Third-Party Development Practices for Mobile Platforms

Mohammad Shoshah

MASTER THESIS 2012
INFOMATICS
Third-Party Development Practices for Mobile Platforms

Mohammad Shoshah

This master's dissertation is done at the School of Engineering, Jönköping University in the subject area informatics. The work is part of a two-year master's program focused in information engineering and management. The authors themselves take full responsibility for the stated opinions, conclusions and results.

Supervisor: Ahamad Ghazawneh
Examiner: Vladimir Tarasov

Credit points: 30 hp (D-Level)

Date: June 2012
Abstract

Information Technology (IT) evolution in the recent decades has been able to grow the mobility of the end-user. A contributing factor has impacted and improved the smartphone area. This, in turn, has changed end-users’ expectations and experiences in the recent years in terms of available services, which have made mobile applications (apps) to find their place in daily life.

In line with the significant growth of smartphones, as the iPhone and Android phones, is also the development of mobile applications exploded. Mobile use has - with this - fundamentally changed and much of what customers used to do on the computer, they do now on their mobile. It makes of course requires that third-parties have a mobile presence and correct utilization, this new arena enormous business opportunities. Hence, as mobile devices become a more visible business tool, it becomes important for organizations to develop applications to meet customer needs and stay competitive.

Third-party development is relevantly in the core of software development nowadays. The introduction of smartphones and tablet devices, the associated products, and services in the form of apps has significantly changed software developments practices for developers. The aim of this research formulated to study the development process and practices associated with third-parties and their effects on dealing with various mobile platforms and boundary resources (SKDs and APIs).

Qualitative study and exploratory design are extensively conducted for data collection and analysis. Primary data is gathered based on experiences, opinions, and insights from the experts in the subject, and secondary data is gathered from literature, which constantly supports the primary data and ensures the research credibility.

The main contribution and results of this study are the factors and development process that third-party developers take into consideration when developing apps for iOS and Android such as ecosystem, programing language, platform, programing model, tools, development process model, supply chain, and submission apps.

The success of the digital ecosystems and platform knowledge adopted by Apple and Google has gained a lot of attention from developers in the last years concerning mobile application development. This, in turn, was the key factor of their success at the smartphones market and the portal for developers from all the world.
Acknowledgements

This project has been an amazing journey from start to finish. I had probably not been able to if it were not for all the support and encouragement I have received.

I would first like to thank my supervisor Ahmad Ghazawneh that has made this thesis possible.

I would also like to thank my teacher Vladimir Tarasov who supported me and always been there if necessary.
Key words

Platforms, Software Platforms, Third-party development, SDK, API, Open Innovation, iOS, Android
Contents

1 Introduction ........................................................................................................... 1
  1.1 BACKGROUND ................................................................................................. 1
  1.2 PURPOSE/OBJECTIVES ...................................................................................... 2
  1.3 LIMITATIONS .................................................................................................... 3
  1.4 THESIS OUTLINE .............................................................................................. 3

2 “Theoretical Background” ...................................................................................... 4
  2.1 OPEN INNOVATION ............................................................................................ 4
  2.2 PLATFORMS ....................................................................................................... 7
    2.2.1 Platform Structure ....................................................................................... 9
    2.2.2 Platform Benefits ....................................................................................... 9
  2.3 SOFTWARE PLATFORMS ................................................................................... 10
    2.3.1 iOS Platform ............................................................................................... 13
      2.3.1.1 Xcode ................................................................................................. 13
    2.3.2 Android Platform ....................................................................................... 13
  2.4 SCRUM ............................................................................................................... 14

3 Methods ................................................................................................................ 17
  3.1 RESEARCH FRAMEWORK ............................................................................... 17
  3.2 RESEARCH DESIGN ......................................................................................... 18
  3.3 RESEARCH APPROACH ................................................................................... 19
    3.3.1 Single Case Study ....................................................................................... 20
      3.3.1.1 Infospread Euro .................................................................................. 20
  3.4 DATA COLLECTION .......................................................................................... 21
    3.4.1 Primary Data .............................................................................................. 21
      3.4.1.1 Interviews .......................................................................................... 22
  1. ABOUT THE COMPANY: ................................................................................... 23
  2. THE DEVELOPMENT PROCESS: ................................................................. 23
    3.4.2 Secondary Data ......................................................................................... 24
  3.5 RESEARCH ANALYSIS .................................................................................... 24
  3.6 RESEARCH CREDIBILITY ............................................................................... 25
    3.6.1 Reliability .................................................................................................... 25
    3.6.2 Validity ........................................................................................................ 25
    3.6.3 Generalizability ......................................................................................... 26

4 Results ................................................................................................................ 27
  4.1 ECOSYSTEM .................................................................................................... 27
    4.1.1 Apple .......................................................................................................... 28
    4.1.2 Google ........................................................................................................ 28
    4.1.3 App Store ................................................................................................... 29
    4.1.4 Google Play ................................................................................................. 29
    4.1.5 Developers ................................................................................................. 29
    4.1.6 Consumers ................................................................................................. 31
  4.2 PROGRAMING LANGUAGE ............................................................................ 31
  4.3 PLATFORM ....................................................................................................... 33
  4.4 PROGRAMING MODEL .................................................................................... 34
  4.5 TESTING AND CONTINUOUS INTEGRATION ........................................... 34
  4.6 RESOURCES .................................................................................................... 35
    4.6.1 API (Application Programing Interface) ..................................................... 35
    4.6.2 SDK (Software Development Kit) .............................................................. 36
  4.7 TOOLS ............................................................................................................ 36
    4.7.1 IDE (Integrated Development Environment) ............................................ 36
    4.7.2 UI Builder ................................................................................................. 37
    4.7.3 Debugger ................................................................................................. 37
4.7.4 Profiler and Heap Analysis ................................................................. 38
4.8 DEVELOPMENT PROCESS MODEL ....................................................... 39
4.9 SUPPLY CHAIN .................................................................................... 39
4.10 SUBMISSION APPS ........................................................................... 40
4.10.1 Submission Apps on Android Platform ........................................... 40
  4.10.1.1 Preparation the App for Release ................................................. 40
  4.10.1.2 Releasing the App to End-User ................................................... 41
4.10.2 Submission Apps to App Store ...................................................... 42
  4.10.2.1 Preparing for App Submission ................................................... 42
  4.10.2.2 App Store Approval Process .................................................... 44
5 Discussion .............................................................................................. 46
  5.1 THIRD-PARTY DEVELOPERS ............................................................. 46
  5.2 APP DEVELOPMENT PROCESS .......................................................... 47
6 Conclusion ............................................................................................... 49
  6.1 SCOPE FOR FUTURE STUDIES ........................................................ 50
7 References ............................................................................................... 51
8 Appendix .................................................................................................. 55
  8.1 INTERVIEWS QUESTIONS ................................................................... 55
  8.2 ABOUT THE COMPANY ...................................................................... 55
  8.3 THE DEVELOPMENT PROCESS .......................................................... 55
List of Figures

**FIGURE 1:** CLOSED INNOVATION MODEL (HENRY W. CHESBROUGH) .......................................................... 5

**FIGURE 2:** OPEN INNOVATION MODEL (HENRY W. CHESBROUGH) .... 6

**FIGURE 3:** PLATFORM-CENTRIC ECOSYSTEMS (TIWANA ET AL., 2010) 12

**FIGURE 4:** RESEARCH FRAMEWORK ................................................................. 17

**FIGURE 5:** APPLE & GOOGLE ECOSYSTEMS .......................................................... 27

**FIGURE 6:** CLIENT.H ................................................................................................. 32

**FIGURE 7:** CLIENT.M ................................................................................................. 32

**FIGURE 8:** CONTENT ASSIST IN ECLIPSE IDE ....................................................... 37

**FIGURE 9:** XCODE MEMORY DETECTION LEAK .................................................... 38

**FIGURE 10:** SUBMISSION PROCESS ON ANDROID PLATFORM ......................... 42

**FIGURE 11:** APP STORE APPROVAL PROCESS .................................................... 45

**FIGURE 12:** APP DEVELOPMENT PROCESS .......................................................... 48
List of Abbreviations

ADT: Android Developer Tools
API: Application Programming Interface
Apps: Applications
DRY: Don’t repeat yourself
IDE: Integrated Development Environment
IO: Input/Output
JDK: Java Development Kit
JDT: Java Development Tools
MVC: Model View Controller
NDA: Non-Disclosure Agreement
OS: Operating System
PC: Personal Computer
R&D: Research and Development
RT: Run Time
SAX: Simple API for XML
SDK: Software Development Kit
S-W: Software
UI: User Interface
XML: eXtensible Markup Language
1 Introduction

This chapter provides background information about the research topic and outlines the scope for this research. It also provides the research questions, purpose, limitations, and thesis outline.

1.1 Background

Nowadays, it is of no secret that technology is a critical element in socio-economic development, and its importance will grow over the years to come. Early technology has benefited only the corporate word and a few rich and/or highly educated categories of people due to the cost and user unfriendliness.

Today technology has gone up a notch and is penetrating society at the speed of light. Innovative development firms are constantly and continuously pouring in countless energy and money in order to improve their existing products and services and generate new ones. The competition in this sector is so intense that a once mega billion companies can become irrelevant instantly, a prime example of this is Netscape which was instant overshadowed by Microsoft’s Internet Explorer at once, or the once mighty AOL.

Tech firms spend about 10 to 15% of revenues in Research and Development; this is due to the rapidly evolving nature of the industry. At the beginning, innovation was internally driven by R&D team but with the advent of open source innovation it is being partly externalized (Chesbrough, Vanhaverbeke, and West 2006). Firms have stated to adopt a new innovation paradigm “Open innovation” (Chesbrough, 2003). Gawer and Cusumano (2002) state that the external sources of innovation are making evolving systems more interdependent.

The sector that is currently leading the Open Innovation boom is the mobile sector and is also one of the sectors where Open Innovation is being used to its full potential. Mobile development firms have fully used innovation paradigm to facilitate the development of mobile applications and services for platforms. The platforms are seen as a set of interrelated specification layers that support interoperability between the technological modules of a system (Baldwin and Woodard 2009). Leading the charge was Apple with the release of its own platform, which immediately became successful. The force of Apple’s platform resides in the company providing a homogeneous platform for developers to create products for its mobile devices. Due to the success and popularity others have followed in Apple’s footsteps in the likes of Google with Android and Microsoft with Windows Phone 7.

