



JÖNKÖPING INTERNATIONAL
BUSINESS SCHOOL
JÖNKÖPING UNIVERSITY

Cross-Platform Mobile Application Development

An Assessment of the Swedish Startups Landscape

Bachelor's thesis within Informatics

Author: Georges Karam

Jamoliddin Daliyev

Tutors: Christina Keller;
Andrea Resmini

Jönköping June 2015

ACKNOWLEDGEMENTS

We would never have been able to finish our project without the guidance of our committee members, help from our friends, and support from our families.

We would like to express our deepest gratitude to our supervisors, Dr. Christina Keller and Dr. Andrea Resmini for their excellent guidance, caring, patience, and providing us with an excellent atmosphere for doing the required work.

We would like also to thank all the doctors at Jönköping University for guiding us through the past three years and helping us to develop our background in informatics.

We would also like to thank our Families. They were and are always supporting us and encouraging us with their best wishes.

Bachelor's Thesis in Business Informatics

Title:	Cross-Platform Mobile Application Development: An Assessment of the Swedish Startups Landscape
Author:	Georges Karam and Jamoliddin Daliyev
Tutor:	Christina Keller, Andrea Resmini
Date:	May 2015
Subject terms:	Mobile development, cross-platform development, mobile operating systems, platforms, compatibility issues, application lifecycle.

Abstract

In mobile computing, there is variety of approaches to develop applications. One challenge is the diversity of mobile devices in the market and targeting more than one mobile device to develop applications. One approach to handle this is traditional, in which for every device there is a project carried out to develop application. Another method is the cross-platform application development which has been introduced in recent years. The purpose of this thesis is to explore if existing cross-platform frameworks can aid startup companies to develop cross-platform mobile apps. The research questions are 1) How can cross-platform frameworks be a viable approach for start-up companies in this respect? 2) What challenges and benefits do cross-platform approaches bring to developers? In order to answer the research questions, respondents from five small-sized and two large companies developing mobile apps are being interviewed. The results shows that cross-platform frameworks are good tools when developing mobile apps for multiple platforms, but are applicable to different degrees depending on the actual case. Challenges and benefits identified by the respondents mainly touched upon decreasing development time, and less coding effort, but also the uniqueness of Apple and Android products and difficulties in making developers switch to new languages.

Table of Contents

1	Introduction	1
1.1	Background	1
1.2	Problem	1
1.3	Purpose and research questions.....	3
1.4	Delimitations.....	3
1.5	Definitions	3
2	Theoretical Background.....	5
2.1	Software Development Methodologies.....	5
2.1.1	Application Lifecycle	5
2.2	Mobile Application Development	7
2.2.1	Techniques for De-fragmenting Mobile Applications.....	8
2.3	Cross-platform.....	11
2.4	Mobile Platforms Review	11
2.4.1	iPhone:	11
2.4.2	Android	13
2.5	Cross-platform vendors and tools.....	15
2.5.1	Phone Gap	16
2.5.2	Sencha/jQTouch	16
2.5.3	MonoTouch/Mono for Android	17
2.5.4	Appcelerator	17
2.6	Conceptual Framework	17
2.6.1	Five stages of Cross Platform Mobile App Lifecycle	17
3	Methodology	19
3.1	Using a Case Study approach.....	19
3.2	Philosophy of the research: Interpretivism	19
3.3	Inductive Research Approach	20
3.4	Data Collection method: Interview.....	20
3.4.1	Open interview.....	21
3.4.2	Pre-coded interview	21
3.4.3	Semi-structured interview.	21
3.5	Data analysis: Qualitative (Summarizing Data)	21
3.6	Time-horizon: Cross-sectional.....	21
3.7	Literature Review	21
3.8	Interviews	22
4	Empirical Findings	23
4.1	Info Spread AB.....	23
4.1.1	Background of the company	23
4.1.2	InfoSpread Results	23
4.2	Netzon AB	24
4.2.1	Background of the company	24
4.2.2	Netzon Results	24
4.3	Frostware AB	25
4.3.1	Background of the company	25
4.3.2	Frostware AB Results	25
4.4	NY AB	25

4.5	Toca Boca AB	25
4.5.1	Background of the company	25
4.5.2	Toca Boca AB Results.....	26
4.6	Hogia Handelssystem AB.....	26
4.6.1	Background of the company	26
4.6.2	Hogia Handelssystem AB Results	26
4.7	Sogeti AB	27
4.7.1	Background of the company	27
4.7.2	Sogeti AB Results.....	28
5	Analysis.....	29
5.1	Parameters for comparing empirical data.....	29
5.2	Parallel vs. Joint development in different stages	30
5.3	Developing cross platform mobile applications using cross-platform frameworks.....	30
5.3.1	Cross platform tools in a web environment.....	30
5.3.2	Process approach.....	31
5.4	Challenges and benefits cross-platform approaches bring to developers	32
6	Conclusion.....	33
7	Discussion	34
7.1	Results discussion.....	34
7.2	Limitations of the study.....	35
7.3	Suggestions for further research	35
8	References	36

Figures

Figure 1.1: Worldwide Smartphone OS Market Share (IDC, 2014).....	2
Figure 2.1: Waterfall Model	5
Figure 2.2: A prototyping lifecycle.	7
Figure 2.3. Spiral model as iterative lifecycle.	7
Figure 2.4. Fragmentation overview (Rajapakse,2008).....	9
Figure 2.5. De-fragmentation techniques. (Rajapakse,2008)	9
Figure 2.6. The architecture of the IOS (Apple, 2012a).....	12
Figure 2.7. The architecture of Android OS (Android, 2012)	14
Figure 2.8. Percentage of users for each tool irrespective of their primary tool (VisionMobile, 2012).....	16
Figure 2.9. Five stages of the cross-platform mobile app lifecycle (VisionMobile, 2012).....	18
Figure 3.1. Inductive Research Approach	20
Figure 5.1. Vision Mobile Cross-Platform Developer Tools 2012	30

Tables

Table 3.1. Interviews included in the study.	22
Table 5.1. Cross-case comparison of findings.	29
Table 5.2 Benefits and challenges of cross-platform approaches	32

Appendix

Appendix A.....	38
Appendix B.....	45
Appendix C.....	48
Appendix D.....	52
Appendix E.....	54
Appendix F	58

I Introduction

I.1 Background

Today, mobile device is one of the central concerns of technology. Mobile devices can already be seen as ubiquitous. As Internet matures and hardware gets stronger by Moore's Law, people are shifting to use mobile devices more often than desktop computers. Indeed, today we can observe that mobile devices are used from simple phone calls to socializing on the internet or even in warehouse management. Beside the wonder of hardware, software is the one that makes the device useful in all application scopes. In the last decade together with mobile devices' progress, mobile computing has become more permeating in software industry.

In order to develop native mobile applications, an important success factor is to take portability into consideration. Apps that run on different platforms are highly desirable by clients but are time consuming to develop. However, they target a larger number of customers in a cost-effective way (Varshney, 2004). This issue becomes very important due to the diversity of the market and the high level of competition between the companies that exists in the mobile industry today.

Undoubtedly, Apple's iPhone made a great change in the mobiles' modern trend after its first release in 2007. One of iPhone's contributions is the progress in developing featured applications for mobile devices. Mobile Software has already boomed so much that no one today would doubt about its future. Yet, it is not mature compared to desktop applications in terms of necessary tools and eloquent frameworks.

Google's Android is another major influence in the mobile industry. Android is an Open Source product differing from Apple's iPhone which gives the product flexibility in terms of use. Two vendors policy of "closed verses open source" also drive their market quite differently. iPhone dictates its market completely, while Android market is open and there are many free apps for Android in the market.

These giant mobile vendors compete with their products by trying to achieve their special features and capabilities in their product, while development companies try to have their share in mobile industry by developing applications running on the operating systems (Google's Android and/or Apple's iOS). Popularity of the mobile platforms puts pressure on developers to develop apps that run on more than one platform. Here developers are intrigued to go with cross-platform software development approach.

To develop cross-platform software, there is usually a framework used that enables building an application targeting more than one platform. That framework offers toolkits to write code and/or compile it (Lyytinen et al., 2004).

I.2 Problem

With quite different mobile platforms existing in the mobile industry, many developers are looking forward to a cross-platform toolkit that enables them to "write codes once and deploy it anywhere". This problem existed also in desktop applications but was solved by Java which allowed developers to write once and run the code on all platforms. Companies dealing with mobile app development today have to decide which platform they desire to target before moving to the designing phase of the actual app. This decision by itself is very hard due to the diversity of the mobile platforms on the market. But, it can be narrowed down to two major platforms-iPhone and Android. According to data from the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone

Tracker, Android and iPhone now hold 96 % of mobile OS market, with 84.7% and 11.7% of the market share respectively.

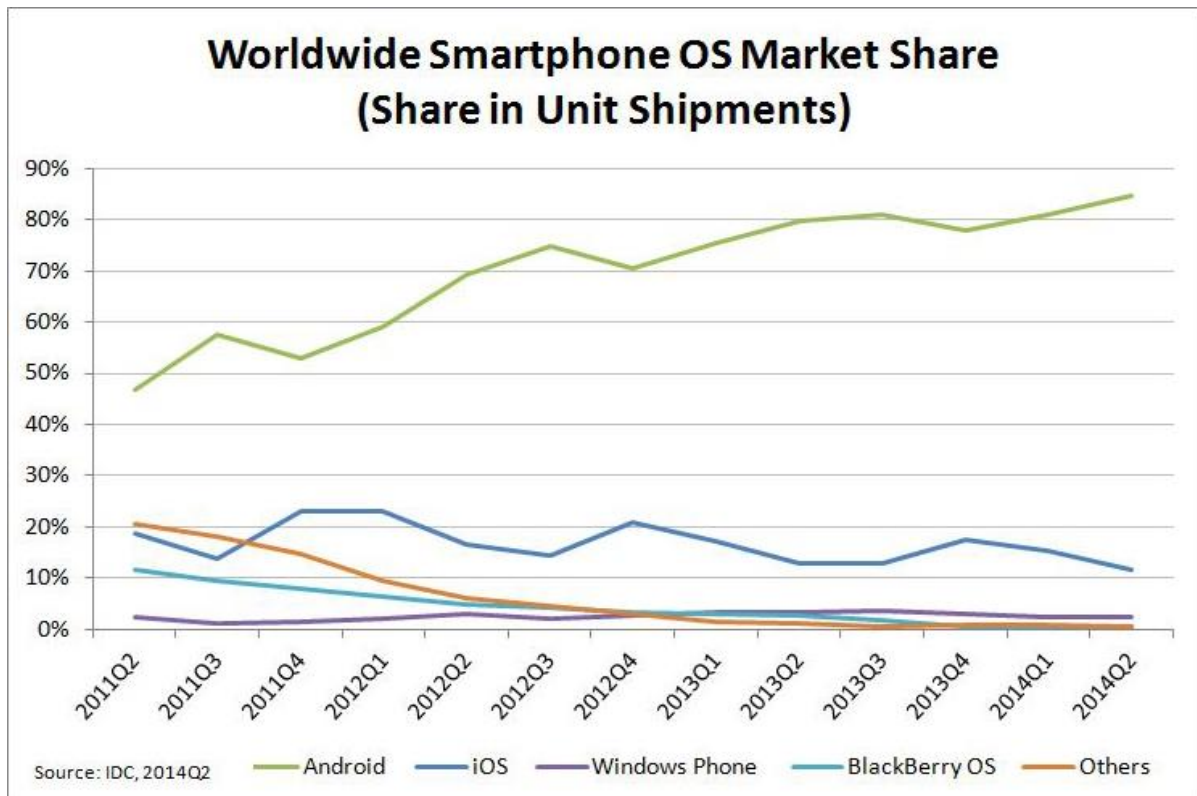


Figure 1.1: Worldwide Smartphone OS Market Share (IDC, 2014).

Targeting more than one platform for software developers is always appealing. However, due to its immaturity, crossing mobile platforms for a developer is a challenge. There is no perfect toolkit or framework that enables developers to build an application which runs on different platforms.

Although some companies claim that they have developed a framework that allows developers to write apps and run it on multiple platforms such as Monotouch/Mono for Android, PhoneGap, and Appcelerator (titanium), these tools are just available for web applications or might exist for native apps as well.

Steve Jobs, former CEO of Apple Inc., mentioned clearly that they are against allowing a third party layer of software to come between the developer and the actual platform since they lack performance and they are very slow in adapting the platform's features. Stating Job's letter "Thoughts on Flash" that has offended Adobe as well as other cross-platform tools:

"We know from painful experience that letting a third party layer of software come between the platform and the developer ultimately results in sub-standard apps and hinders the enhancement and progress of the platform. If developers grow dependent on third party development libraries and tools, they can only take advantage of platform enhancements if and when the third party chooses to adopt the new features. We cannot be at the mercy of a third party deciding if and when they will make our enhancements available to our developers."

This becomes even worse if the third party is supplying a cross-platform development tool. The third party may not adopt enhancements from one platform unless they are available on all of their supported platforms. Hence, developers only have access to the lowest common denominator set of features. Again, we cannot accept an outcome where developers are blocked from using our innovations and enhancements because they are not available on our competitor's platforms.

Flash is a cross-platform development tool. It is not Adobe's goal to help developers write the best iPhone, iPod and iPad apps. It is their goal to help developers write cross-platform apps. And Adobe has been painfully slow to adopt enhancements to Apple's platforms. For example, although Mac OS X has been shipping for almost 10 years now, Adobe just adopted it fully (Cocoa) two weeks ago when they shipped CS5. Adobe was the last major third party developer to fully adopt Mac OS X." (Jobs, 2010)

1.3 Purpose and research questions

Since the mobile industry has become highly concerned about developing apps for more than one platform, the purpose of this thesis is to explore if the existing cross-platform frameworks aid start-up companies to develop Cross-Platform Mobile Apps. Furthermore, this paper will identify the benefits and challenges that arise from using current cross-platform frameworks. The research questions of the thesis are:

1. How can cross-platform frameworks be a viable approach for start-up companies to develop cross-platform mobile applications?
2. What challenges and benefits do cross-platform approaches bring to developers?

1.4 Delimitations

This thesis is conducted to target start-up companies in Sweden that are concerned in developing mobile apps for iPhone and Android platforms. The rest of the platforms were ruled out from this study because of the fact that the above platforms occupy 96% of the market share in the year 2014.

1.5 Definitions

App - Short for Application Software. It is a set of one or more programs designed to carry out operations for a specific application. (Mobile App is short for Mobile Application).

Mobile - In this research, it refers to portable, lightweight computing devices and software for these devices.

Mobile native app - Mobile application that is developed in a dedicated programming language for targeted platform such as, Objective C for iOS, Java for Android and C# for Windows Phone OS.

Hybrid app - This type of application has cross-platform compatibility but can still access a device's hardware. It is developed using platforms such as Sencha, PhoneGap and Mono-Droid/MonoTouch.

Platform - Hardware + Operating System (+ some other software) that supports applications to run. Sometimes this term refers solely to Operating System depending on the context.

Cross-platform - Having ability to run on or being developed for more than one platform, like iOS and Android. Sometimes this term refers to only development process. In that case, the software is developed for more than one platform at once and shipped to targeted platforms.

SDK - Software Development Kit, a set of development tools used to make certain applications on certain frameworks for targeted platforms.

Xcode - Integrated Development Environment (IDE) or simply a tool used to develop applications for Apple devices.

Android – The Android Operating System is based on the open Linux Kernel.

Mobile Platform – Hardware/Software Environment for tablets, smartphones and other portable devices.

Moore's law - It is the observation that, over the history of computing hardware, the number of transistors in a dense integrated circuit doubles every two years approximately.

2 Theoretical Background

This chapter introduces the major frameworks, concepts, techniques and terms used in this thesis in order to enhance the reading of the thesis.

2.1 Software Development Methodologies

In software development, a methodology is a generalized approach that has a set of techniques and notations which support the approach. Avison and Fitzgerald (2006) describe a methodology as a collection of many components. The use of a methodology helps to produce a better quality product, in terms of documentation, meeting the requirements, maintainability, and consistency of software. There are a few famous methodologies being used in software development such as Traditional Waterfall approach, Unified Software Development Process (USDP), Agile approaches like Extreme Programming (XP) or Agile Unified Process (AUP).

