AN EMPIRICAL STUDY ON MOBILE MULTI PLATFORM DEVELOPMENT USING PHONE GAP AND HTML5

Paulo Roberto Martins de Andrade[1] and Fatima Aguiar da Silva[2]

[1]Graduate Studies in Computer Science, University of Regina, Regina, SK, Canada
[2] Faculdade Estácio FIC, Fortaleza, CE, Brazil

ABSTRACT

In a world so connected, it is increasingly common to search for mobile applications to solve trivial problems or have access to information on companies, local and diverse. It is expected that companies, colleges and the government make available such information in a practical way, this new reality. However, doing so is not always something easy or cheap. Many companies spend their resources to offer the best to its customers, so that smaller companies cannot keep up. The emergence of new alternatives to this problem has arisen based on the paradigm of "write once, run anywhere". The multiplatform development has increasingly attracted the interest of researchers and investment firms seeking the best cost-benefit. This work shows how the hybrid development and shows through an empirical study of a real application, this new form of development can be the alternative to this new reality.

KEYWORDS

HTML5; cross platform; empirical study; android; mobile computing.

1. INTRODUCTION

Because of the exponential increase in the need for people to stay connected with everything and everyone through the internet in search of information and communication, increases the need for suitable sites and applications to this new reality. According to the International Data Group [10], since 2010, the amount of mobile devices has increased more than 15% per quarter, which further emphasizes the need for adequacy of companies, government agencies and universities to this new reality. For companies, the lack of technical knowledge has been the obstacle for not implement good ideas due to costs necessary for training.

When it comes to creating mobile applications for multiple platforms, it is necessary for the development team to master different languages and syntaxes. We have today, for example, Google Android, which uses an extension of the Java language created by Google itself. Apps for iPad and iPhone use e Objective-C. Windows Phone has the programming language C#. Among other as BlackBerry (BB), WebOS, Kindle Fire, etc. Given this multitude of languages and paradigms, there are some problems such as: maintain specialized teams on different platforms; control updates; and different defect management in a single application. A multiplatform architecture would be a solution to make this difficult task into something much more affordable, and with the possibility of the development using Web methods or hybrid application development [11].

DOI : 10.5121/ijait.2016.6105

Developing software for mobile devices that are installed and run on multiple platforms can now through the following web technologies HTML5, CSS3 and JavaScript [16]. After the advent of these technologies in the mobile world, there were numerous frameworks and then the division of mobile solutions in three categories: Native, Hybrid and Web Apps [6].

This paper address the architectures for web development and needs for multiplatform development. It presents some frameworks and give more details about the PhoneGap and the Intel App Framework. The following section presents a case study using the two frameworks and the results of empirical tests [3]. Finally, the last section presents the conclusion and suggestion for future work.

2. EASE OF USE

2.1. Sources of Changes

In our lives or in our projects we should always be prepared to deal with and make changes because they always occur. In the development of software applications, we recommend the use of good practices related to Software Engineering. For IT Governance there are several frameworks of best practice, one example would be the COBIT with BAI06 Manage Change process [10].

The COBIT is an IT governance framework maintained by ISACA (ISACA). It is a reference model and is not a ready solution. In other words, it needs to adapt it to the business to meet the needs, and align the information technology sector to the business and can be used only a few controls that seek to improve the processes that comprise the IT governance of each company [10].

There are several causes that give rise to the changes, we can highlight the compliance with laws and regulations such as PCI-DSS (Payment Card Industry - Data Security Standard) [11], SOX (Sarbanes Oxley Act) [16], needs preventive measures in the IT environment, external events of which the IT has to comply in determined period, need for improvements in systems and infrastructure, senior management requests (strategic changes), but the highest incidence of changes in the IT environment noted in the case study in this article are those that originate from problems. In this case, the change is the answer to the problem. Another important part of the change comes from direct demands from any interested party.

2.2. Concepts in the view of Project Management

In order to understand the multiplatform architecture that we first understand the types of strategies used in the development of applications for mobile devices. Figure 1 shows the comparison between the strategies of application development and indicates its multiplatform, increasing fitness from left to right [4].

