“A case study in Mobile Software Engineering: Artemis, a mobile application to combat violence against women.”

By

Douglas Barbosa

B.Sc. Dissertation

SALVADOR, July/2014
Douglas Barbosa

“A case study in Mobile Software Engineering: Artemis, a mobile application to combat violence against women.”

A B.Sc. Dissertation presented to the Computer Science Department of Federal University of Bahia in partial fulfillment of the requirements for the degree of Bachelor in Computer Science.

Advisor: Eduardo Santana de Almeida

SALVADOR, July/2014
I dedicate this dissertation to my parents, friends and girlfriend. Without them I would not have got here. I also dedicate it to all women victim of violence, hoping this work may help them and all people who fight against the violence.
Acknowledgements

First, I would like to thank my mother Vilma and my father Vitomar, they always worked hard to make all of this possible. Every achievement in my life, from the smaller one to the big victories are entirely dedicated to them.

I also would like to thank my friends, especially Danilo Rocha, who was with me through the entire journey at UFBA, and to Sayonara, who gave an insight about my project that made possible to develop Artemis the way it is was, you are awesome guys.

I would like to express my gratitude to Eduardo Almeida, my advisor that worked with me since my beginning at the graduation and to Raphael Oliveira, who made a great first revision of my dissertation.

I would like to thank Ananda Ferreira, Firmiade Venâncio, Márcia Tavares and Ana Virgínia, all of them are brave women that fight everyday against the violence against the women and had the patience to talk with me in order to make me understand this problem.

Finally, I would like to thank Thaíla, my girlfriend. She always was there in the hard times during this journey and gave me the courage to keep going. You are my way home.
It is not our part to master all the tides of the world, but to do what is in us for the succour of those years wherein we are set, uprooting the evil in the fields that we know, so that those who live after may have clean earth to till. What weather they shall have is not ours to rule.

—J.R.R. TOLKIEN (The Lord of The Rings)
A engenharia de software vem enfrentando diversos desafios desde que foi formulada como disciplina, desde então diversos paradigmas surgiram, alguns se consolidaram e outros simplesmente desapareceram com o tempo. Seus conceitos vem evoluindo e sendo utilizado a vários anos, sempre se adaptando a novas realidades, com a computação móvel não poderia ser diferente.

A computação móvel foi introduzida já a alguns anos no contexto da computação, mas apenas de poucos anos para cá, motivada pela rápida evolução dos dispositivos móveis, que houve um grande avanço e popularização deste paradigma. Novas tecnologias foram desenvolvidas e por fim foram lançados os primeiros kits de desenvolvimento para plataformas móveis, o que possibilitou ao desenvolvedor uma forma completamente nova de criar e disponibilizar suas aplicações.

A evolução do software móvel acompanhou a evolução do hardware, que a cada lançamento trazia uma nova possibilidade para o desenvolvedor. Porém essa facilidade em desenvolver e publicar para plataformas móveis cobrou seu preço, as aplicações foram evoluindo e ficando mais complexas, mas pouca ou nenhuma engenharia de software foi sendo aplicada no desenvolvimento dessas aplicações, ou então eram utilizadas ferramentas que não cabem no contexto móvel. E com cada vez mais desenvolvedores buscando entrar no mundo móvel, foram trazidos alguns conceitos baseados no desenvolvimento para a web que não se encaixam perfeitamente com os paradigmas da computação móvel.

Com isso, objetivo deste trabalho é apresentar os principais aspectos da computação móvel e mostrar a necessidade de se aplicar engenharia de software nesse contexto, de forma que aplicações possam ser desenvolvidas explorando todo o potencial das tecnologias móveis. Ainda como parte desse trabalho, uma solução móvel para atuar em um problema da vida real, que é a violência contra a mulher é proposta, dessa forma mostra-se que a computação e o software móvel podem atuar em problemas de nosso dia-a-dia de forma a melhorar nossa vida.

**Palavras-chave:** engenharia de software móvel, computação móvel, desenvolvimento de aplicações, ambientes de desenvolvimento, experiência de usuário, computação ubíqua, problemas da vida real
Abstract

Software engineering has faced many challenges since it was formulated as a discipline, since several paradigms have emerged, some have matured and others simply disappeared with time. Its concepts are evolving and being used for several years, always adapting to new realities, with mobile computing could not be different.

Mobile computing has been introduced in the context of computing for a while now, but only a few years ago, driven by the rapid evolution of mobile devices, there was a breakthrough and popularization of this paradigm. New technologies were developed and eventually were released the first development kits for mobile platforms, presenting to the developer a completely new way to create and deliver their applications.

The evolution of mobile software followed the evolution of hardware, which in each new launch brought a new possibility for the developer. But this facility to develop and publish to mobile platforms took its toll, applications have evolved and become more complex, but little or no software engineering was being applied in the development of these applications, or tools were used that do not fit in the mobile context. And more developers seeking to enter the mobile world has brought some concepts based on web development that does not fit perfectly with the paradigms of mobile computing.

The objective of this dissertation is to present the key aspects of mobile computing and show the need for applying software engineering in this context, thus applications can be developed exploiting the full potential of mobile technologies. As part of this work it is also proposed a mobile solution to work in a real life problem, which is violence against women. Thus it is shown that computing and mobile software can act on problems of our daily life, in order to make it better.

