforms with their respective PhoneGap plug-ins and all the features of crossword puzzle worked correctly without any adaptation efforts to maintain those features except that a few SDK configuration efforts have been made.

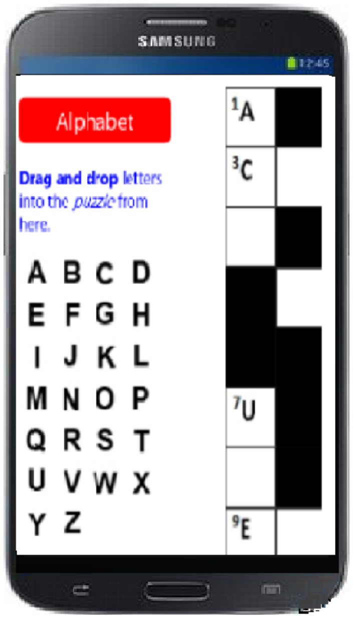


Fig 4. The Crossword Puzzle on Android Platform

For example, among the SDK configuration requirements we encountered that images disappeared both from the alphabet pallet and the puzzle board (see Fig 5) when deploying on the Windows Phone. The' build action' file property of all images is converted from resource to content and the app was rebuilt into the proper features as shown in Fig 6.

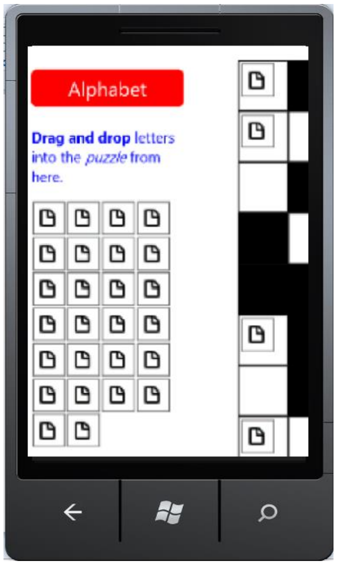


Fig 5. Distortion of Crossword Puzzle on Windows Phone

Our finding pointed out that PhoneGap based cross-platform apps can be ported into other platforms with only limited SDK configuration efforts of the developer and hence usability from the developer viewpoint.

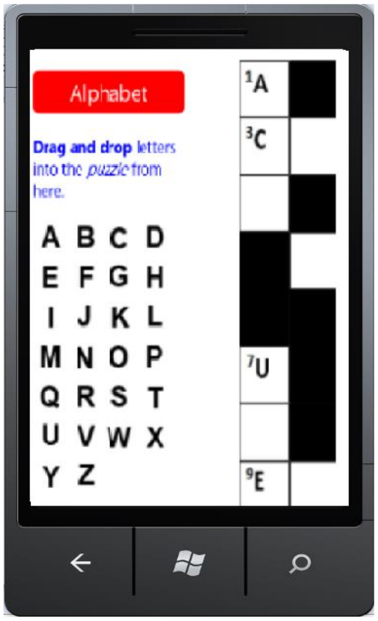


Fig 6. Adaptation of Crossword Puzzle for Windows Phone

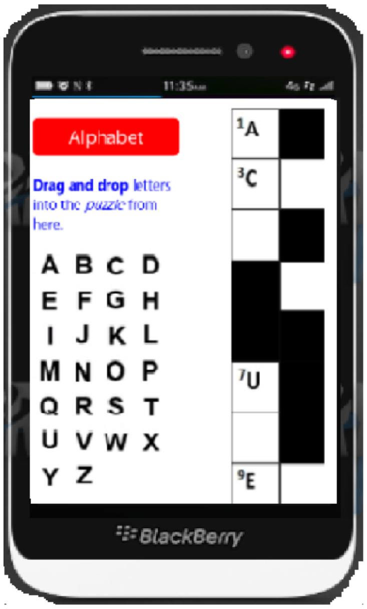
When considering the platform configuration effort as a feature, we observed the following:

- The Android platform required tweaking of the configuration elements such as the cordova.js, cordova.jar, config.xml, and the android manifest. In addition, Google's JavaScript plug-in, and other custom plug-ins are added to the source code.

- The PhoneGap Custom and Starter, Google's JavaScript plug-in, and other custom plug-ins for Windows Phone; and

- The PhoneGap-blackberry, editing the configuration file, and adding JavaScript plug-ins for the BlackBerry.

Thus, we found out that the Windows Phone is easier to configure because the minimal configuration effort required to run the crossword puzzle is less than the Android and BlackBerry platforms.



Drag and drop letters into the puzzle from here.