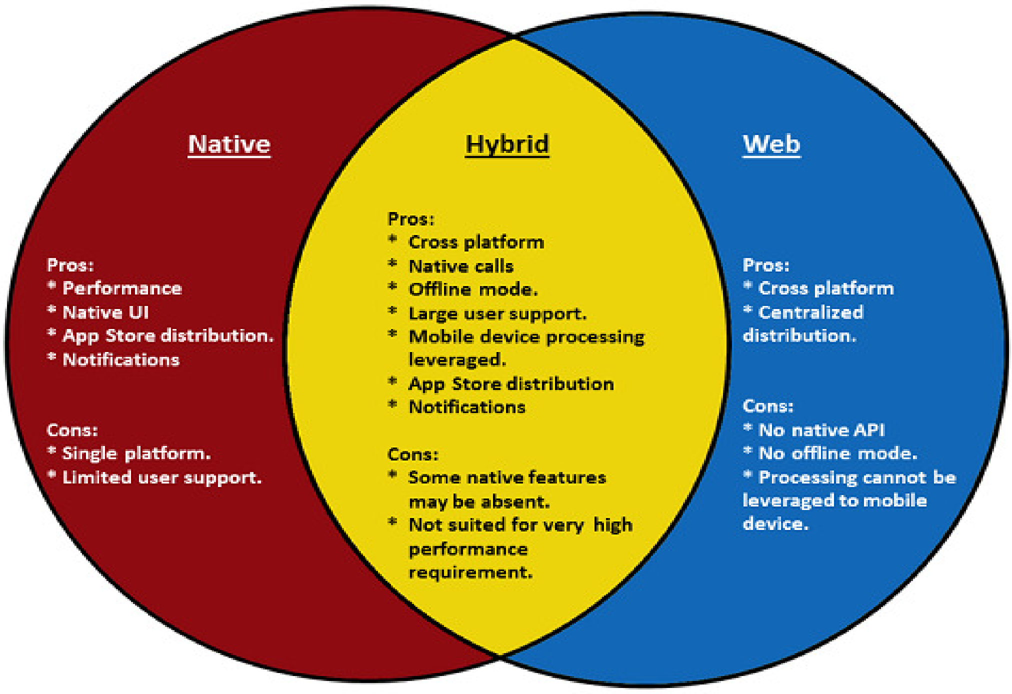


Figure 1. Strategies for mobile development

Hybrid applications combine the advantages of both types of the development of applications (web and native) and the best choice is to create cross-platform applications [1]. The main advantages of using a hybrid technology are:

- Multiplatform & Sharing code: coding once and use it to make the "deploy" on various mobile platforms. Uses the same code "UI" interface for multiple devices;
- Make native calls to hardware using the "Native Shell" through of the JavaScript;
- Offline mode allows access to the same applications that the internet is not readily available.
- Allows a large number of users can get the application due to its multiplatform nature.
- One can avail the mobile processing, which is not possible with web applications.
- Distribution through official stores each platform transmits a native feeling regarding updates.

The web applications also allow a multiplatform fitness, but in a more limited way. Developers will not be able to take advantage of the device hardware and native UI. Furthermore, the local processing means of a browser will not be a good choice for applications whose performance is critical. In case the performance is not a problem and just want to host a web application then this kind of application is sufficient, eliminating the use of a "third party Framework" and further promote the use of existing knowledge of web technology [1].

Native applications offer advantages multiplatform very limited or even nonexistent. For this reason, is that they are not advisable for a multiplatform architecture [6]. Figure 2 illustrates the main pros and cons of each type of application.