The platforms provide a set of common resources for the development of derivative products and services (Robertson and Ulrich 1998). Developers who create software for these platforms are called third-party developers, and they
earn money from the use of their software from end-users/consumers. With every major tech company opening up its own platform the open innovation word was at first fragmented due to the fact of compatibility issues. Thus only the most lucrative platforms drew the most third-party developers. This is due to the fact that developers had to rewrite their software partially or completely in order for them to run on a platform other than the original platform that it was intended for. Therefore, for new upcoming platforms to have a chance in the world, they had to make their platforms as open as possible and require the less changes as possible for existing software to operate on it.

Among software platform owners, it has become increasingly common to nurture communities of third-party developers to build complementary assets to increase the value of the platform. Such assets (e.g., applications and services) promises to extend the software platform’s functionality, offer value to the platform’s users (Huang et al., 2009), and address heterogeneous end-user needs (Adomavicius et al. 2007). Harnessing the collective power of third-party developers is enabled by providing specific platform boundary resources: 1) Technical boundary resources like APIs (Application Programming Interfaces) and SDKs (Software Development Kits) and 2) social boundary resources like incentives, electronic intellectual property rights and agreements, are key strategic resources (Ghazawneh and Henfridsson 2010), and granting access to complementary development (West 2003).

Software development in general is a well-explored area, and much effort has been devoted to develop general and useful principles to improve the development processes. Since 2007 and third-party developers are experiencing new development tools and processes that they have never before. The development of traditional software for desktop and web is slightly different from the mobile applications development. The development for mobile platforms adds new layers and processes that need to be investigated to have a clear image on how this new phenomena is evolving and to understand how third-party developers adopt into it.

1.2 Purpose/Objectives

The development practices for third-party developers in such platforms are immature, and differ when developing for various platforms. Apple’s iOS has its own unique development platform (closed platform), Google’s Android is another platform that is open source, and Microsoft Windows Phone is just one more type.

The purpose of this research is mainly to identify and get insight into the principal factors that third-party developers take into consideration when developing apps for Apple’s iOS and/or Google’s Android platforms.
The research on this area is extremely limited and started to grow up, thus the work at hand will enable us to answer and explore:

- What is the development process by which third-party developers adopt for the development of application in mobile platforms?

The answer to the research question should lead to provide guidance and highlight areas that are important to consider an app development process.

1.3 Limitations

The study will be limited in two main mobile platforms. Apple’s iOS and Google’s Android platforms are highly growing and leading the market. These two platforms are adopting two different approaches for third-party development. While iOS platform is considered as closed and wall-gardened, Android platform is open and widely adopted by many smartphones and tablets manufactures.

In so doing, the research will be able to examine the reteach in question in two different environments and gain more thorough understanding about the studied subject.

1.4 Thesis outline

The rest parts of this research are structured as follows:

The second chapter focuses on the literature review of the research conducted in mobile platforms.

The third chapter focuses on the research methods used for this study. It provides the theoretical framework, the research study and research design used in this study. Moreover, it outlines the different tools and methods used for data gathering and analysis, and describes the research credibility.

The fourth chapter focuses on the results obtained from the primary and secondary study. Subsequently, this chapter consolidates the objectives of the research.

The last chapter of this research affirms the conclusions and drawing inferences from the results of the research.
2 “Theoretical Background”

This chapter outlines the elaboration and the response to scientific abstracts and definitions within the next chapters. At first it begins with historical background about open innovation and how it emerged. Secondly, the platforms definition is presented and explained. In addition, platform structure and platform benefits, software platforms, iOS platform, and Android platform are identified and discussed as well. Finally, the convergence of open innovation concepts and the main concepts involving mobile platforms are the base to understand the whole picture of research project and can pave the way to answer the research question.

2.1 Open Innovation

Since the industrial revolution innovation has been a driving force in the world we live in today. It has greatly contributed to substantially increase our economic and social prosperity in every way. During the 20th century the speed in which innovation occurred has been gaining momentum and hasn’t slowed down since, but with the turn of the 21st century we see that the way innovation itself appears is transforming.

Early on innovation was the fruit of intense R&D by big powerhouses of their sector (Chandler, 1990). These powerhouses had the financial means to take on intense R&D in order to improve or expand their businesses. New products or better performing products resulted in a dominant position in those markets. Internal R&D was a strategic asset for companies (Arora and Gambardella, 2001). This type of innovation is called closed innovation (Chesbrough, 2003), in a closed innovation model, firms generated their own ideas, develop, manufacture, market, distribute and service it themselves. The model is self-reliance one.

For decades and decades this model was dubbed “the right way” thus leading companies injecting as much as 10 % to 15% of profits into R&D and even more for technological and pharmaceutical firms. Because of such investments, these companies were able to: 1) hire the best and brightest minds, 2) discover a significant number of ideas, 3) get to the market first, and thus leading them to reap most of the profits (Arundel, 2001.). They would then reinvest a substantial portion of the profits in R&D, which in term would lead to future breakthroughs, producing a virtuous circle of innovation.

The illustration in “Figure 1” provides a visual explanation of the closed innovation model.
Theoretical Background

Figure 1: Closed Innovation Model (Henry W. Chesbrough)

This model worked till the end of the 20th century, where a shift was seen, innovation started taking place outside big R&D labs. Two main reasons are put forward in order to explain this phenomenon: 1) the increase in numbers and in mobility of high-end knowledge workers and 2) the appearance of venture capitalist (Lerner, Joshua & Tirole, 2002). To illustrate this switch let us take example on Xerox, a company which set up on of the most advanced R&D labs. Xerox had made its reputation on office machines such as printers and copiers. The company became so notorious that “Xerox” became a word.

Xerox being a colossus in the fortune 500 built in 1970 the Palo Alto Research Center (PARC). PARC would turn out to be an innovation factory for Xerox; most if not to say all of the new high-end technologies seen in future products were developed in PARC. The catch with PARC is that it has ripped more benefits to external companies than for Xerox. How is this possible? (Chesbrough, 2003) give us the causes of such a paradigm. Based on a careful analysis of PARC from its structure to its management, he concludes that none of these where mismanaged. He goes on to explain that most project undertaken have beard fruit for Xerox. The firm decided to abandon projects due to the fact that their further development would be expensive and takes money away from more important ones to Xerox. This was true; the money was instead put in another project, which beard more fruits for the firm. Some researchers of axed projects would decide with the consent of Xerox and pursue one their own, as startups or spin-offs. Xerox would in return get a stake in the company as a licensing agreement (Chesbrough, 2002).

Merriam Webster defines Xerox as the action to copy on a xerographic copier.
Theoretical Background

The startups would further develop the products and go from startup to a serious business in the likes of Adobe, 3Com, Documentum and SynOptics, others where used by other startup whom today are among fortune 500 companies. PARC had created the Use-Interface Design concept, which was later picked up by Apple to create the first personal computer and later used by Microsoft in its windows operating system.

The case of Xerox marks the first cases of open innovation. Where innovation starts to take place outside vast R&D labs, this phenomenon would be amplified by venture capitalist (Gruber & Henkel, 2005) a venture capitalist in general is a social entity that has money to invest in young startups, it provides the money to people with ideas so they can develop them and in return are offered partnership. This has been the driving force behind open innovation. These new startups would develop their ideas till they are marketable and in most cases be bought by established companies whom see their innovations as essential to their development.

Nowadays we tend to see more innovation of this king than through R&D labs of big companies.

The illustration in “Figure 2” provides a visual explanation of the open innovation process.

![Open Innovation Model (Henry W. Chesbrough)](image-url)

**Figure 2:** Open Innovation Model (Henry W. Chesbrough)
2.2 Platforms

Today it is of no secret to anyone that we live in a technologically driven world. In this new era a new structure is appearing, well at least to the mainstream knowledge. This new structure in the emergence of platforms, whether industry, sector, or even product based, seem to be the new direction we are on. So what is a platform?

The Oxford English dictionary defines it as “a raised level surface on which people or things stand, usually a discrete structure intended for a particular activity or operation” and goes on to explain that “a platform refers to a concept, an idea that serves as pattern of model. Unlike most word used in the tech world, which has no or little relation with the original use of the word, this here pretty much sums up the techy definition. Although the term is used in more complex ways, notably referring to software (which shall discussed later on) the concept is exactly the same for any field. This concept is that there exists a core component from which complementary components are built upon. The core component may evolve in time but it will always be the socle for the complementary components, which can be as diverse as they want or can be. Tushman and Murmann (1998) state, “the system is partitioned into the core component with low variety and the complementary set of peripheral component with high variety”. Meyer and Lehnerd (1997) come up with the most simple and efficient definition of a platform “as a set of subsystems and interfaces that form a common structure from which a stream of derivative products can be efficiently developed and produced”

The term platform is used in three related fields, product development, technology strategy and industrial economics. We shall proceed to define it in all three fields. In product development a platform product is a new product satisfying the requirement of a bulk of customers but designed to be easily modified into derivatives through the addition, substitution or removal or certain features (Wheelwright and Clark 1992). A prime example of this is the daily used car. In this case, the complementary components are the engine, CD player, type of interior etc.

For technology strategists a platform is a means of control over an industry. If the core component of the platform is successful complementary component can be used to give the user a personalized\(^2\) version. In this case, the platform represents a significant milestone in the future of the product in question or the firm itself. (Bresnahan and Greenstein, 1999) examined the antagonist approach of Microsoft and Netscape. The former opted for “platform groups” strategy and the later chose “cross platform” strategy. Microsoft used its windows operating system (Cusumano and Selby 1995) while the Netscape, then dominant in Internet browser, made its product usable on multiple platforms (Cusumano and Yoffie, 1998). Later on, Windows wound up with

\(^2\) Personalized in the sense of usability, or conformity with the consumer requirements
92% of market share and developed its own Internet browsing software revolving around its core component, Windows and thus driving Netscape of the grid. Even though Netscape had adapted its product as a complementary component of Windows, it was relegated by an in-house product. Even though the tactics used by Microsoft subject to discussion, this shows us the strategic aspect of platforms in an industry. A study by Gawer and Cusumano (2002) also finds that companies such as Intel, Cisco, and IBM... all have used platforms as a strategic aspect of their development.

Platforms were adopted by industrial economist “to characterize product, services, firms or institutions that mediate transaction between two or more groups of agents” (Rochet and Tirole 2003). This type of platform creates network externalities from which the platform owner can benefit from. So there appears to be a dilemma for the owner. Should the platform be open and entry relatively easy or should he/she give it away and await the benefits from the network externalities (Parker and Van Alstyne, 2000)? This is the common chicken and egg problem generally meet by credit card payment networks and shopping malls (Eisenmann, 2008).