Keywords: mobile software engineering, mobile computing, applications development, development environments, user experience, ubiquitous computing, real life problems
## Table of Contents

List of Figures xi  
List of Tables xii  
List of Acronyms xiii  

1 Introduction 1  
1.1 Motivation ............................................. 2  
1.2 Scope .................................................. 2  
1.3 Out of Scope .......................................... 2  
1.4 Statement of the Contributions ....................... 3  
1.5 Dissertation Structure ................................. 3  

2 An Overview on Mobile Software Engineering 5  
2.1 Introduction ........................................... 5  
2.2 Beyond Good Practices, the Complexity and Quality of Mobile Software 7  
2.3 Mobile Computing Aspects ............................. 10  
2.3.1 Great diversity of environments and platforms ......... 11  
2.3.2 Sensors and Device Features ....................... 14  
2.3.3 User Experience ..................................... 15  
2.3.4 Cloud and Ubiquitous Computing ................... 20  
2.3.5 Agile vs. Plan-Based Development .................. 21  
2.3.6 Non-Functional Requirements and Architecture ....... 23  
2.4 Mobile Next Generation ............................... 24  
2.5 Summary ............................................... 25  

3 Mobile Software and Real Life Problems 27  
3.1 Introduction ........................................... 27  
3.2 Data Collection ....................................... 28  
3.2.1 Setting Objectives .................................. 29  
3.2.2 Survey Design ...................................... 31  
3.2.3 Developing the Survey Instrument .................. 31  
3.2.4 Evaluating Survey Instruments ..................... 32  
3.2.5 Obtaining the data .................................. 33  
3.2.6 Analyzing the data .................................. 34
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>Results</td>
<td>34</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Questionnaires Analysis</td>
<td>34</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Interview Analysis</td>
<td>39</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Threats to Validity</td>
<td>41</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Findings</td>
<td>41</td>
</tr>
<tr>
<td>3.4</td>
<td>Summary</td>
<td>42</td>
</tr>
<tr>
<td>4</td>
<td>Artemis - A Mobile Application to Combat Violence Against Women</td>
<td>43</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>43</td>
</tr>
<tr>
<td>4.2</td>
<td>The Development Process</td>
<td>44</td>
</tr>
<tr>
<td>4.3</td>
<td>Requirements Set</td>
<td>47</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Functionalities</td>
<td>47</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Functional Requirements</td>
<td>48</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Non-Functional Requirements</td>
<td>50</td>
</tr>
<tr>
<td>4.4</td>
<td>Architecture Overview</td>
<td>51</td>
</tr>
<tr>
<td>4.5</td>
<td>Architecture Components</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>Contact Controller</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>User Controller</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>Login Controller</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>Artemis Server Handler</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Beacon Handler</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Location Handler</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>WebService Handler</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Artemis Server</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>WebService API and Push Server</td>
<td>56</td>
</tr>
<tr>
<td>4.6</td>
<td>Technologies</td>
<td>56</td>
</tr>
<tr>
<td>4.7</td>
<td>Implementation</td>
<td>57</td>
</tr>
<tr>
<td>4.8</td>
<td>Artemis in action</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Main Screen</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Proximity Detection</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Send Emergency Alerts</td>
<td>64</td>
</tr>
<tr>
<td>4.9</td>
<td>Summary</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>Conclusion</td>
<td>66</td>
</tr>
<tr>
<td>5.1</td>
<td>Research Contribution</td>
<td>66</td>
</tr>
<tr>
<td>5.2</td>
<td>Future Work</td>
<td>67</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>Bibliography</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>Appendix</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>A Survey Instruments</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>A.1 Form for Survey</td>
<td>74</td>
<td></td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Dissertation structure</td>
<td>4</td>
</tr>
<tr>
<td>2.1</td>
<td>Mobile Computing Core Areas</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Samsung devices variety</td>
<td>15</td>
</tr>
<tr>
<td>2.3</td>
<td>Screen Confort Zones</td>
<td>18</td>
</tr>
<tr>
<td>2.4</td>
<td>Example of a Sitemap</td>
<td>19</td>
</tr>
<tr>
<td>2.5</td>
<td>Plan-based and agile developments</td>
<td>22</td>
</tr>
<tr>
<td>2.6</td>
<td>The organization of the MVC</td>
<td>24</td>
</tr>
<tr>
<td>4.1</td>
<td>Spiral process proposed</td>
<td>44</td>
</tr>
<tr>
<td>4.2</td>
<td>Artemis’ architecture conceptual view</td>
<td>52</td>
</tr>
<tr>
<td>4.3</td>
<td>Artemis’ architecture development view</td>
<td>53</td>
</tr>
<tr>
<td>4.4</td>
<td>Artemis main screen - main user</td>
<td>59</td>
</tr>
<tr>
<td>4.5</td>
<td>Artemis main screen - contact user</td>
<td>59</td>
</tr>
<tr>
<td>4.6</td>
<td>Proximity detected - main user</td>
<td>61</td>
</tr>
<tr>
<td>4.7</td>
<td>Proximity detected - contact user</td>
<td>61</td>
</tr>
<tr>
<td>4.8</td>
<td>Main user and aggressor’s location - main user</td>
<td>62</td>
</tr>
<tr>
<td>4.9</td>
<td>Main user and aggressor’s location - contact user</td>
<td>62</td>
</tr>
<tr>
<td>4.10</td>
<td>Main user starting an emergenvy alert</td>
<td>63</td>
</tr>
<tr>
<td>4.11</td>
<td>Contact receiving a push notification about the emergency alert</td>
<td>63</td>
</tr>
<tr>
<td>4.12</td>
<td>Main user location in an emergency alert</td>
<td>64</td>
</tr>
<tr>
<td>4.13</td>
<td>Main user screen after starting an emergency alert</td>
<td>65</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Top Five Smartphone Vendors and Shipments</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Top Five Smartphone Operating Systems and Shipments</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>Cross-platform vs. native development</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Comparative analysis of cross-platform development approaches</td>
<td>14</td>
</tr>
<tr>
<td>4.1</td>
<td>Artemis code statistics</td>
<td>58</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>Mobile Software Engineering</td>
<td></td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
<td></td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
<td></td>
</tr>
<tr>
<td>EXT C</td>
<td>Number of External Calls</td>
<td></td>
</tr>
<tr>
<td>CYCLO</td>
<td>Cyclomatic Complexity</td>
<td></td>
</tr>
<tr>
<td>LOC</td>
<td>Lines of Code</td>
<td></td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
<td></td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
<td></td>
</tr>
<tr>
<td>APP</td>
<td>Application</td>
<td></td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
<td></td>
</tr>
<tr>
<td>IA</td>
<td>Information Architecture</td>
<td></td>
</tr>
<tr>
<td>MVC</td>
<td>Model-View-Controller</td>
<td></td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
<td></td>
</tr>
<tr>
<td>SE</td>
<td>Software Engineering</td>
<td></td>
</tr>
</tbody>
</table>
1

