A Mobile Application Development Strategy-Finding Model

Ghislain Edgard MBAYEN MBAYEN

MASTER THESIS 2013
INFORMATICS
A Mobile Application Development Strategy-Finding Model

Ghislain Edgard MBAYEN MBAYEN

Detta examensarbete är utfört vid Tekniska Högskolan i Jönköping inom ämnesområdet informatik. Arbetet är ett led i masterutbildningen med inriktning informationsteknik och management. Författarna svarar själva för framförda åsikter, slutsatser och resultat.

Handledare:
Examinator:

Omfattning: 30 hp (D-nivå)

Datum:

Arkiveringsnummer:

<table>
<thead>
<tr>
<th>Postadress:</th>
<th>Besöksadress:</th>
<th>Telefon:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box 1026</td>
<td>Gjuterigatan 5</td>
<td>036-10 10 00 (vx)</td>
</tr>
</tbody>
</table>

551 11 Jönköping
Abstract

Nowadays there are several different types of ways to build a mobile application, through web technologies, or through mobile manufacturers’ languages. Thus, in this study we focus on the three major mobile application development strategies: Native, Hybrid, and Web to Native. This plurality of solutions renders the selection of a mobile application development strategy complex. Hence, this study aims at understanding how developers deal with the selection of a mobile development approach. This thesis analyses their behaviour thanks to a survey that collects crucial information about developers searching habits. The latter analysis concludes that general guidelines are not sufficient to provide a tailored and accurate selection of mobile application development approach. Therefore, this study aims at improving the activity of finding a mobile application development strategy. In addition to this, the past and current states of the subject are discussed in a critical manner, which therefore results in using an analytical research approach. Knowing the aforementioned, a design science research approach is used to outline the iteration processes that occurred during this study. Each step of the study is thoroughly treated thanks to a tailored design cycle.

Consequently, a mobile development strategy-finding model has been built by combining literature review and survey’s results. This model highlights the importance of the criteria extracted from the literature review and the survey. To support and allow constant self-evolution of the model, a suggestion of web platform is proposed.

Finally, this model has been compared with a Titanium model. The result of this comparison outlines that the model created in this study seems more usable because it comprises the three major mobile application development strategies whereas the Titanium model only focuses in native solutions and its own hybrid solution. Moreover, the Titanium model focuses on selling its product rather than showing why it is better than the other mobile application development strategies.
Acknowledgements

I would like to thank my supervisor Anders Carstensen for being patient towards my work during this long process of thesis writing and also for giving me thorough comments on how to improve my report. I would like to acknowledge the effort of the Course Manager of the Master Vladimir Tarasov through the different phases of my thesis investigation. In addition, I would like to express my gratitude to all the companies and mobile professionals who accepted to respond to the online questionnaire. Last but not least, I would like to outline my appreciation towards the immeasurable support of my family throughout the entire thesis.
Key words

Mobile; application; development; native; hybrid; web; platform; Guidelines; Analysis;
# Contents

## 1 INTRODUCTION .................................................................................................................. 9
1.1 BACKGROUND.................................................................................................................. 10
1.2 MOTIVATION.................................................................................................................... 10
1.3 PURPOSE & RESEARCH QUESTIONS.............................................................................. 11
1.4 DELIMITATIONS.............................................................................................................. 11
1.5 THESIS OUTLINE............................................................................................................ 12
1.6 KEY DEFINITIONS........................................................................................................... 12

## 2 THEORETICAL BACKGROUND.......................................................................................... 15
2.1 NATIVE SOLUTIONS.......................................................................................................... 15
   2.1.1 iOS ............................................................................................................................ 15
   2.1.2 Android .................................................................................................................... 17
   2.1.3 Windows Phone........................................................................................................ 19
   2.1.4 Others...................................................................................................................... 21
2.2 HYBRID SOLUTIONS - CROSS-PLATFORM NATIVE SOLUTIONS.................................. 22
   2.2.1 Titanium.................................................................................................................. 22
   2.2.2 Phonegap................................................................................................................ 23
   2.2.3 Xamarin.................................................................................................................... 24
   2.2.4 Others...................................................................................................................... 24
2.3 WEB TO NATIVE SOLUTIONS.......................................................................................... 25
   2.3.1 Seattleclouds............................................................................................................. 25
   2.3.2 RareWire.................................................................................................................. 25
   2.3.3 iBuildApp................................................................................................................ 26
2.4 COMPARISON BETWEEN ALL SOLUTIONS STATED..................................................... 26

## 3 METHODOLOGY.................................................................................................................. 27
3.1 RESEARCH FRAMEWORK................................................................................................. 27
3.2 RESEARCH DESIGN......................................................................................................... 29
3.3 RESEARCH APPROACH.................................................................................................... 30
3.4 DATA COLLECTION.......................................................................................................... 31
   3.4.1 Primary data.............................................................................................................. 31
   3.4.2 Secondary data......................................................................................................... 32
3.5 RESEARCH ANALYSIS..................................................................................................... 32
3.6 RESEARCH CREDIBILITY............................................................................................... 33
   3.6.1 Reliability................................................................................................................ 33
   3.6.2 Validity..................................................................................................................... 33
   3.6.3 Generalization......................................................................................................... 33

## 4 PREVIOUS RESEARCH ON THIS SUBJECT..................................................................... 35
4.1 PAPERS IN FAVOUR OF NATIVE SOLUTIONS................................................................. 35
   4.1.1 Native vs. web app development by Abalta Technologies........................................ 35
4.1.2 Mobile application development web vs. native......................................................... 35
4.2 PAPERS IN FAVOUR OF MOBILE WEB SOLUTIONS....................................................... 36
   4.2.1 Mobile web strategy for enterprise success.............................................................. 36
   4.2.2 MobiCloud - Making clouds reachable: A toolkit for easy and efficient development of customized cloud mobile hybrid application........................................... 36
4.3 PAPERS PROVIDING GUIDELINES.................................................................................. 36
   4.3.1 Mobile application development............................................................................. 36
   4.3.2 Native mobile apps: The wrong choice for business............................................... 37
8 CONCLUSION ........................................................................................................................................71
  8.1 Recapitulation of Research Questions ......................................................................................71
  8.2 Further Investigation .................................................................................................................72
9 REFERENCES .....................................................................................................................................73
10 APPENDIX .......................................................................................................................................78
  10.1 Survey Results (1/3) ..................................................................................................................78
  10.2 Survey Results (2/3) ..................................................................................................................79
  10.3 Survey Results (3/3) ..................................................................................................................80
List of Figures

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Android Platform Architecture</td>
</tr>
<tr>
<td>2</td>
<td>Set of APIs included in Windows Phone API</td>
</tr>
<tr>
<td>3</td>
<td>Smartphone Operating Systems, Shipments, and Market Share, 1Q 2013 (Units in Millions)</td>
</tr>
<tr>
<td>4</td>
<td>API available for different types of devices</td>
</tr>
<tr>
<td>5</td>
<td>Comparison chart between all solutions</td>
</tr>
<tr>
<td>6</td>
<td>The framework utilized for this study</td>
</tr>
<tr>
<td>7</td>
<td>The general methodology of design science research</td>
</tr>
<tr>
<td>8</td>
<td>Chart illustrating the comparison between web, hybrid and native</td>
</tr>
<tr>
<td>9</td>
<td>Mobile application framework selection criteria (1/2)</td>
</tr>
<tr>
<td>10</td>
<td>Mobile application framework selection criteria (2/2)</td>
</tr>
<tr>
<td>11</td>
<td>Overview guideline table</td>
</tr>
<tr>
<td>12</td>
<td>Illustration of the survey’s first question</td>
</tr>
<tr>
<td>13</td>
<td>Illustration of the survey’s second question</td>
</tr>
<tr>
<td>14</td>
<td>Illustration of the survey’s third question</td>
</tr>
<tr>
<td>15</td>
<td>Illustration of the survey’s fourth question</td>
</tr>
<tr>
<td>16</td>
<td>Illustration of the survey’s fifth question</td>
</tr>
<tr>
<td>17</td>
<td>Titanium Web Maturity Model</td>
</tr>
<tr>
<td>18</td>
<td>Mobile Application Development Strategy-Finding Model</td>
</tr>
<tr>
<td>19</td>
<td>Decisional schema for choosing a mobile development strategy</td>
</tr>
<tr>
<td>20</td>
<td>MADSF Model Algorithm</td>
</tr>
<tr>
<td>21</td>
<td>First step of the concept of stepstoapp.com</td>
</tr>
<tr>
<td>22</td>
<td>Second step of the concept of stepstoapp.com</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>App</td>
<td>Application</td>
</tr>
<tr>
<td>Apps</td>
<td>Applications</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>WP</td>
<td>Windows Phone</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>BB</td>
<td>BlackBerry</td>
</tr>
<tr>
<td>MVC</td>
<td>Model View Controller</td>
</tr>
<tr>
<td>CPNS</td>
<td>Cross-platform native solutions</td>
</tr>
<tr>
<td>ICT</td>
<td>Information and Communication Technology</td>
</tr>
<tr>
<td>MADS</td>
<td>Mobile Application Development Strategy</td>
</tr>
<tr>
<td>MADSF</td>
<td>Mobile Application Development Strategy-Finding</td>
</tr>
</tbody>
</table>
I Introduction

This chapter aims at providing the scope of the report through several sections including the establishment of the background of mobile application development, the motivation for this research, as well as the purpose and research objectives. It also focuses on the limitations of the report, the thesis outline and keyword of mobile application development.

Nowadays mobile phone are assimilated to personal computer except that they are smaller. This type of phone is commonly called smartphones. Smart is used to illustrate the intelligence of such tiny devices and to remind the calculation power embedded into it. Several constructors of phone jumped into this market after the revolution of the iPhone in 2007. With their venue many of them created their own platform of applications release, which sometimes allow developers to build applications with a specific programming language.

After a couple of years of evolution, new techniques of mobile application development were unleashed from the clouds. Indeed, hybrid solutions such as phonegap or Titanium help you to encapsulate your application into a native application by using respectively web technologies such as HTML, JavaScript, CSS or a custom JavaScript SDK. Recently another type of solution has been launched thanks to great efforts from RareWire and others, which utilizes script technology python, or ruby and an API in between to ease the creation of mobile application via a web based interface.

Previous research on the subject resulted in establishing guidelines, and minimal frameworks for mobile professionals. Our survey revealed that these guidelines are seldom used. Knowing the aforementioned, this study focuses on establishing a new model allowing developers and IT professionals to rely on to find the most suitable mobile application development strategy for their mobile applications projects. This model is called M ADSF, which stands for Mobile Application Development Strategy Finding.

This model is based on the design science research approach due to its problem-solving nature. This approach is subject to improve activities by following a design cycle. The design science research approach is composed of five process steps: awareness of problem, suggestion, development, evaluation, and conclusion. These steps are geared together with circumscriptions knowledge flows. The analysis of the literature review facilitates the precision of the problem aforementioned, which fulfils the awareness problem process step. The suggestion phase of the design science research is based on the assumption stated at the end of chapter 4. The development step is represented by the online survey. The evaluation is identified by the analysis of the survey’s results and the explanation of the new model established called M ADSF model. Finally, the conclusion is drawn to confirm that the research questions mentioned in section 1.3 are answered.

From now on the words application and applications will be, respectively, abbreviated to app and apps.
In order to settle the base of this study we shall trace the history of mobile development from 2007, year of the first smartphone with apps, until now.

1.1 Background

The year 2007 was a glorious period in terms of innovation. Indeed, the iPhone first generation was launched. It was the revolution of the telecommunication and multimedia sector. Thanks to its new sharp design, and simplicity of utilisation, the iPhone reached historical sales record with 1,389,000 devices sold in four quarters [1].

Apple was not the only one with a revolutionary handset operating system. In fact, Google revealed the first Android phone in October 2008. By that time Apple already had a major advantage due to the earlier launch of their smartphone a year and half before. Other competitors followed the smartphone wave after a couple of years for instance windows unveiled its windows phone in October 2010 [2].

The real game changer for all those manufacturers was the opening of their application stores named App Store for Apple iPhone, Google Play for Android, Windows Phone Store for Windows Phone. These application stores allow developers to create apps and provide them on the application store. For a yearly subscription fee developers can access to the platform SDK and publish as much apps as they want after the constructor review of the apps [1][2][3].

However, developers rapidly reached programming language development obstacles. Since constructors used different programming languages developers had to decide on which platform they would focus themselves and improve their skills. Code reuse became a major concern for developers. Therefore, groups of developers working on projects to tackle these drawbacks emerged such as the phonegap project, which started in 2009 [4].

After a couple of years many other solutions were created starting from Hybrid mobile development solutions and lately web app that enables users to build mobile applications by dragging and dropping widgets. Nowadays, developers have a wide range of solutions at the tip of their fingers even though sometimes it can be problematic to choose within the plurality of solutions [5].

1.2 Motivation

Following a work experience in mobile application development, I realized that the work of preparing a mobile application development was time consuming unless being an expert in a dedicated mobile development solution. My work conducted me to choose between several solutions, which took me several days because I had to search beforehand the feasibility of some features depending on each platform. Hence, these difficulties resulted in an idea of creating of model that would render the selection of a mobile development solution easier. Some papers elaborated a set of solutions described in Chapter 4. Nonetheless, none of them appeared to aim at providing a model that combines the three major mobile application development strategies. This being said, there is a Titanium model described in Chapter 6 that
proposes an approach but it is solely focused on native and its own hybrid mobile application development strategy. Consequently, there is a theoretical gap that needs to be fulfilled by the establishment of a model that encompasses the major three mobile application development strategies.

1.3 Purpose & Research questions

The evolution in terms of mobile development solutions has brought several options for developers. Nonetheless, the downside is that new developers have to choose between wide ranges of solutions. Hence, this study will aim to investigate and provide solution to the following statements:

- What processes developers perform for choosing a mobile development architecture?
- What are the criteria for which developers adopt a mobile application development strategy?
- How to provide developers and companies with a model that help them to choose their mobile development strategy?

In order to support this study a survey was carried out to collect accurate information about the mobile development trends from a professional point of view.

Based on the analysis of this survey’s results a new model of Mobile Application Development Strategy Finding (MADSF) has been proposed and thoroughly explained.

1.4 Delimitations

This study shall face internal and external delimitations. In fact, this study will be focused onto three platforms iOS, Android and Windows Phone, which represent 95.5% of the smartphone operating system market share [6] of the first quarter of 2013. Hence, the web site will not cover the other OS because their growth is incomparable to the three giants aforementioned and the forecast are not promising for them.

The second delimitation will concern external difficulties through enterprises barriers. The challenge of the study will be to collect precious data about the current trend in terms of mobile application development solutions used for different types of application. Indeed, mobile application development agencies might not be willing to share this kind of information might be proprietary whether for them or their clients. They might be busy with their daily workload, which will surely postpone their answer.
1.5 Thesis outline

The following section of this research report are structured as below:

The second chapter contours the history and theoretical background of the three giant manufacturers named earlier.

The third chapter provides information about the methodology used for this study. It also explains the research method, the theoretical framework and design employed for in this project. Moreover, it denotes the methods and tools utilized for data collection. Thereby, it supports the purpose of this research.

The fourth chapter depicts the literature review of the research carried out in order to bring a sight into the theoretical background of mobile application development.

The fifth chapter gathers the results of the online survey. Therefore, elaborates on possible combination of factors and analyses the outcome.

The sixth chapter explains the new concept of the M ADSF (Mobile Application Development Strategy-Finding) model in comparison with a current model built by Titanium.

The seventh chapter discusses about the entire thesis study and especially about the reliability and validity of the survey’s results.

Finally, this study is concluded by a summary of the research, deduces the essential results of the research and outlines what could be done in further investigations.

1.6 Key definitions

In order to grasp the subsequent chapters of this study, this section provides a set of definitions of frequent technical terms appearing in this report. Consequently, these key definitions bring an overall understanding and necessary knowledge of the Mobile Application domain to avoid repetition. Therefore, the table 1 below encloses the aforementioned key definitions.