Most platform definitions from the Oxford English dictionary to Meyer and Lehnerd (1997) passing by Robertson and Ulrich (1998) Bresnahan and Greenstein (1999) all focus on the reuse or sharing of core component across complex products, from which we identify a architectural pattern among platforms. Architecture is simply “the scheme by which the function of a product is allocated to physical components, including the specification of interfaces between interacting components” or simply the function to component mapping, but included interfaces in the design rules needed to create a modular system, (Baldwin and Clark, 2000).

One of the main dilemmas of platform resides in the decision to make them open or closed, what is the optimal point. The opening of a platform can lead to its adoption by user because there is no user cost, no lock in effect and the parent company rips benefits from the network externalities, but in contrast it also eliminates switching cost thus increasing competition among platforms rendering the network externalities more difficulty to capture.

The degree of openness of any platform varies from industry to industry and also from product to product. The brand of a company also counts in the openness factor. It is the platform owner’s job to determine this.
2.2.1 Platform Structure

The platform poses both difficulties and opportunities for firms. Basically, the literature suggests that a platform forms a shared structure. This structure is divided into four main categories in (Robertson and Ulrich, 1998) and can be adopted by software development:

1. *Components*: the part designs of a product, the fixtures and tools needed to make them, the programs burned into programmable chips or stored on disks, and the circuit designs. For example, a software development kit (SDK) or an application programming interface (API) are two core components adopted in software development practices.

2. *Processes*: the equipment used to make components and the associated production and development process and supply chain. In software development practices, a verification process applied by a platform’s proprietor on application or service developed by a third-party developer is a one process type.

3. *Knowledge*: technology applications and limitations, mathematical models, testing methods, production techniques, and design know-how. In software development practices, knowledge can have various forms such as development guides, tutorials, documentations, guidelines, human interface guidelines, wikis, standard libraries associated with programming languages.

4. *People and relationships*: teams, relationships among team members, relationships with a network of suppliers and third-parties, and relationship between the team and the larger organization.

2.2.2 Platform Benefits

There are various distinctiveness benefits for firms adopting platforms in their processes, products, and services of the development processes and practices. Adopting platforms in firms appears to be a distinct and successful strategy to create a various assortment of products and services along with efficient uses of resources (Sawhney, 1998; Meyer and Lehnerd, 1997; Meyer, Tertzakian, and Utterback, 1997; Robertson and Ulrich, 1998; Ulrich, 1995) to help firms to cope with customers’ needs for new technologies and innovations overtime to stay competitive more effectively. (Sawhney, 1998) classifies these benefits according to their simultaneous improvements in:

1. *Speed*: the development time for derivative products from a common platform can be dramatically reduced by reusing technology platforms, development processes, component designs, and suppliers working on them. This will create a variety of products and services for the platform.
2. **Cost**: the platform plays a significant role in saving in the design, development, operating, and marketing costs of new products. Shared components and shared designs reduce design costs. Shared development, and shared processes reduce development and labor costs. Shared brands and common customer bases increases customer loyalty, reduce customer acquisition costs, and reduce the cost of brand building.

3. **Design quality**: the design quality of new products can be improved by using a common underlying technology, design, and components, which are thoroughly tested and debugged. Moreover, a performance improvement in the underlying platform will automatically affect all derivative products created using the platform.

4. **Coherence**: Offering products that are managed as platforms have the ability to be extended more logically and coherently and to related products and markets.

5. **Referenceability**: Platform promotes new products to be marketed to a core customer base or customers logically related to the core customer base. The relatedness among the customers of the core product, the derivative products, and the content generators have results in a robust brand advocacy.

6. **Option Value**: Platform and investments in core technologies represent a set of “call options” that a firm can use to exercise in related businesses.

### 2.3 Software Platforms

In software, platforms are springing out of the ground at tremendous speeds, every tech company is jumping on the bandwagon. In the outrageous pace at which technology is developing, a platform seems to be the only way for companies can keep up with the pace of innovation (J. Bosch, 2006). In this case, the software company provides the platform and the core component. The software and the complementary component can be provided either by a company, either by third-party developers or by both (Parker and Van Alstyne, 2008).

Successful software companies such as Microsoft, Apple and Google have all adopted the concept of platform in the early ages of their development (Evans, 2006). By creating a platform from which they themself used to build groups of products, which are complementary to the core product. Today the R&D cost of innovation in order for a company to solely build products around its platform would be too costly and unable to generate a wide variety of products, to meet the consumer demand.

The computer industry is the only industry to this day that has defied the laws of monopoly. This industry is in the hands of a few giant mainly one per sector (Windows in the PC operating system, Intel in the processor, Sony PlayStation...
in game console etc.), but the competition in this industry is relentless. For this paper at hand, we shall focus on software sector and it cut throat competition that has made it evolve in creating software giant and software ecosystems. Even the mighty Microsoft, which still controls more than 75% of the PC operating system, is quivering by how fast the industry is changing and thus rendering it to a company of the ice age (Parker and Van Alstyne, 2008).

A large number of modern products are based on software platforms; the world is shifting into the software, which is also shifting to the cloud itself. We are witnessing a double shift in the tech world. The strategic point is how should companies deal with this shift, how should software platforms be? How open should the platforms be? Should they be totally open with only software companies providing the platform for third-party developers to write applications like in the case of app store or should they be semi open like the current Windows’ model, which already has created a software ecosystem, which constitutes the core platform, and the complementary products are created around this ecosystem. In the former, the platform’s future direction is market driven and in the later it is oriented by the will of the firm controlling the ecosystem. In the software industry we notice that communities of innovation networks, known as ecosystems, have been widely adopted by software platform owners to meet heterogenous user needs (Bergvall-Käreborn, Howcroft, & Chincholle, 2010).

Software platform is defined as “the extensible codebase of a software-based system that provides core functionality shared by the modules that interoperate with it and the interfaces through which they interoperate” (e.g., Apple’s iOS and Google’s Android). A module is defined as add-on software subsystem that interoperate with platform to add functionality to it (e.g., iPhone’s apps and Google’s Android apps). A platform-based software ecosystem is defined as “the collection of the platform and the modules specific to it” (Tiwana et al., 2010). (Baldwin and Woodard, 2009) define software platforms as a set of interrelated specification layers that provide interoperability between the modules and a set of common resources through which to generate derivative assets (e.g., applications and services).

The illustration on “Figure 3” gives a visual explanation of these distinctions.
Apple’s iOS and Google’s Android, and their apps are playing a dominant model for software and services development. Unlike traditional development for software and software-based services, these services leverage the outside expertise of independent third-party developers with skills, capabilities and an insight of user needs that platform owners might not possess.

Software Platforms present a substantial opportunity and challenge for information system research for six reasons (Tiwana et al., 2010). First, the competition is increasingly toward software platform ecosystems, unlike stand-alone systems, which have been the anchor of the literature in the information systems development. Second, the firm boundaries are expanding to harness the outside expertise of a diverse developers community on an unprecedented scale. Third, governing software platforms requires an accurate balance of control by platform owners and independent third-party developers. Fourth, organizing principles and technical architectures of these software platforms together determine their evolutionary trajectories in order to influence software platform differentiation. Fifth, software platforms do not exist in a void, software platform designers’ technical choices impact how well or how poorly they interact with the dynamics of their environment. Finally, software platforms offer the information system discipline an extraordinary opportunity to consider the IT artifact in the core of theory development about how software platforms evolve and to consider their technical design choices to contribute unique distinctiveness from software engineering, strategy, and economics.

Although the software platforms compete for both end-users and third-party developers, in this research at hand I focus solely on the supply (third-party developers) side rather than the demand (consumer) side of software platforms; and on those that are two-sided markets connecting software platform specific module third-party developers to end consumers.
2.3.1 iOS Platform

The iOS platform developed by Apple and launched in June 2007 (Honan, 2007). iOS was released in connection with the iPhone and is now also on iPad and iPod touch. When the iPhone was launched, it was unique with its large touch-sensitive screen and the lack of physical keyboard. The user could interact with the so-called touch screen through various movements of the fingers (West & Mace, 2009).

The development of iOS made against iOS SDK in Xcode development environment, which is an IDE developed by Apple. Using Xcode, it creates the user interface to be pursued and there is also the code behind the user interface. Xcode provides the developers thus all development tools needed to develop their apps for Apple products. In Xcode includes Xcode IDE, Instruments and iOS Simulator. The code is written in the programing language Objective-C. The development of iOS must be done on a Mac computer and then Xcode is only available for Mac OS X (Apple Developer - Tools Workflow Guide for iOS, 2012).

2.3.1.1 Xcode

Xcode is a collection of cutting-edge tools for creating great apps for Mac OS X and iOS and Easy to use, yet capable of building the most sophisticated apps, is an integrated development environment where developer can use a single tool for creating app in multiple languages, such ex. C, C++, Objective-C, Java and AppleScript. Xcode uses a graphical workspace that closely integrates a professional text editor, a robust build system, a debugger and a powerful GCC compiler. The complete tools chain for development to Mac OS X and iOS comes with as part of Xcode, among these tools, iOS simulator, Interface Builder, Instruments, WebObjects framework and a complete reference documentation (Apple Developer - Tools Workflow Guide for iOS, 2012).

2.3.2 Android Platform

In 2007 formed the Open Handset Alliance by Google, which is an alliance of mobile operators of the Android will become a world-leading operating systems. This meant that Google could implement its own search page in the operating system, just like Internet Explorer in Windows (Hall & Anderson, 2009).

Android is a modified version of the Google-core and is an open source operating system, which makes it free. The programing language used to develop an Android is Java and the Java code is converted to Dalvik at runtime. Dalvik is a virtual machine optimized for mobile phones for the smoothest driving (Android Developers, What is Android?, 2012).
In order to develop an Android mainly used programming language Java in combination with XML, Android also supports C and C++ that can be used to develop certain components. The development has access to large parts of the phone's hardware and the same API as the basis apps on the phone uses (Android Developers, What is Android?, 2012).

To set up a development environment for Android, a number of steps performed. These steps are best described in Android (Android Developers - Installing the SDK, 2012) but are presented here in brief. In developing recommended Eclipse, but it is also possible to write code in another IDE or directly into a text editor. Eclipse is recommended that, where direct is to call the other resources required to develop an app (Android Developers - Introduction, 2012) and is also recommended that the ADT (Android Developer Tools) plugin is used. Android can be developed in either Windows, Mac OS X or Linux (Android Developers - System Requirements, 2012). When development occurs in the Java programming language also requires the Java Development Kit installed on the developer's computer. Although Android SDK must be installed which is the most basic tools to work with Android. Thereafter, an SDK is installed for a specific Android platform. The choice of platform depends on which version of Android the developers wish to develop their apps, but at least one platform must be installed. The platform also follows an emulator skin that developer can test his/her app on. The developer can also test his/her app directly into a device running Android. Made progress on Windows requires USB drivers for this (Android Developers – Installing the SDK, 2012).