Introduction

*You’re in trouble, program. Why don’t you make it easy on yourself?*

*Who’s your user?*

—MASTER CONTROL PROGRAM  (Tron)

Nowadays mobile computing is present in many ways in people’s life. Smartphones and tablets are products widely consumed many group ages, from children to older people. With the massification of mobile technology lots of developers jumped into the mobile world, creating several types of applications and selling them on the app stores.

In fact, the modern mobile phone is capable of doing nearly everything the user can do with a desktop computer, but with the potential for more meaningful relevance to daily activities (Fling, 2009). Thus, the applications began to get more complex with the evolution of the mobile devices and their capabilities. Every day people find applications that can help them into their daily tasks or just for entertainment.

With this mobile evolution, the software engineering had to adapt some of its old concepts into the mobile context. The focus of this dissertation is to show the main mobile aspects, that makes it different from other kind of computing technology and to present the place of software engineering in this context. Also, it is shown how the mobile computing can help users to solve real life problems, to achieve this purpose, it will be presented the problem of violence against women and a mobile solution for it.

This chapter presents the main goals of this dissertation and starts by presenting its motivation in Section 1.1. Then the scope of the problem is defined in Section 1.2 and the aspects not covered by the work are presented in Section 1.3. Section 1.4 describes
the main contributions and, finally, Section 1.5 outlines the structure of this dissertation.

1.1 Motivation

While mobile computing evolves, the applications developed and the Mobile Software Engineering (MSE) must evolve in the same rhythm. Development of the mobile application will typically be done within the context of the overall software development effort (Wasserman, 2010). This could compromise the mobile user experience. Therefore, the unique qualities of the mobile environment makes it important not to treat the mobile application as an afterthought, but rather as an independent task with its own software engineering process and product requirements (Wasserman, 2010).

Use mobile technology to solve daily problems is getting more common, so another motivation was to investigate a real life problem. The violence against women was the problem chosen, since it is a serious problem that reaches many women in Brazil. A software solution, especially a mobile one, could help to minimize the problem and help women to avoid violence.

Understand the aspects of mobile computing and MSE is essential to develop applications that can really help problems like the violence mentioned. Also, this understanding is important in order to use the capabilities of the current and the future mobile computing.

1.2 Scope

This work aims to deliver a mobile application to act in the presented real life problem. The application will follow the aspects and constraints of mobile development and apply Mobile Software Engineering (MSE) in the development process. The mobile software functionalities were based on a survey conducted in order to learn about the violence against women and use the knowledge gained to define the requirements of the application.

1.3 Out of Scope

The following aspects were not considered in this work:

- **Multi platforms**

  Although the architecture of the software developed was made considering a cross-platform approach, its development was made for only one platform, leaving the
portability for other mobile platforms as a future work.

• **Mobile network and security**

When describing the aspects of mobile computing it was only presented aspects and constraints regarding mobile development and Mobile Software Engineering (MSE). The aspects of mobile network and security (other important areas of mobile computing) were not considered, leaving it for a future work.

### 1.4 Statement of the Contributions

As a result of the work presented in this dissertation, the following contributions can be highlighted:

- **A study on Mobile Software Engineering**

  This work can show how software engineering can be applied on a mobile context, presenting the main characteristics of mobile development.

- **A study on a real life problem**

  This work presents the problem of violence against women. A survey was conducted in order to understand the problem and to gather information in order to develop an application to solve, or minimize, it.