<table>
<thead>
<tr>
<th>Terminology</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile Application</td>
<td>“A mobile application (or mobile app) is a program, embedded into a mobile device, that performs a specific purpose. They are distributed through software platforms called Application Store, such as the Apple App Store, Google Play, Windows Phone Store, and BlackBerry App World.” [7]</td>
</tr>
<tr>
<td>SDK</td>
<td>“A software developer's kit (SDK) is a set of programs used by a computer programmer to write application programs. Typically, an SDK includes a visual screen builder, an editor, a compiler, a linker, and sometimes other facilities.” [7]</td>
</tr>
</tbody>
</table>
| Framework     | “In computer systems, a framework is often a layered
structure indicating what kind of programs can or should be built and how they would interrelate. Some computer system frameworks also include actual programs, specify programming interfaces, or offer programming tools for using the frameworks.” [7]

<table>
<thead>
<tr>
<th>Platform</th>
<th>“A platform consists of an operating system, the computer system's coordinating program, which in turn is built on the instruction set for a processor or microprocessor, the hardware that performs logic operations and manages data movement in the computer.” [7]</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>“An application program interface (API - and sometimes spelled application programming interface) is the specific method prescribed by a computer operating system or by an application program by which a programmer writing an application program can make requests of the operating system or another application.” [7]</td>
</tr>
<tr>
<td>JavaScript</td>
<td>“JavaScript is an interpreted programming or script language from Netscape. It is somewhat similar in capability to Microsoft's Visual Basic, Sun's Tcl, the UNIX-derived Perl, and IBM's REXX.” [7]</td>
</tr>
<tr>
<td>C#</td>
<td>“C# (pronounced &quot;C-sharp&quot;) is an object-oriented programming language from Microsoft that aims to combine the computing power of C++ with the programming ease of Visual Basic. C# is based on C++ and contains features similar to those of Java.” [7]</td>
</tr>
<tr>
<td>Objective-C</td>
<td>“Objective-C is the primary programming language you use when writing software for OS X and iOS. It’s a superset of the C programming language and provides object-oriented capabilities and a dynamic runtime.” [8]</td>
</tr>
<tr>
<td>Java</td>
<td>“Java is a programming language expressly designed for use in the distributed environment of the Internet. It was designed to have the &quot;look and feel&quot; of the C++ language, but it is simpler to use than C++ and enforces an object-oriented programming model.” [7]</td>
</tr>
<tr>
<td>Silverlight</td>
<td>“A programming model for developing and distributing rich Internet applications (RIA) that use graphics, animations or video within the .NET framework. Silverlight was previously known by its code name, Windows Presentation Foundation Everywhere.” [7]</td>
</tr>
<tr>
<td>iOS</td>
<td>“iOS is Apple's proprietary mobile operating system (OS) for its handheld devices, such as the iPhone, iPad and iPod Touch. The operating system is based on the Macintosh OS X.” [7]</td>
</tr>
<tr>
<td>Android</td>
<td>“Android OS is a Linux-based platform for mobile phones. Android was released under the Apache v2 open source license.” [7]</td>
</tr>
<tr>
<td>Windows Phone OS</td>
<td>“Windows Phone OS is a Microsoft operating system for smartphones.” [7]</td>
</tr>
</tbody>
</table>
| Eclipse | “Eclipse is a Java-based open source platform that allows a
Introduction

software developer to create a customized development environment (IDE) from plug-in components built by Eclipse members. Eclipse is managed and directed by the Eclipse.org Consortium.” [7]

<table>
<thead>
<tr>
<th><strong>Xcode</strong></th>
<th>“Xcode is an integrated development environment (IDE) containing a suite of software development tools developed by Apple for developing software for OS X and iOS.” [9]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Visual Studio</strong></td>
<td>“Visual Studio is a suite of component-based software development tools and other technologies for building powerful, high-performance applications. In addition, Visual Studio is optimized for team-based design, development, and deployment using Team Foundation Service or Team Foundation Server.” [10]</td>
</tr>
<tr>
<td><strong>Bada</strong></td>
<td>“Samsung Bada is a smartphone platform, which is created for wide range of device, unveiled in 2010.” [11]</td>
</tr>
<tr>
<td><strong>Symbian</strong></td>
<td>“Symbian is a mobile operating system (OS) targeted at mobile phones that offers a high-level of integration with communication and personal information management (PIM) functionality.” [12]</td>
</tr>
</tbody>
</table>

**Table 1: Key definitions**
2 Theoretical Background

This section of the report grounds the study by providing an overview of previous studies, research papers, and scientific articles. Thereby, the chapter will be structured through three types of solutions, which are native, hybrid and web to native. In other words, this chapter will primarily cover three types of native mobile application development. Followed by a variety of hybrid solutions. To conclude by, the new tendency, which are web to native solutions.

2.1 Native solutions

Native development remains one of the most used solutions among the main three aforementioned. For a question of clarity, only the main platform will be covered, and the others will only be enumerated.

2.1.1 iOS

iOS represents, at the first quarter of 2013, 17.3% [6] of the world market share in terms of smartphones. It has been ranked 2nd behind Android, which has more penetration around the world. However, iOS Operating System is one of the most used and appreciated among thanks to a wide range of high quality products offered at the time being.

2.1.1.1 History

iOS was launched 5 years ago by Apple and designed by Steve Jobs’ team. The iPhone came along with iOS. The revolution of the smartphone was released for the good of the telecommunication industry. In the beginning, iOS was very light and counted only a few set of applications built by Apple.

In the summer of 2008, Steve Jobs announced the released of the app store. This was another major evolution in this new industry. The app store allows any developers to build new apps and publish them under the Apple platform for a yearly subscription fee of $99USD. Apple withdraws 30% of each sold app and the remaining 70% goes to the developer. This downside did not break developers to jump into the new trend of apps development.

Many updates of iOS were released in order to improve the user experience. At the time being, seven versions of the OS have been launched. It has one of the best update devices in the industry with more than 50% of iOS 6.1 [13].

2.1.1.2 SDK

iOS SDK helps developers by providing tools and resources to enhance their development and knowledge about the edge technology. The Software Development Kit comprise the following layers [14]:

Cocoa Touch

This is the first layer and the most important of the four layers included in iOS SDK. It contains theUIKit framework, which represents the User Interface of every
Theoretical Background

application in iOS. It also comprises vital features such as multi tasking or control the multi touch on the device. Therefore, Cocoa Touch is most certainly more than crucial for iOS to be one the best mobile Operating System. Moreover, UIKit permits to dialogue with the graphical interface of iOS and events such as notifications or accessibility. Plus, Cocoa Touch layer provides some key features such as document support, auto layout, printing, or even gestures recognition.

Media
The Media layer allows developers to reach graphics, audio, and video. The Media layer also handles the assets library framework to access photos and video. The Media layer also comprises the Core Image framework that manipulates image through filters, and the Core Graphics framework for 2D drawing. It handles a set of other frameworks such as OpenGL ES and OpenAL usually for games, AV Foundation, and Core Media.

Core Services
The Core Services framework manages key system services used by iOS applications such as block objects, Grand Central Dispatch, In-app Purchase, or even iCloud Storage. ARC (Automatic Reference Counting) best add on to Core Services has been with the release of iOS 5. ARC handles the memory management for the developer through Xcode. This last framework has enhanced the developer experience by reducing the development time. Core Services also dialogues with the Foundation framework, which handles iOS collections and code sharing.

Core OS
The Core OS is the lowest layer in the SDK. It provides numbers of core features and ensures the security between your application and the core functionalities of iOS. However, it allows developers to modify some low level features through the C-based library named libSystem such as POSIX Threads, DNS Services or BSD sockets.

2.1.1.3 Development Tools
This section aims at enumerating a set development tools most commonly used. Among the existing range of tools for iOS development, four were selected. These tools allows a developer to learn Objective-C via Apple Dev Center, Develop applications through Xcode, test applications by using TestFlight and finally monitor the life cycle of the apps via NewRelic website.

Xcode
Xcode is an IDE (Integrated Development Environment) created by Apple for iOS development. It helps developers create, structure, and publish apps easily. It comprise the Apple LLVM (Low Level Virtual Machine) compiler, a debugger, a interface builder (previously separated from Xcode), a versioning management tool, documentations, iOS simulator, Instruments analysis for testing, and the latest SDKs. [9]
TestFlight
TestFlight is a website that allows developers to test applications and monitor builds, feedbacks from early testers. It comprises a SDK that includes the following features [15]: Sessions users management, crash reports, In-App questions for users, Checkpoints to see how far users navigate into apps, Remote logging, and In-App Updates to prompt users to update their app on fly.

NewRelic
NewRelic is a website that permits its users to monitor their app once on the app store. It includes a set of features that enhance the mobile marketing and data tracking. The main functionalities provided by New Relic [16] are: Http Requests tracks which of your services are slowest, Errors, Versions comparison of your app, Device and Carrier allowing the developer to view performances among the diverse mobile ecosystem, Alerts tracks network issues, Geography response time.

Google Analytics can be coupled with NewRelic to enlarge the monitor features and to get real time data.

Apple Dev Center
Apple Developer Center is a great tool for developers thanks to its wide range of development resources. Indeed, it provides a set of iOS resources such as written tutorials, guides, sample code, references, technical notes, forums, Q&A, video tutorials, a web platform of applications administration where developers can create development certificates, distribution certificates, publish their apps and receive royalties from Apple. Hence, Apple Developer Center represents one of the most important tools that an iOS developer should navigate and take advantage from [17].

2.1.2 Android
Android represents, at the first quarter of 2013, 75,0% [6] of the world market share in terms of smartphones. It has been ranked 1st above iOS, which has the best penetration around the world thanks the high number of devices constructors. The biggest in 2013 is Samsung with 30% [18] of all smartphone sold worldwide.

2.1.2.1 History
Android was originally owned by Android Inc firm. The company launched their activity in 2003 and aimed at creating a mobile platform that would be more aware of the user’s preferences and location. In 2005, Google Inc acquired Android Inc, which became a subsidiary. In 2007, Google decided to add the open source characteristic to the Android project by founding a consortium of developers called Open Handset Alliance.

Android has been designed to improve the user experience by being:

- Free: Android is free of charge as well as its SDK and Eclipse IDE is available on every desktop OS
- Complete: it is secure, robust, easily upgradable, developers can fully blend their applications into Android.
- Open: Android has Apache licensing terms, meaning that developers have access to all features of Android and the device they use [19].

This platform has been widely adopted by developers mainly thanks to its freeness, and openness. Moreover the registration fee for the Google play store only represents $25. This last argument is one of the major assets of Android’s platform.

2.1.2.2 Android Platform Architecture

![Android Platform Architecture](image)

Android is designed to combine together several tools and framework to enhanced devices performances. For this very last reason, Android is more a software than an
Operating System. It has multiple layers starting from the bottom level, which is the robust Linux Kernel representing the core drivers of a mobile OS. On an upper level, Android Runtime helps access fundamental framework such as Dalvik Virtual Machine, Zygote, Android Debug Bridge and encapsulate the Platform Libraries. Above the runtime comes Android Framework where all the major services of Android are found. Finally, the last layer is the Applications level, which comprises common applications of a smartphone such as Home, Settings, Contacts, Phone according to [19], and [2].

2.1.2.3 Development tools

This section aims at enumerating a set development tools most commonly used. Among the existing range of tools for Android development, three were selected. TestFlight was already mentioned earlier for the iOS platform. It will also be available for Android developers. TestFlight team is currently working on a similar SDK for Google’s Platform. However, developers can already utilize the build and update features on the website. NewRelic and Google Analytics are also usable for monitoring an Android app.

Android SDK

The Android SDK includes several tools for developers [2]:

- Eclipse + ADT plugin: Eclipse is an IDE that is usually used for Java development. Since Android apps are mainly written in Java this combination was a no brainer.
- ADT (Android Development Tools) plugin brings a set of options and tools to ease the development.
- Android SDK Tools: comprises a set of development and debugging tools
- Android Platform tools: includes a set of Android versions to download as well as devices simulators.
- The latest Android platform
- The latest Android system image for the emulator [20]

Android Developer website

The Android Developer website (developer.android.com) is one step of the developer learning processes. Indeed, it provides a set of useful articles, guides, tutorials, videos, FAQ and forum. All of these assets allow a Android beginner developer to know all about Google’s mobile OS in few days. [20]

Google Market Play

play.google.com website is the platform where developers upload and publish their Android applications. It disposes of several tools to track sales progress and ranking progress of apps. [21]

2.1.3 Windows Phone

Windows Phone OS represents, at the first quarter of 2013, 3.2% [6] of the world market share in terms of smartphones. It has been ranked 3rd below iOS. The biggest
manufacturer integrating Windows Phone OS in 2013 is Nokia with 83.3% [22] of all smartphone sold worldwide.

### 2.1.3.1 History

The Windows Mobile project started in 2004 under the name “Photon”. However, the project team was working slowly, which resulted in a closed project [23]. After the iPhone revolution, in 2008 Microsoft gathered a new team to work on a new kind of smartphone. The task was even more difficult because of the released of iOS a year ago and Android the same year. The group of work had to imagine a new way of interfacing the graphical appearance as well as think about the user friendliness. [24]

As nearly every IT projects, the Windows Phone OS 7 was supposed to be released in 2009, but due to the development of Windows Phone 6.5 the project was delayed and finally released in 2010. [25]

Windows Phone 7 was not compatible with its previous predecessor WP 6.5 because of one major factor “Time”. According to Larry Lieberman senior product manager for Microsoft Mobile Developer experience at that time, Microsoft was attempting to reach end users as well as enterprise users. [26]

In 2011, Microsoft unveiled an update of WP7 called “Mango” or WP7.5. This last version of the OS included multi tasking of third party apps, IE 9, Twitter sharing, and Windows Live Skydrive capabilities. In the beginning of the year 2013, Microsoft launched an update of the OS called WP7.8 with some features from the next generation of the OS. In between, the new generation of OS WP8 was released in October 2012 [27]. It replaced the Windows CE based architecture with the Windows NT Kernel allowing applications to be portable between versions of OS.

Microsoft also built a Zune applications store for end users to enjoy the wide range of apps developed so far. Developers can register and pay between $19/year and $99/year. [28]

### 2.1.3.2 Windows Phone API

Windows Phone API provides resources to improve the development experience and knowledge about the edge technology. The API 8.0 comprise the following layers [29]:

![Figure 2: set of APIs included in Windows Phone API](image)

---

20
As above illustrated, Windows Phone API contains the .NET API, which is the basic programming language and library for WP development. The current API also provides new features such as Microsoft Phone Wallet or ShareMediaTask [29].

Windows Phone runtime is another fundamental API of the packaged Windows Phone API. It allows developers to access core services and features of WP OS. It is implemented in C++, C# and VB.NET.

The Windows Phone API 8.0 allows developers to use Direct3D or C++ for games and others purposes.

2.1.3.3 Development tools
The Software Development Kit of Windows Phone includes several tools that allow a developer to easily learn the platform. WP SDK comprises the following tools [30][3]:

- **Visual Studio Express 2012 for Windows Phone**: is an IDE that allows developers to build WP apps from scratch or with templates. It has a code editor, a visual designer for UI, and toolbox of controls. Last but not least, it includes simulations, monitoring, profiling, and a WP Store Test Kit.

- **Blend for Visual Studio**: is a powerful IDE for UI design purpose. It allows developers to create complex and rich design for their apps.

- **Windows Phone Emulator**: is used for testing purposes. Developers can simulate nearly all options and behavior as if it was a real device.

- **Simulation Dashboard for Windows Phone**: it allows developers to test their app under critical conditions.

- **Windows Phone Application Analysis**: provides an analysis of WP apps to enhance their quality and performance by using monitoring and profiling tools.

- **Windows Phone Store Test Kit**: helps developers to test their apps for the first publication in WP Store by applying a suite of automated and manual tests.

- **TestFlight, NewRelic, and Google Analytics for mobile**: previously detailed in section 2.1.1.3.

2.1.4 Others
This section aims at giving a grasp of the other actors of the smartphone area. The following charts provides an overview of the actual market share of the smartphone industry:
### Android
- Volume: 162.1
- Share: 75.0%
- Year over Year: 79.5%

### iOS
- Volume: 37.4
- Share: 17.3%
- Year over Year: 6.6%

### Windows Phone
- Volume: 7.0
- Share: 3.2%
- Year over Year: 133.3%

### BlackBerry OS
- Volume: 6.3
- Share: 2.9%
- Year over Year: -35.1%

### Linux
- Volume: 2.1
- Share: 1.0%
- Year over Year: -41.7%

### Symbian
- Volume: 0.1
- Share: 0.0%
- Year over Year: -88.5%

### Others
- Volume: 0.0
- Share: 0.0%
- Year over Year: -83.3%

**Total**
- Volume: 216.2
- Share: 100.0%
- Year over Year: 41.6%

---

**Figure 3: Smartphone Operating Systems, Shipments, and Market Share, 1Q 2013 (Units in Millions) [6]**

As illustrated above, the first three OS detain 92.5% of the entire smartphone industry market share. The following OS are a minority among the aforementioned giants mostly because they had a late switch to the touch-apps-based smartphone area. For instance, Linux only introduced a beta version of Ubuntu mobile OS in January 2013, which is 6 years after the pioneer Apple.