Today there are a large number of devices running Android. These devices have different screen sizes and resolutions. They have different hardware in general with different physical buttons on different devices. That there are many different versions of Android in all of these devices make it a lot to adjust to. This enables the development of devices that Android needs thorough testing be done before publishing to avoid errors. The numerous varieties of mobile device that now have Android make it impossible to test their apps on all devices. On Android Market, a list of which devices are those who have downloaded the app, developed the most. When the user must always be in focus, this list is a guide in which units tests should be performed on.

2.4 Scrum

Scrum is an iterative and incremental agile software development methodology that emerged in Japan in the 80s based on studies performed in the manufacturing industry. This development methodology is considered to be adaptive, fast and self-organizing, which was considered necessary in an area where major changes take place continuously (Abrahamsson, 2002). Variables such as customer requirements, timeframes, technology, and resources are constantly software development processes and will likely change over time,
making the development process is unpredictable. The target is then to consider these variables, which requires a methodology that is flexible (Schwaber, 2002).

The first phase consists of two phases, planning and high-level design. The planning phase will create a product backlog-list that contains all the requirements that are currently known. These requirements can come from different stakeholders. Next, priority requirements for the operation as required. Product Backlog-list is continually updated with new demands or to more detailed forecasts and at each iteration, evaluated the list of team for knowing what to do at the next iteration. This bottom phase also defines project team on potential risks and available resources. The high-level design phase is the planning of the backlog records will be implemented. It also discusses the impact that technological developments have on the implementation. The update of the product backlog-list is done iteratively, but the other processes in the pre-game phase occurs in a predefined way, like the waterfall model (Schwaber, 2002).

The second phase is the phase of Agile Scrum. This phase is described as a black box where unpredictable things will happen. The aforementioned variables are monitored in each iteration, called Sprint, and controlled by such risk management procedures. A Sprint is expected to take one to four weeks to complete, and each Sprint will lead to a refinement of the developed product. Before each Sprint, a Sprint Backlog-list drawn up that defines what must be done during Sprint. Once this list has been established is no change only after the iteration has completed. Each Sprint ends with an examination of the entire team participates in (Schwaber, 2002).

When the team believes that the variables, such as all requirements have been completed, the team goes through to the last phase, termination phase. This happens all the preparations for the release; the software is tested, integrated and documented prior to the handover takes place (Schwaber, 2002).

(Abrahamsson, 2002) In Scrum there are several identifiable roles that are presented below. The three top one is the most important roles, while the last two acts as support roles:

- **Scrum Master**: This role ensures that all the team are focused on the task and that no obstacles in the way of productivity. This role is not a team leader role, Scrum presupposes that the team is self-organized. However, this role is responsible for the project work according to the practices, values, and rules of Scrum. This role should also ensure to hold daily Scrum meetings in about 15 minutes where your progress and obstacles are discussed.

- **Product Owner**: This role is selected by the Scrum Master and other stakeholders and has the task to ensure that priorities are understood by all
involved. The role acts as a voice of the customer and is responsible for product backlog-list and it can be seen and followed. The product owner collects descriptions of how software should work and turn them into clear requirements for functionality. This role has the authority of the team.

- **Scrum Team**: The team is responsible for delivering product and consists of five to nine members. The team must be self-organized and cross-functional and also determines the workload required to complete the tasks.

- **Stakeholders**: Stakeholders contribute through requirements definition to the product backlog-list and is also involved in the audits.

- **Management**: Management has the final word in decision-making situations and is also where the objectives and requirements are defined. Since Scrum itself does not require any specific development practices can be adapted to work with the approach already used within an organization. The major difference arises primarily in the various roles as a project team should consist of. However, Scrum delivers much of the responsibilities for the development team. This development model is thus suited for small teams where flexibility, customer opinions, and adaptation to changing requirements are important factors.
3 Methods

The research methodology encompasses the overall study that is being used of gathering data. For the methodology different tools or methods used to gather data (Dawson, 2002). There will be a discussion, which companies were taken for the interviews and why.

As a result, this section outlines a research framework used for this study. Furthermore, detailed information pertinent to the research design and research approach, different methods and tools for data collection is bequeathed in this section.

3.1 Research Framework

The illustration in “Figure 4” gives the reader a better understanding and a wider point of view of the research at hand. This overview will be analysed and discussed in further sub and chapters.

![Research Framework](Image)

**Figure 4: Research Framework**
After the research question has been formulated, I have to decide what research method I am going to employ, Qualitative research or Quantitative research. A qualitative research inspects behaviour, attitudes and experiences through methods such as focus groups or interviews (Dawson, 2002). A quantitative research generates statistical techniques through methods such as structured interviews or questionnaire (Dawson, 2002). However, for this research no need to mathematical or statistical data to answer the research question. Hence, the qualitative inquiry will be the right choice for this research, and the reasons are explained as the following:

Since the research at hand has a lot of literature review, interviews, and interpretations to get new insights and some experiences about the topic and inquire into the answers of all interviewees. Subsequently, this research aims to identify and explore the development process associated with third-party developers for adopting the development of applications in Apple’s iOS and Google’s Android platforms.

Qualitative research has many types variant from various disciplines, such as education, biology, sociology, psychology, anthropology, and history. For explanatory purposes, five major types have been highlighted: 1) phenomenological studies, 2) ethnographic studies, 3) grounded-theory qualitative research, 4) biographical studies, and 5) case studies - this type is used in many disciplines (Dawson, 2002).

### 3.2 Research Design

The research design will be the general plan of how I will go about answering the research question. In the research methods’ literature the research design is a threefold classification one of descriptive, explanatory and exploratory. However, this research is exploratory design since the question to be answered here is a “what” question. An exploratory study is “a valuable means of finding out of what is happening; to seek new insights; to ask questions and to assess phenomena in a new light” (Saunders, 2009).

This design is being conducted to get better understating of the concepts and to clearly define the problem. It is also being as insights sought within the companies. Subsequently, the research aims to clearly identify and explore the main factors that third-party developers take into consideration when developing apps for Apple’s iOS and/or Google’s Android platforms. Two principal ways will be used to conduct an exploratory design are (Saunders, 2009):

- Interviewing ‘experts’ in the subject
- A search of the literature

Further, detailed information about these ways will be extracted in sections 3.4.1 and 3.4.2.
3.3 Research Approach

The attention here turns to the research approach to be employed. As the research approach at hand can be used for descriptive, explanatory and exploratory research. Semi-structured interviews were asked to the interviewees to get as much valid data as possible. My choice for the research approach will definitely guide the research objectives and question. From this perspective, one research approach will be used from the following: 1) experiment, 2) case study, 3) action research, 4) grounded theory, 5) ethnography, and 6) archival research (Saunders, 2009). This research will employ a case study as a research approach. Case study defined as “a strategy for doing research which involves an empirical investigation of a particular contemporary phenomenon within its real life context using multiple sources of evidence” (Yin, 2003). Miles & Huberman (1994) define a case as “a phenomenon of some sort occurring in a bounded context”. As for the case study, the context is important and influential. In addition, the boundaries between the contemporary phenomenon and the context within a case study are not clearly obvious. Therefore, a case study is an appropriate approach for whom wishing to obtain a greater understanding of the context of a research and the processes being taken place.

The case study is considered to answer the “Why?”, “What?”, and “How?” questions (Saunders, 2009). Relying on the research question “What is the development process (or framework) by which third-party developers adopt for the development of application in mobile platforms?” and the exploratory design as the model for its research design, the case study by then will be a suitable approach and more the concern than other approaches of this research.

(Yin, 2003) distinguishes between four strategies for the case study: 1) single case study, 2) multiple case study, 3) holistic case study, and 4) embedded case study.

A single case study is often selected where it represents a unique or extreme case. Moreover, a single case study it is typical, or it gives you as a researcher opportunity to observe a phenomenon and analyze it from a perspective that few or none have observed before (Yin, 2003).

As multiple cases study, multiple cases are incorporated, the need to set up the findings from the first case for the other cases. As a result, the generalization of these findings will be needed.

As a consequence, the research approach is going towards a single case study.
3.3.1 Single Case Study

A single case study is an empirical inquiry investigating a contemporary phenomenon within its real life context and especially when bounding a specific phenomenon and its context can’t be analyzed in a clear manner or unambiguous one (Yin, 2003).

(Yin, 2003) identifies the characteristics of single case study as follows:

- It defines the operational relation between a set of conditions and its effect. Cause/effect analysis
- Mainly targeting to answer the What/How/Why questions
- Based on causalities
- Finally, it explains by defining the mechanism or mechanisms through which a specific set of relations has been identified

We do have to understand that the main reasons behind choosing a single case study rely on the limited time devoted for the research mainly five months, in addition to the limited resources being used. Furthermore, mobile apps and mobile development is in general a new, relatively unexplored phenomenon, and the researches on it are extremely limited. As (Yin, 2003) mentioned, time constraint is one of the basic reasons researchers opt for a single case study.

Besides, criticism of using a single case study was about its unwillingness to provide generalizing conclusions. Which push scholars to describe its methodology as microscopic due to the lack of meaningful number of cases. However; scholars such as (Yin, 2003) and (Hamel, 1993) argued that the number of cases being in use does not transform the single case study into macroscopic study.

Therefore, a single case study research will be acceptable if meeting the established objectives (Yin, 2003), which I think in my point of view I have satisfied since my main objective is to use findings and data in order to illustrate the framework process.

Section 3.3.1.1 presents the company where the case study was carried out.

3.3.1.1 Infospread Euro

Infospread founded in 2005 with a vision to create mobile applications that make life easier and more fun for users. Now, five years later, they can in good conscience say that they so far have succeeded.

Infospread is a product development company with a development process that moves the boundary between the possible and impossible. Sometimes exclaims one of the team ”eureka”. Then, they know that they succeeded in producing a
prototype that they want to demonstrate to their customers and the world. In most cases, their customers are just as excited as them.

Infospread is working with mobile solutions that are targeted primarily towards public transport. They believe that a mobile application must be simple, while in must do what the user wants. Their products and solutions address a broad group, which makes high demands on the simplicity and flexibility. To reach such a large user base as possible, rely on their products, existing technology and infrastructure. While their prototypes and test applications are always located on the border to techno freak and the most recent.