- **A mobile application to combat violence against women**

  Using MSE and the information gathered from the survey made to understand the real life problem, a mobile application was developed to assist the problem of violence against women.

### 1.5 Dissertation Structure

The remainder of this dissertation is organized as follows:

- **Chapter 2** presents an overview of mobile computing and shows how software engineering can be applied in a mobile context.

- **Chapter 3** describes the survey made in order to learn about the real life problem and define the functionalities and requirements of the application.
CHAPTER 1. INTRODUCTION

- **Chapter 4** presents the software proposed in order to combat violence against women.

- **Chapter 5** concludes the dissertation, providing the main contributions and directions for future work.

Figure 1.1 shows the dissertation structure.

![Dissertation structure diagram]

**Figure 1.1** Dissertation structure
An Overview on Mobile Software Engineering

The code is more what you’d call 'guidelines’ than actual rules.
—CAPTAIN HECTOR BARBOSSA (Pirates of the Caribbean)

2.1 Introduction

The application development for mobile devices is been around for many years. However, an exponential growth can be observed in this field, since Apple’s App Store launched in 2008 (Wasserman, 2010).

Along with the development kits, the developer is holding in his hands tools which would help the development process from designing the application to the release for the final user. In this context, other software and hardware companies opened their own application stores. Examples are Google, BlackBerry, Nokia, Microsoft among others, which entered the mobile market with Software Development Kit (SDK) for their respective mobile platforms.

The mobile application market for these platforms revealed itself as an attractive opportunity for developers, who now had a great variety of environments for their software, each environment with its own structure for application distribution and merchandising, and a well diversified public; plus the high and increasing mobile device shipment number, which grows rapidly year after year. Only in 2013, more than a billion smartphones were shipped around the world (IDC, 2014b), as showed in Table 2.1.

It is important to highlight that the mobile market is not only made of smartphones. Beyond tablets, a popular mobile device among users and developers, nowadays a great
Table 2.1 Top Five Smartphone Vendors and Shipments (Units in Millions) (IDC, 2014b)

<table>
<thead>
<tr>
<th>Vendor</th>
<th>2013 Shipment Volumes</th>
<th>2012 Shipment Volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samsung</td>
<td>313.9</td>
<td>219.7</td>
</tr>
<tr>
<td>Apple</td>
<td>153.4</td>
<td>135.9</td>
</tr>
<tr>
<td>Huawei</td>
<td>48.8</td>
<td>29.1</td>
</tr>
<tr>
<td>LG</td>
<td>47.7</td>
<td>26.3</td>
</tr>
<tr>
<td>Lenovo</td>
<td>45.5</td>
<td>23.7</td>
</tr>
<tr>
<td>Others</td>
<td>394.9</td>
<td>290.5</td>
</tr>
<tr>
<td>Total</td>
<td>1,004.2</td>
<td>725.3</td>
</tr>
</tbody>
</table>

variety of mobile devices are fleeing the conventional, such as watches, cameras, and other devices. Gadgets like Fitbit\(^1\), that tracks the user’s movements, heartbeats, sleep patterns and overall health, and Google Glass \(^2\), a pervasive device that uses augmenting reality to interact with users (Edmondson \textit{et al.}, 2014), are some examples of that. Many of these devices run under Android platform, and/or can interact with apps running on other devices, like smartphones.

The trend is that more and more mobile devices emerge to feed the need of the users to be constantly connected and to have an easy access to software tools and apps which will help with their daily tasks. Thereat, it is believed that mobile computing is just starting, and the next generations of mobile technology are going to be even more pervasive, smaller, and with functionalities not very usual for users right now (Edmondson \textit{et al.}, 2014). In fact, mobile devices are already taking place in computer market and users are buying fewer desktop computers, instead they have been purchasing new mobile technology to complement their personal computer needs (Edmondson \textit{et al.}, 2014).

As mentioned, there are a huge variety of mobile devices and platforms and each platform has its own set of tools and each device with its own screen size, hardware capacity, and specific sensors, such as Global Positioning System (GPS), camera, and accelerometer. These aspects arise one of the main challenge for developers: create mobile applications which can be used on these platforms and sold on their specific marketplaces independently from the device, and exploiting the full set of the device features (Francesce \textit{et al.}, 2013). With that in mind, multi-platform environments were created to try to reach as many of this diversity as possible, such as Phonegap\(^3\) and Titanium\(^4\) are examples of that.

\(^1\)http://www.fitbit.com
\(^2\)https://www.google.com/glass/start/
\(^3\)http://phonegap.com
\(^4\)http://www.appcelerator.com/titanium/
2.2. BEYOND GOOD PRACTICES, THE COMPLEXITY AND QUALITY OF MOBILE SOFTWARE

<table>
<thead>
<tr>
<th>Operating System</th>
<th>2013 Shipment Volumes</th>
<th>2012 Shipment Volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>793.6</td>
<td>500.1</td>
</tr>
<tr>
<td>iOS</td>
<td>153.4</td>
<td>135.9</td>
</tr>
<tr>
<td>Windows Phone</td>
<td>33.4</td>
<td>17.5</td>
</tr>
<tr>
<td>BlackBerry</td>
<td>19.2</td>
<td>32.5</td>
</tr>
<tr>
<td>Others</td>
<td>10.0</td>
<td>39.3</td>
</tr>
<tr>
<td>Total</td>
<td>1009.6</td>
<td>725.3</td>
</tr>
</tbody>
</table>

Table 2.2 Top Five Smartphone Operating Systems and Shipments (Units in Millions) (IDC, 2014a)

Mobile applications present some additional requirements that are less commonly found with traditional software applications (Wasserman, 2010). Therefore, in a software engineering point of view, mobile computing in general, needs a different approach. Even if, in many aspects, traditional software engineering can be applied in mobile computing (Wasserman, 2010), concerns regarding this traditional methodology emerge, such as methods, tools, models, and programming styles (Uskov, 2013).