### 2.2 Hybrid solutions - Cross-Platform Native Solutions

This section aims at providing an overview of the current methods used for hybrid mobile application development. The order of the solutions is not part of a ranking system.

#### 2.2.1 Titanium

Appcelerator Titanium is an IDE for mobile development. It now includes several types of devices, iOS, Android as much for phones than for tablets. However, it does not comprise Blackberry OS. Although, Appcelerator announced few years ago that they would improve the beta version to release a stable version, nothing happened. Titanium is a free software for developers. It also offers a market place to quickly add module or frameworks usually for a small amount of money ($5 on average) [31].

This version was unveiled in the late 2008. It supports of mobile development for iOS and Android, which came quickly after iOS in 2009. A year after, the tablet version of Titanium was launched for iOS and Android. [32][33]

The great advantage of Titanium is the utilization of existing web development skills for developers. Titanium uses its own API, which is different from already known programming languages. Therefore, developers will use JavaScript but will also have to learn Titanium API in order to take advantage of all the features included in Titanium SDK.
Titanium uses a compiler-like to interpret all application source code such as JavaScript core for iOS, or JavaScript Engine for Android. [34]

With more than 50,000 apps built along with Titanium, this platform SDK is one of the most used in the mobile application development industry. Hundreds of developers adopted it mainly because of its ability to reuse code through platforms and its utilisation of JavaScript [35].

2.2.2 Phonegap

Phonegap is a free and open source framework allowing developers to build mobile applications via HTML5, CSS3 and JavaScript technologies [4]. The project has been acquired by Adobe System in 2011 and originally produced by Nitobi.

Phonegap creates native apps by encapsulating web technologies into a native project. Therefore, developers can use their previous competences without having to learn constructors’ programming languages and SDK. Moreover, it allows cross platform reuse code because the main code is written in HTML5, CSS3 and JavaScript. Nonetheless, some developers in the community reported that a simple copy paste from an iOS working project might not be applicable for Android, at least not without some modification more or less heavy [36].

The major advantage of this mobile development framework is its wide range of API available among several devices. The chart below illustrates the aforementioned information.

![Figure 4: API available for different types of devices [37]](image)

The number of mobile applications built and published with Phonegap is estimated at around 23,000 in 2011 [38].
2.2.3 Xamarin

Xamarin is a relatively new firm. Indeed, the engineers of Mono, and MonoTouch have created it in 2011 [39]. This software intends to improve the developers experience in writing mobile applications for the three giants iOS, Android and Windows Phone. Although, Xamarin is recent more than 330,000 developers have adopted it. Several well known companies have selected Xamarin to build their mobile application such as the famous Rdio app [40]. Xamarin Starter package is free for developers. This package comprises Xamarin Studio software and documentation for further technical support. It also offers a market place to quickly add components or frameworks usually free otherwise for a certain amount of money [41].

Xamarin provides a set of different services and software for fresh developers as well as confirmed developers or organisations.

Xamarin Studio

Xamarin studio is a software allowing developers to build, test, and publish rich mobile application for iOS, Android, and WP. It includes Xamarin iOS, Xamarin mobile, and Xamarin Android. This software comply C# apps, through the .NET runtime, into native ARM executable package for iOS or Android platform.

Developers can achieve the following tasks with Xamarin Studio:

- Create complex design through iOS and Android designer tools included
- Create structured apps depending on the platform
- Code revision via a module/github
- Test apps through devices or emulators
- Deploy apps into apps stores

Xamarin continue to grow with a solid developer community and support.

2.2.4 Others

This section aims at giving a grasp of the other actors in the hybrid mobile application development area. Two others software provides the same level of SDK than the above stated.

Corona SDK

Corona is a Software Development Kit specialised into 2D development applications. It also provides extended features to allow developers to build ebooks and business apps. Corona offers a set of useful tools and resources such as elegant APIs for UI purposes reducing drastically lines of code. Corona provides its own powerful and fast simulator, and allows developers to test on their devices. These features are available for iOS, Android, Kindle Fire thanks to Corona cross-platform asset. Its community is active and daily builds are released by Corona [42].
SenshaTouch
SenshaTouch is a HTML5 mobile application framework for building application on iOS, Android, Windows Phone, Blackberry, and more with web technologies. It provides a built-in MVC system, which structures the entire app and speed up the development. It includes over 50 built-in components, themes. Its community is also very active. SenshaTouch provides sufficient resources and tools to create professional cross platform mobile application [43].

2.3 Web to native solutions
This section aims at providing an overview of the current methods used for web to native mobile application development. Web to native means that the solution is a website that helps at building mobile application. The order of the solutions is not part of a ranking system.

2.3.1 Seattleclouds
Seattleclouds is a website that allows developers to build native applications on iOS, Android and Kindle Fire devices via web technologies such as HTML, CSS and JavaScript. It provides a wide range of templates or generic apps. The customer only has to drag and drop elements in order to add or remove then from their application. Therefore no coding is required for a simple application. It allows to access fundamental features such as GPS, Push notifications, or In-app purchase. In terms of packages, a $9.99 basic package permits a work around on one mobile application with 1 update per month. For further possibilities developers will have to spend at least $29.99 [44].

2.3.2 RareWire
RareWire is a website allowing developers to build native applications on iOS and Android platforms. The app builder is a drag and drop based interface, thus no coding is required. The building of application is seamless and easy to complete without any knowledge in programming. RareWire built an XML based WIRE language to allow people to build apps on browsers. Their Engine takes full advantage of the native device in order to render the app written in WIRE. The online Manager helps versioning, warehouse and deliver media resources to apps. The asset of this solution is the ability for customer to update their app with passing by a provisioning profile (for iOS apps). They can push modification directly from their RareWire account. Another useful feature is the collaboration that helps you effortlessly to invite people to your project for teamwork on an application [45].
2.3.3 iBuildApp

iBuildapp is a website that allows its customers, usually not developers, to build native apps for iOS and Android platform through simple clicks. It provides the same options as Seattleclouds [46].

2.4 Comparison between all solutions stated

This section intends to provide an overview of the features and tools available among the solutions cited earlier. Hence, we can identify and rank all mobile application development solutions aforementioned.

The following chart enumerates the features that each solution posses and rank them according to the number of useful functionalities they provide:

<table>
<thead>
<tr>
<th></th>
<th>iOS SDK</th>
<th>Android SDK</th>
<th>WP SDK</th>
<th>Titanium SDK</th>
<th>Phonegap Framework</th>
<th>Xamarin SDK</th>
<th>Seattleclouds</th>
<th>RareWire</th>
<th>iBuildApp</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS app</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Android app</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>WP app</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>BB app</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Tutorials</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Videos</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Samples</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Templates</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Custom</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Teamwork</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Games 3D</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Games 2D</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Update</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Access device</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>APIs MVC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Price ($)</td>
<td>$99/y</td>
<td>$25/subs</td>
<td>$99/y</td>
<td>Free</td>
<td>Free</td>
<td>One fee</td>
<td>9,99/app</td>
<td>Free</td>
<td>Free</td>
</tr>
<tr>
<td>Rank</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

Figure 5: Comparison chart between all solutions

According to the figure 5 above, Xamarin SDK appears to be the best option in overall. Thanks to its multiple features Xamarin is ranked 1st in our analysis of the mobile application development solutions on the market. Closely followed by Phonegap framework, which count a number of useful features and tools. Followed by the three constructor SDKs because of they are well tailored to their platform. The solutions providing fewer options belong to the web to native category.

Basic knowledge about the mobile application landscape has been established with details for three types of mobile application development. This chapter is essentially to provide a better understanding of the notion of mobile platform and their development environment. The following chapter will encompass the methodology used for this study.
3 Methodology

This chapter outlines the research methodology. “Research methodology is a way to systematically solve the research problem. It may be understood as a science of studying how research is done scientifically. In it we study the various steps that are generally adopted by a researcher in studying his research problem along with the logic behind them” [47].

Hence, this section aims at providing details on the research process and structure of this study. Indeed, it explains the framework utilized as well as the design chosen, the approach taken, the data collection is outlined, and the research analysis based on the data previous retrieved.

3.1 Research Framework

![Diagram of research framework]

Figure 6: The framework utilized for this study

The above illustration uses design science research to build a catered framework for this study. This framework is composed of the following sections:
Methodology

- **Research design**: represents the early stages of the research, where the problem area is fixed, research questions are assessed and different research designs are compared to find the most suitable for this study.
- **Literature review**: helps to build up a thorough understanding of the theoretical background and previous research realized in the area.
- **Survey**: it acts as an argument of authority. In other words, it provides real information about the current trends occurring among mobile professionals.
- **Results & Analysis**: permits to delve into details to highlight similarities, which leads to pointing out the missing artifact.
- **MADS Model**: a Mobile Application Development Strategy Finding is created based on the literature review and survey results. This model is articulated in milestones that lead developers to finding the catered MADS (Mobile Application Development Strategy) for their mobile application project.
- **Conclusion & Discussion**: the framework is assessed by several questions. The study is terminated by a summary, which precedes a further investigation section.

Moreover, this framework comprises iterations called circumscriptions. These iterations allowed this study to be thoroughly conducted and helped to complete all required milestones of a thesis. The following statements describe the circumscriptions:

1. The first iteration represented the preliminary foundation of the thesis, which explained in detail the method utilized to bring the model to life. Since, the theoretical concepts were missing an iteration towards the research design was necessary.

2. The second iteration extracted crucial criteria from the literature review and an adaptation of the survey to cater these criteria. Moreover, the suppression of the explanation of the technical solution, which is a web platform, improved the thesis and permitted to focus the study on the theoretical paradigm rather than the practical aspect of the subject. Moreover, the research questions were not representative of the entire study. Thus, a circumscription towards the introduction was needed.

3. The third iteration brought the notion of mobile development model, which helped consolidating the study. This new notion allowed to precise furthermore the research question, the thesis subject, as well as the lead wire of this design science research. In addition to this, the model supported the harmonization of the findings and transformed them into scientific artifacts through the analysis of literature review and survey's results.

The research design approaches used in this study are analytical and qualitative. This is explained in depth in section 3.2 and 3.3. The primary source of data is collected through an online survey; this is discussed in section 3.4.1. The secondary resource of
data collection is through literary source analysis, which is detailed in section 3.4.2. Preliminary results are retrieved from the data collection, which is elaborated in chapter 5, that includes the analysis of the results. Chapter 6 treats with the explanation of the Mobile Application Development Strategy model. A discussion is necessary to wrap up all information and analysis conducted within the study, which is stated in chapter 7. The last piece of this framework concerns the conclusion of the thesis, discussed in chapter 8, which recalls the original research questions, as well as, reminds the questions’ answer and the outcome with possible further work on the subject.

3.2 Research Design

This study follows a design science research approach as mentioned in the previous section. The design science research methodology is structured through knowledge flows, process steps, and outputs [48][49]. Knowledge flows contribute in redirecting the focus on process steps. They are labelled to identify their nature. A flow commences at a process step and is redirected to a former step. Process steps represent the major milestones to follow in order to arrive at a conclusion. Lastly, the outputs constitute the product of each process step.

The framework illustrated in section 3.1 is based on the design science research methodology and general schema below.

![Diagram of the general methodology of design science research](image)

*Figure 7: The general methodology of design science research [49]*

Design science research is “sometimes called “improvement research”, and this designation emphasizes the problem-solving or performance-improving nature of the activity” [49]. In other words, this type of design research focuses on finding a solution to a known problem. It aims at ameliorating existing processes, architectures,
and models in order to gain in terms of performance, comprehension, ease of utilization, and other factors depending on the study. In this case, the thesis focuses on finding a more efficient way of finding a mobile application development strategy by relying on a newly built model. Consequently, this study’s research design relies on the design science research methodology.

With respect of the aforementioned, one can recognize the knowledge flows and process steps used in this study’s framework. The design science research is utilised to organize this study because it identifies the major milestones conducted during the thesis. First of all the awareness of problem consisted in the research design of the study to find the research questions and to precise subject of the study which are outputs of this step. The second step is named Suggestion, because it represents the analysis of the literature review and the theoretical background. The development phase is illustrated by the online survey conducted to retrieve primary data from mobile professionals. The evaluation step represents the results and analysis, which led to the MADSF (Mobile Application Development Strategy Finding) Model. During this phase, the Mobile Application Development Strategy Finding Model was built and compared with an existing model to identify its strengths and weaknesses. The conclusion and discussion step form the result of the study. The circumscriptions are illustrated by iterations in this study’s framework.

On the other hand, the analytical approach consists in using already existing information in the purpose of criticising the material. This method is employed to stress a problem and aims at finding a solution in a critical assessment manner. Using the design science research implies an analytical approach over the literature review and other secondary data.

## 3.3 Research Approach

The research approach is the conceptual structure in which a research is carried on. For this purpose, a research approach has to be chosen between several types of research: descriptive or analytical, applied or fundamental, qualitative or quantitative, conceptual or empirical, among others [47].

The analytical research has been chosen as research design as explained previously.

Applied research aims at finding a solution to a concrete problem that society or an industry is facing. Whereas, the fundamental research approach is tailored for formulating theories and generalisations. Typically, fundamental research is used for mathematical problems in order to prove that a hypothesis is false or no longer correct due to different factors [47]. This thesis addressed several research questions aiming at finding a solution to a concrete and identified issue in the ICT domain. Hence, it is catered for the applied research approach.

Quantitative research concerns finding information that can be counted or enumerated. On the other hand, qualitative research aspires at collecting phenomenon or information related to or involving quality. Qualitative research is often utilised in investigating a targeted group behaviour over a phenomenon. It is essential to use a qualitative approach in behavioural studies to find out what people think about a
certain artefact or phenomenon. This study intends to find a solution to the issue of deciding which mobile development strategy is best for a specific project. For this purpose, we conduct an analysis of the literature and a survey that both provide past and current information about the subject. Thus, this thesis follows a qualitative approach in which the behaviour of mobile developers is assessed [47].

3.4 Data collection

The fundamental methods of data collection are primary and secondary. The research explanatory design is utilized for this research thesis, therefore data is collected through the above methods, gathered and analysed along with a comparison with the previously stated literature review.

3.4.1 Primary data

Online survey is the primary data collection chosen for this study. It tailors the qualitative research approach by providing crucial information about previous mobile application development. Indeed, this survey collects data about the process of choosing a mobile application development method by providing the name of each application, the method of selection, the criteria of selection, the final method and the time needed to find the appropriate development strategy. Since, the research background provided enough information about how to choose between native, hybrid and web mobile or web to native the results of the survey might confirm or not the aforementioned patterns utilized when following a framework or guidelines. More importantly it will provide us with real data about the current market of mobile application development. Thus, the analysis of these data will constitute the ground floor of a future framework used by the web site project built on chapter 6. Hence, the web site will have tangible data right from the beginning of the beta.

3.4.1.1 Sampling

Sampling and choosing the right audience was the most difficult task to achieve for this survey [48]. Indeed, the primary target is “developers and companies that had developed an app or more”. Hence, the sampling is a French engineering schools and a couple of overseas mobile application agencies. It consequently permits to gather tangible data from a solid and trustworthy population. The following French engineering school has been selected for this survey: EFREI. The small sample of mobile application agencies corresponds to the following list: Appsolute (France/Scandinavia), Fewstones (Singapore), Feel&Click (France).

3.4.1.2 Tools

Adobeform.com

In order to offer a well designed and an easy solution to implement the online survey, adobeform.com has been used. It is a web site that helps customers creating nice and professional surveys. Customers can realize their desired survey and then share it via email or embed it in a web site. The later solution was chosen for the purpose of the
survey. This adobe solution also permits to collect responses in their cloud platform and export them into Excel or PDF format.

1and1.com
To host the web survey, a domain name stepstoapp.com has been acquired on 1and1.com host web site. Thus, the survey is embedded into a web HTML page through this following link: survey.stepstoapp.com. Anyone can access it through a web browser.