Infospread success is product MobiTime, is a comprehensive public transport guide app. MobiTime includes smart timetables, bus stop information in real time, travel increases and payment of trip. MobiTime is available for JTME, Android, iOS and Windows Phone 7.

3.4 Data Collection

There are two principal ways of collecting data for a research that are primary and secondary data. Since exploratory research design has been adopted in this research; data is collected through the searching of concerning literature and interviewing the experts and experienced in the subject (Dawson, 2002). Subsequently, secondary data as much as primary data were collected to answer the research question.

3.4.1 Primary Data

Primary data is seen as the first hand investigations and observations (Dawson, 2002). This is usually done when secondary data is unable or not available to answer the research questions. Consequently, data is gathered by using different techniques; documents, personal interviews, or observations to get the required results (Saunders, 2009).

In this research the primary data will consist of qualitative data that will be gathered by interviews. Generally, interviews may be structured, unstructured, or semi-structured. Structured interviews are used in quantitative research and comprise structured questionnaire. Structured approach is used frequently in market research. Unstructured interviews or in-depth interviews the researcher opens an open dialogue with the interviewee to achieve a holistic understanding of the interviewees' situation or point of view. Unstructured approach is used frequently in life history research. Semi-structured interviews are explicitly well-suited for case study research. Using semi-structured approach, the researcher asks predetermined but flexibly worded questions, so that other significant information can still arise while during the interview (Dawson, 2002).
Akin to this, this research has conducted semi-structured interviews, in order to get as much valid data as possible. Questions have asked to collect free experiences, opinions, and insights pertinent to third-party development practices from the interviewees. Crucial questions are identified and interviews are conducted based on these questions. Reputable IT companies in Sweden are identified and the experts from these organizations are the potential targets for this research. Besides, this research has conducted face-to-face and over the telephone interviews, in order to understand the experts’ point of view in regards emerging technology.

However, this method provided an open dialogue to the interviewees, whose point of views are recorded and pertinent data is extracted when needed. Cognizant to this, this research faced challenges pertinent to time, due to the occupied schedules of individual interviewees. In addition, this method proved apt for this research, since it assured abundance of data collection. Hence, assuring the validity of the research, the open dialogue provided the crucial environment for the interviewees to unfold their personal experiences. Furthermore, the direct method of communication, enabled in the collection of detailed point of views, and approved the relation of the gathered data and their theoretical framework at hand.

### 3.4.1.1 Interviews

This section provides an introduction of interviewees conducted in Infospread Euro. The table below outlines the details of the interviews conducted in order to gather primary data.

<table>
<thead>
<tr>
<th>Interview No.</th>
<th>Interviewee</th>
<th>Role</th>
<th>Purpose</th>
<th>Duration</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>iOS developer</td>
<td>Responsible for developing apps for iOS platform</td>
<td>To get general information about iOS platform and App Store</td>
<td>40 min</td>
<td>October 2011</td>
</tr>
<tr>
<td>2</td>
<td>iOS developer</td>
<td></td>
<td>To get detailed information about the programming language and the tools used for developing apps in iOS platform</td>
<td>90 min</td>
<td>February 2012</td>
</tr>
<tr>
<td>3</td>
<td>iOS developer</td>
<td></td>
<td>To get information about</td>
<td>90 min</td>
<td>February 2012</td>
</tr>
<tr>
<td>No.</td>
<td>Interviewee Role</td>
<td>Responsibility</td>
<td>Interview Objective</td>
<td>Duration</td>
<td>Date</td>
</tr>
<tr>
<td>-----</td>
<td>---------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------</td>
<td>----------</td>
<td>----------------</td>
</tr>
<tr>
<td>4</td>
<td>Android developer</td>
<td>Responsible for developing apps for Android platform</td>
<td>To get general information about Android platform and Google Play</td>
<td>40 min</td>
<td>October 2011</td>
</tr>
<tr>
<td>5</td>
<td>Android developer</td>
<td></td>
<td>To get detailed information about the programming language and the tools used for developing apps in Android platform</td>
<td>90 min</td>
<td>February 2012</td>
</tr>
<tr>
<td>6</td>
<td>Android developer</td>
<td></td>
<td>To get information about submission process to Google Play</td>
<td>90 min</td>
<td>February 2012</td>
</tr>
<tr>
<td>7</td>
<td>CEO</td>
<td></td>
<td>To get information about the company and to understand the mobile applications market</td>
<td>30 min</td>
<td>October 2011</td>
</tr>
</tbody>
</table>

Table 1: The interviews conducted to collect primary data

The interviews questions are divided into two groups, and they are explained as follows:

1. **About the company:**
   - Tell us about the company.
   - What is your role at the company?
   - How many development projects have you taken in total?
     - Why do you use these techniques?
     - What projects are you working on right now?

2. **The development process:**
   - What do you use for processes / methods in development?
- Why do you use these techniques?
- How did the development process start to finish this for an example?

- What is your strategy for fully developed apps?
  - How are the rights? (Who owns the code?)
  - How much work is it to register on the App Store and/or Google Play as a company?
  - How is the handover of the product?

- What do you think that different platforms pros and cons?

- Which tools do you use for the development of an app for Apple’s iOS and Google’s Android platforms?

- What platforms have you developed apps to?

3.4.2 Secondary Data

Secondary data is second hand data, it involves information that already exists from other scientific authors and researchers and is passed through the process of appropriate synthesis and analysis (Dawson, 2002). In that matter, research books, articles, journals, and websites of reliable authors and organizations have been employed to collect data and gather information about third-party development process for Apple’s iOS and Google’s Android.

3.5 Research Analysis

Qualitative research analysis will be effectively conducted in this research. Since the research basically focuses on asserting the practices of third-party developers adopting for the development of application in mobile platforms, relevant data pertinent to the development process, and various steps to explain these practices are collected which are detailed in sections 3.4.1 and 3.4.2.

The gathered data is then subjected to qualitative analysis, where a deeper understanding of third-party development practices in terms of developing applications for Apple’s iOS and Google’s Android.
3.6 Research Credibility

It is vitally important to ensure to deliver a high quality data that are being collected when doing a research. Research credibility is defined under three principal aspects, reliability, validity and generalization (Saunders, 2009). These three aspects are eventually judge the applicability and quality of the research. A detailed discussion of each of these aspects shall ultimately help in judging the credibility of the research.

3.6.1 Reliability

Joppe defines reliability as “The extent to which results are consistent over time and an accurate representation of the total population under study is referred to as reliability and if the results of a study can be reproduced under a similar methodology, then the research instrument is considered to be reliable” (Golafshani, 2003).

For the qualitative study at hand, examination of trustworthiness is crucial. Therefore, secondary data is collected from trustworthy and reliable resources such as scientific websites, research books, research articles, research reports that have been used are run by organizations, creators, or researchers that are trusted. Further, primary data is collected from reputable organizations and experts in the subject that are currently active in the area of mobile development. Cognizant to this, care is taken to make sure that the semi-structured interviews are based on the formulated research question.

3.6.2 Validity

Joppe defines validity as “Validity determines whether the research truly measures that which it was intended to measure or how truthful the research results are. In other words, does the research instrument allow you to hit "the bull’s eye" of your research object? Researchers generally determine validity by asking a series of questions, and will often look for the answers in the research of others ” (Golafshani, 2003).

Whether the validity in a qualitative study is concerned there is a threat that interviewees may not be trusted, but instead gives incorrect answers about their information within their organization (Saunders, 2009). Also the answers might be interpreted in wrong way from the interviewers.

Validity is tested based on three criteria, content validity, predictive validity, and construct validity (Saunders, 2009). Since a qualitative study is concerned for this research, content validity can be tested; for, “extent to which the measurement device, in my case the measurement questions in the interviews provides adequate coverage of the investigative questions” (Saunders, 2009).
3.6.3 Generalizability

Attending generalizability has traditionally been related to the qualitative methodology. However, this situation has been changed in recent decades due to many trends such as the growing use of qualitative research in policy-oriented researches and evaluation. Which led to a rise about the awareness for the importance of designing such studies, qualitative, in a manner aiming to enhance its implications in understanding other cases. Thus being able to generalize the findings to other situations (Bogdan & Biklen, 1998).

Mainly three targets for generalization have been identified which are: what is, what may be and what could be. Studying what is, is related to what is typical, ordinary as well as common (Bogdan & Biklen, 1998). Concerning what may be, deals about designing researches in order for them to fit with potential future trends. Finally, the what could be deals with identifying situations that once can expect ideal or in a way exceptional and seeing what happening over there (Becker, 1986).

Generalizability amounts to making predictions pertinent to recurring findings. The issue of whether the researcher can generalize in qualitative research is a matter of validity. Generalizability is a big concern for researchers; it considered one of the criteria for quality case study studies based on the case chosen and being studied. For instance, a case study within a small organization would have difficulty to reason generalization to bigger organizations (Golafshani, 2003).

The objective with qualitative research is the choice of a relevant small number of companies or people from which insights can be driven to a bigger population range and not specifically to generalize (Dawson 2002).

The generalizability concerns existing about my research is about the choice of the company Infospread Euro for interviews that is part of my research. Where the company is representing a specific size or portion of these last. Obviously, if another set of companies was chosen the results and analysis will definitely be different. Therefore, the information driven from the company I choose can determine insights about the behavior of a bigger set of research population.


4 Results

This chapter of study summarizes the factors and development process that third-party developers take into consideration when developing apps for iOS and Android platforms. The collective data from interviews and literature study has provided relevant information needed to understand the apps development process and the perspectives related to it.

4.1 Ecosystem

Based on the collected data from the interviews, this section identifies and explores the different actors in the Apple and Google ecosystems. Meaning that any actors that may be a company, government entity or person can play a role of concepts and ideas development for native applications, which may work on the iOS and/or Android. The illustration in “Figure 5” gives the reader a visual explanation of different actors in the Apple and Google ecosystems.

![Ecosystem Diagram]

**Figure 5:** Apple & Google ecosystems
4.1.1 Apple

In 1976 two brilliant people, Steve Jobs and Steve Wozniak, have started the Apple company, which is an international global company producing and marketing consumer electronics as well as personal computers and computers software. Apple’s products brand extension includes Macintosh computers line, iPhones, iPods and their latest creation the iPad (CNNMoney, 2011).