As Software Engineering (SE) is a discipline concerned with all aspects of software production (Sommerville, 2011), aspects such as limited computing power, security issues (regarding web and mobile networks), and several types of mobile technology stimulate the use of software engineering focused on mobile computing and development, the Mobile Software Engineering (MSE) (Uskov, 2013).

Section 2.2 presents some motivation to use Software Engineering (SE) in the mobile context and shows how the mobile software are evolving. Section 2.3 presents some constraints and aspects of mobile computing and some SE approaches in order to embrace the mobile context in the development process. Section 2.4 gives some insights about the mobile next generation and its challenges. And Section 2.5 summarizes the chapter.

2.2 Beyond Good Practices, the Complexity and Quality of Mobile Software

It is possible to have access to official guides of the mobile platforms, mainly Android, iOS, and Windows Phone, which compose the main mobile platforms in the market (see Table 2.2). These guides address practical aspects of each platform, which involve concepts of programming, energy consumption, frameworks and Application Programming Interface (API), user interaction, and even design patterns.
In essence, these guides represent the good practices of each platform and technical aspects of its SDK. These set of tools were one of the responsibles for the growth and popularization of mobile development since it became very useful for the individual developer, who would like to develop applications as quick as possible.

This kind of developer represents a substantial part of the mobile developers. A survey of 352 Application (APP) developers showed that forty percent of them develop apps outside of their main job; twenty-one percent work on apps part time; and thirty-nine percent made their living through app development (Mojica et al., 2014). In other words they need to develop fast, and the guides and SDK provide the tools needed to accomplish that.

This developer, despite the inclination of following the recommended set of best practices, rarely applies any kind of formal development process or methodology (Wasserman, 2010). This can be justified by the powerful tools provided by the SDK, which let the development process much faster and simplified, allowing the developer to have apps on the stores as fast as he/she can write the code. So this new generation of tools encourage single developers to adhere to the principles of abstraction and modularity, built into the platforms architecture (Wasserman, 2010).

However, the mobile development is transcending the individual developer frontier. Even more software solutions are being proposed to attend more diverse and complex needs, leaving the inexpensive recreational applications. In this case, it will be essential to apply software engineering techniques to assure the development of secure and high-quality mobile software (Wasserman, 2010). Pervasive computing (which will be discussed in more details later in this chapter), for example, is increasing the complexity of mobile applications, and as mobile devices quickly gain improved features, the difficulty in developing mobile software increases (Huang and Mangs, 2008). Even if platform developers used years of software engineering evolution to create architectures and SDK available to developers, these only technical aspects will not be enough to face the challenges of large-scale mobile applications (Wasserman, 2010).

Another aspect largely used by mobile developers, which can allow them to develop applications quickly, is the practice of software reuse. It is one of the aspects that keep mobile developers to create successful apps, even with the lack of formal training in software engineering (Mojica et al., 2014).

There are several types of software reuse, such as inheritance; code reuse; and framework reuse (Mojica et al., 2014). In mobile development is common the use of entire third party frameworks, mainly because the native SDK does not provide directly
2.2. BEYOND GOOD PRACTICES, THE COMPLEXITY AND QUALITY OF MOBILE SOFTWARE

access to some features, or these features are hard to implement. Thus, developers rely on libraries and frameworks to implement features which are more laborious to implement and that would take a considerable amount of time.

Mobile apps seem to belong to one of these types of framework reuse (Mojica et al., 2014):

- Reuse of private closed source classes owned by companies for their own purposes
- Reuse of private closed source classes owned by companies to develop solutions for their clients
- Reuse of a public open source collection of libraries
- Use of automatic mobile app builders.

Software reuse has many benefits: it can encapsulate the knowledge of application specialists in some reusable components and reuse it every time that knowledge is needed in some specific task. Common application standards, such as user interface, can also be implemented as a set of reusable components bringing a system to market as soon as possible. Reusing software can speed up system production in terms of development and validation time (Sommerville, 2011). These three mentioned aspects are the main reason for mobile developers reuse software, mainly frameworks and libraries. The development time seems to figure in one of the top reasons third party frameworks are used.

However, software reuse brings some usual problems as well, for example: if the source code of a reused software or component is not available, the maintenance costs can increase because the reused elements embedded in the system may become incompatible with system changes; and reused software components have to be discovered, understood and, sometimes, adapted to work in a new environment (Sommerville, 2011). These two problems fits very well on the framework reuse categories described above.