Figure 2. Pros and cons of each type

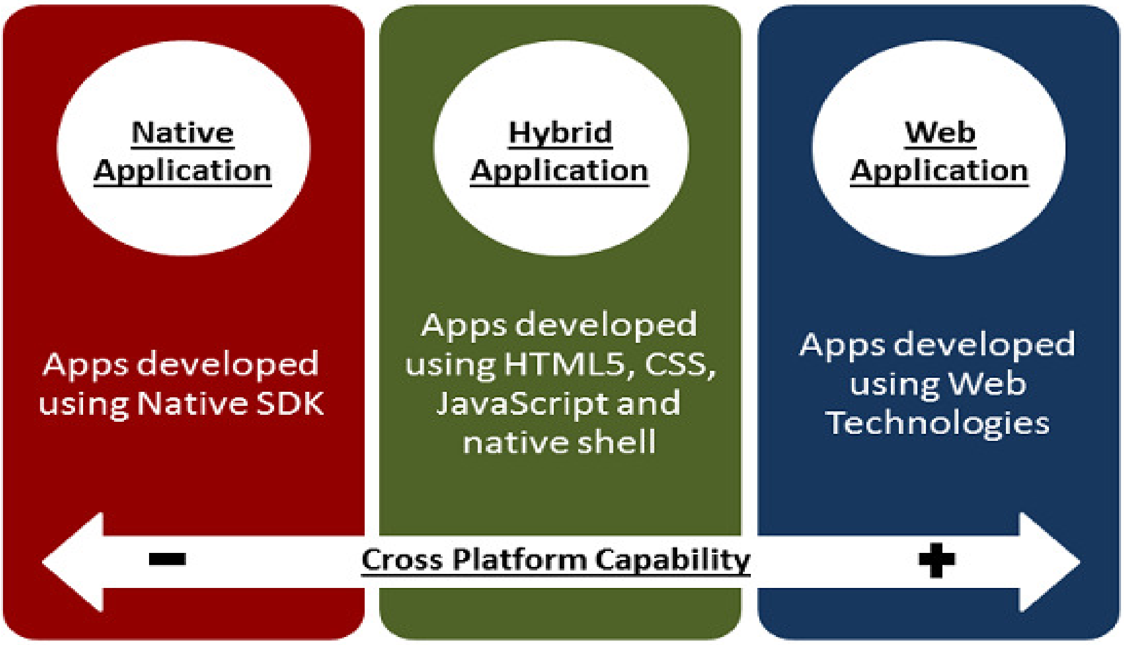

Native Application
Hybrid Application
Web Application
Apps developed using Native SDK
Apps developed using HTML5, CSS, JavaScript and native shell
Apps developed using Web Technologies
−
Cross Platform Capability
+

Figure 2. Pros and cons of each type

In a multi-platform architecture using hybrid methods, the development of an application uses Web technologies such as HTML5, JavaScript and CSS but that run inside the "Native Shell" of the Mobile platform. Thanks to the increasing sophistication of multi-platform tools, performance has improved dramatically, and both the look and feeling that we have to use the application is almost as good as the native UI.

3. Frameworks

The frameworks for development of hybrid applications are separated into two types, those responsible for the interface (UI - User Interface), and those who package the application for different platforms and allow it to access the phone's features (Native Shells) [7].

3.1 UI Frameworks

The UI frameworks, are responsible for defining the look that will have the application, it being a native application, Web App or hybrid, all have a user interface. The visual part of the application is the one that gives the first impression and that can cause greater acceptance or greater rejection of the application. An application with friendly interface and easy to use, has greater chances of winning the preference of users compared to one that, even though the same functionality or to others, is presented with a difficult to use and unintuitive interface.

The interface design for interactive systems is such a relevant task that has become one of the subfields of human-computer interaction (HCI) which, in turn, aims to study, plan and understand how people and computing devices can interact so that the their needs are reflected in the most effective way possible.

For native applications, there is some standardization of interfaces, as they obey the rules of the platform, while for the app and hybrid web, can be used for web development frameworks, such as Intel App Framework, jQuery Mobile, Bootstrap or among others. It is not necessary that the framework is specific to mobile devices even though a specific could provide better experience, both in performance and as for usability.

3.2 Frameworks for Native Resources Access

A multiplatform framework is a set of source files, libraries and tools that supports more than one platform, at least two different platforms. In this definition, iOS and Android are two different platforms, iOS 5 and iOS 6 are not, and allows development without branches of source code, that is the same code for all platforms.