Apple is the world’s admired company, offering devices that play a crucial role in the ecosystem, making it by far the leader in the mobile market industry. To have an idea about how Apple’s second-quarter 2012 financial results are, the bellow numbers will illustrate them (Apple Press Info, 2012):

- Apple sold 35.1 million iPhones, 7.7 million iPods, and 12 million iPads
- Apple’s quarterly revenue of $39.2 billion, and quarterly net profit of $11.6 billion

Being able to create and build an international reputation, positioning Apple as the leader in its industry is no matter of chance. Since, Apple’s products are well recognized for their product integration, where quality, design as well as hardware and software are self-created. Having a futuristic vision and being keen on the choice of materials and design features, Apple’s strategy consists on handling their devices with a high level of care, giving attention to small details that makes big difference such as: their font choice and icons designs and offering it in one package that make a difference and fit together.

4.1.2 Google

Nowadays, Google has one of the largest hardware infrastructure as well as computing power worldwide. This fact was possible via the saving information strategy Google created, making users able to search for information through all the saved pages as well as relevant results instantly (BusinessWeek, 2005).

According to BusinessWeek, Google has purchased Android Inc. on the 17th of August 2005, a move from Google to create a new line extension by targeting the mobile phone market. With the large infrastructures developed by Google and its ability to store huge amount of data, Google for Android has monopolize some of the features and especially such as Google Search, Gmail, and etc..
4.1.3 App Store

The App Store is a digital application distribution platform for iOS developed and maintained by Apple, it is accessible from the iPod touch, iPhone, iPad, and iTunes via an iOS application.

4.1.4 Google Play

Google Play, formerly Android Market is a digital application distribution platform and multimedia-content developed and maintained by Google which includes an online store for books, music, movies, cloud media player, Android applications and games. Play Store is preinstalled on Android OS devices that allows consumers to download various applications created by Google or third-parties developers, and it is accessible from the web, Google TV and Play Store on Android.

Regardless what is done to enhance the App Store and the Google Play, the challenge for third-party developers and consumers will always be the same, the concern for apps developers is how to get their apps remarked, while the concern for consumers is how to find desired app easily among all the apps in App Store and/or Google Play.

Based on the data collected from the interviews, it is obvious that many developers doing apps for iOS and Android platforms strive at what price to sell their apps. Free apps can bring more downloads for them but might not bring them much profit. Another perspective that apps developers are adopting is known as in-app purchase. However, this can be practiced from a freemium app to drive sales to a paid version of apps, or developers can build add-on options into their paid apps to boost the sales of their apps.

4.1.5 Developers

Based in the collected data from the interviews, in today’s IT on the go world, third-party developers started playing a crucial role in the development of applications in mobile platforms. They are representing substantial homogenous group with to the edge coding experience, with at least one published application.

The CEO of the company said “third-party developers group age may vary from 14 to 50 years old, however, most of them are between 22 to 33 years old. However, the field is still dominated by males where the number of women developers is still in a away insignificant so far.”

Third-party developers have acquired tremendous number of skills in terms of platform know-how as well as publishing experience spanning from 1 to more than forty recognized published applications. The majority of the developers of
the company and in order to focus on their expertise have opted for single platform development, and it was obvious that interviewees’ focus and expertise were tended to one platform. Nevertheless, some opted for multiple platforms development. Third-party developers have most of the time worked on outsourced applications that were commissioned by other third-party organizations on their behalf; still most of them have ownership over their developed applications.

The CEO of the company said “Concerning revenue for third-party developers it has always depends on how active and experienced they were and this last ranged from $120,000 to $100 for experienced to those who had released seven applications. Still, a multi-platform developers enrolling in a Stockholm start-up was able to generate 25,000SEK from the least successful applications he developed.”

One of the issues third-party developers are encountering is the lack of distribution, which is a major factor in promoting their artifacts, even if they did have ownership over their products and the associated intellectual property rights. Apple and Google have granted the access to the markets and marketing for third-party developers, which however has been viewed as an advantage, still was entirely within the control of both Apple and Google. This means that third-party developers are operating as a “commodity producers” when fortune technology firms in the industry are controlling the stores showcasing and promoting their applications into merchandise for a worldwide global consumers. Since mobile applications are suffering from high quick rate of obsolescence and short life cycle, it makes the market full of unpredictability and flops in a market that requires and needs a continuous constant stream of creative new products that have huge potential for mass distribution. To have an idea on some of the issues third-party developers are facing I would love to say from what a developer said from his own experience in iPhone development that: “One of the issues I was confronted with is from the App Store approval system inconsistency. One time, the App Store has rejected one of my application features, which apparently was not functional. However, one can notice from their server’s logs that Apple has tested the application and it functioned, and still they were sure it did not. The reason behind this mistake was the fact that they run it quickly without given the deserved time and attention to assess it and judge it. Besides, that is only one case scenario and I do know developers that their applications have been rejected for countless irrational reasons.”

In one hand, if we take into consideration the end-users, there exist few information available apart from the number of downloads, when it is feasible to have user feedback from applications stores. However, these stores don’t offer any interface for third-party developers to respond to end-user feedback. In the other hand, and from the third-party developers point of view, being able to interact with the end-users (consumers) can help them ameliorating upgrades
and the need for potential products, while in real life there exist a miner dialogue engagement with the end-users community.

The mobile applications are proof of the contemporary environment where programing is very distinct from use. One can argue on the fact that one of the reasons behind Apple and Google investing limited effort in determining links between consumers and third-party developers may be due to the fact that many applications are cheap, and that 1% of users downloading a specific application may become long-term user.

4.1.6 Consumers

A consumer who uses the App Store or the Google Play has certain options to look for and find apps before a purchase. The consumer can either use the App Store or the Google Play to find what is he/she searching for or use personal sources such as asking friends or professionals, or use public source that is, searching on various websites. In addition, the consumers can evaluate the apps before a purchase by reading what other customers have already written about the app in the customer reviews, seeing what other customers have bought, and even seeing the ratings of the app.

As a result to vast completion in the App Store and the Google Play, it is crucial for third-party developers to consider how the consumers behave in the App Store and the Google Play. Through understanding the consumer’s behavior and needs, the developers can address their market to the right audience. Further, if we apply the purchase decision process that is need recognition, information search, alternative evaluation, purchase decision, and post-purchas behavior on app consumers we get structure to describe the sights and sounds of the market place from the consumer’s perspective.

4.2 Programing Language

Based on the collected data from the interviews, in one hand, iOS programing language of choice is “ Objective-C”, which is the C language with extensions for what called object oriented concepts, for example: interfaces, messages, typing, inheritance and more… In the other hand, Android development is based on the Java language, which actually does not compile to byte codes, which, it does compile to Dalvik executable files.

In case of the company, the developers have been impressed by the Objective-C language and some of its features such as named arguments, categories and message passing, which is an alternative to calling methods. However, Objective-C language is an 80’s one and do have many flows that makes us as third-party developers not of a huge funs. Some of these flows can be described
as: time consuming problems concerning for example, split header, violation of DRY, and implementation files. These issues have both a cost in navigation as well as the fact that the sense of context has to be recreated in every opened file concerning the caret, selection and the developer position in the file and how it is organized.

DRY means every piece of knowledge must have a single, unambiguous, authoritative representation within a system.

Concerning DRY, we as third-party developers must assign five steps to just declare a property. These steps are:

- Declare in the class definition
- Declare getter/setter
- Initialize in the “init” method
- Synthesize in the implementation, and
- Finally, release in “dealloc”

“Figure 6” and “Figure 7” show how these steps can be applied in Objective-C language.

```objectivec
@interface Client: Updatable {
    NSString *name_; // declare the property
}
@property (nonatomic,retain) NSString *name_; // declare the property again
@end

Figure 6: Client.h

@synthesize name_; // implement getter/setter
-(void) dealloc {
    [name_ release]; // -- release memory
}

Figure 7: Client.m
```
Java language has a similar problem with properties, nevertheless not quit so bad; and the IDE helps the developers to write the setter/getter.

Moreover, the pointers in Objective-C language are time consuming. Hence, this is one of the areas where Java overpowered it using its garbage collection and third-party developers were continuously considering if the allocated objects are freed considerably. The code flow is really weak because of the fact that the application logic is littered by memory management, which is not of big helps when trying to localizing where mistakes have been made and try to fix them or making it not visible when accessed a part that has been deallocated. Besides, there are several other issues from developers, which are confronted with such as deallocating or autoreleasing objects two times.

Another issue in Objective-C language is the patterns to follow such as: many rules to know by hard, implementing the right init and dealloc methods is not trivial, @synthesized setter/getter for some proprieties with retain may not called in such models. Objective-C language import/forward declaration, @class is too complicated, and even thought the same issues are relevant to Java programming, however; Eclipse’s JDT is very impressive allowing developers to easily write an import. By just using “Ctrl+space” to auto-complete a class name or just “Ctrl+Shift+O” for example to organize an import.

### 4.3 Platform

Based on the collected data from the interviews, on Android third-party developers argued that they can use Java runtime, at least some of its standards such as Java RT classes which are available on Android, since the majority of Java IO, regex libraries as well as networks are affordable. Besides, Android RT classes are based on a stable class that is Harmony. In the other hand, getting the needed functionality with iOS is kind of hard, because of the poor organization of classes and methods. In terms of lacking of some functionalities for regular expressions such as XML handling and string handling, developers opt for Regex Kit Lite. Concerning XML parsing developers implement Libxml with a parser abstraction over it, which can be done easily with NSXMLParser that is SAX alike.

Concerning iOS developers have to turn to Google hoping to find solution to the same problems they faced, a technique that was applied by Apple´s first NDA policy. Since, due to this policy the iPhone content is limited on the Internet, developers have to turn to guesswork and experimentation as a counter solution.

Developers prefer sometimes to work with Android because of it is open source, which can be fully downloaded with all its code on their system, and
Results

give them the possibility to rebuild the SDK from those sources ensuring a match to their emulator’s runtime classes. This enabled developers to have an idea on how the implementation was done in the Internet but also discover why their codes didn’t produce the expected results.

Finally, the organization, naming conventions, and layout of Android platform classes are predictable and consistent, which make it easier to learn.

4.4 Programing Model
Based the collected data from the interviews, the iOS platform has done a considerable job in promoting MVC design pattern. With such pattern design build in to the iOS platform, being able to build an UI was easier without a need for developers to figure out the organization of the UI component design. Since all sample codes are organized in the same manner.

The Android platform offers kind of the same quality with design patterns using a completely different variation than the one used in iOS. Android’s platform allows for support to intents and Activities, with its support for multiple component reuse and processes. Resulting in a better user experience, which however does lead to a kind of problematic for developers, meaning that when trying to start an activity from another one, an intent is used to communicate any parameters. Which are only passed by value and not by reference. Whereas on the iOS it is simple to share the same data structure through screens, which is kind of hard to have with Android.