Since mobile developers heavily rely on external libraries, someone must also understand the behavior of the reused framework or library, complicating the program comprehension and maintenance. The usage of the external libraries implies in the growth of an app, which can be observed based on correlations of Number of External Calls (EXTC) vs. Cyclomatic Complexity (CYCLO), and EXTC vs. Lines of Code (LOC), which is causally connected with this kind of reuse (Minelli and Lanza, 2013).

In order to understand the evolution of mobile software, in a recent study researchers investigated the applicability of Lehman laws of software evolution in mobile applications
(Zhang et al., 2013). The Lehman laws consist in a set of laws related with software evolution. In the study, Zhang et al. (2013) used three laws:

i) **Continuing Change**: Software has to continually change if the software is to remain useful.

ii) **Increasing Complexity**: Unless complexity checks are in place, software will get more complex over releases.

iii) **Declining Quality**: As the software evolves, the software quality drops.

To perform the study, Zhang et al. (2013) compared the mobile and desktop versions of two distinct software, using software evolution metrics to examine whether Lehman laws apply to mobile apps. Then, they compared the metrics for the mobile apps with their own desktop version.

The metrics used in the research could give an insight of the applicability of Lehman laws on mobile software. The three laws could be observed in the mobile versions, even with some reservations due the accuracy of numbers of commits metric, used to verify the law of increasing complexity. However, the LOC metric, shows the presence of the law of increasing complexity in both mobile applications (Zhang et al., 2013). The applicability of these laws illustrates the increase of software complexity on mobile applications.

While a mobile software is resumed in a simple single developer APP, all the good practices and software reuse can be enough to get the work done. However, as soon as the applications get more complex and mobile computing concepts are getting more important in the development process, the use of software engineering becomes essencial to ensure the software development with quality, scalability, and with a controlled budget.

### 2.3 Mobile Computing Aspects

Mobile development is just one of many layers of mobile computing, which is composed of important aspects that must be included in any mobile project, especially large and complex projects. The fully understanding of these aspects are fundamental to the mobile development process and to apply software engineering on it.

Since mobile devices, such as smartphones and tablets, are changing internet access and communication patterns, making possible today’s ubiquitous consumption of web and multimedia content (Benincasa et al., 2014), the developer must understand how to
create applications that can fulfill the needs of the users, and it is important to comprehend the aspects of mobile computing in general.

According Uskov (2013), Mobile computing can be grouped in four main core areas:

![Diagram of Mobile Computing Core Areas](image)

**Figure 2.1 Mobile Computing Core Areas**

i) Development of mobile software systems and/or mobile applications.

ii) Software engineering of mobile systems.

iii) Mobile and wireless networks.

iv) Security of mobile systems and mobile networks.

Although these main areas are heavily attached (almost every app uses concepts of them even if encapsulated by mobile SDK, going unnoticed by developers), the focus of this work is on mobile development process and mobile software engineering. Therefore, only these two areas will be detailed discussed, presenting some important aspects in mobile computing that will show how mobile development is different and how software engineering can be applied in a mobile context. These aspects will be present in the following sub-sections.

### 2.3.1 Great diversity of environments and platforms

As earlier mentioned there are several development tools and platforms in mobile development, these tools fit in two main categories: Native tools and cross-platform tools (Madaudo and Scandurra, 2013).
Native tools are designed to create applications that run on specific platforms (Madaudo and Scandurra, 2013). Such tools bring its own SDK, that works with specific programming languages, frameworks, Integrated Development Environment (IDE), and best practices, provided by programming and design guides. To create iOS APPs, for example, the developer will use Cocoa Touch framework\(^5\), Objective-C language or the recently announced Swift language\(^6\), and Xcode IDE\(^7\).

Choose to develop using native tools for different platforms require a depth knowledge of them and their SDK, increasing development cost, time to release the app, and brings updates issues (Dalmasso \textit{et al.}, 2013).

On the other hand, cross-platform tools can provide developers the flexibility to create multi platform applications, according to “write once run everywhere” principle (Madaudo and Scandurra, 2013). Several tools with different cross-platform approaches are available. Later in this section all different approaches and their respective cross-platform tools will be addressed.

However, this method compromises the user experience, given that the User Interface (UI) is usually implemented with HTML5, CSS, and JavaScript, simulating native behavior. Furthermore the access of many of the devices and platform functionalities can be restricted. Therefore, this process is helpful only when a developer is willing to compromise user experience, giving more importance to the launching of the app in multiple platforms to reach to maximum users, reducing the development costs and time to market (Dalmasso \textit{et al.}, 2013).

Many factors must count when choosing between native or cross-platform approach, such as: the audience of the app; development costs, including the experience of the development team; how the app is going to be distributed; and the functionalities of the app. Other more specific factors can be addressed to the choice of mobile development approach. Table 2.3 evaluates the two approaches based on some of these factors.