Thus, frameworks should have a unifying language that is the same for all platforms. APIs that allow one to access a resource in the same way on an Android OS or a Firefox, and which enables the application to access native resources system. During the study found several frameworks such as the Titanium [14], the Trigger.io [15], the Sensha Touch [12] and the Phonegap (or Apache Cordova) [2].

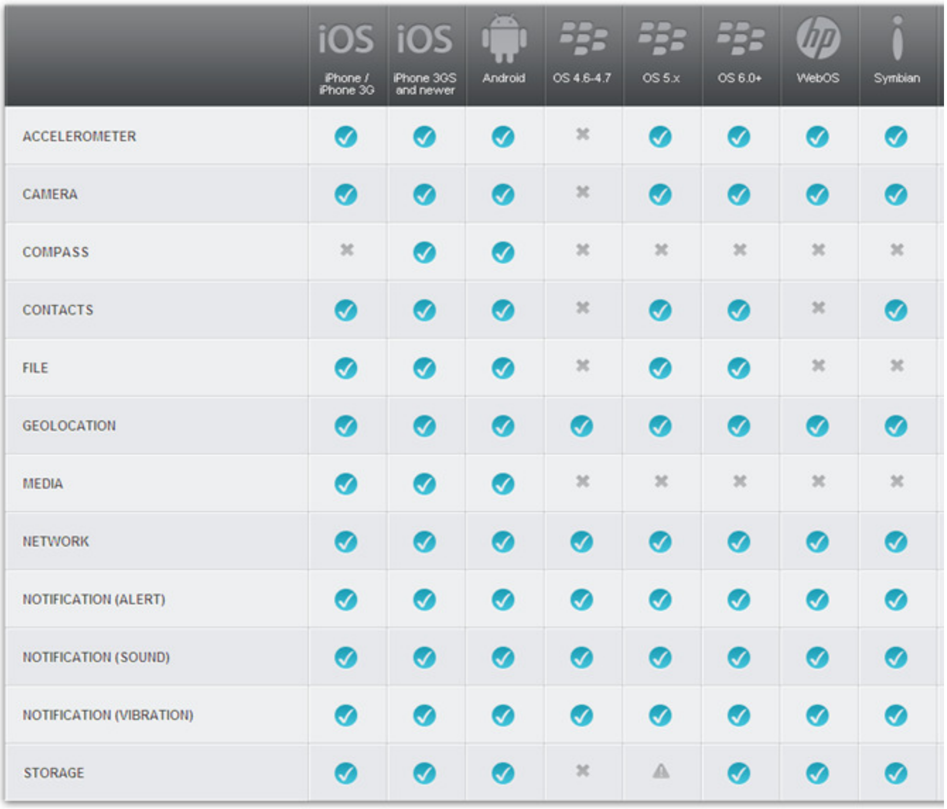
After the analysis of the ways of development of each platform, their language and other characteristics, we chosen for the case study, the Phonegap framework. The decision was based on the fact that even using web languages like HTML, CSS and JavaScript also has a greater number of supported platforms, by having greater popularity (having a larger number of developers), and good documentation [11].

3.2.1 PhoneGap (Apache Cordova)

The PhoneGap is a framework that has the function of "Native Shell". It created by Nitobi in 2008 as an open source solution for building cross-platform mobile application, passing the first to support the iPhone, Android and Blackberry 4 later in the Symbian and WebOS and Windows Phone 7 finally in 2011, Adobe acquires Nitobi Software. In October 2011, PhoneGap has donated to the Apache Software Foundation (ASF), under the name Apache Cordova. Through the ASF, the development of PhoneGap passes to ensure open government project. Remaining free and open source under the Apache License Version 2.0 [11].

The Apache Cordova uses the methodology write-once, where the idea is to write the code only once and by importing, exporting it as a native application. Currently supports all major platforms such as iOS, Android, Blackberry 10, Windows Mobile, Windows Phone (7 and 8), Firefox OS, Ubuntu and Tizen. Through its library and the native code generated. It allows API calls for smart phone hardware [2].