Each methodology has procedures, techniques, tools and documentation aids that help the developer in his or her development process. There is also a lifecycle that organizes procedures.

2.1.1 Application Lifecycle

In software development, a systematic approach is needed to control the project lifecycle. There are different models for software development lifecycle. They divide the development process procedures into smaller tasks that can be managed more easily.

Waterfall Model

There are many variations of the waterfall lifecycle (Sommerville, 2007), differing in their phases and activities allocated to them. One specific feature and benefit of this model is that its phases have explicitly defined deliverables. Sommerville and Thomson (1992) suggests a series of deliverables produced by different phases of the model as shown in figure 2.1.

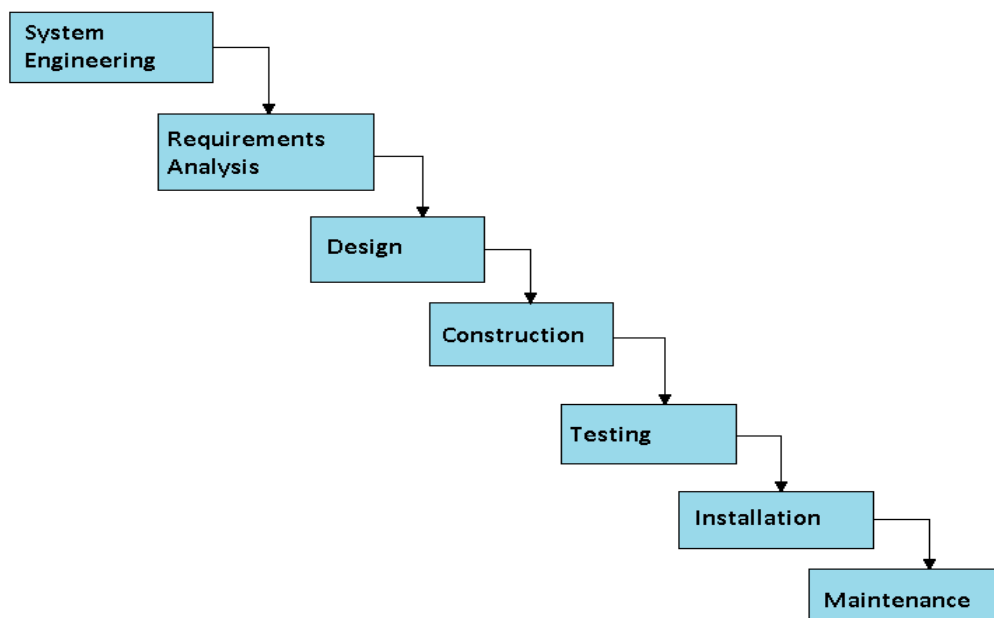


Figure 2.1: Waterfall Model

The model as illustrated in figure 21 have the following phases and typical activities in each phase (adapted from Sommerville, 1992):

System Engineering – high level architecture specification.

Requirements Analysis – requirements specification, functional specification, acceptance test specification.

Design – software architecture specification, system test specification, design specification, sub-system test specification, unit test specification.

Construction – program code.

Testing – unit test report, system test report, subsystem test report, acceptance test report, completed system.

Installation – installing system/make it ready to execute.

Maintenance – correct faults, change requests, change request report.

The waterfall model suits projects where requirements are stable and not likely to change during the development process and where there is less need for user involvement in development. Therefore, it assumes that it is not necessary to see some parts of the software (e.g. prototype) during the development process. However, these features are usually essential to successful application development, and today many projects have moved away from the waterfall lifecycle model. Instead, developers are adopting approaches that encourage greater user involvement, incremental delivery, iteration, and possibly prototyping (Bennett et al., 2010).

Prototyping Lifecycle

When waterfall model is used, the final product is produced at the end of the project. This approach has a significant difficulty in that users experience how the application works after it has been delivered. The prototyping approach helps to overcome this problem.

In software development, prototype is a partially completed application that is quickly built to explore some aspects of software requirements in early phases of the development process.

The followings are the main stages to prepare a prototype as illustrated in figure 2.2:

- Perform an initial analysis
- Define prototype objectives
- Specify prototype
- Construct prototype
- Evaluate prototype and recommend changes
- If the prototype is not completed, go to the “Specify prototype” stage.

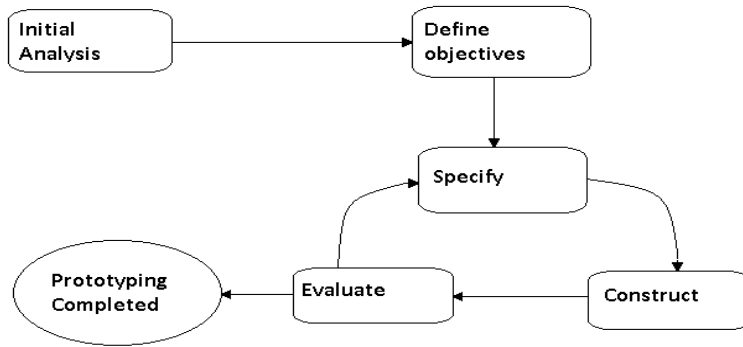


Figure 2.2: A prototyping lifecycle.

Iterative Lifecycle

Iterative software lifecycle is the most common approach today. In this approach, the project is made up of a series of development activities that are repeated. Each of these repetitions is a mini-project producing new successive or better artifacts. Gilb (1998) suggests that a large software product starts out as a small product that grows incrementally. Boehm's (1988) spiral model can be seen as an incremental delivery as illustrated in figure 2.3.

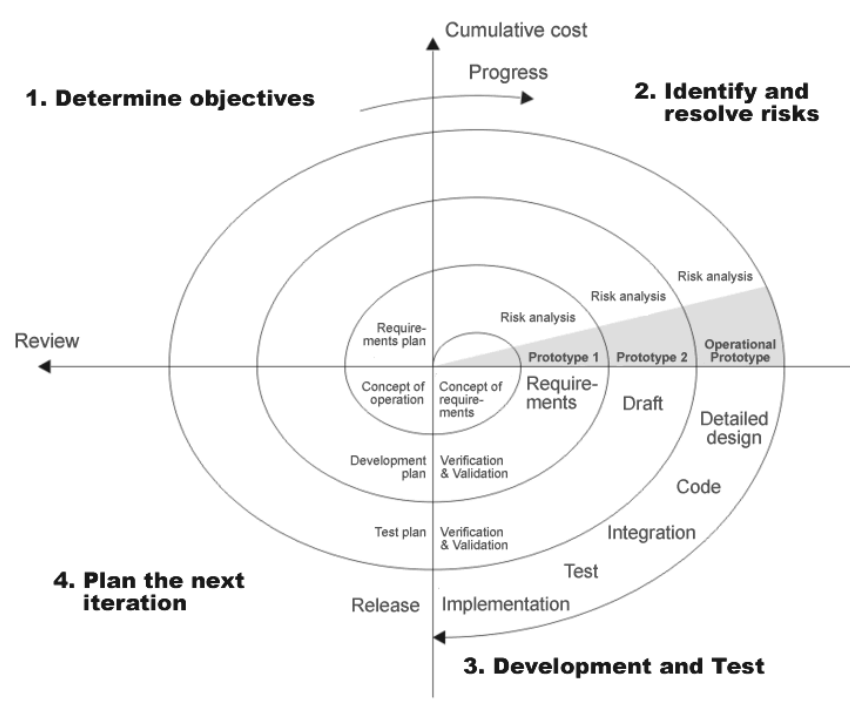


Figure 2.3. Spiral model as iterative lifecycle.

2.2 Mobile Application Development

In order to develop mobile application, mainly we need to understand the slight difference between mobile platforms and mobile operating system and their relation to mobile applications (Rajapakse,

2008). Fling (2009) refers to mobile platforms as hardware/software environments that allow services to be run on smartphones and mobile devices such as Palm, Android, iPhone, Windows Mobile, and Blackberry. Whereas, mobile operating systems “OS” is the software that communicates with the hardware and provide tools that allow application to share data and services. Examples of mobile operating systems would be Symbian, Palm OS, Mac OS X, Android, and Windows Mobile (Rajapakse, 2008).

Further, mobile application frameworks provide developers an interface to build applications using a set of class libraries. Examples of application frameworks include MonoTouch, Monodroid, Java ME, Cocoa Touch, Android SDK and Web Kit; hence, applications are built using particular application frameworks for different platforms (Rajapakse, 2008).

2.2.1 Techniques for De-fragmenting Mobile Applications

Rajapakse (2008) defines fragmentation in the context of mobile applications as the inability to “write once and run anywhere” because of the diversity of operating contexts (see Figure 2.4). The operating context (OC) for an application is described as the “external environment that influences its operation”, thus causing fragmentation. Further, operating contexts may differ from one another in the following ways:

- Hardware diversity
- Software diversity
 - Platform diversity
 - Implementation diversity
- User-preference diversity
- Features variations
- Environmental diversity

The third-party mobile application developers dealing with multiple platforms often see the diversity of operating contexts as a major problem. Thus, the development process requires more time, additional effort to keep the application updated and synchronized for different platforms, and additional knowledge of all the different device platforms. In order to fight this problem, a number of approaches and techniques of cross-platform development have emerged. (Rajapakse, 2008) in his article “Techniques for De-fragmenting Mobile Application” presents a taxonomy of de-fragmentation techniques based on the basic approach each technique uses to tackle the problem. The cross-platform development approaches illustrated in this paper can be classified by using this taxonomy. Figure 2.5 shows an overview of the de-fragmentation techniques.

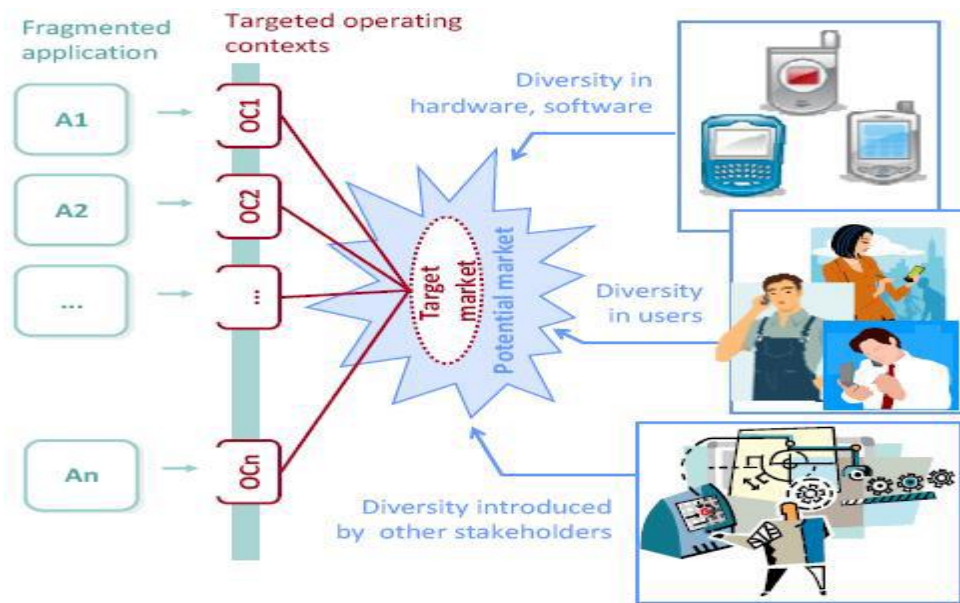


Figure 2.4. Fragmentation overview (Rajapakse,2008).

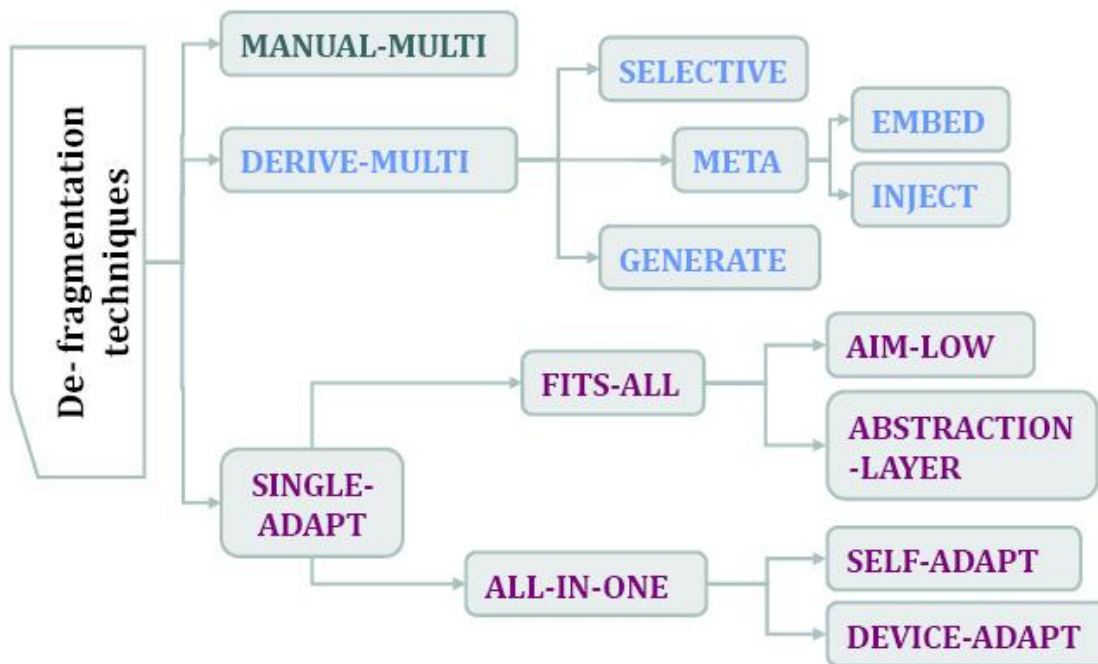


Figure 2.5. De-fragmentation techniques. (Rajapakse,2008)

Manual-Multi approach

This approach is seen as the most primitive way of de-fragmenting mobile applications by developing multiple versions of the applications manually to suit different operating contexts. The manual-multi approach implies duplication of work in several aspects of software development such as fixing the same bug in hundreds of different versions.

Derive-Multi approach

In this approach, third party developers derive multiple versions of the application from a single code base. Even if it still ends up in multiple versions of the same application, there is still only one code base to work with; thus minimizing the effort required to fix bugs and keep the application updated. The derive-multi approach is however divided into three sub-approaches referred to as Selective, Meta and Generate.

Selective

This approach localizes variations into interchangeable components such as classes and files. It also uses a build script or what so calls a “linker” to create a unique version for each operating context. The selective approach is often used when developers want images of different resolutions to fit different screen sizes.

Meta

The Meta approach specifies the way to derive Operating context specific versions of the application by using meta-programming and similar code manipulation techniques. It can be achieved by two different ways, either using an Embed approach or the Inject approach.

Embed: This approach embeds operating context specific variations into the source files of the application by using preprocessing tags or directives.

Inject: This approach involves writing the operating context code and the application code as two separate projects. By using aspect-oriented techniques, it enable developers to weave operating context code into the common application code; thus, creating an operating context specific version.

Generate approach

This approach uses an intelligent generator that is able to adapt a generic application to suit a specific operating context; thus, generating multiple versions of the application automatically.

Single-Adapt approach

Simply, this approach builds one and only one version of the application that functions on multiple operating contexts. This approach is however divided into two sub-approaches called: Fits-all and All-in-one.

Fits-All

This approach creates what so called “one-size-fits-all” application that is suitable for all operating contexts. Further, it can be achieved by two different ways, either Aim-Low or Abstraction-Layer.

Aim-Low: This approach uses what is common to all operating contexts or the lowest common denominator. For example, the images will be designated so they fit the smallest screen of the targeted devices.

Abstraction-Layer: This approach is achieved by hiding the variations of different operating contexts in an abstraction layer.

All-in-One

This approach involves the creation of a single application, which in its turn adapts to a certain operating context at run-time. It is however divided into two sub-approaches called the Self-Adapt approach and the Device-Adapt approach.

Self-Adapt: This approach enables the application to discover information about the operating context programmatically; thus, adapting itself to the operating context at run-time.

Device-Adapt: In this approach, developers write the application in an abstract manner, where the device itself decided how to adapt it to the targeted operating context at run-time.