Cross-platform approaches can be divided into four groups: Web, hybrid, interpreted, and generated APPs. None of them is neither prevalent nor the best solution to the problem of developing cross-platform mobile applications (Xanthopoulos and Xinogalos, 2013). To make this choice one must be aware of the limitations, advantages, and disadvantages of each one, evaluating the best approach for the app to be developed. Each group can be described as bellow:

- **Web**

\(^6\)https://developer.apple.com/swift/  
\(^7\)https://developer.apple.com/xcode/
2.3. MOBILE COMPUTING ASPECTS

<table>
<thead>
<tr>
<th>Factor</th>
<th>Native</th>
<th>Cross-platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI User Experience</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Performance</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Device-specific features</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Distribution via app-store</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Multiple platforms deployment costs</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Developers support</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Security</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Timely access to new OS innovations</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Code reusability</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>Design challenges</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>Availability of programming expertise</td>
<td>low</td>
<td>high</td>
</tr>
</tbody>
</table>

Table 2.3 Cross-platform vs. native development (Madaudo and Scandurra, 2013)

Web applications are browser-based applications in which the software is downloaded from the web (Xanthopoulos and Xinogalos, 2013). Since web applications use popular web technologies, such as HTML5 and JavaScript, the knowledge to develop an web APP is very common among web developers, facilitating the mobile development for a team of web developers, reducing costs of hiring another team. However, web apps have limited access of device’s hardware and data, and they cannot be installed physically in a device (Xanthopoulos and Xinogalos, 2013). This issue difficults the use of some platforms and device features and its sensors.

- **Hybrid**
  Hybrid apps still use web technologies such as web apps. The main difference is they execute their code inside a native web container, and can be installed on a device, allowing the application to access device hardware and data using specialized API (Xanthopoulos and Xinogalos, 2013). PhoneGap\(^8\) is an example of an hybrid tool.

- **Interpreted**
  Interpreted apps generate native code to implement the User Interface (UI), while all the application logic is implemented using several technologies and languages (Xanthopoulos and Xinogalos, 2013), this may also reduce the need of a specialized team, and the use of native code improves the user interaction, reducing the perils

\(^8\)http://phonegap.com
CHAPTER 2. AN OVERVIEW ON MOBILE SOFTWARE ENGINEERING

<table>
<thead>
<tr>
<th></th>
<th>Web</th>
<th>Hybrid</th>
<th>Interpreted</th>
<th>Generated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketplace deployment</td>
<td>No</td>
<td>Not guaranteed*</td>
<td>Yes**</td>
<td>Yes**</td>
</tr>
<tr>
<td>Widespread technologies</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Hardware and data access</td>
<td>Limited</td>
<td>Limited</td>
<td>Limited</td>
<td>Full access</td>
</tr>
<tr>
<td>User interface</td>
<td>Simulated</td>
<td>Simulated</td>
<td>Native</td>
<td>Native</td>
</tr>
<tr>
<td>User-perceived performance</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
</tr>
</tbody>
</table>

Table 2.4 Comparative analysis of cross-platform development approaches. * E.g. simple web clippings content aggregators or a collection of links, may be rejected by Apple. ** Apps can be distributed without difficulty but the experience that the app provides must comply with the app store development guidelines (Xanthopoulos and Xinogalos, 2013)

of a cross-platform approach. Appcelerator Titanium Mobile\(^9\) uses this approach for multi platform apps, with a JavaScript SDK.

- **Generated**

Generated apps give native code for each platform, so the disadvantages of cross-platform apps are drastically reduced. However, this is not a popular approach, since the use of generated native code is difficult because of its automated structure (Xanthopoulos and Xinogalos, 2013). Applause\(^10\) is an example of a generated APP tool.

Table 2.4 shows the capabilities of each cross-platform aproach regarding some of the aspects.

### 2.3.2 Sensors and Device Features

There is a large number of devices running on all kinds of platforms, each of them with their own hardware characteristics. Mobile devices must support applications that were written for all of the varied devices supporting that platform (Wasserman, 2010). Developers must keep in mind the hardware diversity that is available in the market for the chosen mobile platform, just as features like screen size, computing power, battery capacity, and sensors availability, that may vary between devices. Figure 2.2 shows an example of the great diversity of devices.

Several electronic devices such as Personal Digital Assistant (PDA)s; digital cameras; and Global Positioning System (GPS) are being substituted by mobile hardware like

---

\(^9\)http://www.appcelerator.com/titanium/
\(^10\)http://www.applause.com
smartphones, making them all-in-one ubiquitous computing devices (Abolfazli et al., 2013). Most of them include sensors such as an accelerometer, luminosity detector, a touch screen, a GPS, a microphone, one or more cameras, and multiple networking protocols (Wasserman, 2010). Nowadays, even biometric sensors, such as fingerprint readers and eye motion detector, are been embedded on mobile devices.

All these sensors enhance the user experience, feeding applications with data that can allow it to identify the user context, an essential concept that will be addressed later in this chapter, that makes a rich user experience available, in addition to serve as ubiquitous input source. Sensors also make possible applications to use them to perform tasks like geo-localization, motion sensitive features (extensively used in games), and augmented reality, provided by cameras.

While gestures, sensors, and location data may be used in traditional computers, they play a dominant role in many mobile applications (Wasserman, 2010), especially regarding the user experience, that envolves many mobile practices.