3.4.2 Secondary data

The literature review collected through e-search in the fourth chapter of this report outlines previous research on the mobile application development sector. This secondary data collection is essential for the understanding of the study because it represents the basement and premises of the mobile application development emergence. They are included into several categories, such as scientific publications, articles, or white papers. However, some articles are similar in terms of content. Therefore, it is crucial to denote, through several type of scientific papers, the role, overview of mobile application development. This dichotomy results in four categories of point of view: papers in favour of native solutions, papers in favour of cloud web solutions, papers providing guidelines, and papers dispensing frameworks. Each section includes at least a review of two articles.

Both primary and secondary data collection help at building groundwork for the rest of the thesis, especially for the analysis of the survey’s results and the establishment of the web site solution.

3.5 Research Analysis

This research, as previously outlined, is conducted through an online survey and the analysis of literature review. The research analysis is detailed in the next chapter of this study. Indeed, the Preliminary results of the survey are ased through graphics and tables illustrating the data collected from survey.stepstoapp.com. This analysis helps understanding which pattern can be identified easily thanks to a thorough data assessment. The software Excel is utilized to process the data with ease. Therefore, this software delivers specific graphics according to the results’ questions.

On the other hand, the analysis of the literature review conducted in the previous chapter resulted in providing a table depicting the main parameters to consider when choosing a mobile development strategy. These parameters are utilized in the chapter 4 to compare them with the survey’s results. Therefore, we legitimate the establishment of these essential criteria.
3.6 Research Credibility

Credibility is, generally, determined by three factors: reliability, validity, and generalization. These terms in most cases decide whether or not a study is credible [50][51]. The following explanation will help to understand their true meaning.

3.6.1 Reliability

According to Joppe, the reliability is described as following “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” [50].

Similar to this, secondary data is collected through research articles, white papers from existing companies therefore from trustworthy references. Akin to this, primary data is gathered from existing mobile application development companies currently working in several places in the world.

3.6.2 Validity

According to Joppe, “validity determines whether the research truly measures at which it was intended to measure the research results. 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” [50].

Validity is tested through the three following criteria, content validity, criterion-related validity, and construct validity [47], [51]. Since, this study utilizes a qualitative method of collection, being an online survey, the content validity is proven since the sample is composed of developers, and enterprises are currently in activity. Hence, the argument of content validity can easily be verified and tested by any outside representative from the thesis.

3.6.3 Generalization

Generalization “is the real value of research lies in its ability to arrive at certain generalizations” [47]. Generalization allows extrapolating a research to commence from specific facts and to terminate by focusing its research in a research topic, meaning finding a solution to the research question enunciated in a study’s introduction.

Three interrogations are subject to identify generalisation, which are: what is, what may be, and what could be. The study of what is leads to analysing what is common or ordinary [52]. What may be concerns the fact of designing researches to improve
them in order to generate the best research quality for potential future trends. Finally, dealing with what could be aims at triggering specific facts that might seem exceptional and follow them to figure what is their outcome.

Generalisation aspires to provide plausible and accurate prediction to recurring artefacts. Generalisation is an important factor assessing researches quality. Therefore, in case of a survey supporting data collection, the survey’ sample should be from different sources and not focused on a certain type of population. As explained earlier in section 3.4.1.1 the sampling of the survey is motley because it comprises several types of respondents such as companies and mobile freelance professionals. Hence, based on this knowledge, this thesis comprises generalization.
4 Previous research on this subject

This section treats with the previous researches realized before this study. The goal is to gather fundamental data on the subject in order to compare previous solutions to the one created through this thesis. In this section a number of scientific articles, and white papers are grouped by categories of subject. In fact, some authors aimed at focusing on native solutions, some others focused on web solutions, and some papers advocates that hybrid solutions are the best among all. Few papers contented themselves with providing guidelines on mobile application development. Certain authors built frameworks helping companies to understand and choose one of the main three solutions.

4.1 Papers in favour of native solutions

4.1.1 Native vs. web app development by Abalta Technologies

Abalta technologies firm published this white paper in 2012. This article aims at stating the pros and cons of using HTML5 programming language to build mobile applications. It advocates that HTML5 is an alternative if the app developed does not have the following features: complex graphics, sophisticated algorithms, sophisticated UI and transitions with several animations, or very native looking UI, controls and behaviours. Plus, HTML5 still has some issues since it uses devices’ web browser because of the lack of homogeneity among the bevy of web browser behaviour. Hence, it leads to slow applications with performances issues, which deteriorates the user experience. However, sometimes the combination of HTML5, JavaScript and phonegap can accomplish wonders such as the famous LinkedIn iPad app. Nonetheless, application such as Facebook switch from HTML5 to native mobile applications due to low performances. Consequently, Abalta technologies company sustains that HTML5 can be well used for application displaying texts and images without complex interactions and algorithms; otherwise native technologies should be utilized. For now, native apps are still more used than web based apps [53].

4.1.2 Mobile application development web vs. native

This article brings fundamental information about the history of mobile handset as well as key concept to take into account when analysing devices features such as browser performances, design of the device, and user experience. It compares programming languages among the following OS: iOS, Android, Blackberry, MeeGo, Windows Phone, Symbian, Bada, and HP Palm webOS. This paper concludes its analysis on whether web mobile application is equivalent to native by stating that in a couple of year web technologies might outpace native technologies. However for now it they are still weak because of different factors such as the instability of framework used from a mobile browser to another or even their performances. Thus native development still provides the best user experience, the best UI or design [54].
4.2 Papers in favour of mobile web solutions

4.2.1 Mobile web strategy for enterprise success

This paper outlines the emergence of mobile web technologies by explaining the advantages and drawbacks of this solution. Nevertheless, the pros of the mobile web solution seem to outpace its downsizes. Indeed, the author provides a list of all Key Values as well as technical constraints while developing a mobile web application, without forgetting the design considerations. The paper contours key recommendations for mobile web development. It also stresses the fact that cloud mobile web application is a fast growing market because of the many non-negligible benefits that it brings to mobile project. Among these benefits, the author states, “Foster effective collaboration among applications and people.” Which implies that the cloud allows deploying several collaboration tools such as email, web conferencing, instant messaging, and team collaboration. In real business world a legitimate example would be the Facebook app, which was at first built with HTML5 technology and web technologies [55]. This combination was working at the time, however it was not fast and smooth enough according to the customers community. [55] The challenges of mobile cloud computing are raised at the end of the analysis. One of the main challenges is the interoperability between different browsers and standards they utilise.

4.2.2 MobiCloud - Making clouds reachable: A toolkit for easy and efficient development of customized cloud mobile hybrid application

MobiCloud is an initiative from the research group Knoesis launched in 2010. In their paper they outline the benefits of using a cloud web solution for developing a mobile application. The main advantages are: the modification of information can be made whenever needed from the server side. Moreover, all the logic of the application is stored on the cloud. Therefore only cached data is saved in the device. The authors explain their solution called DSL (Domain Specific Language) script CMH (Cloud-Mobile Hybrid). It consists on providing a generic code script for any platform such as Android, Blackberry, Amazon EC2, or Google App Engine. Thus, developers can use this tool to develop simple app with very few programming code lines see their example [56]. This paper also explains how their web site works, which also gives more information about the processes used to achieve the transformation from DSL to native app [56].

4.3 Papers providing guidelines

4.3.1 Mobile application development

This paper aims at acknowledging the lectors about the mobile application landscape by structuring the article in four chapters: the mobile application development options: native, portable, and hybrid, challenges posed to mobile application development strategies by a fractured marketplace, the mobile application
development landscape, the players of the mobile code. In the first chapter the author depicts the three main ways of developing a mobile application. Indeed, the paper is more explicit on the strategy of each of the solutions by providing useful graphics. The author wanted to stress the market share in terms of mobile platforms around the world in 2011. Through the third chapter the paper establishes several keys points to manage when launching an app. The conclusion of the report indicates that no schema of mobile application development can always be chosen because it depends on certain criteria such as the price the firm is willing to pay, the time needed to develop the application, or the mobile platforms targeted among others [57].

4.3.2 Native mobile apps: The wrong choice for business

This paper follows the previous one aforementioned by addressing disadvantages and advantages of using native, web or hybrid through a table and criteria. The author stresses the advantages of choosing native app development such as the graphical performances, the app store which helps to advertise apps, and the total integration of the device API. It also outlines the drawbacks of the native development by stating the importability, which was treated earlier in this study, the expensive development cost, a simpler maintenance process, the platform instability (a platform can disappear within a few year example Blackberry) apps are controlled by Apple on the app store. Of course the author reiterates the same pattern of stating the advantages of utilising mobile web technologies for application development by alleging the cross-platform possibility, the inexpensive development because of the reuse of the code, the simple maintenance thanks to the same programming language over all the platforms, the future proof because in a few year the mobile web browsers will allow more features and combinations, instant update because web apps are located in the server side, and a freedom over the app store control. The disadvantages illustrated in the paper are against the moderation of device integrated API since as for now browsers cannot access the device’s camera, contact list and microphone for instance, and there is a limitation in graphics because web browsers cannot handle heavy graphics. Hybrid solutions are not left on the side of criticism, the author outlines four advantages of choosing this alternative: the native feel and look without its cost, the device integration since this solution offers “native-only” features, the app stores availability since hybrid apps are transformed into native apps after compilation, and in fine the inexpensive cross platform development. In terms of downsize the paper states the following artefacts such as the requirement of knowing the mobile framework as hybrid language, and limited graphics. The latter is less true now since mobile frameworks such as corona SDK or Xamarin help enhancing the user experience and app look and feel by providing tools to improve the graphical aspect of mobile apps developed. The article concludes by providing guidelines about how to approach the mobile app development. Indeed, it states that a company should first of all analyse the web mobile development to determine if it fits their business, then it should continue on analysing the hybrid mobile development for the same reason, and finally focus itself on native mobile development if all other alternative are not sufficient enough [58].


4.3.3 Native web hybrid

The following table sums up all the information gathered in this paper.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>WEB APP</th>
<th>NATIVE APP</th>
<th>HYBRID APP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code &amp; Design</td>
<td>Can have a single code base</td>
<td>Requires multiple code base</td>
<td>Single code base with wrappers for OS device</td>
</tr>
<tr>
<td>Cost</td>
<td>One time cost across platforms</td>
<td>Additional cost for every platform</td>
<td>Between Web &amp; Native apps</td>
</tr>
<tr>
<td>Speed</td>
<td>Instantaneous</td>
<td>Subject to App store approval process</td>
<td>Subject to App store approval process</td>
</tr>
<tr>
<td>Device</td>
<td>Limited APIs access by web browser</td>
<td>All device APIs like GPS, Camera, address book</td>
<td>All device APIs like GPS, Camera, address book</td>
</tr>
<tr>
<td>Interactions</td>
<td>Limited by browser capabilities</td>
<td>Device centric impressive experiences</td>
<td>Limited by browser capabilities</td>
</tr>
<tr>
<td>Distribution</td>
<td>Smart phone users with internet access</td>
<td>All device APIs like GPS, Camera, address book</td>
<td>All device APIs like GPS, Camera, address book</td>
</tr>
<tr>
<td>Edit</td>
<td>Complete freedom</td>
<td>App store rules</td>
<td>App store rules</td>
</tr>
<tr>
<td>Pricing</td>
<td>Any pricing model with 100% revenue capture</td>
<td>Free or paid with app store capturing 30%</td>
<td>Free or paid with app store capturing 30%</td>
</tr>
<tr>
<td>Suitability</td>
<td>Media content, Social collaborations, games</td>
<td>High bit-rate games, camera apps, multimedia processing</td>
<td>App that can be worked with web frameworks</td>
</tr>
<tr>
<td>Heuristic</td>
<td>Great for new ideas</td>
<td>IF performance matters</td>
<td>Logical step of web app</td>
</tr>
</tbody>
</table>

Figure 8: Chart illustrating the comparison between web, hybrid and native [59]

The author also provides guidelines on how to choose between the three mobile development solutions. These guidelines join the one previously stated in other articles. Although, The author insists on the fact that no solution is better over another, but the features of the app being development are critical to establish before starting the actual project. These functionalities will lead to the mobile development method. The paper also adds fundamental consideration to take into account when possessing a mobile application project, especially in terms of mobile strategy. Clarice Technologies enterprise is at the origin of this paper. They provide Mobile product
engineering expertise, Mobile user experience expertise, and Mobile ecosystem expertise [59].

4.3.4 Choosing the Right Mobile Development Method

RDA Company, author of this paper [60], only treats of two mobile development methods, which are the native approach or the hybrid solution. The goal is to bring awareness among businesses wanting to launch an app. This article furnishes a wide description of each one of the solutions above mentioned. As several others, it focuses its analysis on the pros and cons of these methods. Therefore, these explanations will not be treated in this study, as they are similar to previous articles’ analysis. However, this paper brings a good analysis on the assets of the hybrid solution. It insists on the monetization of the apps created through different means. Indeed, it explains the importance of having a hybrid development solution, which is the fact of having the ability to publish your application on several app stores using one code for all the platforms. As a result, the gain from the different app stores will probably outpace the gain of only one app published in one store.

4.3.5 Web-based mobile business apps may be good - but are they good enough

Sybase by SAP firm, author of this article [61], introduces a new concept and term “Mobile Enterprise Application Platform (MEAP)”. MEAP empowers hybrid applications thanks to its structured architecture. Indeed, MEAP represent containers that are completely integrated into the platform environment. It facilitates the deployment of mobile hybrid applications by managing the content received on the device and sent from the application. It also allows enterprise’s employees to use a hybrid app that can easily separate data from personal information to business information. This kind of system improves the push of new hybrid apps process to employees thanks to its agility. The article goes deeper into the subject by providing and elaborating on a real life case featuring a major global fast food brand’s survey application. The author details the entire processes and describes the survey app in order for the reader to understand more easily the migration of a mobile app based on a hybrid technology. The app catered for their worldwide auditors already worked on windows based laptops. Nevertheless, it needed some modifications because of the tremendous growth of mobile devices, which are more adapted to audits thanks to their lightweight and small size. This explanatory paper concludes on sharing some guidelines on when to opt for native or hybrid. These guidelines resemble at the one stated at the section 3.4.2.

4.3.6 Native, web or hybrid mobile-app development IBM

This well structured paper [62] realized by IBM describes the mobile development dilemma of enterprise wanting to launch an app by providing essential information about native apps, mobile-web apps and hybrid technologies. The real add-on among the community is the explanation of the different levels of the Application Programming Interface API of native devices, with details about low-level and high-level of the API as well as information about the Graphical User Interface GUI. The
mobile-web section of this article is worth noting because it approaches the web mobile applications from a different angle than above papers. Indeed, it adopts the customer point of view and simulates an installation of a mobile-web app, which no other articles presented up to here. Hence, it explains that mobile-web apps can be added as native apps with the same icon app shortcut. They can even be launched offline, which is a feature usually encountered for hybrid apps since they are packaged into native apps. Nonetheless, this is a major add-on to the entire study because this last solution was not taken into account. This mobile-web app solution was not previously mentioned because this thesis aims at providing a grasp on the current solutions for mobile development solution resulting in creating a native application. Although, the shortcut and the offline aspect of the mobile-web app are argument demonstrating that they have similar behaviour with native apps, nonetheless they are not packaged through native compiler therefore they do not belong to the same family. This white paper aims at harnessing the benefits of each development approaches in order to stress the essence of the subject, which is to bring crucial information about three kind of development solution for mobile apps with a set of guidelines. For this purpose, the author adds a section about when to choose one solution over another depending on predefined scenarios.