2.3 Cross-platform

In this thesis, the term cross-platform is defined in terms of Information Technology. In this context, cross-platform refers to software or hardware that is able to work on more than one platform. In our case, we consider only cross-platform software that can run on multiple mobile platforms. It may run on as few as two or as many as existing platforms. The term platform itself may need a definition to the reader. In IT, it is usually a Device and an Operating System, which together supports applications, such as iPhone and iOS, Samsung Galaxy and Android. Thus a platform is device itself and the operating system such as iOS or Android.

There are two types of Cross-platform software: one is built individually or compiled for each platform, and the other one runs directly on target platforms without special preparations (PC Magazine, 2015).

2.4 Mobile Platforms Review

In this section, the authors present the platforms of iPhone and Android. The two platforms are compared by means of parameters such as foundation, architecture, learning curve, development, supported platforms, development tool availability, and functionality.

2.4.1 iPhone:

Apple had plans to work on a mobile phone after releasing the iPod. In 2007, Steve Jobs introduced the iPhone and was pleased with the result as Apple succeeded to combine a widescreen iPod with touch controls to create a revolutionary mobile phone and a breakthrough internet communicator. They called it the iPhone. Apple invented a new technology called the Multi-Touch which ignores unintended touches and provides multi-finger gestures. The iPhone runs on iOS and can actually be described as a scaled back version of the stationary Mac OS X.

Foundation:

The iPhone OS is primarily based on a programming language called Objective-C, which is used by developers to develop native applications for iPhone. Objective-C is a composition of the C programming language together with surplus features supporting the object-oriented programming. Further, Objective-C can be fully integrated with both C and C++ since it is based on the American National Standards Institute for C language (ANSI C) (Alessi, 2010).

Architecture:

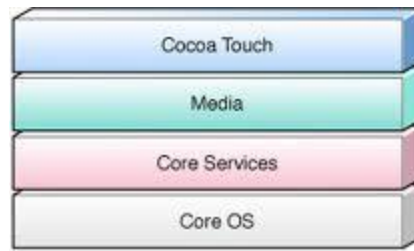


Figure 2.6. The architecture of the IOS (Apple, 2012a).

- **Core OS, Core Services:** These layers provide low-level services like file I/O and network sockets (Tilak, 2008).
- **Kernel:** The Kernels' architecture is based on the Mac architecture in Mac OS X. Communication between the applications and the underlying hardware is established through a set of well-defined system interfaces that protect the application from hardware changes (Apple, 2012a).
- **Media:** In this layer, services used to support 2D and 3D graphics. Also, audio and video are provided.
- **Cocoa Touch:** Fundamental infrastructure used by applications is provided in this layer. Additionally, it provides an object-oriented programming support for file management and network operations as well as access to the device user's profile information and other features (Rajakapse, 2008).

Moreover, developers have two different kinds of frameworks to write their codes. Higher-level frameworks that provide object-oriented abstractions makes it a lot easier to write a code since they decrease the amount of code that has to be written and encapsulate complex features such as sockets and threads; however, they are mostly used in lower-level constructs. On the other hand, lower-level frameworks give developers more control when higher-level frameworks are insufficient (Apple, 2012a).

Learning Curve:

According to developers dealing with iOS, developing native and web applications for iPhone means that the developer has to know Objective-C which is just used by Apple. Even though developers have a rich background or knowledge concerning both C and C++, they still have to practice Objective-C since structures and syntax are different making it quite challenging for developers.

Development:

Objective-C is the programming language used in X-Code which is the main tool to develop applications for iOS (iPhone and iPad). It provides whatever a programmer needs to create applications for Mac, iPhone, and iPad. Though, it is tightly integrated with the Cocoa and Cocoa Touch frameworks, it is considered to be powerful enough since the same tool is used to produce Mac OS X applications as well. In addition to that, this toolset includes an instrument analysis tool together with more supporting developer tools such as a debugger and an Apple LLVM compiler fully integrated (Apple, 2012b)

Supported Platforms:

Application development using X-Code is considered to be a closed environment, since applications developed on X-Code can only run on Mac OS X and Apple iOS.

Development Tool Availability:

Tools such as X-Code for developing applications for iPhone are available to download for free. However, developers have to join the iPhone Developer Standard Program in order to obtain a license for developing iPhone applications which costs 99 Dollars. On the other side, enterprises pay a price of 299 Dollars in order to join the iPhone Developer Enterprise Program. By doing that, developers will be able to develop and distribute their applications on App Store. (Apple, 2012c).

Functionality:

In the old iPhones' Software Development Kits (SDK), some functionalities were not applicable such as scanning for Wi-Fi network, network interface selection, battery life, level of available RAM. However, they were enabled after the release of the iOS 4.0. Location has always been supported using a GPS or Wi-Fi triangulation. The iPhone platform supports secure Internet connection by using Hypertext Transfer Protocol Secure (HTTPS) and Secure Sockets Layer (SSL). Storage is also possible using file I/O and an integrated SQLite database. Further, Application Programming Interfaces (APIs) enable developers to access data stored on the device such as media and contacts.

2.4.2 Android

Android is an Open Source Operating System based on Linux kernel, designed for mobile devices by Android Inc. which was founded in 2003 by Andy Rubin, co-founder of Android Inc. The company was acquired by Google in 2005. Since then Android is supported by Google. It is Google who released Android as an Open Source software although most of the android devices ship with a combination of open source and proprietary software including the ones developed by Google (Amadeo, 2013).

Today, Android has taken the mobile industry by surprise with major 3rd-party handset manufacturers and network operators around the world embracing Android. The leading device shipped with Android today is Samsung mobile phones. According to Statista (Statistics Portal) the shipment of android OS summed 1.6 billion devices in 2014. According to Gartner (2015), the world's leading IT research and advisory company, Android will continue to grow in 2015, with an increase of 26% year over year.

Foundation

Android is built on the Linux kernel. Android applications are developed in java and run on JVM. Developers use Google's Android SDK to build their android apps. Android NDK (Native Development Kit) is also accessible for programmers to implement parts of their apps using native-code languages like C or C++.

Although Android is an Open Source software, most of the useful services offered by Google are proprietary which makes the OS entirely not open.

Architecture:

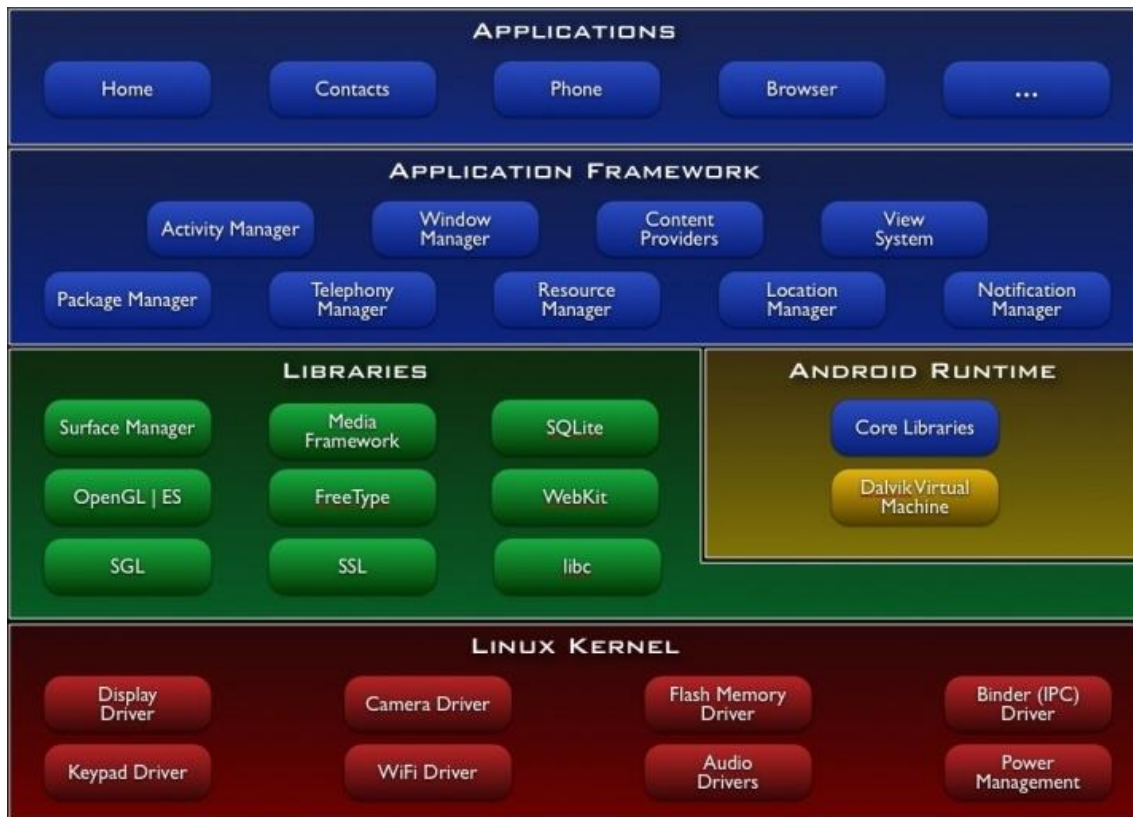


Figure 2.7. The architecture of Android OS (Android, 2012)

- **Linux Kernel:** The Linux Kernel basically handles core system operations and services and acts as a Hardware Abstraction Layer between the actual device and the Android SDK (Tilak, 2008). Examples on its functionalities are Wi-Fi Driver, Audio Drivers, memory management, process management, and other related services as shown in the figure above.
- **Android Runtime:** It includes the Dalvik Virtual Machine (DVM) and the core libraries. Based on Java VM, the optimization of Dalvik design has been achieved for the low memory requirements allowing multiple virtual machines to run concurrently (Android, 2012).
- **Native Libraries:** Those native libraries consist of shared libraries written in C or C++, assembled for the particular hardware architecture used by the phone. The major libraries are SQLite database for structures data storage and OpenGL for 2D and 3D graphics (Ibid).
- **Application Framework:** In this layer, high-level building blocks are provided with which applications are created. Some of the important classes would be the Activity manager that controls the applications' life cycle, and Resource manager and Location manager that provide location sensing options (Ibid).

Development Tool Availability

Concerning Android application development, nearly everyone uses the open source Eclipse IDE as a connector that integrates the tools of Android and emulator inside Eclipse. The Android developers even use Java programming language environments as Netbeans, Android SDK, and ADT.

Supported platforms

The cross platforms are supported from Android which uses many platforms which in its turn use higher-level libraries to simplify and organize the diverse devices.

Development tool availability

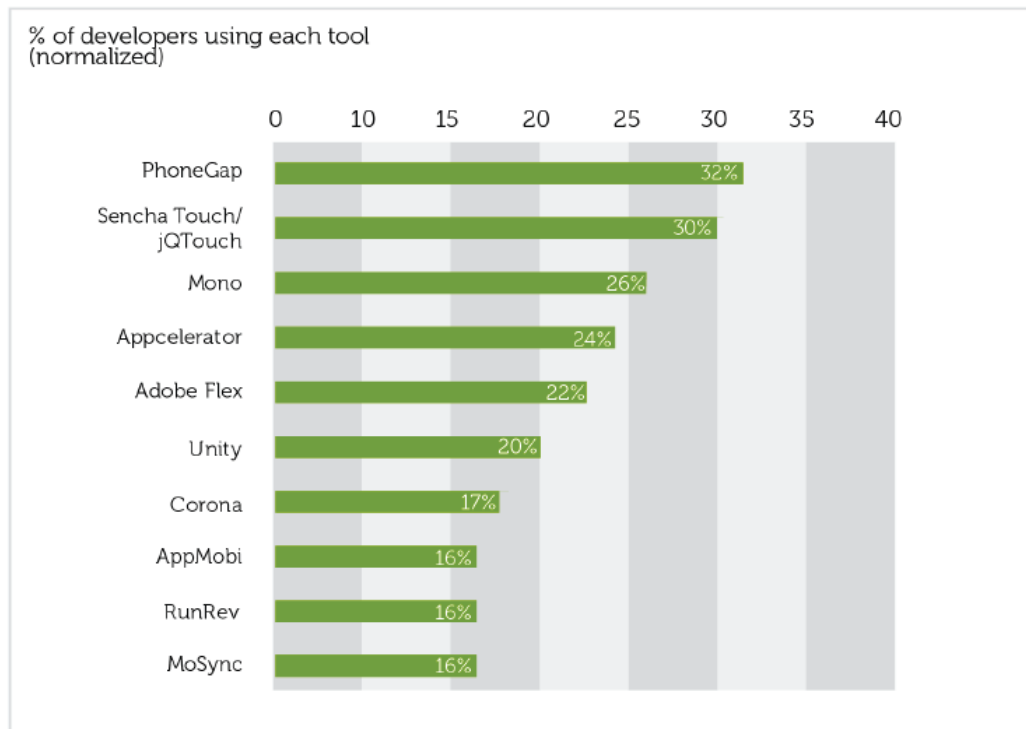
As an Android developer, you can use free tools because Android has one of the biggest and most modern companies behind it such as Google. Most programmers use Eclipse IDE which is also free to download. Publishing the app created by developers is an easy task. A \$25 fee must be paid in order to upload the app.

Functionality

It is possible for developers to get permissions in order to access hardware features or phone data in Android applications; however, basic applications do not need any permission to access phone data or hardware features.

2.5 Cross-platform vendors and tools

Researchers in Vision Mobile recently released their report about cross-platform development tools in the mobile industry (Jones et al., 2012). They listed 15 of the most popular cross-platform vendors and tools in regards to their technology, positioning and market share. They also pointed out which tools are being used most and which are dropped out by developers. According to their research, PhoneGap and Sencha shine are at the top of the list, used by 32% and 30% developers respectively. The third place is taken by Xamarin's MonoTouch/Mono for Android and the followings are Appcelerator and Adobe (Flex). The list shows percentage of users for each tool irrespective of their primary tool. The list of 10 most used tools is illustrated in figure 2.8.



Source: Cross-Platform Tools 2012 | www.CrossPlatformTools.com | February 2012

Licensed under Creative Commons Attribution 3.0 License

Sponsored by: Marmalade, Runrev LiveCode, Verizon Developer Community, Xamarin | Supported by webinos project



Figure 2.8. Percentage of users for each tool irrespective of their primary tool (VisionMobile, 2012).

The first top four tools listed above will be reviewed below. The authors decided to skip the fifth tool since Adobe Flex lately announced their retirement in the mobile industry.

2.5.1 Phone Gap

PhoneGap is an open source project originally founded by Nitobi and purchased by Adobe Systems. The global community supporting it includes freelance contributors and developers in big companies such as IBM, RIM and Microsoft. It is an HTML5 app platform that allows developing native applications with web technologies and get access to APIs and publish apps to app stores. JavaScript is the supporting language in PhoneGap while HTML and CSS are used for building user interface. The resulting application is a hybrid app. Supported platform is iOS, Android, Windows Phone, Symbian, WebOS and Bada (PhoneGap, 2012).

2.5.2 Sencha/jQTouch

jQTouch is a mobile web application framework which enables programmers to develop mobile applications with a native look for the devices of iPhone, Android using HTML, CSS, and JavaScript. It emulates mobile platforms like iPhone SDK as much as possible. JQTouch is an Open Source project founded by David Kaneda and maintained by Jonathan Stark. Sencha supports Android, iOS, Bada, BlackBerry, Symbian, WebOS and Windows Phone (VisionMobile, 2012).

2.5.3 MonoTouch/Mono for Android

MonoTouch and Mono for Android are commercial native app frameworks developed by Xamarin. Both are .NET like frameworks mimicking Microsoft's .NET framework and supporting C# programming language.

MonoTouch allows developing apps for iPhone devices while Mono for Android is used for Android app development. Mono for Android users can also use Visual Studio.

Since both frameworks are .NET based and use C# language for coding, developers can reuse their code between platforms which makes them cross-platform tool as Xamarin claims (Xamarin, 2012a).

2.5.4 Appcelerator

Appcelerator Titanium is a mobile application framework for mobile, tablet and desktop applications using web technologies developed by Appcelerator Inc. It supports Operating Systems of iOS, Android, BlackBerry, WebOS, Bada, Symbian and Windows Phone. The framework allows developing native mobile apps using HTML, CSS and JavaScript along with Titanium API (VisionMobile, 2012).