To understand better, the Apache Cordova is a set of APIs that allows the developer to access the native device functions like camera, calendar, GPS and other by JavaScript, HTML5 and CSS3, instead of device-specific languages such as Objective-C. The development is like any site and therefore offers greater ease of learning. Dispensing, for example, long hours of dedication to more complicated systems, such as Java and reduce project costs [13]. Figure 3 we can see all the APIs supported by the framework.



	iOS iPhone / iPhone 3G	iOS iPhone 3GS and newer	Android	OS 4.6-4.7	OS 5.x	OS 6.0+	WebOS	Symbian
ACCELEROMETER	✓	✓	✓	✗	✓	✓	✓	✓
CAMERA	✓	✓	✓	✗	✓	✓	✓	✓
COMPASS	✗	✓	✓	✗	✗	✗	✗	✗
CONTACTS	✓	✓	✓	✗	✓	✓	✗	✓
FILE	✓	✓	✓	✗	✓	✓	✗	✗
GEOLOCATION	✓	✓	✓	✓	✓	✓	✓	✓
MEDIA	✓	✓	✓	✗	✗	✗	✗	✗
NETWORK	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (ALERT)	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (SOUND)	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (VIBRATION)	✓	✓	✓	✓	✓	✓	✓	✓
STORAGE	✓	✓	✓	✗	⚠	✓	✓	✓

Figure 3. Apache Cordova API's

The creation of a generic and offline using HTML, CSS and JavaScript application and its testing in several screen sizes, test the use of imported APIs in its development (as Geolocation, Camera and Notification APIs) is possible using an emulator for Google Chrome called "Ripple Emulator". After creating the application, it is possible in Apache Cordova v.3.0+ create projects for each platform through command line using “Node.js”. When creating a project, the generated files are accessible natively in each program to perform the necessary customizations for each platform and test on emulators.

A particularity is found that for iOS apps that you need use a MAC for continue you development, because the generated project should be opened with xCode and requires its own libraries as well the project for Windows Mobile needs be done using Windows 8.0 or higher. Projects for Android, Blackberry OS and Firefox OS work on any platform. An alternative is to use PhoneGap Build Online and perform the conversion in native code independent of the operating system being used [13].

3.2.2 Intel App Framework

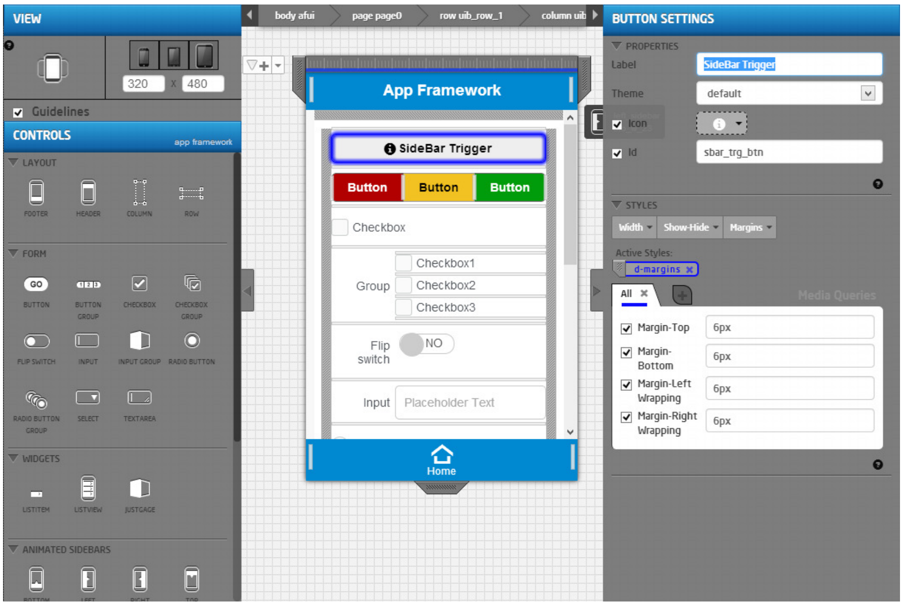
The Intel Company may still be known for creating the chips that power the world's desktops, but the company has been trying to expand into the world of software for many years. The Intel App is one of their latest designs. The Intel App Framework is a framework for building cross-platform mobile applications using HTML5 technologies. He came when based jqMobi, a mobile optimized version of jQuery, which was created by the team behind appMobi. Intel acquired the tools and personnel jqMobi in February 2013 [9].