Finally, iOS and Android platforms allow for user preferences declaration in XML. Both platforms also provide a default UI for those preferences editing. Besides, iOS developers intending to customize preferences will need to implement a UI from zero, whereas XML format for Android is extensible allowing for custom UI components integration, which makes a breeze for user preferences.

4.5 Testing and Continuous Integration
The developers of the company for iOS and Android platforms agree on the fact that all the programing efforts have to contain unit tests. In addition, bigger teams must have or adopt Continuous Integration.
4.6 Resources

Based on the literature and the collected data from the interviews, Apple has done a considerable job of providing a lot of resources needed for developers to build apps for iOS platform. The iOS essentials, techniques, and best practices are explained in online videos, which makes grasping ideas and techniques easier for them. In addition, it provides as well many coding samples and sample applications in order to give an idea about the API usage.

One iOS developer said “The Development Center for iOS give access to the developers with materials such as the development know how, design as well as technical information that will be of support in order to develop using the latest technology. It also helps us as developers with applications programming, documents, videos, development guides, discussion forums as well as human interface guidelines.”

In the other hand, Android developers also have access to many resources such as the guide and API reference which are installed through SDK, making the information and resources available offline which very helpful for developers working in transit. Third-party developers seem to prefer Android development resources because of their outstanding organization and amount of time needed to search for them, which is acceptable, meaning less time. One example is the Apidemos sample application, which is a relevant starting point for developers seeking resources and APIs usage.

Apple´s NDA policy seems to luck from open sources for iPhone, however; Android seems to have an advantage over that fact in my point of view because of the possibility to download open source projects for Android architecture and APIs usage.

4.6.1 API (Application Programing Interface)

API is a structured way and a set of routines, tools, and protocols to access data and functionality from different systems. It can be a contract between those who own the data and those who want to use this data. APIs are used by developers to help them to build mobile applications for iOS and Android platforms.
4.6.2 SDK (Software Development Kit)

SDK is a set of tools that are supposed to make it easier for developers writing applications for iOS/Android as well as iOS/Android platform itself. SDK is often accompanied by nondisclosure agreements and various other obligations, and often have a high price. Nevertheless, Apple’s iOS SDK is available for free for those have membership at apple developer center. In contrast, Android’s SDK are available for free for any developers who want to download it, and willing to learn it.

4.7 Tools

Based on the collected data from the interviews, this section summarizes different tools used in iOS and Android development such as: IDE, UI builder, debugger, and profiler.

4.7.1 IDE (Integrated Development Environment)

With every Eclipse installation for Android development, JDT tools are stocked in it as well as standard to it. Third-party developers for Android have preference for its IDE because of its abundance model of the source code.

Third-party developers seem to favor Android JDT because of its incremental compiler, which allows for instant feedback providing errors and warning as you develop. This means that as a developer is typing he or she can have immediate feedback on the committed mistakes in his or her development, allowing for changes to occur before finishing, which allows in a way flexibility in development and typing. This advantage allows for no waiting for compiler, which in many cases breaks the programing flow. The following are some of the main features making Eclipse favorable to work with: 1) organize imports, 2) refactoring, 3) content assistance, 4) fast-fixes.

Finally, Eclipse seem to have many advantages because of the fact that all concepts and methods are instantly available as developers are writing codes, in addition to availability of their documentation. Hence, if developers need to learn an unknown API, the content assist and integrated javadoc are one of the best ways to do it.

“Figure 8” shows the content assist with integrated javadoc in Eclipse IDE
Some of the developers have agreed on the fact that Xcode is not an efficient development tool, and tried to offer recommendations on how to make Xcode a trusted IDE. In my point of view and through my experience the following is a list of what I believe need to change:

1- Changing the content assist for Xcode
2- An effective window/editor management system, allowing for managing multi- windows at once
3- Being able to sort files in an alphabetical order in a project tree view
4- An integrated API documentation with IDE

4.7.2 UI Builder

Mobile application developers have some preferences concerning the UI builder that seems to defer from iOS to Android. Concerning iOS, it has been commonly argued that it has better qualities and is easy to work with, since it can model outstanding UIs making it in a way flexible to their needs. On Android, one developer said “it seems to have some difficulties because of the fact that: 1) lack of efficiency on the UIs due to displaying them in their basic way, and how they appear 2) need to code directly UIs in XML source view of the UI builder.” However, the content validation and assist are good enough to build UIs.

4.7.3 Debugger

The developers of the company using the Java debugger noticed a difference between the Eclipse and the Xcode IDEs, since in Eclipse rather than in Xcode, these developers can see as well as modify variable values. This affects the developers debugger usability, because of native code affairs state while debugging.

Concerning Xcode, the developers of the company agreed that it get confusion to the types of an object, presenting them with just pointer value and no explicit
details. However, in Eclipse, developers can simply drill down via object’s graph in a friendly user manner.

Finally, in Xcode, debugger UIs is not easy to use, since many windows are opened when needing to show code by clicking on stack. Besides, the watch does not work quite often.

### 4.7.4 Profiler and Heap Analysis

One iOS developer said “*One of the areas in which Apple development tools seem to excel is using profiling and heap analysis, such tools are agreed by developers to be mature enough and friendly to use.*” With this kind of tools a developer does not need a lot of experience in order to have good understanding of his application in a short time, by finding and fixing many memories leaks and improving performance.

“*Figure 9*” shows the tool used in Xcode for memory detection leak.

![Figure 9: Xcode memory detection leak](image)

One Android developer said “*Most of Android developers need to use traceview application for Android, which himself agreed in the fact that it does require a lot of efforts in order to operate and configure. Besides, it seems also that source-coding need to be altered to have the needed trace files for analysis.*”
Finally, the developers of the company are quite doubtful about the fact that Android is able to give heap dumps in “hprof format” and in case if it does analyzing heap usage can be easily done via MAT tool.

4.8 Development Process Model

Based on the collected data from the interviews, they have been using Scrum as development model or a modified version of it. Something that all interviewees have touched on is the importance of an iterative development process and collection of feedback, especially from the end-user as the app is meant for.

One Android developer said “we as developers seek to use Scrum to the development process because we believe it is a modern and agile ways of doing software development, but in some cases we have to adapt to customer's development processes and other requirements and deadlines which render less suitable Scrum.”

Interviewees also believe that the best results are obtained when the members of project teams may have specialized roles, such as a specialist for each mobile platform, but that the members has to contend with an understanding of others' actions. Otherwise, be app development projects are usually in the same way as any other software development projects at any time. On the other hand, a slight difference from ordinary software development that was presented, however; the close collaboration between graphic designers, interaction designers and developers: because these roles are so dependent on each other need a close and iterative collaboration take place between them to allow work to proceed.

It is also important to continue dealing with an app after it published. New versions of iOS and Android and new devices make the apps under maintaining to work well. It is therefore impossible to simply abandon the app when it is published. To get feedback and work on it is a part that can be done in the same process with the maintenance of new versions and new devices. Setting up a channel in which end-users can provide simple feedback on the app is recommended. This can be another good reason for the use of Scrum model, where development constantly seen as unpredictable moment.

4.9 Supply Chain

Based on the collected data from the interviews, the supply chain processes, production as well as development and practices of Apple’s iOS and Google’s Android product platforms begin by an application idea or solutions from third-parties using the SDK’s and API’s components to develop the idea, which is
followed by handing the developers application to the associated personnel platform’s in order to be reviewed.

One iOS developer said “if the application is being agreed on it, will be than available to be downloaded and purchased by iPhone’s consumers via the App Store. In case if it is rejected the developers will have the opportunity to modify it and redesign it and be able to reenter the process.”

4.10 Submission Apps

Based on the data collected from the interviews, submission is the process that makes developers’ iOS and Android apps available to consumers. Cognizant to this, different procedures and policies undertaken by Apple and Google to publish the apps in their stores. However, these procedures will be discussed in sections 4.9.1 and 4.9.2.

4.10.1 Submission Apps on Android Platform

This Section summarizes how Android applications can be published to end-users. Submission apps to Google Play involves two main tasks:

- The developer prepares the app for release
- The developer releases the app to end-user

These tasks are discussed in details in sections 4.9.1.1 and 4.9.1.2.

4.10.1.1 Preparation the App for Release

During the preparation task the developer build a release version of the app, which end-users can download and install on their Android devices. This task is divided in five main tasks as follows:

- Gathering materials for release which includes End-user License Agreement (EULA), cryptographic keys, and application icon
  - *EULA*: Considering a EULA helps the developer to protect his/her person, organization, and intellectual property, it is recommended to provide one with developer’s app
  - *Cryptographic keys*: The Android system requires a digital certificate for any installed app to be signed that is owned by app’s developer. Each certificate has a private key which is hold by the developer as a means of identifying the author of an app
Results

- **Application Icon:** Each app must have an application icon and that it fulfill the recommended Android icon guidelines. Recently, Google Play requires a high resolution version for app’s icon

- Configuring application for release which includes removing logging and debugging settings, configure manifest settings, and clean-up project
  - **Removing logging and debugging settings:** The developer has to make sure that a suitable package name has chosen before distribute the app to users. In addition, the developer has to make sure that logging and debugging settings are disabled before build the app for release
  - **Configuring manifest settings:** The developer has to make sure that manifest items such as <uses-permission> element, android:icon attribute, and etc. are set correctly
  - **Cleaning-up project:** The developer has to clean-up his/her project and make sure it meets the directory structure described in Android projects

- Building application for release: After finishing configuring the application the developer can build it into a release-ready .apk, which is optimized and signed. The Android SDK contains the tools for compiling and optimizing the .apk file. Hence, the JDK contains the tools for signing the .apk file

- Preparing remote server production use and updating resources for release: If the app relies on an external server, the developer has to make sure the server is secure as well as that it is configured for production use. In addition, If the app fetches content from a real-time service, the developer has to make sure that content he/she is providing is ready for production and up to date

- Testing Application for release: The developer should at least test his/her app on a one handset-sized device and one tablet-sized device to ensure that app’s UI are sized correctly, app’s performance, and battery usage are acceptable

### 4.10.1.2 Releasing the App to End-User

During the release task the developer publish, sell, and distribute the release version of his/her app to end-users. This is can be done on several ways as follows:

- Releasing app on Google Play
- Releasing app on developer’s website or own server
Releasing app through email: The developer can send his/her release version of app (.apk file) to a user through email.

The illustration in “Figure 10” briefly summarizes the tasks for publishing an app on Android system.

![Diagram](image)

**Figure 10**: Submission process on Android platform

### 4.10.2 Submission Apps to App Store

This section summarizes the steps necessary to submit an iOS application to App Store to be available for end-users. Submission apps to App Store involves two main tasks:

- Prepare for app submission
- App Store approval process

These tasks are discussed in details in 4.9.2.1 and 4.9.2.2 sections.