2.6 Conceptual Framework

This section presents the conceptualized framework that will be used in this research as a basis to form a suggested approach to develop multi-platform mobile apps. The authors will use this concept to answer the first research question that is to find a preferable practical approach of developing cross-platform mobile apps.

2.6.1 Five stages of Cross Platform Mobile App Lifecycle

VisionMobile researchers (see section 2.4) suggest the five stages of cross-platform mobile app lifecycle: Develop, Integrate, Build, Publish and Manage (see figure 2.9).

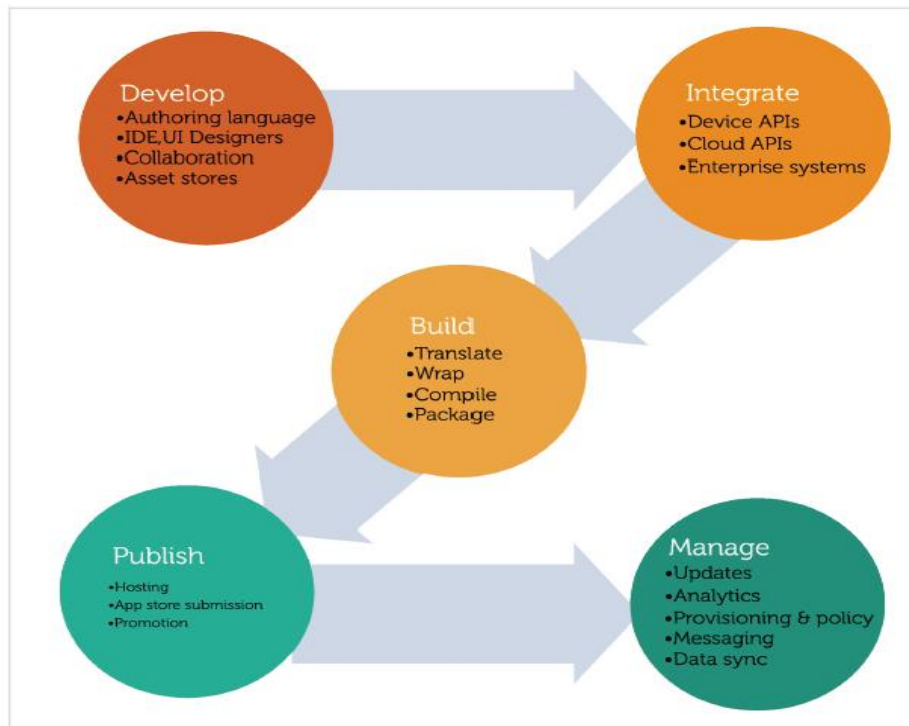


Figure 2.9. Five stages of the cross-platform mobile app lifecycle (VisionMobile, 2012)

1. **Develop.** There are many cross-platform tools to use, from entry level, simplified languages like LiveCode or Lua (e.g., Corona, DragonRad, Moai), to web languages HTML, CSS and JavaScript (e.g., Appcelerator, Application Craft, Kony, Spot Specific) to mid-level languages like Java and C#/.NET (e.g., Xamarin MonoTouch and Monodroid), or more low level languages like C++ (e.g., Qt, MoSync, Marmalade, Aqua Platform). Some of them offer a visual drag-and-drop environment (e.g., Games Salad, Yo Yo Games Maker, Appcelerator, and Spot Specific) unlike others offering only a restrictive template-based app development process (e.e., iBuild App, Wix Mobile).
2. **Integrate.** This stage involves integrating with native device capabilities, cloud APIs and enterprise databases.
A popular approach is to use PhoneGap alongside with JavaScript to integrate app with native device capabilities. This approach will produce a hybrid-native application.
3. **Build.** This stage is where cross-platform development occurs. The two most popular ones from many approaches of building apps are: compiling the code and UI templates directly to native platform binaries; and packaging code in a native shell and integrating through a runtime, which could be a “naked browser” packaged with the code, or even the device’s own browser rendering engine.
4. **Publish.** This stage involves submitting app to the app store (e.g., Apple App Store, Android Market or Windows App Hub). Many tool vendors help to manage publishing apps to the store. Some of them such as Appcelerator LiveCode and Corona also showcase apps on their websites.
5. **Manage.** App management is usually offered by enterprise-targeted apps. This stage may also include push messaging and data flow management.

3 Methodology

3.1 Using a Case Study approach

Case study is chosen to be the research approach in the study as it excels at bringing us to an understanding of complex issues. Researchers have used the case study research method for a long time across a variety of disciplines. In particular, social scientists have widely used this qualitative research to examine contemporary real-life situations. According to Yin (2003) case study research is appropriate when the focus is on a contemporary phenomenon with a real-life context, and when "how" or "why" questions are being posed. Cross-platform mobile application development is a contemporary issue and our research questions attempt to cover how and why cross-platform frameworks can be an option for start-ups in this area.

Case study method fits this study because it studies the phenomenon as it exists and uses multiple sources in the shape of multiple interviewed companies. The phenomenon in this study is the process of mobile application development, particularly, cross-platform mobile application development process. To explore the phenomenon the researchers interviewed 7 software companies.

According to Yin, there are three types of strategies for case study analysis: exploratory, explanatory and descriptive case studies. This research conducts an exploratory case study according to the research questions and the purpose of the research stated before.

We adopt the six steps of case study method proposed by Saunders, et al. (2009) based on the works of mentioned case study researchers.

1. **Determine and define the research questions**
2. **Select the cases and determine data gathering and analysis techniques**
3. **Prepare to collect the data**
4. **Collect data in the field**
5. **Evaluate and analyze the data**
6. **Prepare the report**

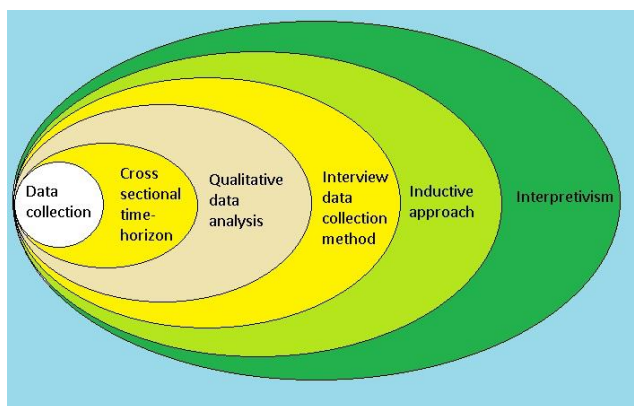


Figure 3.1. Research Approach

3.2 Philosophy of the research: Interpretivism

Interpretivistic approach of epistemology is chosen to be the philosophy for this study. Researcher who is adopting Interpretivism assumes that reality is socially constructed, meaning that our understanding of reality is not as simple as it appears. It is rather something that people in societies

form (1) their interpretation of reality, which is influenced by their values and their way of seeing the world, (2) other people's interpretation, (3) the compromises and agreements that arise out of the negotiations between the first two (Fisher, 2007).

The research uses this approach because our research analysis highly depends on the thoughts and explanations of the interviewees. They might see a single phenomenon differently and even contradict their own beliefs. If necessary, instead of simply following the answers of interviewees, the authors will try to examine the phenomenon deeper with the help of facts rather than their beliefs.

3.3 Inductive Research Approach

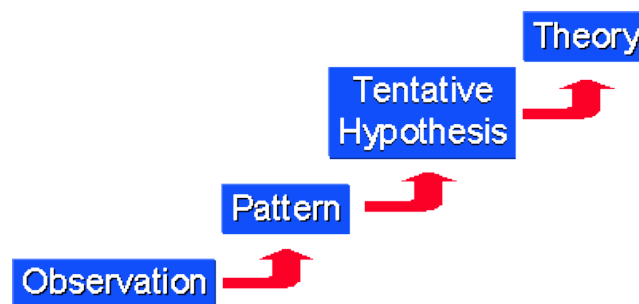


Figure 3.1. Inductive Research Approach

We are using a conducting inductive approach in our thesis. Inductive reasoning is a bottom up approach, moving from specific observations to broader generalizations and theories. As displayed in figure 3.2, first the field is observed, then from the collected data a pattern is built and a hypothesis is formed. In the end, the hypothesis is tested to develop a final concluded theory. Inductive approach is probabilistic as opposed to deductive. Usually a conclusion is drawn from past experience or experimentation. The assumption is made that because things have always been so, then that is how they will be in the future. The strength of inductive argument is often weaker than that of deductions. However, the more supporting evidence there is, the stronger the probability that the inference is true (Fisher, 2007).

In inductive studies, usually qualitative data analysis is conducted. We will also use qualitative data analysis as we will collect the data by interview method.

3.4 Data Collection method: Interview

“The great tragedy of science - the slaying of a beautiful hypothesis by an ugly fact” - Thomas H. Huxley (1870).

As mentioned before this study is an exploratory case study which is the method which may involve interviews, observations and documents for data collection. However, interviewing is the most commonly used method in exploratory researches. Interviewing method of data collection is the most suiting for our research because **it is the only technique available for us in terms of time, fund, and accessibility limits**. It can provide deep information about the existing situation. There are three techniques of interviewing that Fisher (2007) describes shortly: Open interview, pre-coded interview, and semi-structured interview.

3.4.1 Open interview

In open interview, the interviewer engages in informal conversation with the interviewee about a particular area of interest. The interviewer may drive the conversation by picking up the themes or concepts mentioned by the respondent. However, in general, the respondent leads the direction of the interview.

3.4.2 Pre-coded interview

In this technique the interview is controlled by the researcher. The interviewer reads from a prepared questions in the script and does not deviate from it. The questions are organized into a logical sequence. In pre-coded interview, for most questions the respondent is given a series of options and asked to choose any number of the given options.

3.4.3 Semi-structured interview.

This technique is in between the other two extremes. The interviewer has a schedule to remind the respondent of the main issues and topics that need to be covered by the respondent. However, the respondent has a freedom to answer the questions in the ways that is sensible to them.

The authors chose the open interview technique as appropriate for the research area and the focus of the problem. As we do not have the clear variables to measure or fixed questions to answer we have to go deeper in the field and develop them first. At this, we rely on companies in our case. We will give the respondents freedom to explain their point of views and the problems they faced or they consider.

3.5 Data analysis: Qualitative (Summarizing Data)

This study uses “Summarizing/Condensation of Meanings” analysis type of Qualitative data. The summary will compress long statements into briefer statements in which the main sense of what has been said or observed is rephrased in a few words (Kvale 1996). In this process we try to identify the relationships between the interviewees' and/or experiences and current conditions on the phenomenon in interest. The beliefs of the interviewees are also considered as affecting the nature of the data. However, the experiences of the interviewees or target companies are considered as closest to facts after the established facts we collected as a secondary data as we don't have the capability to observe the phenomenon in real event in this study.

3.6 Time-horizon: Cross-sectional

This research is cross-sectional study in cross-platform mobile app development. In this case, data is gathered just once over a period of 16 weeks in order to answer the research questions and fulfill the purpose of this paper.

The cross-sectional study is much more relevant than the longitudinal one in this case; it is due to the fact that the researcher's concern is not a long-term behavior of the companies. A longitudinal study would have suited more if we were studying the change of the developers' behavior in the long term after the foundation of a cross-platform tool (Saunders, Lewis & Thornhill, 2009).

3.7 Literature Review

Literature review was one of the main works in this research. The purpose of literature review is to form a background about our researched topic and most importantly, theories and methods that the work relies on.

Since our research area is new, the authors found the data gathering process quite difficult. Therefore, the research is not heavily based on any previously introduced concepts, although a recently introduced model was taken into consideration. The authors will explain this part in the Conceptual Framework section.

The literatures used are the following:

- Journals
- Research papers and old dissertations in the same field
- Scientific reports
- Web Sites

Search terms: Mobile platforms, iOS vs Android, mobile application development, cross-platform frameworks, development tools, application lifecycle.

3.8 Interviews

Table 3.1 displays the time, method and role of the interviewee for all the companies we gathered data from. As the table tells, most of the interviews were conducted by Skype and majority of interviewees are managers. Managers are the most suitable for gathering realistic data since they can see their company in a holistic approach and at the same time they can reach out the information that is detailed enough which the research may need.

Table 3.1. Interviews included in the study.

Company	Role of interviewee	How interview was performed	Time duration
Info Spread AB	iOS developer	In Person	30 minutes
Netzon AB	Project Manager	Skype	40 minutes
Frostware AB	Project Manager	Skype	35 minutes
NY AB	Developer	Email	----
Hogia Handelssystem AB	Regional managing director	Skype	45 minutes
Toca Boca AB	Project Manager	Skype	30 minutes
Sogeti AB	Senior Developer	Skype	1 hour

4 Empirical Findings

This chapter presents four case studies and two experts' point of view, which provide the reader with a practical view on the challenges and benefits of using a cross-platform development tool, as well as it highlights the approaches adopted by small enterprises dealing with mobile application development for multi-platforms. The units of analysis are from the same industry but with different categories of mobile applications, which includes mobile payment systems (Hogia Handelssystem AB), transports applications (InfoSpread AB) and others such as games (Toca Boca AB). This chapter portrays a general background for each company, which aim to present the companies and experts point of views in relationship to challenges and benefits regarding cross-platform tools.

4.1 Info Spread AB

4.1.1 Background of the company

Infospread is a start-up company located in Jönköping and is constituted of 5 employees. It works with mobile solutions which are orientated chiefly towards the public transport end of the market. They believe that a mobile application must be simple while still doing exactly what the user wants it to do. Their solutions and apps address a wide group, making simplicity and flexibility two important factors behind the success of the app. Infospreads' products relies on existing technology and infrastructure in order to reach the largest userbase possible, while keeping prototypes and test applications hover around the border between technical “nerdery” and the “latest”.

MobiTime, a complete public transport guide, is one of the successful apps developed by InfoSpread. It contains smart tables, buss-stop/ station information in real time, travel searches as well as providing the capability of paying for journeys.

4.1.2 InfoSpread Results

In their initial stages, they just developed J2ME applications, however today they develop apps for Android, iPhone, and Windows Phone 7.

Developers at InfoSpread AB use X-Code, Eclipse, and visual studio targeting iPhone, Android and Windows Phone 7 respectively.

The developer gave us an insight about the structure or the approach they follow from the point where they get the order from the client until they publish the application. The R&D management who is above the developers. The management collects the requirements and hand them back to developers in a written form. Later, developers start with their developing processes each sitting on different platform and working independently, where they only collaborate with the management. All the developers together with the management meet once a week where they present their work and take the next task.

The requirements given to developers are high-level requirements, where each developer work further on them according to his targeted platform. For example, if they have a requirement which says the user should be able to find their ways by a map. An iPhone developer will not have any problems to do that since the solution is right there and Google Maps feature is already implemented on the iOS. On the other side, other developers working on different platforms must find their own solution. So, the difference is only in the use case at a lower level of abstraction. Shoshah pointed out that using the SCRUM method makes their application development iterative and incremental.

When asked if the management checks the requirements after being specified, Shoshah said that developers never start writing codes without the managements' approval. Moreover, they do not share their solutions with each other rather than just with the management.

Along the interview, the interviewee stated that he suggested a cross-platform tool (titanium) to cut down the development time and to be more efficient but the idea was rejected by the management. He claims that they did not want to risk. One major reason is that developers are used to their own tool and language and they are good at it, so it is very hard to force them use a different tool. Another reason is that it will become problematic to target multi-platforms which have different form factors and different features embedded. For example, Android itself has various packages which are modifiable. Additionally, device vendors are now modifying Android for their devices' specific and unique features. Hence, after the discussion with the management, he does not advocate cross-platform tools anymore.

To view the full Interview Q/A, see Appendix B.

4.2 Netzon AB

4.2.1 Background of the company

Netzon is a new and fast growing company that is headquartered in Stockholm. It has also two other offices in Nyköping and in Davao City (Philippines) operating in the international market in the field of application development. Their solutions range from native to web apps targeting multi-platforms depending on the clients' needs.

Netzon is composed of a team that share strong interests in technology and design, who in various ways complement each other, offering a wide range of services and technology solutions while keeping the high quality.