4.4 Papers dispensing frameworks

4.4.1 Cross-platform mobile development

Tribal is a Great Britain enterprise specialised in mobile application development for e-learning purposes. Their team works on a project called Mole [5], which is a mobile application that enables professionals and customers to access crucial information in the most difficult environments thanks to its offline ability of delivering essential information. While working on this project Tribal’s development team accomplished some research on how to build the best mobile application without having to raise tremendous amount of money. Thus, they came up with this white paper gathering analysis about essentially hybrid mobile development solutions. Indeed, the article focuses its study on three major actors among the hybrid mobile development community. Phonegap, Rhodes and Titanium are the solution detailed. The author brings a particular interest into an overview, supported platform, strengths, weaknesses, and a chart of device access capabilities for each one of the aforementioned solutions. Hence, this paper completes the analysis made on the previous analysis of papers, such that it adds some value to each platform device access capabilities and pros/cons. However, these information are now, for some of them, obsolete because of the fast pace update of these frameworks features and supported devices. Nonetheless, the strengths and weakness still remain somehow relevant due to the involution of the platform in some cases. For instance, Phonegap still does not provide a sufficient amount, or inexistent, of documentation, tutorials, videos, and native UI components, design patterns, and development tools featured. Along the entire article, Tribal links hybrid mobile development to its core activity and e-learning mobile project. They insist on the fact that e-learning and the mobile environment are two fast growing domains. They also advocate that the combination of both of them will bring another dimension to the way people learn thanks to mobile devices interactive features. They use a new term “Mobile learning”, which has been previously described.
4.4.2 Building a Mobile Application Development Framework

Intel is proprietary of this white paper [63] covering the mobile application framework development created by their labs. In the first paragraphs, the author states the objectives of the article as well as the challenges their team faced. Intel launched this project mainly due to its huge number of employees that utilize different types of personal mobile devices to access to enterprise’s information or in house services. Therefore, the Intellectual Property question was asked at the beginning of the project, which end it up being a non negligible requirement to fulfil in order to address the level of security the company aimed at achieving. Mobile device management was another reason because it would help the firm manage what content could be seen by whom through solutions like calendar, and email. The deciding deploying mechanism phase of the project brought discussions about the repartition of mobile application development methods from Hybrid (Native and web such as Phonegap) to Virtualized platform dependent solutions through Mobile Enterprise Application Platform MEAP or Web browser only platform independent. Advantages and disadvantages are depicted for each one of the solutions except web browser only. Consequently, Intel analyses the different alternatives to elaborate on two of them, which are the MEAP and virtualisation solutions. The reason being that MEAP is the best fit when there is only one platform to focus on, and the virtualisation is the second best alternative when the application requires to be delivered on several platforms. The author later emphasises on the core blocks of the framework, which represent the following elements: guidance documentation, enabling capabilities, supporting resources. In which, guidance documentation provides the following services: Governance, Value, Security, Development, and Provisioning. On each sub phases Intel furnishes crucial guidelines to the developers. Concerning the enabling capabilities, they comprise: a web access portal, virtual private networks, authentication, device management, MEAP, and an enterprise app store. All of the latest is giving by Intel infrastructures, which improves the mobile developer’s working experience. Intel also provides complete supporting resources through a series of tools such as the application workshop group, or even engineering roadmaps enabling developers to track their development phases among projects. The conclusion of this white paper reveals that Intel IT’s mobile application development framework is complete thanks to five years of consistent work. Their framework was successful in number of cases such as the deployment of customer relationship management, social media, travel tools and many others since the publication of this thesis.

4.4.3 Framework selection for mobile enterprise applications

Sierra System company is author of this white paper [64] dedicated to the mobile enterprise applications. It ties with the previous article from Intel because it outlines the same business need for mobile apps and targets companies’ employees. The exception resides in the firm first target. Indeed, the Intel’s framework was originally established for Intel employees’ devices whereas Sierra System’s framework for selecting existing frameworks targeted for a wider audience, basically any enterprises. This article stresses several key points for enterprise to go for a mobile application starting by the usability and security, the cost, the fine tuning apps, and the performances. The paper elaborates on the analysis by providing example such as aiming for only one mobile platform or many, where for the later Mobile Device Management should be used for security purposes. In terms of cost, the author
Previous research on this subject advocates that developers, tools, and runtimes are the major investment into mobile application projects. If the developer is a native skilled expert the amount will be more likely increase significantly. For instance, the average salary for an objective-c developer in San Francisco bay is in 2013 near 140,000 Dollars per year [65]. Moreover, on average the cost for native application oscillate between 15,000 and 500,000 dollars [64]. The fine tuning app treats of which mobile application development solution to use. This section elaborates on how to choose the best-fit development method, which in this case is the hybrid solution featured. The author emphasises on the cross platform tools and method that help developers to be more productive and efficient since they reuse major parts of the code thanks to the combination of HTML5, CSS, and JavaScript technologies. Concerning the framework selection criteria, they are very similar to previous one featured in earlier sections of this study. Hence, they will not be treated to avoid repetition. However, the tables below illustrate the questions to think of when starting a mobile application project. These tables are mentioned for the lector understanding.

<table>
<thead>
<tr>
<th>Selection Criterion</th>
<th>Description</th>
</tr>
</thead>
</table>
| Native and web architecture support | Determine if your app requires:  
- interaction with device sensors other than GPS and phone (i.e. camera, accelerometer, compass, audio)  
- complex interaction with other mobile apps (e.g. process data for other mobile apps) or with the operating system (e.g. process data while in the background)  
If the answer is positive then the framework must support the development of native or hybrid mobile apps, otherwise mobile web architecture support is sufficient. |
| Platforms                   | What platforms are supported by the framework? What versions of these platforms are supported? |
| Design productivity aids    | Does the tool included in the framework provide design aids allowing you to extend the existing enterprise apps (i.e. design page transitions, design page layout, bind the data model to the mobile user interface, specify declaratively the field and form validations)? |

*Figure 9: Mobile application framework selection criteria (1/2) [64]*
The only value added to previous frameworks is the establishment of a task list to follow in order to take the best advantage from Sierra mobile app framework criteria table [64]. This list is replicated hereby:

1. “Set parameters for each of the criteria to give precision to the evaluation and provide a common unit of evaluation (e.g. cost) that will help with the scoring.”
2. “Decide which criteria you cannot be flexible about – these become your mandatory requirements”
3. “Create a short list of frameworks that fully meet the criteria”
4. “Score the frameworks on the short list using the remaining (rated) criteria”
5. “Pick one of the frameworks at the top”
6. “Through a proof of concept, evaluate the framework in depth”
7. “Repeat the steps 5 and 6 until satisfied, or as time allows.” [64]

According to Sierra System expertise, the choice of the architecture is the most crucial decision to settle because it leads to one framework of another. Later in the paper, the author applies the above criteria to existing frameworks through a table comparing web app frameworks, already covered earlier, and native/hybrid frameworks, also featured in previous articles.
4.5 Main criteria extracted from the literature

The analysis of the above articles permitted to collect the most common criteria selected when choosing a mobile development strategy. The following section aims at gearing up similar criteria encountered in the literature. Hence, it constitutes a summary of the literature review clarifying which components represent key aspect of the selection of mobile development strategies.

4.5.1 Selected criteria

Each criterion is described in the following sub sections.

4.5.1.1 Performance

This criterion is one of the most critical if not the most critical for choosing a mobile development strategy. As technology advances, powerful responsive smartphones are released and mobile users are increasingly focused on details. Thus, mobile applications developers must keep the fast, almost frenetic, pace of evolution to please their clients. Performance is therefore one of the most criticized criteria for mobile application rating.

Performance is mentioned in several articles and enunciated in section 4.1.1, 4.1.2, 4.3.3 and 4.4.3. It usually concerns poor performances noticed on smartphone when utilising web hybrid mobile development strategy.

4.5.1.2 User Interface

Design is another criterion fundamental from the users’ point of view. Therefore, the User Interface is considered crucial because handset furnish a high level of screen quality not only in terms of colours but also in terms of design possibilities. User Interfaces evolve towards simplicity. Hence, this criterion has to be dealt with thoroughness.

User Interface is mentioned in several articles and stated in section 4.1.2, 4.3.3, 4.3.6 and 4.4.3. It concerns the low level of interfaces quality noticed on smartphone when utilising any of the native, hybrid or web to native mobile development strategies.

4.5.1.3 Cost

Cost is undeniably a major criterion to be accounted for. Usually, hybrid mobile development is preferred if the company has a limited budget to input on the mobile application development. Companies, usually, think that hybrid solutions will sustain for the debut of their project. In fact, it is not the case because performances do not meet customers’ expectations. Consequently, Cost is a criterion that can be an obstacle for start-ups or SME wanting a native mobile application if they do not already possess the necessary skills.
Cost is mentioned in several articles and depicted in section 4.3.1, 4.3.3, 4.4.1 and 4.4.3.

**4.5.1.4 Core device API**

Core device API is another criterion that can be extracted from the literature review. Core API means any core functionality that deal with the lowest layer of APIs built by smartphones’ manufacturers. It can include, 3D graphics, Bluetooth, Accelerometer, NFC, Vibration, among others. Applications using the aforementioned functionalities have to be implemented using native development.

Core device API is mentioned in several articles and depicted in section 4.3.3, 4.3.6, and 4.4.3. These articles outlined the incapacity of web hybrid mobile to access core device API.

**4.5.1.5 Code update**

Code update is only utilized for web mobile applications or cloud mobile solutions. It is possible thanks to the fact that the code is held on the server side. This functionality can be useful for corporate mobile application demanding a fast code update pace. IBM created their own mobile development framework to build corporate mobile applications, among other range of applications that comprised code update ability.

Code update is mentioned in several articles in section 4.2.2, 4.3.2, and 4.3.3.

**4.5.1.6 Time**

The time criterion is subjective because it depends on the number of developers working on the project and on their experience with the mobile platform SDK. However, it has been enunciated in the article covering section 4.3.1 therefore it constitutes a potential criterion to acknowledge in our study.

**4.5.1.7 Portability**

Portability is a critical issue encountered for enterprises having a restricted budget. This criterion has to be accounted uniquely if the company desires to provide its mobile application on several platforms. The most utilised solution is yet the native development strategy as mentioned in the online survey by mobile professionals in section 5.

Portability is analysed in the following set of papers of section 4.3.1, 4.3.2, 4.3.3, 4.3.4, 4.4.1, 4.4.2, and 4.4.3.
4.5.1.8 Platform Stability

Platform stability is mentioned in the article of section 4.3.2. This criterion is useful due to the fast growth pace of the mobile application development domain. New technologies, frameworks, software development kits are constantly released which contributes in regulating the innovation rhythm in the market. Therefore, mobile application developers are constrained to follow up and provide the best user experience according to the latest technologies. Companies that intend to build an app should consider the platform stability in order to be visible by the maximum amount of active users. Blackberry RIM has lost more than 35% of the mobile market share in devices shipment over the last three years as mentioned in figure 3. Thus, if a company still bets on Blackberry it means that they might face a decrease of Blackberry’s users or even the disappearance of the brand.

4.5.1.9 Suitability

The suitability aspect is covered in the article analysed in section 4.3.3. It specifies the best suitable use of native, hybrid, or web app solutions. It provides a general guideline for mobile application development. It should be one of the first criteria to focus on when choosing a mobile development strategy.

4.5.1.10 Heuristic

The heuristic criterion illustrates, as mentioned in section 4.3.3, the best possible utilisation of a mobile application development strategy. It constitutes a non-negligible criterion because it contours the development purpose of the application. In other words, if a company has an idea that is feasible through a web hybrid solution for the purpose of a demonstration this criterion needs to be consulted to limit mistakes in development choices.

4.5.1.11 Development resources

Documents and platform support is essential for the sustainability of a mobile developer. The attraction of new mobile professionals is emphasized when this criteria is met, which means that the development resources are sufficient to build simple to complex applications. This criterion is discussed in several papers illustrated in section 4.4.1, and 4.4.3.

4.5.1.12 Development skills

The development skills criterion is enunciated in section 4.4.3 in order to warn companies in their internal hiring process in case they are interested in employing in-house development skill, which means hiring a mobile developer. Indeed, enterprises should choose their mobile development strategy before having to search for a developer for their application. Otherwise, if a modification occurs in the mobile
development strategy after hiring a mobile expert in certain programming languages that do not match with the new strategy a major issue will have to be handled.

### 4.5.2 Overview guideline table

The below depicted table illustrates and sums up the explanations given on the previous section.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Native</th>
<th>Hybrid</th>
<th>Web mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Good</td>
<td>Relatively good</td>
<td>Poor</td>
</tr>
<tr>
<td>User Interface</td>
<td>Good</td>
<td>Good</td>
<td>Fairly good</td>
</tr>
<tr>
<td>Cost</td>
<td>Expensive</td>
<td>In between Native and Web mobile</td>
<td>Inexpensive</td>
</tr>
<tr>
<td>Core device API</td>
<td>Accessible</td>
<td>Accessible in most cases</td>
<td>Restricted access</td>
</tr>
<tr>
<td>Code update Time</td>
<td>Forbidden</td>
<td>Forbidden</td>
<td>Possible</td>
</tr>
<tr>
<td></td>
<td>Relatively short</td>
<td>Relatively short</td>
<td>Short</td>
</tr>
<tr>
<td>Portability</td>
<td>1 code 1 platform</td>
<td>Cross-platform</td>
<td>Cross-platform</td>
</tr>
<tr>
<td>Platform stability</td>
<td>Good</td>
<td>Fairly good</td>
<td>Fairly good</td>
</tr>
<tr>
<td>Suitability</td>
<td>High performance games, complex algorithms</td>
<td>App that can work with dedicated frameworks</td>
<td>Media content, social content, games</td>
</tr>
<tr>
<td>Heuristic</td>
<td>If performance matters</td>
<td>In between</td>
<td>For prototyping</td>
</tr>
<tr>
<td>Development resources</td>
<td>Many</td>
<td>Relatively good</td>
<td>Many</td>
</tr>
<tr>
<td>Development skills</td>
<td>Less than web</td>
<td>Less than Native</td>
<td>Many</td>
</tr>
</tbody>
</table>

**Figure 11: Overview guideline table**

These criteria represent the secondary data collection according to the methodology of this study, which is thoroughly expounded in section 3.4.2. With respect to the aforementioned section, this table is established after a concrete and extended analysis of the literature review, which implies that the above guideline table was built using a analytical research approach.

The analysis around previous researches on the subject ultimately offered a wider overview about the actual state of art. One can understand that four different types of papers argue the best possible way of acknowledging developers and companies to the right mobile development strategy. Among all, the third solution seems to dominate in the scientific area, with six papers advocating the benefits of guidelines. Hence, we can assume that guidelines constitute a tool helping developers to choose the suitable mobile development strategy. Therefore, the following survey will confirm the above stated hypothesis. Moreover, this survey will use the aforementioned criteria to identify which one are the most important so that they can be re-employed differently to aim at finding the suitable mobile development strategy.
5 Survey

This section aspires at providing the results of the online survey conducted from September 12th to September 20th of 2013. The section is structured as following; the first sub chapter aims at establishing the boundaries to the questions posed and also an explanation of the chosen interrogations. The second sub chapter provides explanation about the techniques employed to get in touch with reliable responses. The third sub chapter underlines the results of the survey by focusing on each question supported by charts.

5.1 Boundaries and establishment of the questions

5.1.1 Boundaries

The online survey had to be structured so that organisations and freelance developers could answer immediately to the questions. In order words, questions had to be straightforward and easy to understand. The relevance of the questions to the subject of thesis was another fundamental point to assess when realizing the questionnaire. The number of questions was also a major criterion. If the survey was composed of many questions, responders might have been reluctant to reply. The availability on the web and the tool employed was another concern to include in the preparation of the questionnaire’s opening.

5.1.2 Establishment of the questions

Each following sub section focuses on a question for this study and explain the nature of the interrogation. This survey comprises seven questions.

5.1.2.1 What is the name of your mobile application?

This is the first step to the identification of an app project. It helps tracking a specific project through the rest of the questions. It also brings an identity to each app rather than a number among others. Lastly, it provides an element of verification to avoid spam or false projects’ application. Therefore, it is a relevant question for the conducted study.

5.1.2.2 Which processes do developers conduct before applying a mobile development architecture?

This question aims at collecting information about how developers proceed to find the right mobile application development architecture/strategy. Eight responses are available to the respondent. Answers highlight different methodology of research from seeking frameworks to investigation through guidelines.
5.1.2.3 What are the criteria for which developers adopt a mobile application development strategy?

Several criteria significantly affect the choice of mobile application development strategy. This question is one of the most important because it depicts what enterprises are looking for and also how developers react upon those requirements.

5.1.2.4 How long did you take to choose your mobile development strategy (Native, Hybrid, or web)?

This question has been established in order to grasp the approximate time needed to determine the most catered mobile application development architecture for a specific project. Thereby, statistics can be retrieved and compared to identify the average time needed to narrow down the tailored solution. Thereafter, we can compare the average time needed with the web platform tool built on stepstoapp.com.

5.1.2.5 Which mobile application development solution/IDE have you used to develop your mobile app?

The above question is also fundamental for the study because it represents a critical element of mobile strategy modification. In other words, this question provides to the study insight into the outcome of developers’ researches.

5.1.2.6 What is your email address?

This question was set up for a matter of identifying mobile developers or app owners and reliability. An email of acknowledgment will be sent at the end of the study as well as a web link comprising the results of the online survey.