The Intel App Framework is a free and open source under an MIT license. Combined with its lightweight JavaScript library, it provides a basic MVC structure and various components of the user interface. Instead of imitating the appearance and behavior native, he chose to offer their own style for use on all platforms. Over time, styles added to mimic the display of native apps, at the request of users, being free to choose the style to use, and maybe carry a different style according to the operating system of the mobile device where the application is running. We can also customize the styles (css and html) using the "Style Builder" tool. The Intel App Framework can be used together easily with PhoneGap (Apache Cordova) to create native applications instead of web applications, which is their standard.

One of the most notable differences is a smaller library to manipulate the DOM, which offers the most important features of jQuery without the slower functions. Intel claims that its library is faster and more robust response to the mobile internet. The functionality and the structure is almost the same as jQuery, and has a perfect compatibility with it, allowing you to install a plug-in jQuery [9].

The best part may be the largest collection of tools that includes a website and a Java-based build and test their applications client. There is also a Windows executable for those developers who want everything running locally. The entire structure is marked in HTML, and JavaScript does the job. In the Intel Style Builder tool, just choosing the colors, it generate a new CSS with the style for your application. The Framework keeps some of the elements used less, like the carousel, as separate plug-ins that you can add when you need it. This keeps the resulting code smaller and therefore easier to deploy. This framework has its own native APIs to call the hardware but usually use the phonegap together to use the hardware, so, the Intel App Framework just assuming the role of HTML framework.

Is also possible to use the Intel App Starter, an online tool, for help you create an initial project. You can drag and drop DIVs and components in the right place and test the results in your browser. You will create all the basic structure of your application, such as pages, menus, choose transitions, simpler forms and customizations header. Upon completion of construction, it is possible to download the generated html code and use it to continue the application [9]. Figure 4 shows the interface of this tool.



BUTTON SETTINGS

PROPERTIES

Label: SideBar Trigger

Theme: default

☑ Icon

☑ Id: sbar_trg_btn

STYLES

Active Styles: d-margins

☑ Margin-Top: 6px

☑ Margin-Bottom: 6px

☑ Margin-Left Wrapping: 6px

☑ Margin-Right Wrapping: 6px

Figure 4. Intel App Start

4. Case Study and Results

The Cagece (a company of basic sanitation Brazilian public sector) taking into consideration the strategic planning and goals, needed a quick and convenient way to access the strategic indicators the analytical form company, showing thus information on amount of water connections, the number of sewage connections, indices of customer satisfaction, indices of EBITA margin, among others. One of its purpose was to facilitate meetings between company managers and provide a more transparent management for your employees.

The Cagece decides to create a mobile app as a solution for this case, called "SEI App". The scope and functionalists of this app are:

- Load the indicator data using a web service;
- Divide the State of Ceará into business units and filter through the indicators per unit;
- Filter the bookmarks by city;
- Loading data indicators by month and compare with the previous month;
- Load data indicators per year and compare with the previous year, making analysis of variation;
- Show trend graphs;
- View registered action plans on the server.

After the creation of the project documentation and the prototype, we presented the idea to the board of the company. The board approved the idea and authorized the implementation. We used the Agile PDD [8] and the PM5 [17] methodologies to plan, execute and manage this project. With this methodology, the stakeholders could follow the development of more closely project. Because the diversity of mobile devices encountered in the company, was also raised the need for the app should work on smartphones and tablets with iOS or Android, which could generate a higher cost of resources and time to develop.

The major problem was the lack of skilled labor within the company to develop for iOS. Initially it was thought to be held in an internal training tool, however only one application did not justify the investment and have a small learning curve. Another option was to perform the outsourcing for development through a software factory.