#### 4.10.2.1 Preparing for App Submission

During the preparation task the developer has to make sure that he/she meets the Apple’s regulations and policies. These are explained as follows:

- Setup user account in iTunes connect: iTunes connect is a suite web-based that provides developers with tools needed to submit and manage their applications for distribution on App Store. In iTunes connect the developer will be able to:
  - Check the status of his/her contracts
  - Setup his/her tax and bank information
Results

- Manage users, metadata (icon art, screen shots, and description), applications, and in-App Purchase catalog
- Request promotional codes
- Obtain financial and payments reports

- Gather information and assets for developer’s application: Besides to the binary file, several attributes should be assigned for the app, including:
  - App name
  - App description
  - App URL
  - App rating
  - Primary and secondary catalog
  - Subcategories
  - End User License Agreement
  - Copyright
  - Keywords
  - SKU number
  - Screen shots
  - Pricing
  - Available date
  - Support URL
  - Support email address
  - Large app icon 512 x 512 for iPhone apps
  - Large app icons 1024 x 768, 768 x 1024, 1024 x 768, 1004 x 768, or 768 x 1004 for iPad apps
  - Small app icons 72 x 72 and 50 x 50 for iPad apps

In-App Purchase offers the customers to make purchase within developer’s application. A paid application contract in effect with Apple should be assigned by the developer.


4.10.2.2 App Store Approval Process

During the app approval process Apple ensures that apps are follow the design, technical, and content specifications in order to protect consumers from explicit and offensive content, protect consumer privacy, and avoid apps that abase the core experiences of iPhone, iPad, iPod touch. Once a developer has submitted an app to Apple for approval, several processes are applied by Apple:

1. Prepare for Upload
   Appears as the first status for an app. The developer should enter metadata, screen shots, etc., to prepare his/her app for upload to the App Store.

2. Waiting for Upload
   Appears once the developer has completed entering his/her metadata and ready to submit the metadata. However, the developer has not finished uploading the metadata through Xcode’s organizer. The app must be Waiting for Upload for the developer to be able to deliver the metadata through Xcode’s organizer.

3. Upload Received
   Appears when the metadata has been received through Xcode’s organizer.

4. Waiting for Review
   Appears once the developer has completed submitted a new app or update to Apple. This means that the app has been added to the app review queue, but not yet begun the review process.

5. In Review
   Appears when Apple is now reviewing the app prior to the app being rejected or approved. It takes on average 5-10 business days, however, the process may take longer time. This estimated period based on past experience with Apple’s review process.

6. Processing for App Store
   Appears when the app being processed and will be Ready for Sale within 24 hours.

7. Pending Developer Release
   Appears once the app has been approved by Apple and the developer has chosen to set his/her Version Release Control. Release to the App Store when the developer is Ready.

8. Ready for Sale
Appears once the app has been approved and posted to the App Store.

In case of rejection by Apple, third-party developers will receive a communication from App review in the Resolution Center regarding the reason for rejection. For this reason, many developers have faced problems with Apple during the approval process regarding the rejection, and the delay in the review process. So, an experienced developer will minimize the chances for dealing with rejections.

The illustration in “Figure 11” briefly summarizes the App Store approval process.

Figure 11: App Store Approval Process
5 Discussion

Data collected from the literature and professionals in mobile applications development, is subjected to qualitative analysis. In order to simplify the discussion and to obtain effective results, this chapter is divided into sections 5.1 and 5.2.

5.1 Third-Party Developers

In relation to third-party developers, all of them have platform knowledge and extensive code experience. Their publishing experience extends from one app to many apps, with around half of them have already published more than one application to App Store and/or Google Play. Most of the third-party developers tend to opt for one platform development in order to focus their experiences, although some of them tend to work for multiple platforms.

While third-party developers own their product in form of app and the associated intellectual property rights, they do not own the distribution channel, which is key to publishing their artifacts. Third-party developers access to the App Store and Google Play for the sale of apps is opened up by Apple and Google, and, while this is viewed definitely by many developers, it is completely with governance of Apple and Google.

Mass consumers/end-users are the primary audience of apps and services. Hence, third-party developers offer free or paid apps, with free apps more often receiving largely downloads. Paid apps conform defined price structures but are mostly low cost.

Third-party developers widely agreed that interaction with consumers could be beneficial for enhancing upgrades and future products and services. They lack such a sufficient dialogue with consumers. Besides, consumers’ reviews impact the rating system which influences potential downloads.

The integrated development environment and other tools such as SDKs, APIs, debuggers, and simulators provided to write, test, and deploy apps on the target platform, which are hosted by Apple and Google to enable third-party developers to work on their platforms.

On Android platform, the tools used are superior, leveraging Java language and the Eclipse IDE are main winning factors for Android. Apple’s developers’ tools are poor by comparison. The language Objective-C and platform APIs are poorly organized. These factors combined make app development more expensive when developing for iOS, the only area where Apple’s developers tools beat is in profiling and heap analysis.
Apple’s App Store from a consumer’s standpoint and from a worldwide coverage standpoint is superior. In contrast, Android’s Google Play is poor.

Consequently, for iOS platform, many developers started to develop on this platform since it preceded Android platform, and, hence, these developers need a strong rationale to persuade them to switch. In contrast, for Android platform, many developers are drawn to app development because of the open source aspect, which makes it easy for them to share and reuse code. Third-party developers choose Google’s Android over Apple’s iOS for some reasons or factors, and they are:

- iOS is too controlled with many conformance rules
- Mac computer is essential for app development
- May not always developing an app leads to approval for the App Store

In contrast, Android platform does not offer the smooth ecosystem that Apple can offer, whereby the iPhone, the iTunes connect, the App Store seamlessly integrate.

Finally, Mobile application developers not only focus on coding, interaction design, graphics, and animations, but as well as on customer care activities and idea generation.

### 5.2 App Development Process

Based on the results, mobile application development takes longer time than other development arenas, and this is not just in terms of writing code. Third-party developers should follow a framework or a development process when developing on iOS platform or Android platform. Apple and Google have similar processes and concepts such as SDK, APIs, guidelines, simulation, UI design and development, s-w design and development, application certificate, mobile device testing, and submission to marketplace. Apple differs from Google in terms of review process, which Google doesn’t have for Android platform, and this means that Apple has more control over its platform than Google on its Android.

The illustration in “Figure 12” explains the development process for Apple’s iOS and Google’s Android platforms.

As a result, Scrum perfectly fits this essential ingredient of iteration that is inherent in the app development on iOS and Google platforms.
Figure 12: App development process
6 Conclusion

The development for mobile platforms adds new layers and processes such as ecosystem, programing language, platform, programing model, tools, development process model, supply chain, and submission apps, which have been identified and explained to give the reader a clear image on how this new phenomena is evolving and to understand how third-party developers adopt into it.

Denominator from both a technical and an organizational perspective and third-party developers use as a quick source of expertise. The existing development organization needs to adapt to app development by making use of more agile methodologies. By making use of expertise from an outside supplier, their own experience of the industry and software development as well as a feedback from end users creates a good basis for meeting the challenges that exist within the app development.

Scrum is a highly agile development model that assumes that software development is done in an unpredictable and changing environment. This description has been shown to agree well with the current iOS and Android environment, where change occurs rapidly and in a manner difficult to predict. The theory that revolves around a Scrum development process emphasizes that creativity is enhanced if the developers can try things out and experiment instead of working for normative processes and predefined requirements.

Mobile applications become successful after more experimentation and adaptation. The process of refining and getting to what consumers like occurs every day, as apps are in the hands of consumers who post comments and ratings on the App Store and/or Google Play, and apps themselves. Scrum perfectly fits this essential ingredient of iteration that is inherent in the development of mobile applications.

The success of the digital ecosystems and platform knowledge adopted by Apple and Google has gained a lot of attention from developers in the last years concerning mobile application development. This, in turn, was the key factor of their success at the smartphones market and the portal for developers from all the world.

With its iPhone, iPad, and iPod touch, Apple has created a successful products family that facilitated their entry into the crowded smartphones market. At the same time, it has changed the conception of the smartphones, demonstrated the significance of smartphones in driving demand and need for new mobile services. This, in turn, led to digital ecosystem where Apple has innovated.
With its open platform Android, Google has created a wide range of smartphones and tablets that run Android, and that is made Android the market leader for smartphones in terms of market share and number of third-party developers by which developing applications in Android platform. In relation to third-party developers, writing an application for Apple will work on only three devices iPhone, iPad, and iPod touch. Writing an application for Android has the ability to work on hundred devices.

6.1 Scope for Future Studies

Although this study has highlighted the development process, practices, and factors by which third-party developers adopt when developing an app for iOS and Android platforms, this study was intended to come up with some insights into this area of research which could be improved and expanded upon in future work.

This study can pave the way for future work to employ this work to cover Microsoft Windows Phone 7 and RIM platforms, and how these platforms can compete Apple’s iOS and Google’s Android in the future.

On a more holistic plan may also be of interest to study the app from a lifecycle perspective, given the brisk technological taking place in this field. The apps and even the mobile units’ rapid pace of development should be compared to traditional software development projects that develop over long periods.
7 References


References

Bergvall-Kåreborn, Birgitta; Howcroft, Debra; and Chincholle, Didier (2010), "Outsourcing Creative Work: a Study of Mobile Application Development", ICIS 2010 Proceedings


References


Golafshani, Nahid (2003), "Understanding Reliability and Validity in Qualitative Research", The Qualitative Report, 8 (4), 597-607


References


West, Joel & Mace, Michael (2010), “Browsing as the killer app: Explaining the rapid success of Apple’s iPhone”, Telecommunications Policy, Vol. 34

8 Appendix

8.1 Interviews Questions
This is a qualitative study, semi-structured interviews were carried out. The questions were divided into two groups, and they are explained in sections 7.2 and 7.3.

8.2 About the Company
- Tell us about the company.
- What is your role at the company?
- How many development projects have you taken in total?
  - Why do you use these techniques?
  - What projects are you working on right now?

8.3 The Development Process
- What do you use for processes / methods in development?
  - Why do you these techniques?
  - How did the development process from start to finish this for an example?
- What is your strategy for fully developed apps?
  - How are the rights? (Who owns the code?)
  - How much work is it to register on the App Store and/or Google Play as a company?
  - How is the handover of the product?
- What do you think that different platforms pros and cons?
- Which tools do use for the development of an app for Apple’s iOS and Google’ Android platforms?
- What platforms have you developed apps to?