Fig 7. Adaptation of Crossword Puzzle for BlackBerry

The result of usability evaluation from the end user viewpoint (Table 3 above) also indicated that the usability of the cross-platform app remains an affected across the individual platforms when ignoring the impact of the form factor of each device.

The Windows Phone version, however, is found to be more usable than the other two versions.

VII. Conclusion

It is observable that the cross platform mobile application development frameworks are benefiting developers to build applications for multiple platforms. However, there is a little doubt on the behavior of the resulting cross-platform applications (in terms of usability from the viewpoint of the developer as well as the end user) on the native platforms.

Usability of the resulting applications from the developer viewpoint is seen in terms of the effort required in lines of code, and platform configuration to adapt into the respective native platforms. For the end user, on the other hand, we applied ten questions usability questionnaire.

Our finding showed that the usability of cross-platform smartphone applications remains an affected when deployed on the respective native platforms. In addition, we observed that the cross-platform development tools such as PhoneGap require only minimal configuration effort to deploy the cross-platform app to the specific platforms. HTML5 and related web technologies together with cross-platform tools would offer considerable opportunities to enhance usability of developer tools.

Thus, our future work will consider usability of existing Web APIs to compose cross-platform smartphone applications and the prospect of HTML5 to enhance usability in the composition of REST/SOAP web services for smartphone applications.

REFERENCES

[1] Liane Cassavoy, What Makes a Smartphone Smart? http://cellphones.about.com/od/smartphonebasics/a/what_is_smart.htm, Accessed on July 23, 2013

[2] William Jobe, Native Apps vs. Mobile Web Apps, iJIM 2013

[3] Anna Isakow, Hao Shi, Review of J2ME and J2ME-based Mobile Applications, International Journal of Computer Science and Network Security (IJCSNS), VOL.8 No.2, 2008.

[4] Tor-Morten Grønli, Jarle Hansen, Gheorghita Ghinea, Muhammad Younas, Mobile Application Platform Heterogeneity: Android vs Windows Phone vs iOS vs Firefox OS, Submitted 2013

[5] https://developer.apple.com/xcode/.[Accessed: 20 January 2013]

[6] Henning Heitkotter, Sebastian Hanschkem, and Tim Majchrzak, Evaluating Cross-Platform Development Approaches for Mobile Applications, Springer 2013

[7] Isabelle Dalmasso, Soumya Kanti Datta, Christian Bonnet, Navid Nikaein, Survey, Comparison and Evaluation of Cross Platform Mobile Application Development Tools, IEEE 2013

[8] Damianos Gavalas and Daphne Economou, Development Platforms for Mobile Applications: Status and Trends, IEEE 2011

[9] Manuel Palmieri, Inderjeet Singh, and Antonio Cicchetti, Comparison of Cross-Platform Mobile Development Tools, IEEE 2012

[10] Henning Heitkotter, Sebastian Hanschke, and Tim A. Majchrzak, Evaluating Cross-Platform Development Approaches for Mobile Applications, Springer 2013

[11] Feifei Hang, Liping Zhao, HyperMash: A Heterogeneous Service Composition Approach for Better Support of the End Users, IEEE 2013

[12] Sommer Andreas, and Stephan Krusche, Evaluation of cross-platform frameworks for mobile applications, Proceedings of the 1st European Workshop on Mobile Engineering 2013

[13] John M. Wargo, PhoneGap Essentials: Building Cross-platform Mobile Apps, Addison Wesley 2012

[14] Enrico Marino, Federico Spini, Alberto Paoluzzi, Fabrizio Minuti, Maurizio Rosina, Antonio Bottaro, HTML5 Visual Composition of REST-like Web Services, IEEE 2013

[15] Andreas Holzinger, Peter Treitler, and Wolfgang Slany, Making Apps Useable on Multiple Different Mobile Platforms: On Interoperability for Business Application Development on Smartphones, International Federation for Information Processing 2012

[16] Guidance on Usability: ISO 9241-1, International Organization for Standardization1998

[17] Carol Barnum, Usability Testing Essentials, Elsevier 2011

[18] Jacob Nielsen, Usability Engineering, Orlando, FL, USA: Academic Press. 1993

[19] Jeff Sauro, Quantitative Usability, Statistics, and Six Sigma: Measuring Usability, http://www.measuringusability.com/sus.php, Accessed on February 18, 2014

[20] Andrew Faulring, Brad A. Myers, Yaad Oren, Keren Rotenberg, A Case Study of Using HCI Methods to Improve Tools for Programmers, IEEE 2012.

[21] GitHub, Inc. (US) https://github.com, Accessed on February 15, 2014

[22] Minhaz Fahim Zibran, What Makes APIs Difficult to Use, International Journal of Computer Science and Network Security, 2008