Within the company, there was a web development team, which proposed to study the possibilities and present to the business area, proposals to make the project viable. After a few weeks of study and comparisons of available technologies and the need for each, the team chosen a hybrid application development using HTML 5 as the basis of the application, PhoneGap v.3.6 as "Native Shell" and the Intel App Framework v2.1 as the User Interface. The web service created for the communication between the oracle database and the mobile app was implemented using the Php and slim framework.

The entire development of the hybrid application lasted 4 weeks. It was the second application developed by the team using these frameworks, but with different versions. For make a comparison, in parallel, the Java development team, with some experience in mobile development, has also developed a version of the same application, taking 5 weeks to complete.

After finishing the two developments, were selected 48 employees of the company, experienced in the use of Android devices, to use the application. We chose the Android OS because the fact that in some models has more limited performance. Were selected 24 tablets Samsung GT-P3110 and 24 mobile phones Samsung S-i9000. Each employee received one of the devices. In the first round of tests, only 12 devices of each type had the native application, while the others had the hybrid application. The table 1 illustrates the division.

TABLE I
THE DIVISION OF DEVICES IN THE ROUND ONE

Devices	Native App	Hybrid App
GT-P3110	12	12
S-i9000	12	12

The instructions for the employees were to use the application for 2 weeks and give your feedback on the experience. This feedback was collect using an online survey. The questions is about the use experience, the visual, the stability and the performance.

TABLE II
THE DIVISION OF DEVICES IN THE ROUND TWO

Devices	Same Native	Changed Native to Hybrid	Same Hybrid	Changed Native to Hybrid
GT-P3110	4	8	5	7
S-i9000	5	7	4	8

After use, half the amount of devices were reversed, but was told to all employees that they would receive a device with a different version of the test first, applying the placebo effect. Employees would use two more weeks to give a new feedback. The table 2 illustrates the new division of devices.

After the second round, a new online survey (with the same questions) was applied with the functionaries and the data was compared with the first round. In the second round, the users could send a review or suggestions about the app.

As a result, 87.5% (42 of 48) of the users reported that they have not noticed a difference in the use of the application, stating that had the same experience. The other 12.5% users reported that they perceived that an application was relatively faster over another in their response to use with the native applications. On the other hand, 50% of this users that noticed a difference between the rounds, used a different app, the another 50% (4 of 8) reported this using the same app in the same device. Because of the expectation of change, the users had the impression its initial implementation used was faster than the second application, even if they were the same. As the result, we can see that only 8.33% (4 of 48) users realized the difference in performance between the native and hybrid apps [5].

This comparison helps to see that the hybrid application, even taking a more time in this case, is a viable alternative for companies when the need is to create applications for different platforms. The performance of the current frameworks has increasingly approached the native applications. Another positive point is a shorter learning curve for new applications by taking advantage the common knowledge of the most WEB developers, not making it necessary for learning a new language. This comparison also shows that native development is a viable option only when a deeper use of specific APIs, frameworks and not supported by harnessing the full power of the displays and UIs is required.

5. Conclusion

Native applications can provide a good user experience, but maybe the companies had no money or expertise to develop natively. A hybrid approach offers a simple solution for developing applications for smart phones and tablets. Write the code once and deploy to different operational systems will help companies to quickly launch their mobile applications and reduce maintenance costs. Hybrid structures are suitable options for the real benefits in the use of applications for business or for education.

We suggest as future work more detailed metrics on the learning curve, performance and savings generated for hybrid applications and different sizes and needs.

Acknowledgements

This work is supported by the National Counsel of Technological and Scientific Development of Brazil (CNPq - Conselho Nacional de DesenvolvimentoCientífico e Tecnológico) and the Ceará Water and Sewage Company (Companhia de Água e Esgoto do Ceará – Cagece).

References

[1] Andrade, P. R. M. de; Albuquerque, A. B.; Frota, O. F.; Silveira, R. V.; Silva, F. A. da, “Cross Platform App: A Comparative Study”, in International Journal of Computer Science and Information Technology, pp. 120-123, February 2015.