4.2.2 Netzon Results

Patrik works for Netzon. Mostly, they develop applications for iOS but recently it is becoming more for Android. He says that clients use to ask for the iPhone first and then for Android, but now they ask for Android first and then the iPhone since there are more and more Android phones on the market. Netzon have cases where they develop for both platforms at the same time and are getting a lot of requests concerning tablets as well. Patrik mentioned that they do cross-platform development, but with that he meant that they reuse the code that has been developed for iPhone first and then apply it on Android with some adjustments; however, they do not use cross-platform tools. Instead they use X-code for iOS and Eclipse for Android.

Navigation is one of the challenges that the developers at Netzon face when targeting both platforms, the back button on Android does not exist on the iPhone so they have to find their ways around that. Even though they reuse the code, they cannot do copy paste and have an iPhone design on an Android phone because a lot of Android users do not like that. According to Patrik, many users say that the app looks better on an iPhone than it does on the Android- design wise. Another challenge would be that developers targeting iOS are not allowed to assign a function to record phone calls because of the Apples' restrictions and policies; whereas, they can pretty much do whatever they want on an Android platform since it is an open platform.

To view the full Interview Q/A, see Appendix C.

4.3 Frostware AB

4.3.1 Background of the company

Frostware AB is a consulting company established in summer 2010 and located in northern Sweden. They design and develop applications for iPhone, iPad, and Android devices starting from the first idea until the final publishing process.

Frostwares' mission is to develop apps that are fun, interesting, and offer more functionality than a simple shell for a website. They try to make best use of the features the platform provides such as the camera, GPS, and graphics card to develop creative apps benefiting both the user and the company owning them.

In October 2010, they released their first application or game in the Apple's App store, namely Skycat and the Star children. The game was well received with reviews from Touch Arcade, SlideToPlay, the portable gamer, and the Swedish magazine "Allt om iPhone" among others. Skycat has now been downloaded 180,000 times so far and has an average score of 4.5/5 on the App Store. Since then, they have developed applications targeting both Android and iOS.

4.3.2 Frostware AB Results

Martin Gudmundson is an Android developer and the CEO of FrostwareAB. They develop applications for Android and iOS for the moment and they started with Windows Phone 7 but haven't really had big projects for this platform. Martin mentioned that they do not use cross-platform tools since the apps they develop are very advanced and more than shell-apps or standard-button apps, so they cannot write the same code targeting both platforms. Instead they use Eclipse, X-code, and visual studio for Android, iOS and Windows phone 7 respectively.

4.4 NY AB

NY AB, is a company developing mobile Apps on both Android and IOS platforms.

NY AB team confirmed that eventually the clients will need cross-platform mobile applications starting first with an IOS application and after IOS successful release, the clients refer back to the company ordering an Android version for the same app.

The team major concern is the reusability of the code which is in most cases difficult because of the unicity of some features in each mobile platform. In addition, the Titanium tool used in NY AB produces a native code after converging the original Hybrid code and the developer's team does not have full control over the generated version. Even after huge efforts to enhance the original code, when the application gets bigger, the problem will relatively multiply.

In their opinion, *"For more advanced app that has lot of nested views, is built on a lot of graphics or need to do some heavy calculating like image/video processing then native is the only way to go"*

To view the full email conversation, see Appendix F.

4.5 Toca Boca AB

4.5.1 Background of the company

Toca Boca is a play studio that develops games or digital toys for kids. Their games are educational and tend to stimulate the kids' imagination.

Toca Boca's team consists of experienced play designers, art directors, developers, interaction designers, and marketers. They were established in September 2010, and their first products were launched in March 2011. Their games target the iOS platform but they are however planning to start developing for Android as well.

After being a part of the Bonnier Group, a family owned business that started as a book publisher in 1804, the group has grown to include newspapers, magazines, television, cinemas, and digital media.

4.5.2 Toca Boca AB Results

Jack Nutting differently approaches this subject, he benefits from the power of internet and tried to exploit his server/ backend usability to the very limit by minimizing the amount of data available on the devices and putting the load on the server.

Jack Nutting is the only interviewee who puts the spotlight on company's policies and does not believe that technology is the real problem, as he states: *"So technology wise it's possible to different software to share same technology layer development in top of them but they are not really interested in doing that"*.

On the other hand, he finds in a perfect cross-platform development tool a huge advantage for the industry since a great amount of coding effort time could be spared and a lot of developers familiar with html5 and CSS could find their way easily into the mobile development industry.

To view the full Interview Q/A, see Appendix D.

4.6 Hogia Handelssystem AB

4.6.1 Background of the company

Hogia Handelssystem AB was founded in 1980 by Bert-Inge Hogsved. Hogia has been dealing with general software for finance and personnel for many years, but have recently started to invest heavily in industry-specific software such as accounting firms, real estate companies, construction companies, retail, fleet operators, and public transport. Today Hogia is the world's largest supplier of standard software for booking and administration in the ferry service.

Hogia has customers in four continents and maintains an active internationalization of the business in the area of transport systems. Hogias' performance continues to improve with a turnover of 400 million Swedish crowns amounted to 2009, after nineteen consecutive years showing positive performance figures.

Hogia has a mobile app development department that deals with both mobile payment systems, transport and logistics. They target iOS and Android, while keeping windows phone as a learning project since there were not a lot of customers investing in it (Hogia Handelssystem AB, 2012).

4.6.2 Hogia Handelssystem AB Results

Walenius considers cross-platform tools to be challenging more than being beneficial since they are not completely mature yet. He pointed that it is hard for the customers to understand why the development of a complex mobile application should be done in two separate projects, one for Android and the other for iOS when cross-platform tools exist. *"Of course, there are tools and platforms to develop codes and applications that run on both platforms but they are difficult to use when you either need capacity*

or you need a quick response in the UI or if you need specific hardware services and features to be used. The frameworks are getting better and better to have similar implementations but still they are on different code bases since they use Java and C variant, so it needs to be done twice if done natively” (Walenius, 2012).

Walenius further gives us an insight or structure of the process approach when developing a native app targeting more than one platform. He considers the agile environment to be soft and easy going. In the demand specification, the manager sits down with the client and tries to understand what he is trying to accomplish. When they have understood what the client wants, the developers create something they call the backlog, and from the backlog they pick up the most important issues for the customers and start to implement those for one sprint. A sprint regularly takes from 14 days up to one month. When the first sprint is done, they move into that kind of iteration process and get back to the backlog where they strike out what they have accomplished so far and bring again the prioritization list and pick up the important issues for the second sprint. Clients are involved in that process, as well as in the development process as they can see, test and give direct feedback if the development is on the right track or not.

When asked if there is any nominated language or tool that is going the Javas’ virtual machine way by being the first ideal tool to cross the platform natively, Walenius answered that natively he sees no hope of having different device manufacturers coming together in one platform language, which is a part of the game according to him. *“Manufacturers are trying to make something unique that will give them a market preference and Apple has been successful in this” (personal communication, Walenius, 2012).* Looking at Apple apps and comparing them to Android apps, there is a complete difference in them and that is that most iOS applications are looking much more professional than many of the Android are. There is also a difference in who pays for applications and where will there be money to make and where there is no money (Walenius, 2010). However, the only candidate for that is the HTML5 standardization, but they still have a couple of years left to go. Their intention is right and what we see in the market with the tools is also right.

Hogia investigated different tools and platforms for cross-platform development. They tried out PhoneGap, Appcelerator, and evaluated Mono in theory. Some developers at Hogia had previously used Mono in hobby projects at home and they said that it was not really that stable to be used in a production environment. They did not adapt this tool since they neither knew what it would create nor the risks behind it. Instead, they still separate the projects when developing the same app for multi-platforms. Walenius stated that there are efforts being done to provide developers or start-ups with an ultimate tool that allows cross-platform development. *“But my conclusions after I have been doing that for several years is that there is no simple way in general. There might be a possibility of reusing codes and using tools that allow cross-platform development but that is decided upon the task and the size of the project” (Walenius, 2012).*

To view the full Interview Q/A, see Appendix A.

4.7 Sogeti AB

4.7.1 Background of the company

Sogeti is a leading provider of professional technology services. It is specialized in Application Management, Infrastructure Management, and High-Tech Engineering and Testing. Sogeti operates as a consultant company as well, that enables its clients to leverage technological innovation and achieve maximum results by working closely with them. Sogeti is composed of more than 20 000 professionals distributed in 15 countries and is present in over 200 locations in Europe, the US and India. Sogeti is a wholly-owned subsidiary of Cap Gemini S.A., listed on the Paris Stock

Exchange. Despite providing a team of experienced IT professionals, they deliver practical services based on more than 40 years of global experience.

Sogeti is also the founder of ViNT (The Institute for the Analysis of New Technology), which interprets the implications and impact of emerging technologies in business and in day-to-day life. ViNT has published a large number of books, since its creation in 1994 (<http://www.sogeti.se/sv/In-english/About-Sogeti/>).

4.7.2 Sogeti AB Results

Mr. Chris Forsberg broke down the development of a mobile application for top to bottom passing through the needed platform for this application, the design and general layout of it and finally to the development of this application.

Mr. Forsberg points also to the fact that mobile platforms are becoming more and more demanding regarding the unity for each (iPad, iPhone, Android Phone and Android Tablet). Each of the mentioned channels needs a special treatment in the development which makes it almost impossible for the developers to group all of these platforms in one unified code.

In this interview, another point was spotted which is the market trend and user preferences in design and functionality in each and every application. *“For example, if I put a back button that looks like an iPhone button in an Android app, the Android customer would then remove the app from the device”.*

He continues this point and make it stronger by stating that hybrid applications can be used but they have to be almost identical which is not applicable nor acceptable for both Android and iPhone customers.

To view the full Interview Q/A, see Appendix E.

5 Analysis

In this section, we answer the research questions based on our empirical data. Since we have multiple-interviews with different companies, our main path of analysis will be comparing. While comparing the companies' approaches and methods, we try to find a pattern helping us to form a theory answering our research question.

5.1 Parameters for comparing empirical data

We choose parameters for comparing empirical data. Parameters should help to narrow the data down towards the research questions. The parameters we chose are the following:

1. Size of the company. The size is categorized in three types – small, medium and big.
2. Platforms to develop app.
3. Tools used.
4. App type: Native, web or hybrid app.
5. Developing methodology they use.

Table 5.1. Cross-case comparison of findings.

Company	Company size	Platform to develop app	Application type	Methodology used	Tools used
Info Spread AB	small	iPhone, Android	Native	Waterfall	X-Code, Java Eclipse
Netzon AB	small	iPhone, Android	Native, Web	Agile, Waterfall	X-Code, HTML5
Frostware AB	small	iPhone, Android, Windows	Native	Waterfall	Java, X-code, .Net
NY AB	small	iPhone, Android	Hybrid	Agile	Titanium
Hogia Handels-system AB	big	iPhone, Android	Native, web, hybrid	Agile, Waterfall	X-Code, HTML5, CSS
Toca Boca AB	small	iPhone, Android	Native	Sprint (like Scrum)	X-Code, Java Eclipse
Sogeti AB	big	iPhone, Android	Native, Hybrid	Agile	Appcelerator

5.2 Parallel vs. Joint development in different stages



Figure 5.1. Vision Mobile Cross-Platform Developer Tools 2012

The companies' development phases can be compared against the Cross-platform Mobile Application Lifecycle described in the conceptual framework. As the interviews show companies seem to start with universal approach to all platforms in the early stage and later on split the project as the platforms require different implementation.

We can illustrate the pattern in the diagram similar to the one of five stages of cross-platform mobile app development. Figure 5.1 shows which companies keep the project of different platforms together in which stages and where they split them.

5.3 Developing cross platform mobile applications using cross-platform frameworks

5.3.1 Cross platform tools in a web environment

Cross-platform tools work perfectly when developing web mobile apps for multi-platforms but there are still some differences there as well. That is due to the HTML5 standard that is not really finalized yet and that shows in the code, as the developer needs to put a specific code segment that are pointing for different web engines. For instance, if the developer is targeting the iOS environment, he needs to target some HTML5 tags with Safari specific code and the same goes for Firefox and Chrome (personal communication, Walenius, 2012). Hence, for a while longer they need to be kept as two separated projects and know which browser the developer is targeting for his application at the moment.

Furthermore, there is a huge difference between a native app and an app developed using a web language but executed in a native shell (personal communication, Forsberg, 2012). This shows in

the user experience. When the user starts to use the web application, they get directed to a web browser and at first they see an address field that is actually going out to the internet. This address field might disappear if the programmer has used the option of hiding it, but the users still have a feeling that they are in a web browser. However, the graphical user interface possibilities with the HTML5 are getting much richer, allowing the app to look more like a native app (Walenius, 2012).

5.3.2 Process approach

Regarding the process used at Hogia, first they get a rudimentary demand specification from the client, which mostly do not have a deep down drilled requirement specification and that is a problem. If the developers move on with that very overview picture that the client provided, they might end up with something that is not what the customer really wanted. The first and very important step is to drill down the demand specification by using either the agile or waterfall development. The waterfall project provides standardized phases with demand specifications, design phases, and development phases and can directly be transformed to a mobile project. On the other side, if the consultant goes with the agile development, they can directly move from the demand specification and involve the client with the development phase very early; thereby, they can together with the client get much better possibility of finalizing the project to the initial requirements that the client wanted while keeping the possibility to adapt the project to any change in demands if any during the development phase (Walenius, 2012). According to Walenius, the choice of the structure approach lies behind the history of the client as well. For example, a telecom or a manufacturing company are customers to waterfall projects since many years back and that is often what companies use in large projects today, but there is a mixture between the two models since parts of the larger waterfall projects may be agile as well. After all, the consultant needs to investigate what the customer preference is. If there is a possibility of getting the client to understand or try out the agile development model, then he should definitely go for that since it is more productive.

Both iOS and Android are in the same requirement meeting or workshop, since a mobile application should not be as it was for a couple of years ago- device specific. The needs of the clients' customers are probably universal, so the requirements should be developed and deployed on any platform if the client has the time and the money to do it. Producing the backlog or the requirement specification documentation should be the same for all platforms.

Each and every start-up company needs many features to succeed in their field. For instance, in mobile application development field, cross platform development can be very viable for these companies to reach their goals and achieve a decent start-up since it can help them get the best of the most important key features of a good start.

In fact, cross platform allows these companies to develop application that could be exposed to the majority of the public, reaching different mobile platform and especially Android and IOS which possess up to 96% of the mobile market. This exposure will make it easier for them to make their market breach and enter the market from a strong point with a well-known application.

In addition, when the vast exposure is accomplished, marketing is an easier task since the marketing campaign will be viewed by wider audience, and this will automatically lead to a larger number of potential clients which propels the company to a better turnover and expansion.

Furthermore, the new clients will not create additional cost for the company even though they require developing the same application on different platforms since the development is done once. Moreover, application updates are also developed once and deployed everywhere which reduces the cost of maintenance for the developing company as long as the client.

Finally, all platform users will have the same experience in using the application regardless of the used platform especially if no special features are included in the application.

5.4 Challenges and benefits cross-platform approaches bring to developers

Table 5.2 summarizes the challenges and benefits that cross-platform approaches bring to developers. These results are based on the interviews conducted with experts and developers.

Table 5.2 Benefits and challenges of cross-platform approaches

Company	Benefits	Challenges
Info Spread AB	Cut down development time and be more efficient	There is a high risk in forcing developer to switch from their own tools and languages to adopt new tools.
Netzon AB	Re-using the same code with some modifications instead of re-writing all the code	Users of a specific platform do not like the look of any other platforms. Apple restrictions and policies, navigation button and phone call recording are good examples
Frostware AB	Can be used for simple button apps	Advanced and shell applications require native coding languages
NY AB	Lighter and less advanced applications can be developed much faster than with native codes.	Uniqueness of some features in each mobile platform.
Hogia Handelssystem AB	Perfect for mobile web applications	Not completely mature. Manufacturers always try to make something specific to get market edge
Toca Boca AB	Less coding effort. HTML5 and CSS developers can enter the mobile development market.	Manufacturers company's policies are highly against cross-platform development.
Sogeti AB	No benefits	Mobile platforms are increasing their demand regarding uniqueness.