5.2 Methods used to reach responders

The most difficult aspect of this study was to convince freelancers and professionals to answer the questionnaire for free. Indeed, usually online surveys are remunerated by a small amount of money susceptible to generate an interest. Unfortunately, this study was not sponsored nor financed. Thus, the only solution to tackle this issue was to make use of social networks and relationships. The developers network group from EFREI, French engineering school, was contacted, and therefore contributed to the completion of this study. Many professionals from different countries have also been selected and participated in filling the online survey. After some time the use of mobile developers’ forums was employed to collect more replies. Every three days reminders were sent by email in order to keep track on the professionals that did not have time to answer. In the end, in less than ten days thirty answers were collected.
5.3 Results of the survey

The results are organised by following the order of the online survey questions. Some of the questions allowed selecting several choices.

5.3.1 What is the name of your app?

* 30 total responses, 100% of submissions

The above chart illustrates the success response rate on the particular question. Every professional answered to the question, which improves answers reliability. After a brief search on the respective app stores, all apps featured in this survey’s results happen to be effectively present.
5.3.2 Which processes do developers conduct before applying a mobile development architecture?

<table>
<thead>
<tr>
<th>Process</th>
<th>Percentage</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research of similar projects</td>
<td>33%</td>
<td>10</td>
</tr>
<tr>
<td>Research of frameworks</td>
<td>50%</td>
<td>15</td>
</tr>
<tr>
<td>Use of in-house guidelines</td>
<td>30%</td>
<td>9</td>
</tr>
<tr>
<td>Use of external guidelines</td>
<td>13%</td>
<td>4</td>
</tr>
<tr>
<td>Feasibility studies</td>
<td>20%</td>
<td>6</td>
</tr>
<tr>
<td>Exploration of Forums</td>
<td>3%</td>
<td>1</td>
</tr>
<tr>
<td>Use of developers expertise</td>
<td>37%</td>
<td>11</td>
</tr>
<tr>
<td>Comparison analysis with previous projects</td>
<td>10%</td>
<td>3</td>
</tr>
</tbody>
</table>

* 30 total responses, 100% of submissions

**Figure 13: illustration of the survey’s second question**

As depicted in the above graphic, developers rely more on research of frameworks, similar project and their own competences to decide on which mobile development strategy they should utilize. Indeed, these three processes have been selected 36 times by mobile professionals. On the other hand, a small amount of external guidelines and exploration of Forums processes have been judged useful from the developers and companies point of view.

These processes have been selected after the analysis of the literature review. The latter pointed that guidelines were more inclined to have an impact of mobile professionals, their number outpacing other types of papers advocating the use of framework or specific mobile development strategy.
5.3.3 What are the criteria for which developers adopt a mobile application development strategy?

<table>
<thead>
<tr>
<th>What are the criteria for which developers adopt a mobile application development strategy?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
</tr>
<tr>
<td>User Interface</td>
</tr>
<tr>
<td>Portability</td>
</tr>
<tr>
<td>Dev. skills</td>
</tr>
<tr>
<td>Core device API</td>
</tr>
<tr>
<td>Dev. resources</td>
</tr>
<tr>
<td>Cost</td>
</tr>
<tr>
<td>Time</td>
</tr>
<tr>
<td>Quality</td>
</tr>
<tr>
<td>Code Update</td>
</tr>
</tbody>
</table>

* 30 total responses, 100% of submissions

**Figure 14: illustration of the survey’s third question**

The graphic demonstrates the predominance of three criteria for choosing a mobile application development strategy: Performance, Quality, and User Interface (UI). They represent a combination of 50 selections by mobile developers. However, Core device API nor the Dev. skills seem to be less considered.

These criteria are a combination of the one stated in section 4.5.2, with more understandable terms for the responders, and other criteria that seemed logical to engrraft.
5.3.4 How long did you take to choose your mobile development strategy (Native, Hybrid, or web)?

<table>
<thead>
<tr>
<th>Time Frame</th>
<th>Percentage</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 2 days</td>
<td>17%</td>
<td>5</td>
</tr>
<tr>
<td>2 - 4 days</td>
<td>60%</td>
<td>18</td>
</tr>
<tr>
<td>4 - 6 days</td>
<td>13%</td>
<td>4</td>
</tr>
<tr>
<td>+ 6 days</td>
<td>10%</td>
<td>3</td>
</tr>
</tbody>
</table>

* 30 total responses, 100% of submissions

*Figure 15: illustration of the survey’s fourth question*

More than 18 developers stated that they needed between 2 to 4 days to decide whether they would choose a native, hybrid or web mobile development approach. In total 25 mobile professionals admitted to reflect on the choice of their mobile development for more than 2 days. The remaining 5 developers affirmed that they only needed between a few hours to 2 days to complete this task.
5.3.5 Which mobile application development solution/IDE have you used to develop your mobile app?

<table>
<thead>
<tr>
<th>Solution/IDE</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xcode</td>
<td>60% (18)</td>
</tr>
<tr>
<td>Eclipse</td>
<td>53% (16)</td>
</tr>
<tr>
<td>Visual Studio for Windows Phone 7/8</td>
<td>7% (2)</td>
</tr>
<tr>
<td>Xamarin</td>
<td>0% (0)</td>
</tr>
<tr>
<td>Titanium from Appcelerator</td>
<td>0% (0)</td>
</tr>
<tr>
<td>Phonegap</td>
<td>20% (6)</td>
</tr>
<tr>
<td>Corona SDK</td>
<td>0% (0)</td>
</tr>
<tr>
<td>Android Studio</td>
<td>3% (1)</td>
</tr>
<tr>
<td>Web app</td>
<td>7% (2)</td>
</tr>
<tr>
<td>Other</td>
<td>3% (1)</td>
</tr>
</tbody>
</table>

5 additional choices not shown

* 30 total responses, 100% of submissions

**Figure 16: illustration of the survey’s fifth question**

This chart outlines the fact that the Apple mobile platform is still the most utilized, with 60% of the apps developed for iOS. Closely followed by Android with around 53% of the apps implemented for the Google mobile platform. Phonegap framework is utilized in 6 cases. Merely 7% of the apps are deployed on the Windows Phone platform.

The tendency stressed in the introduction and theoretical background, according to which the market is broken such that Android devices are the most shipped in the world with nearly 75% of the entire mobile market, followed by iPhone devices and Windows Phone in third position, is almost confirmed at the exception of the Windows Phone. Nonetheless, according to ABI research, that published a report in March 2013, “56 billion smartphone apps will be downloaded in 2013”. Among which 58% will be Android and 33% from Apple mobile platform [66]. Moreover, the number of mobile apps in both Apple and Android app store has exceed 800 000 according to Canalys [67]. Hence, the study sample is fairly representative of the real mobile application market distribution.
5.4 Analysis of the survey’ results

This analysis aims at providing answers to the research questions stated in section 1.3. Data collected from mobile professionals is subject to qualitative analysis. Thus, each survey chart has been examined to retrieve responses to each research questions.

5.4.1 Processes

The figure 13 illustrates which processes are utilised to determine the suitable mobile development strategy for a specific project. These processes provide a better understanding on how developers proceed to find a mobile development solution. This graphic highlights three categories of processes below described.

First of all the major category is represented by the Research of frameworks, the Research of similar projects, the Use of in-house guidelines, and the Use of developers’ expertise. These processes are above the average with more than 9 selections up to 15. If we were to combine all selection the amount of selection would be 59, and these processes added together give 45 selections. Thus, the first category of processes is the most utilised with a high percentage of 76.3% usage by mobile professionals.

The second category is composed of three processes: Use of external guidelines, Feasibility studies, Comparison with previous projects. Indeed, these processes are considered by developers, however they are not priority compared to the first category. Consequently, these measures cannot be solely used to accurately define the best mobile development strategy for an app. Akin to this, the first category of processes is more frequently utilised by mobile professionals. Hence, perhaps a combination of both processes will constitute a thorough research (see Appendix Survey table 1 and 2 for details).

The third category is only composed of one process: Exploration of Forums. This method has solely been selected once which means that it cannot be considered as efficient as the aforementioned processes. Therefore, Exploration of Forums is an obsolete process for researching a mobile development strategy.

Criteria are taken into account after the processes of choosing the tailored method of investigation that will lead the developer to the catered mobile development strategy.

5.4.2 Criteria

Most of the criteria are defined in section 4.5.2 by analysing literature review. By assessing this chart, four categories of criteria were highlighted. The latter is described below by explaining how these groups are formed.

One can clearly notice the first category that combines Performance, User Interface, and Quality. They are dominant criteria by far with more than 14 selections each,
compared to the remaining criteria these three are above the selection average. These criteria are dominant because of their fundamental characteristics. In other words, a mobile project should always be in favour of providing the best User Interface possible, as well as high performance and an overall outstanding quality of service. When these criteria are met at the time in a mobile application, the latter should be well rated by users.

The second category deals with Time, Portability and Code Update, which are considered to be important in the process of choosing a mobile development strategy. Indeed, Time is an essential factor for clients because the quicker the application is released; the better it is for the customers.

The third category concerns the following criteria: Cost, Dev. resources. Developers are less concerned about these criteria; nevertheless they represent non-negligible factors. Indeed, the Project Cost affects the type of mobile development strategy retained. Dev. resources are fundamental for every mobile application development strategy chosen; they allow developers to quickly find the resources they need.

The last category combines the least used criteria among mobile professionals. Dev. skills and Core device API form the fourth category of criteria. Usually these factors weight less than the one previously mentioned due to the focus over a type of mobile professionals by the clients. For instance, clients will contact a mobile agency having a diversity of developers’ skills such that Dev. skills are not an issue and therefore neither a criterion is that case. On the other hand, Core device API would be useful when the clients’ application needs to use specific devices’ drivers such as 3D transformation utilised in games, Accelerometer, Internal processes, Audio/Video drivers.

5.4.3 Time of research

The figure 13 depicts the time of research needed for each mobile project in order to find the suitable mobile development strategy. This graphic outlines that 18 projects spent between 2 to 4 days to complete the research of the mobile development solution. For an entire mobile project, this research of solution is a crucial phase due to effect it can cause in the long term during the implementation phase. Nonetheless, 2 to 4 days of research represent a steep price because this time could be allocated in building the product. Consequently, all methods used to find the tailored mobile development strategy have not been enhanced enough to provide a faster way of choosing between the considerable amount of mobile development solutions.

As a result, this study will provide a suggestion to overcome these drawbacks. The following chapter is dedicated to the improvement of the research method of mobile development strategies through the use of a model.
6 Mobile Application Development Strategy-Finding Model

This section aims at explaining the model suggested for responding the research question “How to provide developers and companies with a model that help them to choose on a catered mobile development strategy?”. Indeed, this chapter will attempt to provide detailed information about previous research on mobile development strategy models to prove the inexistence of such a MADSF Model. Then an explanation of the need for criteria to be part of the model will be given. Followed by the description of the model on itself. The transition between the method and the model will be accompanied by the depiction of the output algorithm from the decisional schema stated in the section 6.3.

6.1 Titanium - Mobile Development Strategy Model

Few scientific papers or articles treating of this subject were realised. Indeed, a selection of the closest papers in terms of meaning and area of research is depicted in the following section. Titanium “4 steps to creating a Mobile Development Strategy” [68] is analysed by interpreting their model step by step.

This paper is the closest to this thesis subject and objective. Indeed, it contours the subject of mobile application development by offering a model approach that is composed of a Web Maturity Model, which separates the major phases encountered when developing mobile applications.

Titanium proposes 4 steps to accomplish the task of choosing a MADS (Mobile Application Development Strategy) starting by deciding on how you want to interact with your mobile customer. Secondly, this paper suggests prioritizing on your platform. Thirdly, it advocates that the evaluation of your development resources should be dealt with. Finally, it recommends choosing a scalable development technology.

It offers three types of phases that are about: Exploration, Acceleration, and Innovation.

**Exploration** aims at testing a mobile strategy, which means choosing a mobile platform and a target group to focus on. This phase usually implies to outsource the development of the application.

**Acceleration** deals with engaging a community by making use of social tools. It also means the expansion of mobile platform targeted at two. The mobile development starts to be handled inside the company, or is managed from the company.

**Innovation** is the ultimate phase that does not end as long as new mobile Operating Systems keep on being released. This phase aspires at expanding to all the major
mobile platform existing and at centralizing the mobile development within the company.

Such architecture of model allows the comparison between goals. Every step represents a row of the model.

![Titanium Web Maturity Model](image)

**Figure 17: Titanium Web Maturity Model [68]**

The following statements explain each steps of Titanium’s model:

1. **Decide how you want to interact with your mobile customer**

   This step aims at establishing the foundation for further investigation into the best mobile development strategy. The first item is “Inform (Brochureware)” for the Exploration goal. It targets mobile applications that provide information to users, such as news, blogs, and magazines applications. The second item, linked to the Acceleration goal, refers to mobile applications targeting social communities. The third item, linked to the Innovation goal, presents mobile applications that use e-commerce and sophisticated cloud services features.

2. **Prioritize your platform development**

   This step suggests that the customer should prioritize its app development strategy on one platform for Exploration, two platforms for the acceleration phase and multiple platforms for the innovation phase. This point is verified by the tremendous ascension of Whatsapp mobile application. Indeed, the company started its Exploration phase by launching their application for the iPhone, then continued its Acceleration with the Android platform, and now the app is represented in several platforms such as iPhone, Android, BlackBerry, Nokia S40, Symbian and Windows Phone.
3. **Evaluate your development resources**

The following step aspires at bringing awareness around the development resources. It states that the mobile application development should be outsourced for the Exploration goal, Mixed for the Acceleration goal, and In-House for the Innovation goal. The paper claims that scalability is the key aspect of the decision to possess In-House developers, Outsource or have a mix of both.

4. **Choose a scalable development technology**

The last but not least step concerns the technology to use for a mobile application development. The “Native” technology is used for the Exploration goal, as “Reusable Modular Components” are utilized for the Acceleration goal, and “Integrated Mobile Architecture, Cloud Services” are employed for “Innovation” purposes. The cross-platform, scalability and time-to-market are the main three problematic encountered in this step.

This model (figure 17) helps to define the right customer for your application as well as finding the right number of platform to focus on depending on your goal. This paper also permits to establish a development staff strategy in terms of mobile development. Finally, it allows customers to choose a suitable mobile development technology/strategy for their mobile application business.

### 6.2 Use of Criteria

On one hand, criteria extracted from the literature review constitute an overview of fundamental artefacts that can help in the decision of a mobile development strategy. Even though the choice of a mobile development strategy might only be based on the developer expertise, criteria provide a subjective point of view that can enhance the process of choosing a MADS. Criteria can be transformed into questions that structure the research to find the best fit for a specific mobile project.

On the other hand, criteria are utilised by mobile professionals, which means that they are not only useful for complex projects but also for any type of mobile projects as long as it is tailored for a mobile platform.

Consequently, the use of criteria seems to be essential to the determination of a tailored MADS. In order to construct a comprehensive model we used a system of steps that include crucial criteria.

### 6.3 M ADSF Model

MADSF stands for Mobile Application Development Strategy-Finding Model. This model is the result of the analysis of the literature review and the survey’s results. Thanks to several iteration within the design science research, the below model was created. It represents the contribution of this study in the mobile development area.
The MADSF Model is composed of five milestones that conduct in finding the tailored MADS. This model represents the conceptual view of the decisional schema illustrated in section 6.3.2. In fact, each milestone is present and visible in the decisional schema. Each milestone is constituted of several questions that need to be answered in order to continue to the next milestone. The responses of the inner-milestone determine the path to follow in the decisional model. For instance, if to question “Is your app built for a demo?” the developer answers “Yes”, then the MADSF Model will eliminate the “No” path of the possible responses. In other words, the developer will only have a half of the decisional schema to focus on. Nevertheless, the model user will have to go through the 4 other milestones to complete the process of finding the catered MADS.

6.3.1 Model

The following model comprises five milestones. Each milestone is characterised by its name and an order number.

![MADSF Model Diagram]

*Figure 18: Mobile Application Development Strategy-Finding Model*

1. **Mobile Application Purpose** aims at determining if the mobile application is designated for a business marketplace of mobile applications or rather for a demonstration. Therefore, the question asked at the 1st milestone would be:
   - Will the mobile application be used for a demonstration purpose?
2. **Mobile Application Device Target** aspires at finding more information about the platform target of the mobile application. Indeed, knowing the type of device would determine the nature of the mobile application project. Akin to this, the questions asked for this milestone would be:
   - How many mobile platforms will the app target?
   - What are the mobile platforms targeted?