[2] Apache, PhonegGap [online], http://phonegap.com/ (Accessed: January 20, 2015).

[3] Ciolkowski, M.; Laitenberger, O.; Vegas, S.; Biffi, S., “Practical Experiences in the Design and Conduct of Surveys in Empirical Software Engineering”, in Empirical methods and Studies in Software Engineering, R. Conradi and A. I. Wang, Eds., ed: Springer, 2003.

[4] Dalmasso, I.; Datta, S.K.; Bonnet, C.; Nikaein, N., “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, 1-5 July 2013.

[5] Falessi, D.; Ali-Babar, M.; Cantone, G.; Kruchten, P., “Applying Empirical Software Engineering to Software Architecture: Challenges and Lessons Learned”, Empirical Software Engineering, Vol. 15, pp. 250-276, 2010.

[6] Francese, R.; Risi, M.; Tortora, G.; Scanniello, G., “Supporting the development of multi-platform mobile applications”, Web Systems Evolution (WSE), 2013 15th IEEE International Symposium on, vol., no., pp.87,90, 27-27 Sept. 2013

[7] Gangundi, R., “Smartphone Application Development using Cross Platform Frameworks”, Proceedings of the National Conference on Information and Communication Technology, NMIMS University, Mumbai, India, 2010.

[8] Herden, A.; Farias, P. P. M; Andrade, P. R. M. de; Albuquerque, A. B, Agile PDD – “One Aproach to Software Development Using BPMN”, in 11th International Conference Applied Computing, Porto, Portugal, 2014.

[9] Intel, Intel's App Framework [online], http://app-framework-software.intel.com/ (Accessed: 20 October 2014).

[10] International Data Group, “Market Analysis Perspective: Worldwide Mobile Phone Market” [online], http://www.idc.com/getdoc.jsp?containerId=245240 (Accessed: October 18, 2014).

[11] Mahesh Babu, R.; Kumar, M. B.; Manoharan, R.; Somasundaram, M.; Karthikeyan, S.P., “Portability of mobile applications using PhoneGap: A case study”, Software Engineering and Mobile Application Modelling and Development (ICSEMA 2012), International Conference on, pp.1,6, 19-21 Dec. 2012

[12] Sensha, Mobile App Development Framework [online], http:// http://www.sencha.com/products/touch, (Accessed: January 20, 2015).

[13] Tian, L.; Du, H.; Tang, L.; Xu, Y., “The discussion of cross-platform mobile application based on Phonegap”, Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on, pp.652-655, 23-25 May 2013.

[14] Titanium, Mobile Development Environment [online], http://www.appcelerator.com/titanium, (Accessed: January 20, 2015).

[15] Trigger, The Simplest Way to Build Amazing Mobile Apps [online], http://trigger.io, (Accessed: January 20, 2015).

[16] W3C, W3C Draft: HTML5, A vocabulary and associated APIs for HTML and XHTML [online], http://dev.w3.org/html5/spec/Overview.html (Accessed: 18 October 2014).

[17] ANDRADE, PAULO R. M.; ALBUQUERQUE, A. B.; FROTA, O. F. ; SILVA FILHO, J. F. . PM5: One approach to the management of IT projects applied in the Brazilian public sector, 2015, Las Vegas, Nevada, USA. Proceedings of 13th International Conference on Software Engineering Research and Practice - SERP. San Diego, California, USA : Waset.org, 2015. v. 1.

AUTHORS

Paulo Roberto Martins deAndrade is a PhD Student in the University of Regina, Associated Professor in the "Universidade de Fortaleza" and Research in the Software Engineering. Actually, his focus is in Project Management. He has 11 years of experience in IT, with some international certifications, like PMP and Cobit. He has a master in Computer Science, MBA in Project Management and MBA in IT Management.

Fátima Aguiar da Silvahas a degree in Business Administration and MBA in Project Management. She has the PMP Certification and shehas been working for over 6 years in the field of Process Management, with an emphasis on IT Governance and Project Management.