6 Conclusion

Cross platform application development is viable for startup companies since it enlarges their exposure to their new market leading to a higher turnover. In fact, cross platform development is assuring the same user experience in all mobile platform resulting in better and enhanced marketing campaigns.

Although developing mobile applications using native approaches is the ideal approach when the application requires full platform features and high potential of hardware specs, when developing standard applications that do not require high graphics potentials nor intensive resources, other approaches might also be applicable to avoid high costs. Since native approaches are usually much more expensive than other approaches, developers tend to take the cost factor into consideration when deciding which approach suites best to meet the applications' and business requirements.

Therefore, the cross-platform application development tools, which made some of the applications single code writing possible. These tools resolved a part of the problem because they provided this feature for applications that doesn't require the use of the majority of the resources nor the hardware of the smart device. The search is still active for the best solution to replace native code with a unified code. In fact, the cross platform development is progressing each day and becoming more stable and mature since HTML5, CSS3 and JQT (jQuery Touch) are evolving and integrating new features for mobile development day in day out. Moreover, HTML5 today supports many features like Offline and storage, file and hardware access, graphics and multimedia. On the other hand CSS3 is more and more advanced to integrate graphics and shapes, rotations and circular forms which give the developers and designers to advantage to be creative without worrying about the limits of cross-platform development. Furthermore, the integration of web browser in all mobile operating systems makes the developers fearless about approaching any mobile application through the web platform development. Many development tools were created for this type of development such as PhoneGap, Titanium, etc. Most of these tools are used by the companies interviewed throughout this study and has been considered as an alternate solution for native application development. These tools can easily be considered as a benefit for any rising company for many advantages such as:

- Developing with already standardized web technologies.
- Code once; Deploy anywhere
- Reduced cost and time of development
- Fast prototyping and code reusability are now possible

On the other hand, Apple IOS and Google Android are now the two major operating platforms that are competing to make the bigger and better influence on the developers and Mobile application developers. That is by adopting and hiring creative and innovative developers to create exclusive applications for their platform, this so called war between these two giants is considered as the biggest problem facing the cross platform approach. In fact it is a huge advantage and a considerable win situation for one of them to create and publish a new application. This war also affects the way the hardware is manufactured with closed environment where only a dedicated language and a specific type of coding can exploit the full power and capabilities of the device's hardware (VisionMobile, 2012).

7 Discussion

7.1 Results discussion

After revising and taking into consideration all the interviews and opinions of the esteemed experts, one can deduct that cross-platform immaturity is still nowadays presenting huge problems for developers from which we state the following:

1. Cost and time losses developing the same application in both native languages.
2. Long design processes to satisfy both customer bases for Android and iOS in each and every channel for each application.
3. Immaturity of hybrid and third party application in comparison of the native code (design wise, less control of resources, etc.)
4. Company Policies nowadays not ready to be cooperative with each other to facilitate the process of mobile application development.

Although all these features are already present and functional, the professionals are still facing the same issue regarding the applications in need for the communication with the core of the operating system and exploiting the hardware of the mobile devices. In the latter case all the professionals interviewed in this paper referred back to the native applications. On the other hand, they all have high hopes for the breakthrough to be done with cross platform development but they are concerned that at the end of the day, cross platform are always on a different code base of the native application as Mr. Walenius clearly states: *“Of course, there are tools and platforms to develop codes and applications that run on both platforms but they are difficult to use when you either need capacity or you need a quick response in the UI or if you need specific hardware services and features to be used. The frameworks are getting better and better to have similar implementations but still they are on different code bases since they use Java and C variant, so it needs to be done twice if done natively”*

On the contrary of all the experts this paper mentioned, one has gone on the other side of the problem. The majority of the experts blamed the technology for the immaturity of the cross-platform development and they were helpless in face of huge applications. They all preferred to refer back to the native coding for these type of application. This expert spotted an important point while other tried to overlook it or even didn't pay attention to it, it is simply these two giants will to cooperate and make their hardware compatible with each other resulting in a much easier, more effective, and cost less for the development firms alongside their clients.

Furthermore, these manufacturers find the entrance of the cross platform development to the mainstream of the mobile development a very harmful issue for their market share. As an example, if cross platform was able to complete its breakthrough and stand out between the native mobile application developments, some less popular platforms will have the chance to re-enter the market and become serious competitors such as Windows and RIM blackberry. That's why, these two will do whatever it takes to maintain control over the mobile development business and create a developer lock-in in order to have the best developers on their side. In addition, if the cross platform emerges, windows is more than capable of catching up and maybe even becoming better than the two dominant companies nowadays (VisionMobile, 2012).

That being said, one can easily blame the companies for their harsh decisions about the difficulties faced in the design and development of any application in need of the phone/tablet resources (Memory, Processor, Multi-Touch Screen, etc.); although, the combination of HTML5, CSS3 and JQuery Mobile can, in a couple of years, present a permanent replacement for native application

development resulting in a more effective way to produce and publish applications on both platforms at the same time. However this solution could be applicable depending on Apples' and Androids' will to cooperate and unify system resources on all mobile phones and tablets.

Finally, one can only hope such cooperation exists and the cross-platform development reaches a level of maturity in order to be able to handle large and complicated applications. However this is all relatively dependent of these two giants' choices.

7.2 Limitations of the study

This research was started in early 2012. It is now finalized in 2015. There was an interrupt for about 3 years because of some issues. The data was gathered in the midst 2012. Since IT industry changes very fast today our gathered data can be a little outdated.

Another limitation of our research study was the means of interviewing the companies. Due to the lack of budget and time we interviewed them from distance, mostly via Skype. That said, we did not have any chance to participate or observe any event during data gathering. Yet our methodology is based on Interpretivism which requires researchers to closely learn the interviewees and their relationship to the phenomenon to better understand their way of thinking and explaining the phenomenon.

7.3 Suggestions for further research

This research can be seen as a gist in learning this topic. The further research should be strengthen the quality of gathering data. Having chosen qualitative approach the researchers need to examine the phenomenon by participating the event and observing it. To compare the events in different companies their culture and way of thinking also should be learnt. Comparison of the culture of the companies involved in the research will help researchers tremendously to objectively reveal the phenomenon. To sum up, instead of just interviewing, other closer ways of gathering data also should be used. Our suggestion would be that giving the method of observing the highest importance even rather than participating suits the aim of the research.

8 References

- Alessi, P. (2011). *Beginning IOS Game Development*. John Wiley & Sons.
- Amadeo, Ron. (2013). Google's iron grip on Android: Controlling open source by any means necessary. Retrieved May 20, 2015 from <http://arstechnica.com/gadgets/2013/10/googles-iron-grip-on-android-controlling-open-source-by-any-means-necessary/>
- Android. (2012). *Android M developer's preview*. Retrieved June 18, 2012 from <http://developer.android.com/>.
- Appcelerator. (2012). *Native apps. Mobile APIs. Real-time analytics. One Platform*. Retrieved July 13, 2015 from <http://www.appcelerator.com/developers>
- Apple. (2012a). *Developer* Retrieved July 23, 2015 from <http://developer.apple.com>
- Apple. (2012b). *Xcode: The complete toolset for building the right app* Retrieved July 23, 2015 from <https://developer.apple.com/technologies/tools>
- Apple. (2012c). *iOs dev center*. Retrieved July 23, 2015 from <http://developer.apple.com/devcenter/ios/>
- Avison, D. & Fitzgerald, G. (2006). *Information systems development: Methodologies, techniques & tools*. London: McGraw Hill.
- Simon Bennett & Ray Farmer. (2010). *Object-Oriented Systems Analysis and Design Using UML*. London: McGraw-Hill.
- Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, 21(5), 61-72.
- Cosgrove, T. (2015). *Mobility has a major impact on IT support*. Gartner.
- Fisher, C. (2007). *Research and writing a dissertation: a guidebook for business students*. 2nd ed. Harlow: Prentice Hall.
- Fling, B. (2009). *Mobile Design and Development: Practical concepts and techniques for creating mobile sites and web apps*. O'Reilly Media, Inc.
- Gilb, B. T. (1998). Process Out Quality In. In *Sixteenth Annual Pacific Northwest Software Quality Conference*.
- Huxley, T. (1870). *Critiques and Addresses*. Published in *Collected Essays by Appleton ed* in 1893.
- International Data Corporation (IDC). (2014). *Worldwide Smartphone OS Market Share 2011-2014*. Retrieved May 12, from <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- Jobs S. (2010). *Thoughts on Flash*. Retrieved March 12, 2012 from <http://www.apple.com/hotnews/thoughts-on-flash/>
- Jones, S. (2012). *Cross-Platform Developer Tools 2012*. Retrieved on May 24, 2015 from <http://www.visionmobile.com/blog/2012/02/crossplatformtools/>
- Kvale, S. (1996). The 1,000-page question. *Qualitative inquiry*, 2(3), 275-284.
- Lyytinen, K., & Rose, G. M. (2004). *How agile is agile enough? Towards a theory of agility in software development*. Sprouts, Association for Information Systems.
- Phonegap. (2012). *Easily create apps using the web technologies you know and love: HTML, CSS, and JavaScript*. Retrieved August 2, 2015 from <http://www.phonegap.com>

- Rajapakse, D. C. (2008). Techniques for De-fragmenting Mobile Applications: A Taxonomy. In *SEKE* (pp. 923-928).
- Saunders. M., Lewis, P., & Thornhill, A. (2009). *Research Methods for Business Students*. Harlow: Prentice-Hall.
- Sommerville, I., & Thomson, R. (1992). Configuration specification using a system structure language. In *Configurable Distributed Systems, 1992., International Workshop on* (pp. 80-89). IET.
- Sommerville, I. (2007). *Software engineering 8*. 8th ed. Harlow: Pearson Education Limited.
- Statista (2014) Smartphone OS worldwide by installed base in 2014 (in millions). Retrieved on May 10, 2015 from <http://www.statista.com/statistics/385001/smartphone-worldwide-installed-base-operating-systems/>
- wsdl2objc. (2012). *Generates Objective-C (Cocoa) code from a WSDL for calling SOAP services*. Retrieved on May 10, 2012 from <http://code.google.com/p/wsdl2objc/>
- Varshney, U. (2004). Vehicular mobile commerce. *Computer*, 37(12), 116-118.
- VisionMobile. (2012). *Cross-Platform Developer Tools*. Retrieved September 12, 2012 from http://VisionMobile_Cross-Platform_Developer_Tools_2012.
- Xamarin. (2012a). *Monotouch/Mono for Android*. Retrieved on June 29, 2012 from <http://xamarin.com/monotouch>
- Xamarin. (2012b). *Create native iOS, Android, Mac and Windows apps in C#*. Retrieved on June 29, 2012 from <http://xamarin.com/monoforandroid>
- Yin, R. K. (2003). *Case Study Research Design and Methods*. 3rd ed. Thousand Oaks, CA: Sage Publications.

Appendix A

Questions of interview 1 conducted on April 30 with Michael Walenius, regional managing director and member of the Board at Hogia Handelssystem AB. Walenius has previously worked as IT management and strategy consultant, project manager, and a former CIO at The Sahlgrenska academy at University of Gothenburg.

1. Do you have any department that deals with mobile web development?
2. What platforms do your applications target?
3. Do you usually have cases from clients where you should develop the same app targeting multi-platforms such as Android, iOS, and Windows phone?
4. Do clients care about the development phase or the tools used or just about the end results?
5. The preliminary question for most of developers today is whether to develop a native or a web application. However, we know that hybrid apps combine the best of both. Do you decide what format the app will have or upon the customers' request?
6. When developing a web app, do cross-platform tools support all the different platforms with no problems or are there still some differences?
7. Are these frameworks helpful for developing native apps as they are for web apps?
8. What are the challenges that arise from using cross-platform tools?
9. What are the benefits that these tools provide?
10. Coming back to the native applications, can you give us insights on the process approach when developing single native app targeting more than one platform?
11. I see that you can use both agile development (Scrum method) and waterfall model. On what do you base the choice of your approach?
12. Requirement specification, is that phase done universally for all the platforms or do you separate them between the departments or separated workshops for different platforms?
13. Is there any nominated framework that is going the Java Virtual Machine way? (that will cross the mobile platforms as JVM did with desktop applications)

The interview with the regional managing director at Hogia Handelssystem AB-*Michael Walenius*, was conducted via skype and recorded on a recording machine on April 30, 2012 and lasted for 45 minutes.

Interview with Michael Wallenius:

Q: Which company do you work for?

M: Today I am working with a company called Hoggia handelssystem, I would translate it to Hoggia retails systems. So they manufacture and market business systems for retail companies.

Q: So you have a department in your company that deals with mobile app development?

M: Yes, we both deal with mobile payment systems and also have a section of the company that is dealing with transport and logistics.

Q: Usually for which platforms do you develop apps for?

M: I refer now to my former position as a regional manager of the consultancy company, and in that position we mainly developed services and applications for Android and IOS (iPhone and iPad).

We studied windows phone but there weren't any customer projects interested in investing in that so we had it more as learning projects.

Q: But you have a good significant experience in the other platforms then.

M: Definitely, yes!

Q: Do u usually have the case where you should develop the same app for multi-platforms. I mean the same application but the client wants it for both Android and IOS and maybe windows phone as well?

M: Yeah, this is a problem that is hard for the customer to understand in the beginning. That if you want to do a complex mobile application then you need to do it in two separate projects, one for android and the other for IOS. Of course there are tools and platforms to develop code and applications that run on both platforms but they are difficult to use when you either need capacity or you need a quick response in the UI or if you need a specific hardware services to be used. The frameworks are getting better and better to have similar implementations but still they are on different code bases since they use Java and C variant. So you need to do it twice if you want to do it natively.

Q: Yeah, but maybe clients don't care about the development phase if you have separate code or what tool do you use. But my question is, are there cases that clients just want similar requirements or application but they want it for IOS and Android as well but they don't care how you do it or which tool you use.

M: Yeah, mostly they don't care at all which tools you use. They care about the end result. In that case, as a consultancy company you need to be aware of the possibilities that you have in the different options. Because of course you should use a framework that is possible to use on both platforms if you can. There is no reason to do two development projects in android and IOS if you can do the same in the JavaScript environment. Definitely not; that is something that the customer needs to understand that why you choose different tools or platforms. They have a hard time to understand why it sometimes cost them 200 000 crowns to develop a project when in another situation almost similar maybe cost them half the price.

Q: Talking about native applications; the preliminary question today for many developers is whether to develop a native or web application. However, we know today that there is something called hybrid apps which combine the best of both. Do you decide what format the application will have- native, web or hybrid; or you do it upon the customers' request?

M: If it is possible to develop a mobile web client, then definitely you should go for that. But there are different types of applications that are suited to be built in a web environment and there are others that are more suited to be developed in a native environment. So the most important thing that a developer needs to sit down with is the initial discussions with the client and getting both the client and themselves to understand what is actually that we need to do and what are the requirements. There is no need to build a native application if there is only an ordinary type of application where there is an answer and a response sending back and forth. But if you need to get in contact with cameras and GPS devices, then it is pretty quick gets complicated and slow if you are using like the PhoneGap development environment, JavaScript based etc.. But with the HTML5 and JavaScript and the CSS3 environment that is coming now, it is getting more and more possible to be done in a web environment.

Q: When developing a web app, the cross-platforms tools support all the different platforms with no problems. I mean do those cross-platform frameworks work perfectly when developing a web app for an IOS, Android and windows 7 phone or there are still some differences?

M: There are still some small differences and that is due to the HTML5 standard that is not really finalized yet and that shows in the code, as you need to put a specific code segments that are pointing for different web engines. For instance, if you are targeting the IOS environment then you need to target some HTML5 tags with Safari specific code and the same goes for Firefox and Chrome. They have some differences so for a while longer you need to keep them separate and know which browser you are using for your application at the moment.

Q: But still cross-platform app development (frameworks) somehow works for web apps. Isn't that so?

M: Yes, they are perfect. You can do fairly much with a web environment on a mobile phone; definitely. There is no discussion about.

Q: In contrast with native apps, the difference is still much bigger. Right?

M: Yeah, I think the difference is in the User experience. When the user starts to use the web application they get directed to a web browser and at first you see an address field that is actually going out to the internet, and then if the programmer had used the possibility of hiding the address field it disappears. But still you have a feeling that you are in a web browser. But the graphical user interface possibilities with the HTML5 are getting much richer; so it is getting closer to the native application.