3. **Mobile Application Core Device API Utility** intends to bring light to the core essential features of the mobile application. In other words, the determination of the Core Device API is crucial for the app development. The following question would be stated:
   - Would the mobile application use on the following Core Device API feature?

4. **Mobile Application Update Process** aims at discovering the mobile application main update steps. Major update would affect the design and content of the application. Hence, it demands to be conduct via two solutions whether the application will be updated remotely from a server side base or via the usual mobile application marketplace. The following questions could be asked:
   - Will the mobile app be updated through a server side?

5. **Mobile Application Quality Standards** is the last but not least milestone to pass by in order to determine which MADS is catered for a specific project. Even though Quality is subjective concept, in the mobile application environment it can be defined as following. High quality of mobile application provides smooth user interface as well as high performance, which mean that all transitions are perfectly inline with actions triggered. Medium quality of mobile application aims at compromise between high quality and cost. Whereas, Low quality of mobile application can allow small malfunctions, minor design details or visible inhomogeneity, and low responses time actions triggered. Consequently, the following questions would be asked:
   - What level of quality the mobile application tends to follow?
   - From which concept could the app be separated?

By following each step enumerated above the developer will reach the end point being the MADS catered for its mobile application project. Although these five milestones suffice to determine a suitable MADS, some other criteria count in the process of selecting a MADS. Indeed, some criteria extracted from the literature review and utilised in the survey can be part of the selection. Thus, it means that independent variables can interact with the MADSF Model. These criteria are the following: Cost, Time, Documentation, and Code Expertise. In other words, the cost can fluctuate, as well as the development time, the amount of documentation which depends on the chosen development solution, and the code expertise can be an aspect of the selection of a MADS. The variation of those four criteria will not be illustrated on the decisional schema.
6.3.2 Decisional schema

Based on the criteria established in section 4.5.2, we have combined the theoretical results in order to build the decisional schema below designed with entity objects. Connections are illustrated by lines, decisions are depicted by diamonds, and squares represent the main three possible MADS. Akin to this, “YES” and “NO” represent the path given for both cases according to a decision made. For instance, Prototype means that “Will the mobile application be used for a demonstration purpose?”, Cross-platform is related to “What are the mobile platforms targeted?”. The other questions are linked to the aforementioned milestones. Therefore, the same type of questions can be constructed from the green decisions illustrated below.

![Decisional schema for choosing a mobile development strategy](image)

**Figure 19: Decisional schema for choosing a mobile development strategy**

The development skill parameter from the original table is not represented in the schema because it depends on several factors such as the country of the developers and whether or not the development skills are in part of the company.

This schema is simplified such that the essential information is retained among the chosen criteria. Nonetheless, a number illustrates milestones. Not all criteria are considered in the schema due to their dependence to external factors. For instance, the project’s cost is influenced by the number of features, the time of development, as well as the country it is implemented in. Moreover, all criteria illustrated in the schema are also part of the determination of the total price. Nevertheless, the project’s cost should not be the first argument to establish a comparison of mobile development strategies because it usually draws the project at a low quality delivery. Time, Platform stability, and documentation are not treated for the same reason as earlier
stated for the Cost, but more importantly they do not fit in a fixed decisional schema. Indeed, they can only interact in the MADSF Model as stated in the previous section.

This above schema subjectively describes the process of the algorithm. In other words, the web platform will convey the same order of question in terms of conditional blocks to retrieve the most suitable MADS. Developers will have to answer to all the preliminary questions such as following:

- What is the name of your app?
- Is it a prototype?
- Does your app have to be developed in several mobile platforms?
- Does your app include Core API devices features?
- Does your app look and feel have to be remotely updated?
- Do you intend to provide an exceptional user experience towards a high quality product?

The backend collects all answers to retrieve the best mobile development strategy according to the above-illustrated schema. Then the user is conducted to the second step of the process through a series of questions that caters the user’s mobile project such as selecting the targeted platforms, select the needed features. The next page will retrieve a list of development tools, viewed in chapter 2, to be used for the user’s specific project. After these three milestones the user is shown a list of freelance and mobile experts that have the skills to accomplish their project. The preceding statement explains one possible scenario of the web platform that could support our MADSF Model.

From this decisional schema an algorithm can be deduced. A conditional algorithm is applied for this purpose.

6.3.3 From decisional schema to the Algorithm

The decisional schema is organized by decisional steps such as questions. Questions are translated into conditional blocks in terms of software development methods.

For visibility purposes the algorithm is illustrated on the next page.
Different colours below referenced represent the milestones seen in figure 17:

1. **Mobile Application Purpose** is represent in Orange
2. **Mobile Application Device Target** is represent in Dark blue
3. **Mobile Application Core Device API Utility** is represent in Green
4. **Mobile Application Update Process** is represent in Purple
5. **Mobile Application Quality Standards** is represent in Turquoise

This algorithm will be performed through several steps explained in the next section.
6.4 Method to support the model
The method used to support the MADSF model would be a web platform hosted on the cloud. The following chapter provides an overview of the solution. The objective of the web platform is explained, as well as its concept to conclude with a comparison between Titanium model and the MADSF model.

6.4.1 Objective of the web platform
As mentioned earlier, the website is built for developers and enterprises that desire to commence the implementation of a mobile application.

On one hand, the literature review discussed in the third and fifth sections significantly stressed the difficulty for companies to decide on which mobile strategy they will employ. The reason for this lack of confidence in the choice of a MADS include several factors such as the cost, the timing, and overall performance of the app. The different solutions comprise advantages on one hand and drawbacks on the other. Therefore, the choice becomes a serious issue if the developer or the enterprise in question is not familiar with mobile application development. In addition to this, even if the company is expert with these types of projects, a project manager or several engineers have to combine their forces for some time to come up with the best available solution for the amount quoted for the project.

On the other hand, the result of the online survey depicted some patterns in mobile application development that can be used for the good of others. Indeed, the chapter five gathered fundamental data that are integrated parts of the backend structure of this website. In fact, these crucial data constitute the ground information of the database of the website solution for choosing between different mobile application development methods.

Consequently, this solution aims at providing a web platform that will help developers to decide more efficiently the best mobile development strategy for a given project. The analysis of several scientific papers (see chapter 4) participated in establishing the most accurate questions that will lead to find the suitable mobile development strategy.

6.4.2 Concept
The concept of the website called “stepstoapp.com” is fairly simple to understand. The information previously gathered thanks to trustworthy developers and companies are used as example for other similar projects. In other words, if an enterprise has an idea of mobile application that resembles to one the mobile projects of the database, the solution used (SDK, Framework, and IDE) will be provided as a result of search on the website.
To be more explicit, the illustration below depicts how stepstoapp.com works:

![Diagram](image)

**Figure 21: First step of the concept of stepstoapp.com**

Users access stepstoapp.com and start by selecting their targeted mobile OS. The second step is to enter their desired app parameters. A backend process scrawl the database produced from the online survey to search a match with the desired mobile application project. It then retrieves the best possible match or the closest solutions utilized in similar projects included in the database. For instance, a user wants a radio app with a budget that allows covering for the best quality app available. If we follow the criteria table given in section 4.5 here would be the outcome:

- Performance expected: Good
- Cost: < $50k
- Core API device: Not necessary
- Code update: Not necessary
- Time: Flexible if less than 3 months
- Portability: iOS and Android
- Platform Stability: Safe
- Document support: Sufficient enough for this app
- Development skills: externalized

Therefore, the best possible fit for the mobile app development strategy would be first the hybrid approach because it fulfils the aforementioned criteria. It is robust enough to sustain the app maintenance; the user experience will be preserved, as well as the possibility of developing a cross platform application on iOS and Android for the expected budget. The second plausible approach would be to use native development except that the cost will be slightly more expensive.

To generalize, the companies or developers see all the different alternatives with an explanation on each solution with its pros and cons such that they can compare and choose the best possible fit for the project.

Advantages and disadvantages about development methods will be provided by the literature review constituted in the fourth chapter. Hence, stepstoapp.com will provide its user with complete answers and guidance for the development of their mobile application targeting single or cross-platforms.
The second goal of stepstoapp.com is to sustain the same level of data reliability. To achieve this, a second selection process is offered to the user, this selection represents whether the mobile application agencies or freelance mobile developer. The selection of one of the agencies/freelance will send a demand of quotation with the user’s project characteristics (OS/platform targeted, its features, and the development solution if selected). These data will help the developer or the mobile application agency, to determine an amount for the development of the above-mentioned project. The important aspect of this step is the confirmation from the mobile agency or the freelance developer of the development solution utilized for the project in question. Thereby, stepstoapp.com will update its database according to trustworthy professionals. Hence, the data is always up to date.

The illustration hereafter describes the above stated:

![Diagram](image-url)

**Figure 22: Second step of the concept of stepstoapp.com**

### 6.5 Comparison between MADSF Model and Titanium Model

The comparison between Titanium Model named “4 steps to creating a mobile development strategy” and this study MADSF Model reside in five milestones. First of all, the strengths and weaknesses of Titanium Model are assessed. Secondly, the same analysis is conducted to highlight the advantages and downsizes of this thesis MADSF Model. Lastly, a conclusion over the outlined artefacts is stated.

#### 6.5.1 Strengths of Titanium Model

Thanks to the analysis in section 6.2 of the paper mentioning the Titanium Model, the following advantages of this model can be extracted:

- **Overall structured paper**: every step is clearly stated by numbers and a model using slick graphics for a better understanding.
- **Tailored information**: this article objective’s is to drag customers and professionals to Titanium products. Enough information is provided for a customer to grasp a picture of the overall process of choosing a MADS.
- **Vulgarization of terms**: this means that the model is usable by non-developers people, typically CIO, CEO, marketing and business management people.

### 6.5.2 Weaknesses of Titanium Model

The analysis in section 6.2 of the paper mentioning the Titanium Model allowed the following drawbacks of their model to be extracted:

- **Business focused**: this paper is tailored to everyone. Nonetheless, the model is simplified to reach business-oriented people and therefore losses its technical precision.
- **Analysis in surface**: the paper does not emphasis more on the pros and cons of each MADS. It solely states them briefly. The reader does not have a clear picture of the overall causes of choosing a MADS over another.
- **Inflexible**: the model cannot be easily updated since Titanium should keep its knowledge up to date, which is difficult to realize in this fast pace evolution field.

### 6.5.3 Strengths of MADSF Model

Thanks to the analysis in section 6.2 of the paper mentioning the Titanium Model, the following advantages of this model can be extracted:

- **Ease in update**: thanks to the users input in the web platform, this model can be sustained and is able to evolve by itself.
- **Simple structure**: the enumeration of the milestone, the choice of the terms used, the slick and clear graphical aspect provides anyone none technical to understand the model with the help of the figure 17.
- **Thorough analysis**: made from the theoretical background to the literature review. It provides a complete understanding of the MADSF.

### 6.5.4 Weakness of MADSF Model

The analysis in section 6.2 of the paper mentioning the Titanium Model allowed the following drawback of their model to be extracted:

- **Deeper analysis of MADS**: the actual MADSF Model does not delve into frameworks and software development methods used for each MADS. Its only purpose it to redirect users to the tailored MADS.
6.5.5 Which Model to choose?

Both Models have their place in the ICT community, however they target different people and they have a different purpose. Titanium Model has been built to advocate the advantages of using their technology rather than another one. Instead, the MADSF Model provides people a clear way of finding their MADS. Moreover, This thesis aspires at combining several artefacts such as the theoretical background, the literature review and the survey’s analysis to provide a thorough overall dossier that developers/IT skilled people can utilise for their future mobile applications.

Consequently, if the users’ purpose is solely to find a brief paper that scratches the surface of the area, enough to provide users with an answer on how to find the catered MADS, the Titanium’s model should be considered. On the contrary, if the users’ objective is to the find their tailored MADS but also understand why one MADS is better than another for their specific case, they should utilise the MADSF Model.
7 Discussion

7.1 Reliability of the survey’s results

“Reliability refers to the extent to which your data collection techniques or analysis procedures will yield consistent findings. It can be assessed by posing the following three questions” [69]:

1. Will the measures yield the same results on other occasions?
2. Will similar observations be reached by other observers?
3. Is there transparency in how sense was made from the raw data?

The following sections elaborate on the aforementioned questions.

7.1.1 Will the measures yield the same results on other occasions?

The mobile application is a fast-pace evolution area such that Apple, at the time of this thesis is written, launched 7 evolutions of its mobile operating system in a matter of 6 years, the same process is followed by Android and Windows Phone. Indeed, they constantly need to redesign their products in order to bring innovation in people’s hands.

Thus, the survey’s results are tailored for a period of time that will be determined by a complete change in the Mobile Development Strategy finding process, which can happen at any time. Therefore, the sustainability of the result is confirmed until that innovation in Mobile Development Strategy finding.

7.1.2 Will similar observations be reached by other observers?

The sample taken for the survey is small nonetheless it represents a fairly good overview of the situation in terms of Mobile Development Strategy. Generalisation has been made based on these 30 responders because they come from different countries, they have different status such as mobile engineer, CEO, CTO, mobile developer and come from different professional domains. Hence, the sample found can be exchanged with other observers and the results will significantly be the same with some slight changes.

7.1.3 Is there transparency in how sense was made from the raw data?

Yes due to the analysis yielded after the raw data has been illustrated. Moreover, transparency definition is the quality of being truthful, honest and clear about something. The time and date of the responders’ answers is stamped in the table in Appendix 1, which means that transparency is met.

The aforementioned three questions are covered by this study. Therefore, Reliability is part of this thesis.
8 Conclusion

The mobile application domain is in constant evolution such that in the first quarterly of 2013 tablets exceeded traditional desktop devices for conversion rates for the first time [70].

Reports on the trends of mobile application development strategy, showcase the use of processes and criteria for finding a tailored MADS. These processes and criteria were extracted from the literature review and my professional experiences.

This study utilised different research approaches in order to build a solid research framework. The analysis of the literature review was conducted in a critical manner, which is called an analytical approach. Qualitative data collection has been conducted through an online survey in order to understand the behaviour of the responders towards the process of selecting a MADS. The design science research approach is utilised because of its improvement characteristic and problem-solving nature which correspond to this study. Indeed, the objective was to contribute in the mobile development area by bring a new model called MADSF to facilitate the process of finding a mobile application development strategy.

A recapitulation of the research questions shall assert the processes and criteria used by developers for choosing a MADS, and the suggestion of a MADSF Model. Ultimately, the scope for further research is highlighted in order to state possible enhancements of the Model.

8.1 Recapitulation of Research Questions

This section shall confirm and verify that the three research questions have been answered.

1. What processes developers perform for choosing a mobile development architecture?

This question is answered through the survey. The figure 12 illustrates the results of this same question. It showcases that 50% of the responders research frameworks, around 30% of them also utilize in-house guidelines and research similar projects, lastly almost 40% also use their own development skills to choose a MADS. These findings led to think that external guidelines are not often used with around 15% of the responders.

2. What are the criteria for which developers adopt a mobile application development strategy?

This question is answered through the literature review in section 4.5. The figure 10 illustrates the list of criteria that should be taken into account when seeking for a MADS. The list counts 12 criteria; each one of them has items targeting one of the three MADS focused. These criteria were extracted from
the literature review. Then they are utilized in the survey to verify if they are plausible criteria towards mobile applications professionals.

3. **How to provide developers and companies with a model that help them to choose their mobile development strategy?**

This question is answered in the previous chapter. The figure 18 depicts the conceptual model suggested for finding a MADS efficiently and without hassles. In addition to this, the decisional schema (Figure 19) determines through several steps questions the most suitable MADS for a specific mobile development project. From this decisional schema an algorithm was constructed. Finally, thanks to this same algorithm and the MADS decisional schema a plausible web platform could be extracted. The latter represents a method for supporting the model.

All questions found an answer through the conduction of this study.

### 8.2 Further investigation

Although this study has outlined the processes and criteria used by developers for choosing a MADS, and a MADSF Model to improve the process of choosing a MADS, scope exists for further research. Indeed, the model only provides a MADS. Nonetheless, each one of them has a multitude of frameworks and software development methods. The following future study can be investigated: Enhance the MADSF Model by delving into which frameworks and software development methods to employ for a MADS.
9 References


Center of Excellence in Knowledge-Enabled Computing, Wright State Univ., Dayton, unpublished.