Q: So now coming back to the native applications, if you have a case that you have to develop a single native app targeting more than one platform. Can you give us an insight or structure of the process approach?

M: I would say in general that in first you get a very rudimentary demand (requirement) specification from the client. Mostly they don't have a deep down drilled demand specification and that's the problem. If you as a developer move on with that very overview picture what the client needs, you may end out in a situation where you have given a prize on something that is not really what the customer wanted. The first and the most important step is to drill down the demand specification. There are 2 possibilities, the agile and the waterfall development. There (waterfall) you have standardized phases with demand specifications, design phases, development phase. If you investigate what a waterfall project is you can directly transfer that to a mobile project. But if you go in to the agile development, you can go directly move from the demand specification and involve the client with the development phase very early. And you can together with the client get much better possibility of finalizing the project to the requirements that the client wanted or if it has changed during the development then you have the possibility to adapt it to the change demands.

Q: I see that you can use both ways now. Do you base the choice of the development framework according to how big the project is?

M: Yes, but you need to investigate what the history of the client is. So if you come to a telecom or car manufacturing company. They are since many years back a customer to waterfall project and that is what they are using in large projects today. But parts of the larger waterfall projects may be agile as well so there is a mixture. But you need to investigate what the customer preference is. If you have a possibility of getting the client to understand or try out the agile development model, you definitely should try that since it is more productive.

Q: So Requirement specification, is that phase done universally for all the platforms or do you separate them between departments or separated workshop for different platforms?

M: they are both in the same requirement meeting or workshop, since a mobile application shouldn't be as it was for a couple of years ago (device specific). The needs of the customers' customer are probably universal so those requirements should be developed and deployed on any platform if they have the time and the money to do it. I would say producing the back log or the requirement specification documentation should be the same for all platforms.

Q: Do you think that there is any framework that is going JVM way, that will cross the platforms before all the others.

M: I think the only candidate for that is the HTML5 standardization but they still have a couple of years left. However, the intention is right and what we see in the market with the tools is also right. I compare where we are today with mobile tools and mobile development is the same like it was in 96, 97 with the web. They were trying to develop standards and being browser agnostic and everything but still there are differences between browsers. And I believe that will continue to be for a while but the HTML5 standardization process is going to close the changes for the HTML5 when 2 of the major browser engine producers have come along on one set of implementations. We will end up in a good way in a couple of years (on the web level). Natively, I see no hope of having different device manufacturers coming together in one platform language. That is a little bit of the game. They are trying to make something unique that will give them a market preference and apple has been successful in this. Looking at apple apps and comparing them to android apps, you see a complete difference in them and that is that most IOS applications are looking much more professional than many of the android are. And you also see a user difference in who pays for applications and where there will be the money to be made and where there is no

money.

Q: Google maps on both platforms?

M: Google bought the company Android and started to develop it as an iOS and making it public. What happens after that is that the Android OS has been handed over to an organization called the Open Handset Alliance. Open handset alliance is developing the Android OS today supported by Google and the others. Apple is not on that track at all, they are developing their own operative system environment. And yes the iOS frameworks for developing is a little bit more mature, a little bit more easy to use. The android on the other hand is a little bit more open. You have more capabilities, more freedom to do whatever you like. That is a little more complex, that freedom comes with complexity. Apple does a lot for you with their framework which Google doesn't. you have to do much more urself in the Android enivornments.

Q: There are quite a lot of device specific differences which create issues and problems for developers if they want to use cross platform tools to develop apps targeting multi-platforms?

M: Yes definitely.

Q: Can we come back to the development phase? In Agile development. How does it go ?

M: The agile environment is a little bit soft or easy going where you sit down with the customer and try to understand what he is trying to accomplish. And when you have understood what he wants, the developers create something they call the backlog, and from the backlog they pick the most important issues for the customers and start to implement those for one sprint. And one sprint can be 14 days or 1 month most regularly. And then you move in that kind of iteration process. When the first sprint is done you get back to the backlog and you strike out what you have accomplished and you bring again the prioritization and you get a list of what is most important at the moment (right now) and you do this together with the client. So the client is always involved in the development process, he can see the process, he can test everything that is coming out of the developing group which gets direct feedback that we are on the right track.

Q: How do you solve platform differences? Do you use any framework that allows you to reuse the codes or you totally separate them.

M: We separate them definitely. If you have a project that is supposed to do an iOS and an Android application and you have decided that you want to do that as native projects, you need either to have a developer that is capable of both writing objective-C code and java code. Or you need to have 2 developers doing the same application but separately.

Q: Have you ever thought about using cross-platform frameworks? Some tools like monotouch/ mono for android claim that they provide u with everything u want to build a native app targeting multi-platforms. How much truth is there?

M: We investigated different tools and platforms for cross-platform development. We tried out PhoneGap and a couple of other tools. We evaluated Mono in theory. We had a couple of developers that had used mono in hobby projects at home and they said that it wasn't

really that stable to be used in a production environment. So we didn't bother doing more on mono. We didn't know actually what it would create and the risks behind it.

Q: So still, you have to separate the projects? Our purpose is to find the most preferable approach for start-ups that desire to develop native apps targeting multi-platforms.

M: Yes, there are efforts being done on this. "But my conclusion after I have been doing that for several years is that there is no simple way in general. There might be a possibility of reusing code and using tools that allow cross platform development but that is decided upon the task that the project is about".

Differences and possibilities with different operative systems:

As a consultancy company, which is basically the main objective of all development companies, whether start-ups or large enterprises, they always need to see what the customers are really asking for. They need to try to understand if the customer is asking for something that he does not really know about. It often happens when discussing the requirements with the clients that you know what they want more than they do (Walenius, 2012). When the company has understood that picture, they can target the platform that suits the application the most. Besides that, they can decide if the application should be deployed in mobile web or in native Android. There are actually things that developers can do in Android but not in an Apple environment. For instance, in an iOS environment, applications cannot send a text message without the device's user confirmation, but it is possible to do that on an Android without even informing the owner. This is a security risk as well, but if the client wants to have that functionality, there is only one platform that allows the development of that function in the application. So, developers also need to understand the differences and the possibilities with the different operative systems

So if you need to have quick responses, you need to have access to cameras, GPSs and other hardware that is to be used in the application, then I would say no you don't have any other option than developing a native code. Since native codes gives the best user experience, you get the best interfaces, you get the most rapid response and much more flexibility. What you don't get is the low price and you don't get tons of developers being capable of writing C# codes. You need to have specific developers that are also aware of the limitations and the possibilities of the mobile platforms. So you need to have a specific developer that is interested to do coding for mobile platforms because it is very different.

Q: They have to have same skills?

M: Yes but for different platforms, it is completely different to build an Android environment than building for an iOS environment. In iOS u need to be more capable in memory planning, memory handling and emptying recycle bins, which is automatically done in java. But that is also a positive result because the iOS applications are generally more stable because you have a developer that is actually caring more on how he is writing his code and what happens.

The importance of having access to additional platforms. CPT selection criteria again:

Yeah. Its pretty simple also. There is one question you need to ask to the clients. Which are your customers? Which devices are there using. Can you afford to develop three different native applications? And with three I mean iOS, Android and windows phone. Those are

three major platforms that are going to live for the next ten years definitely. If they cannot afford that, ok then which platforms are the most interesting to you. By doing that kind of questioning, you narrow it down to what you are capable of doing. So in the end, you have got the answers to all the important demand questions. Then you can say, ok now we have these options. We can only do it natively on Android or you have another set up of questions ending up in ok we have all flexibilities: This is not very complex application, we can do it as a web application or we can do it as JavaScript, PhoneGap application development project. And if you come to that situation, it is mostly up to the customers saying ok this is price for a native application and this is the price of a PhoneGap application. And you need to compare the pros and cos with the different platforms and involve the customer in choosing the best platform with the consultants' help.

if the customer is looking for market share, the only option is to go for iOS and Android. And if they only can afford one platform, I would say iOS is the only platform if you are looking to make some money out of your application development, because the iOS users are more reluctant to buy applications than the Android users. iOS are pretty comfortable to pay through iTunes, they have an eco-system for buying and delivering and paying for applications that is much more used than an Android environment.

Q: What do you say about the future of windows phone 7?

M: I think that windows phone has definitely got a future, no doubt that they will do everything to come back and take market share. They will have around 5 years period before they are back in the game again. There are a lot of limitations in the operative system environments. The best thing with windows phone is user interface. It is nice, smooth, and good looking interface. That is really attractive, but the functionality and the user base is way behind Android and iOS. The amount of applications possible to download is way beyond the Android and iOS amount of applications. So Microsoft has a lot to do to come back to the game again, but together with Nokia, they have a pretty good standing point.

Appendix B

Questions of interview 2 conducted on May 1 with Muhammad Shoshah, current iOS developer at Infospread AB and former Informatics student at Jönköpings University holding a Masters degree.

Interviews.

Company: InfoSpread AB

Located: Jonkoping

Company type: small

Questioner:

Q

Interviewee – Muhammad Shoshah (iPhone App Developer): M

Q: What platforms do you develop apps for?

M: If you are asking about the company, iPhone, Android, Windows Phone 7 and J2ME. I develop for iPhone devices.

Q: Which platforms did the company target at the beginning of its existence?

M: Only J2ME.

Q: Do you have the cases where you need to create the same app for different devices?

M: Yes, usually we develop the same app for more than one platform.

Q: What is your approach in that case? Can you give us the picture about your work, the structure and how it goes?

M: We have Research and Development Management who is above us developers. The management collects the requirements and give us in written form. There are developers for each Mobile platform and the developers on each platform work independently. They only collaborate with the management. The project is conducted with a SCRUM method. We have a meeting every week where we developers show what we have done to the management and take the next task.

Q: So the requirements given to the developers are the same?

M: Yes.

Q: How does it work for different devices? Isn't there any problem in platform difference and the same requirements?

M: The requirements given to us are the high level requirements. When we receive them we work on them further according to our targeted device.

Q: Can you give an example?

M: Sure. Let's say we have a requirement which says the **user should be able to find their ways by map**. For me working on iPhone, the solution is right there. The iOS has the Google Maps feature implemented. The other developers must find their own solution.

This is a difference in finding a solution. That is why the difference is only in the use case in a lower level of abstraction.

Q: After you have specified the requirements further does the management check it?

M: Of course, we start writing codes only after the management's approval.

Q: You said you use SCRUM method of project management. That means, your application development is iterative and incremental.

M: Yes.

Q: On the way of your software development lifecycle, do the developers in different platforms share their solutions for certain problems or the state of their work?

M: Not really, we developers discuss it with the management.

Q: What tools do you use in coding?

M: For iPhone I use Eclipse, for android - Xcode and for Windows Phone 7 - Visual Studio.

Q: I bet Visual studio is the perfect option for Windows Phone 7.

M: No doubt, it is the best.

Q: How do you think this approach of multi-platform mobile app development is working for you?

M: We are happy with it so far.

Q: What product can you give us as an example of success for different platforms?

M: You can check MobiTime.

Q: Is that a native app?

M: Yes.

Q: Have you ever thought about Cross-platform tools to use and go in a different direction. I mean, have you ever thought about developing cross-platform apps for the sake of efficiency?

M: Yes, I personally suggested to use Titanium which enables Code-based approach?

Q: What was the result of your suggestion?

Appendix

M: The management rejected it after discussing. They didn't want to risk. One problem is that we have developers who use their own tool which they like and they are good at. It is very hard to force them to use different tool. Another problem is that it will be problematic to target so much heterogeneous platforms which have different form factors and different features embedded. For example, android itself has a various packages which are modifiable. Now device vendors are modifying android for their device's specific and unique features. Now let alone differences between different Operating Systems and devices. It is unlikely that cross-platform tool gives us more efficiency regarding these issues. After the discussion, now I do not advocate cross-platform approach.

Q: Thank you very much for your time and help!

M: You are welcome!

Appendix C

Q: Which company do you work for patrik?

P: Netzon

Q: Which platforms do you usually develop apps for the most?

P: In the beginning, it was more usual to develop apps for the iPhone but now people are moving more and more to the android. We had customers asking for the iOS first and then for the android but now most of the customers ask for the android first and then for the iOS.

Q: Are there cases where clients ask you to develop apps for both platforms or multi-platforms?

P: Yes, more and more. We have even got customers asking for tablets now.

Q: How do you take care of such cases?

P: If we make changes on the app it goes on the both platforms at the same time. First we do that for ourselves and save the codes on a special system that we have, since the iOS, java are not that similar as u know. We do it for the iPhone first, and then the customer wants it for the java. We made a system to cut some time for ourselves that allows the reusing of the code on both platforms.

Q: What frameworks/tools do you currently use when developing cross-platform apps? What challenges do you face when creating apps for iOS and android at the same time?

P: Well, mostly the navigation on the android is not that similar to the iPhone. You don't have the back button on both platforms. Many users say that the iPhone looks better design wise so you have to find your way around that. But at the same time you can't have an iPhone application on an android phone because the android users don't like that. They would look at it and say ooh that's an iPhone so you have to find your way to let it look like an android app. But as far as native function, we have faced problems with recording. When you record phone calls and stuff Apple has more control over you than android does. It happened that the platform refused to give the permission for the developer to add the recording function to his app in iOS "illegal move". On the other hand, it is as simple as a piece of cake on an android platform.

Q: You can pretty much take advantage of reusing the code between two platforms?

P: Yes.

Q: We would really want to get your answer about the tools that u use if u mentioned that u use cross-platform development platforms.

P: I think it's both yes and no. We have been sitting on the project for like a year and I don't think that anyone else has exactly what we have right now, that u have a system where u can administrate both platforms. But if the client needs a new native function that we don't have it saved in our system, then we have to build it separately for each platform and then integrate it in our admin so we can reuse it later on with other customers or projects. However, basic

functions for almost all the apps are saved in our admin and can pretty much be used in all the apps.

Q: From what you've said, it seems that cross-platform app development is nothing more than reusing the code on two different tools not like JVM or something like that. What benefits could you gain from that?

P: You save a lot of development time. Otherwise, you have to start from scratch. You know how you want the app to behave if you start with the iPhone but you still have to start from scratch for the android and vice versa. But now we have like code structure. Let's say that you can reuse 70% of the codes for basic functions or the templates and all that. It cuts a lot of development time for us plus it makes it much cheaper for the client.

Q: Do you have different developers for different platforms or is it the same developer or section for all the different platforms?

P: No, we have one developer for android and another one for iOS. And now we even have 3 web developers doing like web- based applications. So the apps we build today are more likely to be 50-50 between native and web applications. They look like native apps but they collect the information from different websites or servers.

Q: Patrik do u think that you reached what you maximum with the cross-platform app development or are you trying to reach somewhere?

P: No of course we haven't reached our maximum yet, we can easily get to another level. We have recently released our software and now we started to work with reseller on that and we are getting feedbacks from them all the time.

Q: Patrik can u give us an insight about the structure in mobile app development. How does the app lifecycle looks like. What methodology do u use?

P: Many of our clients don't have the basic knowledge on app development. They know what they want, but they don't really know if it is possible or not. And for us we need the information on how big the project is, what APIs it needs. What we usually do is that we sit down and have a study session with the client for like 4 to 5 hours taking notes. Then, I go back to the client with a flowchart. This is how I see ur project, is this correct?

From that we make another document called specification, that's like the second page of the flowchart. When they confirm the flowchart, then I start making specification for every point of the flowchart. And they have to accept that or say no that's a mistake. When all this is set, then we have a specification order project, and we can give the client the predicted time and cost for the project and that's written in stone pretty much.

Q: What about later on? Do u also get back to the client showing them prototypes or something similar?

P: Yeah sure, we try to help the clients with different things coming out and say well, this will work perfect for ur app, but we try to sit down with them and make some editing as well. But also as u said, we try to keep in contact with them and then help them out of course.

Q: What about windows phone 7? Do u take this platform into consideration, at least for learning?

P: Yes, we have it on the table already. We have a client that will most likely come back to us and ask for the same app but for windows phone for their users, but for the moment it is not a priority. Its very few projects since there are not a lot of them in Swedish OS market. But half a year from now, it will be more and more. We have some clients in London now asking about the project.

Q: What about HTML5 and CSS3?

P: In our platform, u can have just a mobile web app if u want. And its based on CSS and html5.

Q: When it comes to picking tools and frameworks, what do u rely on? Your background as a developer or what the actual tool offers?

P: Our project manager has a developer background. I have a design and UI background. He checks it from a developer point of view, while I look why should it be like this for the end-user if its not logical. But we really try to help the clients with that, u might be able to build a very advanced project, but ur clients don't want it probably. So we usually sit down, me and the project manager and check both sides there.

We have built a system that generates that code. If we are building a new app, then we build it either in eclipse or in x-code and in the system at the same time. So later on, the system re-generates the codes so u can reuse it. We have a lot of codes already done in the system now.

Q: Do u think that there will be a tool that will cross the platforms before all the others?

P: Not really, that's why we built our system. Cz as u guys know, apple are not that friendly with other languages. We wished it for a year ago, now I don't because we have our own. But I don't really think that apple will allow that; however, it will be great for everyone, us developers, the clients and the end-users. Html5 is still going the right way even the 100% cross platform will not happen.

Flash is almost dying now. Those 2 giants right now, it might be windows phone later on. But they gonna keep on development on their own and they are not going to get close to each other rather than split even more and more. Apple is closed, it is closed but unfortunately I think it will even get more closer in the future.

Q: Do u have big projects or just small projects?

P: We have a huge variety of clients, and we have very big clients that we have been sitting with for a year now. A project might vary from 2 weeks to a year. It might cost from 20 000 to 500 000 crowns. It's a big variation, we estimate our time to complete the project and then we give the price.

Q: Do u use agile or waterfall development models?

P: We use waterfall for big projects and agile for small ones.

Q: How do you cut ur costs?

Appendix

P: Like now, we can set up an app with standard basic functions in a few a hours on both platforms and then comes the new development on top of that. But we had offers from different companies and there were a lot of companies competing with us but their price was 50 to 70% more than ours. I was thinking why Is that? It should take us the same time as them, but we have our platform that gives us the kick start. So other companies might need 200 hours for each platform while we need 200 hours for both platforms probably.

Appendix D

Jack Nutting

Q: do you have any cases where you have to develop the same app for multi-platforms or at least for two?

J: Not in toca boca where he works now, but he did previously. They develop apps (games) for iOS. However, he has previously worked with cross-platform development where they developed web applications for iphone, ipad, and android.

Q: what was your approach in those cases? How did you solve the multi-platforms issues in your previous experience? Did you separate the projects or same project?

J: we tried to put as much as possible on the server, since they were using the same server/backend. All users must have internet connection in order to get access to those apps.

Q: How did you solve the developments issues?

J: we started with the iphone apps first and then 2 or 3 month later we start with the android, after getting an idea on how the application will look like. So it was a bit of an iphone app on an android phone if we say so.

Q: Ongoing from your experience with mobile apps development, How do you see the future of the cross-platform development toolkits for native apps?

J: I think the best way to judge this is to look at the computing history and what happened before. In every computer platform, there were never any real cross-platform tool that could run the same app on a windows computer and a mac computer. You always had to set a special hardware configuration. I think it is in the interest of any company to create something that is unique to it or to its platform. What apple does is that they provide you the iOS and its own tool with it not an unknown device. You know that you have this machine to run your apps on. Whereas android, you can more or less create your own machine and run your apps on it.

Q: Apparently, it is not technology issues being the obstacle or different company policies?

J: Yes, I think so. Historically, the only situation I can remember when you could run the same software on two different devices was in the late 80s or early 90s. It was the windows OS and the IBM OS2 and the reason for that is that both companies had some kind of license restrictions since they developed certain softwares together so they were kind of stuck with each other for a while until Microsoft started developing windows in new directions that IBM couldn't copy. So technology wise its possible to different softwares to share same technology layer development in top of them but they are not really interested in doing that.

Q: coming back to your company projects, how big are your projects? I mean how much time do you spend coding?

J: usually most of our applications take from 3 to 4 months from start finish for a complete project to be done. We have a team of about 3 people, project manager, game designer and a programmer. Usually programming for one game is 2 to 3 months. But we spend more time on graphics than programming to make it look good for the kids.

Q: Do you use the SCRUM method or waterfall when start with your projects? Can you give us an insight on the entire process from the idea to delivering the product?

J: we use scrum or something really close to it. We have sprints. Each sprint takes about a week or two and then we sit back again before next sprint and check how far have we came and what need to be done and so go on.

Q: what about Microsoft being part in the phone software battle?

J: Microsoft is a huge company and they have the money and the patience. They have being working on that project for more than ten years, so I don't think that we will just leave it apart. At some point, this platform will be worth it, either because the users are so Microsoft centric or simply because they like the phone or its user interface.

Q: if someday we will have a perfect cross-platform toolkit for mobile development, can you summarize the benefits and challenges that would arise from this technology evolution?

J: Apparently, we would spend a lot of development time. Another benefit is that you have a lot of developers that are familiar with html5 or javascript, and those people will have a much easier step into mobile development. The challenges are around hardware support and OS support.

Q: Hybrid apps can be developed using toolkits that already exist and perform perfectly on different devices. Why hybrid apps aren't the best solution rather than native apps, knowing that they combine the best of both native and web apps?

J: There are a lot of mobile features that you cannot access using an html5 tag. There is a lot of discipline error as well. Like the facebook application, it is a hybrid app. But if you are on the app and you get a sudden connection failure, it will get the html scripts but it won't get the CSS scripts. If you want to be safer and looking for great performance, like a game or an application that plays music at real time you have to be closer to the hardware level.

Q: Will developers be able to create native apps targeting multi-platforms using html5 or CSS3 or even javaScript one day or it will always be web development with these tools?

J: html5 is a good candidate. You can already today create widgets on the iOS using html5 tags. But I mean again it will be apple doing it their way.

Q: thank you Jack Nutting for your time.

Appendix E

Chris Forsberg

C: The general approach in building apps is not that different from building any other IT systems. The usual phases that we have, we have a simple model where we have a definition phase, where we collect prerequisites, design guidelines that the client has decide on (branding guidelines or any strategy that they have both IT & Market strategy). We define also the scope of the app and also what u r after, we define what platforms the app will support. In Sweden, we mostly build for iOS and android because it is like 96% of the market(two third is iOS and one third is android). The only platform that will probably change this trend in the near future is probably windows phone 8 which is coming at the end of this year and the device will probably come around Christmas time. Microsoft had some really hard time and also Nokia, but they are not that far away since the platform they are offering is really a solid platform. At the moment it is basically the 2 platforms. This does not make the choice easy anyway because if we look at iOS we more and more have a demand to support both iphones and ipads. So we need to have a solution in order to support both form factors which is on the iOS called universal app. It is harder on the Android since there are 300 to 400 different devices so what we need to do is selecting a number of these devices that we can guaranty that it will work on, and also the orientation that supports the landscape or only one of them. Now with Android four, we need to definitely support tablets as well. This is what is done in the define phase.

In the next phase (the design phase), we set up the real contents of the app (how we want it to look) and also go through graphical stuff to see exactly how we want it to look, in addition to deciding if we need to connect it to a server besides defining all needed interfaces . At the end of the design phase, we have a clear picture about the way it should work and the way it should look.

Moving to the development phase, different teams working on different platforms do so in a parallel way. Recently, we have discovered the differences between iphone and Android with the new design guidelines from Google 4.0 we need to make specific designs for Android. Thus, we are moving to a direction where the differences are bigger than before. We need to make the apps more iOS specific and more Android specific to be correct. The iOS developers or user wants an iOS experience in the app. For example, if I put a back button that looks like an iphone button in an Android app, the Android customer would then remove the app from the device because he/she is used to something that is Android and not iOS and they want it to look android. Concerning the cross-platform, what we see today is that there is huge desire from apps customers and more specifically the IT departments to move to a cross-platforms development strategy. Yet, the problem is that to make something that looks right on each platform is as hard as working with a specific tool for each platform. To conclude, I have been looking at several tools for a couple of years after clients' request. Just to mention few, IBM has recently bought an Israel company called 'Worklight' and they are purely focusing on transforming IBM by starting to work on Worklight cross-platform tool which is a nice tool but has an entry of 30,000 Dollars which is good for large app projects costly for small app projects. To name another tool, Appcelerator from Titanium is a common one based on Java script that has a lot of built apps. Our problem concerning these tools as a consultant company (SOGETI) is that we work on many projects with different platforms for different clients and standards. So we try to make our consultants learn the core skills such as Objective C for iphone and Java for Android. Html5, CSS3, Java Script are also core skills. The problem with Appcelerator is that you create developers that only

know Appcelerator and they will neither be able to write a native app for iOS using X-code nor a native app for Android using Java because they only recognize the Java Script supported by Appcelerator . Working on mobile app cross-platform development is making us move towards the web technology. The most common project used is JQueryMobile which is an extension of the jquery project which is an open source project.

Q: How can developers develop native applications using JavaScript on Appcelerator?

C: They have a pre-compiler, so what happens is that the developer writes the code in java script and then the pre-compiler transforms all this JavaScript code into native objective-C for iOS and it actually transform it not directly to java I mean for android but it creates a jar file, so there is actually no translation into source code in the java case but for iOS they didn't have a choice because they need to build the executable with x-code so they actually generate a complete x-code project from the java script. However, the problem with java script is that if you have an experienced developer using it you will probably be doing fine but if you have a beginner using it you will probably have problems because the language is very forgiving so he can make whatever he wants there more or less. So if you don't know what you are doing you can create a mess.

Q: Why don't you use appcelerator for mobile app development in your company if java script is valid and you have the skills?

C: As a consultant company, we work with large customers and a lot of them work with IBM. Others are Microsoft centric or even linux. We have to have the core skills in our company. We cannot afford 10 Appcelerator developers just because one client wants one project. Java script is valid but using it in a tool like appcelerator is a problem because we create appcelerator developers and we prefer much more to use a core html5 simple library like JQueryMobile where you actually code in java script and you can even package it as a native app where you put the scripts in a shell like PhoneGap. Phone gap is available now for iOS, android, blackberry and windows phone. You just put the web app inside that shell and it works like a normal native app that you can upload to the app store or android market or wherever you desire.

Q: Do developers working on different platforms interact with each other during the development process or they work separately from each other in two separate projects?

C: Different developers from different departments interact together during the design phase and development while working on various projects. The reality is that we work too close to the client. They negotiate between each other but not on a deep technical level. We will solve it like this on the iOS, how will you do it on the android. So there is a lot of communication between the teams through the whole process. The hard choice is to do something that feels consistent across the platforms but also that is true for each platform. The important person here is the solution architect, he is the one that take all these decisions but he is always talking to the developers.

What is Worklights' problem besides that it has an entry charge of \$30 000? Does it allow the development of cross-platform apps?

C: I would say that it is good in comparison to appcelerator because it doesn't introduce you to a new development language or API or something. For example, you can take a project built with JQueryMobile which is a pure html5 library and you can put it inside worklight.

What worklight brings which is nice is that it does the things that we have to do manually like the hands on activities or adapting the app to a specific platform when we use JQuery. In woklight they take the whole responsibility for the whole process. For example, you have a structure of files and you have a common library with html5 or java script sources and then you want something different for another platform, you make copy of that file and you put it in a library for that platform. It means most of the code will be common but only the things that differ will be in its library for a specific platform or folder. They have a complete solution for the whole system not only the apps. Their solution considers distribution, push notifications, users feedback and even update app.

Q: What about monotouch/monodroid? They claim that you can code once and then apply it everywhere. Is it true?

C: Personally, I am a Microsoft person. I mean I have a Microsoft background. The .Net library is a huge library and you have millions of expert .Net developers outside. Java and the other languages are not even close to it. But the problem is that there is no huge company behind mono (Microsoft is supporting windows phone). It was created by Novell but now it is bought and supported by Xamarin which is not really huge company yet. The idea of mono is great. However another problem is that they are charging a fee of almost 500\$ which should be much cheaper at least for the Indian developers (it should even be for free for Indian developers, I think this will help).

Q: People complain about cross-platform apps that they lower the performance of the device that is already not that great compared to desktop applications?

C: In all these cross-platform solutions you should have something that makes it common or looks the same, but the problem with it is that we do not want them to look the same (The need for something that makes it common, and at the same time avoid making different platforms look the same). Again android developers and users expect something looking android not iPhone to them and vice versa. That is the biggest problem for the cross-platform tools, you still need to adapt to the platform you are targeting. For example, we are now working on developing an application for a Swedish company and they are developing an app with jQuerrymobile in PhoneGap. We still need to spend a lot of time and energy in theming the application. I mean to make it look like android on android and like iPhone on iOS. That is almost as hard as separating the two projects and developing them natively. But the gain that you get from developing cross-platform apps is probably not so much in the first app. If you do it right, then you have to maintain a single code base because the maintenance normally is almost 80% the total cost of the app and the initial development is 20%. So you back the money in the update and maintenance of the app not in the implementation. (Interesting answer)

Q: Anatomy of a cross-platform app life cycle according to SOGETI?

C: 1. Define 2. Design 3. Development (implementation occurs here) 4. Deployment (roll out or put the app as part of a system) 5. Maintenance (maintain the app and keep it updated. In case of big clients, maintenance might include developing new apps and the cycle starts over).

The benefits from cross-platforms start when an app is developed in order to be used for different events. This happens as long as the app is well maintained since the development

of the app makes up 20% of the app cost while maintenance makes up 80% OF THE APP COST.

The fourth phase is deployment where the app is put in the system.

The last phase is maintenance where updates of new versions or new features are maintained. Also, in some cases, new apps are developed within the maintenance phase which makes the cycle starts over again.

Our advantage is that we have developers for different platforms off-shore consisting a team in Bombay and a team in Bangalore

Is javaScript rich enough to create native apps?

The only problem with javaScript is that it very forgivable language, you can pretty much write what u want.

Appendix F

Interview with NY (questions sent and received by email)

1. What platforms do you develop apps for?

Android and iOS

2. Do you have the cases where you need to create the same app for different devices?

Yes. In all our cases our customers have only wanted an iOS app to start with, then when the iOS app has been on the app store for a while the customer has returned and wanted an Android-app as well. Many of these cases have been pilot-projects for testing a third party platform to develop apps targeting both Android and iOS. In these cases we have compared Titanium, Phonegap and Corona, which are all application frameworks for cross-platform development.

2.1 If yes, what is your approach in this case?

It depends on the project. First we look at what the app must be able to do and then we decide how we should develop the app. It comes down to how advanced the app will be.

2.2 Do you use a specific tool?

In all of the cases we have used Titanium.

2.3 What issues do you face when developing multi-platform apps?

Reusing codes. The whole idea behind one code base for multiple platforms is that you shouldn't write too much target specific code but instead reuse as much code as you can for your both platforms. But this isn't always easy because one of the targets will always have something unique that the other one doesn't have. Like Android's physical menu button that doesn't exist on iPhone/iPad.

And that it's going good in the beginning of your multi-platform project when you haven't implemented much graphics or nested views etc, but the longer you get on your project you start getting problems with stability and memory management since you don't have any control of the native code that's being produced.

2.3 How do those tools help you tackle multi-platform issues?

It helps you develop faster. Native developing will always be the better choice if you have unlimited time and budget. But if you can use tools like Titanium and target multiple devices within the same code base then obviously that the development will go much faster.

2.4 Can you give examples with your products?

For the moment we only have published our cross-platforms apps on Google play, and that is for a simple reason. When we have made an Android-version of an iOS-app using a third party framework and target both platforms, the iOS-app has never

been as good as the Native developed one. We have spent many hours trying to optimize our apps to get the same feeling and stability as the native developed app for iOS but without succeeding (not the same with Android). But some of the multi-platform apps that you can download from Google Play are the Swedish apps: Monark, Crescent and Holmgrens bil.

3. What factors issues do you consider when choosing a tool for mobile app development? What do you focus on the most?

How advanced the app will be. For more advanced app that has lot of nested views, is built on a lot of graphics or need to do some heavy calculating like image/video processing then native is the only way to go.

For lighter and less advanced apps that take advantages on the core functions of each targeted platform and don't have to do any memory consuming tasks then you should take a look at third party platform like Titanium.