## 10 Appendix

### 10.1 Survey results (1/3)

<table>
<thead>
<tr>
<th>Time Submitted</th>
<th>What is the name of your mobile application?</th>
<th>Which processes do developers conduct before applying a mobile development architecture?</th>
<th>What are the criteria for which developers adopt to choose your mobile development strategy?</th>
<th>How long did you take which mobile application development solution/IDE have you used to develop your mobile app?</th>
<th>E-mail Your email will not be used for any commercial purposes.</th>
<th>Do you have any comments on how to improve this questionnaire?</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 20, 2013</td>
<td>myAudi France</td>
<td>Research of frameworks; Use of developers expertise</td>
<td>Performance; Time; Quality</td>
<td>2 - 4 days</td>
<td>Xcode</td>
<td><a href="mailto:william@appsolute.fr">william@appsolute.fr</a></td>
</tr>
<tr>
<td>3:52 PM</td>
<td></td>
<td>Research of similar projects; Research of frameworks</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 20, 2013</td>
<td>Google Keep</td>
<td>Research of frameworks; Use of developers expertise</td>
<td>Performance; Time; Quality</td>
<td>2 - 4 days</td>
<td>Eclipse</td>
<td></td>
</tr>
<tr>
<td>2:50 PM</td>
<td></td>
<td>Research of frameworks; Research of frameworks</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 19, 2013</td>
<td>was not published</td>
<td>Research of frameworks; Use of developers expertise</td>
<td>Performance; Code Update; Portability</td>
<td>2 - 4 days</td>
<td>Eclipse; Phonegap</td>
<td><a href="mailto:remi.cambuzat@gmail.com">remi.cambuzat@gmail.com</a></td>
</tr>
<tr>
<td>10:48 PM</td>
<td></td>
<td>Research of frameworks; Use of developers expertise</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 19, 2013</td>
<td>Matrix</td>
<td>Research of frameworks; Use of developers expertise</td>
<td>Performance; Quality; Code Update</td>
<td>2 - 4 days</td>
<td>Eclipse</td>
<td></td>
</tr>
<tr>
<td>9:47 PM</td>
<td></td>
<td>Research of frameworks; Use of developers expertise</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 19, 2013</td>
<td>Just Word It</td>
<td>Research of frameworks; Use of developers expertise</td>
<td>Performance; User Interface; Quality</td>
<td>4 - 6 days</td>
<td>Xcode</td>
<td></td>
</tr>
<tr>
<td>5:46 PM</td>
<td></td>
<td>Research of frameworks; Use of developers expertise</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 19, 2013</td>
<td>Wallet</td>
<td>Research of similar projects; Use of developers expertise</td>
<td>Time; Quality; User Interface</td>
<td>2 - 4 days</td>
<td>Eclipse</td>
<td><a href="mailto:a.darivon@gmail.com">a.darivon@gmail.com</a></td>
</tr>
<tr>
<td>10:45 AM</td>
<td></td>
<td>Research of similar projects; Use of developers expertise</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 19, 2013</td>
<td>Total</td>
<td>Research of similar projects; Use of developers expertise</td>
<td>Time; Portability; Performance</td>
<td>2 - 4 days</td>
<td>Eclipse; Phonegap</td>
<td><a href="mailto:a.darivon@gmail.com">a.darivon@gmail.com</a></td>
</tr>
<tr>
<td>8:44 AM</td>
<td></td>
<td>Research of similar projects; Use of developers expertise</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 18, 2013</td>
<td>Co-Voiturage</td>
<td>Research of frameworks; Performance; Dev. resources; Portability</td>
<td></td>
<td>4 - 6 days</td>
<td>Eclipse; Phonegap</td>
<td><a href="mailto:a.darivon@gmail.com">a.darivon@gmail.com</a></td>
</tr>
<tr>
<td>7:13 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 18, 2013</td>
<td>Roland Garros</td>
<td>Research of frameworks; Performance; Dev. resources; Portability</td>
<td></td>
<td>2 - 4 days</td>
<td>Xcode; Eclipse; Phonegap</td>
<td><a href="mailto:a.darivon@gmail.com">a.darivon@gmail.com</a></td>
</tr>
<tr>
<td>6:42 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 18, 2013</td>
<td>City Anecdotes</td>
<td>Research of frameworks; Use of in-house guidelines</td>
<td>Performance; Dev. resources; Quality</td>
<td>2 - 4 days</td>
<td>Xcode</td>
<td><a href="mailto:maxence@ugly-ducklings.com">maxence@ugly-ducklings.com</a></td>
</tr>
<tr>
<td>3:49 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September 18, 2013</td>
<td>Shahoots</td>
<td>Research of frameworks; Use of in-house guidelines</td>
<td>Performance; Time; Quality</td>
<td>4 - 6 days</td>
<td>Xcode</td>
<td><a href="mailto:fradow@gmail.com">fradow@gmail.com</a></td>
</tr>
<tr>
<td>3:40 PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## 10.2 Survey results (2/3)

<table>
<thead>
<tr>
<th>Time Submitted</th>
<th>What is the name of your mobile application?</th>
<th>Which processes do developers conduct before applying a mobile development architecture?</th>
<th>What are the criteria for which developers adopt a mobile application development strategy? (Native, Hybrid, or Web)?</th>
<th>How long did you take to choose your mobile development strategy? (Native, Hybrid, or Web)?</th>
<th>Which mobile application development solution/IDE have you used to develop your mobile app?</th>
<th>E-mail Your email will not be used for any commercial purposes.</th>
<th>Do you have any comments on how to improve this questionnaire?</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 18, 2013</td>
<td>Karate Quiz</td>
<td>Research of frameworks; Use of in-house guidelines</td>
<td>Time; Quality; User Interface</td>
<td>+ 6 days</td>
<td>Xcode</td>
<td><a href="mailto:fradow@gmail.com">fradow@gmail.com</a></td>
<td>Do you have any comments on how to improve this questionnaire?</td>
</tr>
<tr>
<td>11:39 AM</td>
<td></td>
<td>Research of similar projects; Exploration of Forums</td>
<td>Performance; Cost; Quality</td>
<td>2 - 4 days</td>
<td>Eclipse</td>
<td><a href="mailto:monjaud@efrei.fr">monjaud@efrei.fr</a></td>
<td></td>
</tr>
<tr>
<td>September 18, 2013</td>
<td>EdT Efrei</td>
<td>Use of in-house guidelines; Comparison analysis with previous projects</td>
<td>Cost; Time; User Interface</td>
<td>2 - 4 days</td>
<td>Eclipse</td>
<td><a href="mailto:carlier.yannick@gmail.com">carlier.yannick@gmail.com</a></td>
<td></td>
</tr>
<tr>
<td>11:25 AM</td>
<td></td>
<td>Research of similar projects; Feasibility studies</td>
<td>Performance; User Interface; Quality</td>
<td>+ 6 days</td>
<td>Xcode</td>
<td><a href="mailto:nicolas.marsaud@feelandclic.com">nicolas.marsaud@feelandclic.com</a></td>
<td></td>
</tr>
<tr>
<td>September 17, 2013</td>
<td>None released yet</td>
<td>Research of similar projects; Use of in-house guidelines</td>
<td>Performance; User Interface</td>
<td>2 - 4 days</td>
<td>Xcode</td>
<td><a href="mailto:mariaisa.galbert@gmail.com">mariaisa.galbert@gmail.com</a></td>
<td></td>
</tr>
<tr>
<td>3:00 PM</td>
<td></td>
<td>Comparison analysis with previous projects</td>
<td>Performance; User Interface</td>
<td>2 - 4 days</td>
<td>Xcode</td>
<td><a href="mailto:mariaisa.galbert@gmail.com">mariaisa.galbert@gmail.com</a></td>
<td></td>
</tr>
<tr>
<td>September 16, 2013</td>
<td>Getboox</td>
<td>Research of similar projects; Feasibility studies</td>
<td>Performance; User Interface; Quality</td>
<td>+ 6 days</td>
<td>Xcode</td>
<td><a href="mailto:nicolas.marsaud@feelandclic.com">nicolas.marsaud@feelandclic.com</a></td>
<td></td>
</tr>
<tr>
<td>10:48 PM</td>
<td></td>
<td>Research of similar projects; Use of in-house guidelines</td>
<td>Performance; User Interface</td>
<td>2 - 4 days</td>
<td>Xcode</td>
<td><a href="mailto:mariaisa.galbert@gmail.com">mariaisa.galbert@gmail.com</a></td>
<td></td>
</tr>
<tr>
<td>September 16, 2013</td>
<td>Coinche gratuit</td>
<td>Research of similar projects; Use of in-house guidelines</td>
<td>Performance; User Interface</td>
<td>2 - 4 days</td>
<td>Xcode</td>
<td><a href="mailto:mariaisa.galbert@gmail.com">mariaisa.galbert@gmail.com</a></td>
<td></td>
</tr>
<tr>
<td>2:29 PM</td>
<td></td>
<td>Research of similar projects; Use of in-house guidelines</td>
<td>Performance; User Interface</td>
<td>2 - 4 days</td>
<td>Xcode</td>
<td><a href="mailto:mariaisa.galbert@gmail.com">mariaisa.galbert@gmail.com</a></td>
<td></td>
</tr>
<tr>
<td>September 16, 2013</td>
<td>belote gratuit</td>
<td>Research of similar projects; Use of in-house guidelines</td>
<td>Performance; User Interface</td>
<td>2 - 4 days</td>
<td>Xcode</td>
<td><a href="mailto:mariaisa.galbert@gmail.com">mariaisa.galbert@gmail.com</a></td>
<td></td>
</tr>
<tr>
<td>11:28 AM</td>
<td></td>
<td>Research of similar projects; Use of in-house guidelines</td>
<td>Performance; User Interface</td>
<td>2 - 4 days</td>
<td>Xcode</td>
<td><a href="mailto:mariaisa.galbert@gmail.com">mariaisa.galbert@gmail.com</a></td>
<td></td>
</tr>
<tr>
<td>September 16, 2013</td>
<td>belote Coinche</td>
<td>Research of similar projects; Use of in-house guidelines</td>
<td>Performance; User Interface</td>
<td>2 - 4 days</td>
<td>Xcode</td>
<td><a href="mailto:mariaisa.galbert@gmail.com">mariaisa.galbert@gmail.com</a></td>
<td></td>
</tr>
<tr>
<td>10:45 AM</td>
<td></td>
<td>Research of frameworks; Use of external guidelines</td>
<td>Performance; User Interface; Quality</td>
<td>+ 6 days</td>
<td>Xcode</td>
<td><a href="mailto:mariaisa.galbert@gmail.com">mariaisa.galbert@gmail.com</a></td>
<td></td>
</tr>
<tr>
<td>September 14, 2013</td>
<td>Sush.io</td>
<td>Research of frameworks; Use of external guidelines</td>
<td>Dev. resources; Code Update</td>
<td>2 - 4 days</td>
<td>Xcode</td>
<td><a href="mailto:mariaisa.galbert@gmail.com">mariaisa.galbert@gmail.com</a></td>
<td></td>
</tr>
<tr>
<td>12:21 PM</td>
<td></td>
<td>Research of frameworks; Use of external guidelines</td>
<td>Performance; Quality; Code Update</td>
<td>2 - 4 days</td>
<td>Xcode</td>
<td><a href="mailto:mariaisa.galbert@gmail.com">mariaisa.galbert@gmail.com</a></td>
<td></td>
</tr>
<tr>
<td>September 14, 2013</td>
<td>Quickscan</td>
<td>Research of frameworks; Use of external guidelines</td>
<td>Performance; Quality; Code Update</td>
<td>2 - 4 days</td>
<td>Xcode</td>
<td><a href="mailto:mariaisa.galbert@gmail.com">mariaisa.galbert@gmail.com</a></td>
<td></td>
</tr>
<tr>
<td>11:20 AM</td>
<td></td>
<td>Research of frameworks; Use of external guidelines</td>
<td>Performance; Quality; Code Update</td>
<td>2 - 4 days</td>
<td>Xcode</td>
<td><a href="mailto:mariaisa.galbert@gmail.com">mariaisa.galbert@gmail.com</a></td>
<td></td>
</tr>
</tbody>
</table>
## 10.3 Survey results (3/3)

<table>
<thead>
<tr>
<th>Time Submitted</th>
<th>What is the name of your mobile application?</th>
<th>Which processes do developers conduct before applying a mobile application development architecture?</th>
<th>What are the criteria for which developers adopt a mobile application development strategy?</th>
<th>How long did you take to choose your mobile application development strategy (Native, Hybrid, or web)?</th>
<th>Which mobile application development solution/IDE have you used to develop your mobile app?</th>
<th>Do you have any comments on how to improve this questionnaire?</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 14, 2013 10:20 AM</td>
<td>Viewer de documentation technique dans le domaine de l’aéronautique</td>
<td>Use of developers expertise; Comparison analysis with previous projects</td>
<td>Performance; Code Update; Portability</td>
<td>2 - 4 days</td>
<td>Xcode; Eclipse; Phonegap; Web app</td>
<td><a href="mailto:jegouic@efrei.fr">jegouic@efrei.fr</a></td>
</tr>
<tr>
<td>September 13, 2013 10:23 PM</td>
<td>Autimo</td>
<td>Research of frameworks; Use of external guidelines</td>
<td>Time; Quality; Core device API</td>
<td>2 - 4 days</td>
<td>Xcode; Eclipse</td>
<td><a href="mailto:fradow@gmail.com">fradow@gmail.com</a></td>
</tr>
<tr>
<td>September 13, 2013 9:24 PM</td>
<td>Blagues de merde</td>
<td>Use of external guidelines</td>
<td>Performance; User Interface</td>
<td>0 - 2 days</td>
<td>Eclipse; Visual Studio for Windows Phone 7/8</td>
<td><a href="mailto:olivier@blaguesdemerde.fr">olivier@blaguesdemerde.fr</a></td>
</tr>
<tr>
<td>September 13, 2013 5:38 PM</td>
<td>Social Tv Livebattle</td>
<td>Research of frameworks; Comparison analysis with previous projects</td>
<td>Performance; User Interface; Portability</td>
<td>4 - 6 days</td>
<td>Xcode; Phonegap; Web app</td>
<td><a href="mailto:a.darivon@gmail.com">a.darivon@gmail.com</a></td>
</tr>
<tr>
<td>September 13, 2013 4:48 PM</td>
<td>Jeu de mémoire</td>
<td>Research of similar projects; Use of in-house guidelines</td>
<td>User Interface; Dev. resources; Platform stability</td>
<td>0 - 2 days</td>
<td>Eclipse; Bada library</td>
<td><a href="mailto:thomas.guillard.efrei@gmail.com">thomas.guillard.efrei@gmail.com</a></td>
</tr>
<tr>
<td>September 13, 2013 11:38 AM</td>
<td>Ucky in Paris</td>
<td>Feasibility studies; Use of developers expertise</td>
<td>Cost; Time; User Interface</td>
<td>+ 6 days</td>
<td>Xcode</td>
<td><a href="mailto:maxence@lerigner.com">maxence@lerigner.com</a></td>
</tr>
<tr>
<td>September 12, 2013 6:37 PM</td>
<td>Bloomee - Culture Générale</td>
<td>Research of frameworks; Use of external guidelines</td>
<td>Performance; Portability; User Interface</td>
<td>0 - 2 days</td>
<td>Eclipse</td>
<td><a href="mailto:hugues.malinie@gmail.com">hugues.malinie@gmail.com</a></td>
</tr>
<tr>
<td>September 12, 2013 6:35 PM</td>
<td>NSoul, ...</td>
<td>Feasibility studies; Use of developers expertise</td>
<td>Portability; Quality; Code Update</td>
<td>0 - 2 days</td>
<td>Xcode; Eclipse; Visual Studio for Windows Phone 7/8; Android Studio</td>
<td><a href="mailto:morgan.collino@gmail.com">morgan.collino@gmail.com</a></td>
</tr>
<tr>
<td>September 12, 2013 11:45 AM</td>
<td>Plan me up</td>
<td>Use of developers expertise; Feasibility studies</td>
<td>Time; Performance; Quality</td>
<td>0 - 2 days</td>
<td>Eclipse</td>
<td><a href="mailto:edgard@mbayen.com">edgard@mbayen.com</a></td>
</tr>
</tbody>
</table>

This application was made as part of a junior enterprise bid.