### 2.3.3 User Experience

Using a mobile device is different from working with a desktop or laptop computer (Wasserman, 2010). Besides the using of a large set of sensors, that already makes a great difference in mobile user experience, the small display of the devices and different styles of user interaction also have a major impact on mobile applications design. All of these design particularities influence directly on the application development (Wasserman,
Unlike traditional paradigms that are based in windows, icons, menus, and pointers, mobile user interface is based in touch, physical motion, and keyboards (Wasserman, 2010). Many applications have customized keyboards to different input data. Mobile user interface can also have many context dependencies, such as physical location, and proximity to other mobile devices (Wasserman, 2010), provided by the sensors discussed.

In order to create a great user experience, the developer must be aware of three main items that compose the mobile experience: context, Information Architecture (IA), and visual design (Fling, 2009).

- **Context**

  Context is probably the most used, underestimated, and misunderstood concept in mobile (Fling, 2009). Many developers use context in their applications even without notice it, but what if the context changes? How to adapt the app to the current context? Keep the context in mind when developing an application and answer questions like that can bring much value in any app user experience.

  In this perspective, context is to understand circumstance, how to aggregate value from something the user is current doing (Fling, 2009). It is also to understand the environment around the user in order to enhance the user experience with that information. In mobile computing, context can be geographical coordination of a user, weather humidity, and band width availability (Abolfazli et al., 2013). All these information can be retrieved by sensors or features embedded in modern mobile devices, which give them great capacity to create contextual information (Fling, 2009). In other words, mobile devices can detect how people behave when they are at certain locations, surrounded by certain people, and so on (Guilleén et al., 2014).

  To better understand how to address context into an application, Fling (2009) proposes a set of questions:

  i) **Who are the users?** What do you know about them? What type of behavior can you assume or predict about the users?

  ii) **What is happening?** What are the circumstances in which the users will best absorb the content you intend to present?

  iii) **When will they interact?** Are they at home and have large amounts of
time? Are they at work where they have short periods of time? Will they have idle periods of time while waiting for a train, for example?

iv) Where are the users? Are they in a public space or a private space? Are they inside or outside? Is it day or is it night?

v) Why will they use your app? What value will they gain from your content or services in their present situation?

vi) How are they using their mobile device? Is it held in their hand or in their pocket? How are they holding it? Open or closed? Portrait or landscape?

- Information Architecture

The second item in mobile user experience is Information Architecture (IA). The mobile IA is not much different then in traditional web applications. It is made especially tricky mainly because different devices have different capabilities and therefore different interaction models, contextual capabilities (already discussed in this section), and needing to display on a smaller screen as much information as we would on a desktop (Fling, 2009). A good example of an application of IA in a mobile context, aiming the devices screen sizes, are the three different touchable zones that can be mapped in a mobile device screen (Abolfazli et al., 2013) in order to define where to place interface objects, such as buttons. These touchable zones are the Reach, Easy, and Medium zones, as seen on Figure 2.3.

However, the mobile devices have different screen sizes, and these comfort zones cannot be applied to all of them. So each device needs its own IA.

In general, IA defines how the information will be structured and how the user will interact with it (Fling, 2009) and, especially in a mobile application, how various possible contexts can be addressed. Simplicity seems to be the key for a good IA, since mobile users are often seeking to quickly complete a simple task (Wasserman, 2010), and to accomplish that the developer can make use of some tools to help the design process, such as site maps, wireframes, and prototypes (Fling, 2009).

Site maps are used to show how the information will be organized. They visually represent the relationship between contents and provide a map for how the user
will navigate through the informational space (Fling, 2009). This space represents all information that can be accessed through the app. Figure 2.4 shown an example of a sitemap.

Wireframes show how the user will interact with the information physically, how the information and aspects of navigation will appear on the screen, they add some more practical perspective to the site map. However, wireframes are often static and cannot bring some real usage perspective. So another important tool in Information Architecture (IA) and in general mobile design are the prototypes. This tool can bring dynamic views of the application workflow, making possible to visualize possible issues in the IA. There are several ways to create a prototype, from paper prototypes to more complex HTML models or even superficial UI implementations using some mobile SDK.

In a mobile software engineering point of view, these practices can figure important steps in the development process, once they can provide a solid IA and consequently a good user experience. Therefore, they must be part of a mobile software development planning.

• **Visual Design**
Figure 2.4 Example of a Sitemap (Fling, 2009)
CHAPTER 2. AN OVERVIEW ON MOBILE SOFTWARE ENGINEERING

The last item in the user experience layers is the visual design, the most subjective layer. There is no tool that guarantee a good visual design, that is mostly based in the “look and fell” concept. However, as already discussed, mobile platforms have their own UI libraries and guidelines, so native applications for a device will share a common “look and feel” (Wasserman, 2010). A platform user expects to have a similar visual experience in all its applications, thus following the platform UI guidelines is a good way to reach an acceptable visual design.

2.3.4 Cloud and Ubiquitous Computing

Cloud computing is a term that represents any internet based computing. Nowadays many services are available such as real time communication, file sharing, email services, video streaming, and even the mobile marketplaces, where developers can make their APP available.

From a SE point of view, cloud computing is a paradigm shift (Yu and Yuan, 2011). With application development and deployment moving toward the “cloud”, this new computing paradigm can affect development processes, tools, and application architectures (Wasserman, 2010). When applied in a mobile context, cloud computing can be considered as an expansion of the mobile device that supplement its limited computing resources by accessing shared computing power or storage on demand (Yu and Yuan, 2011).

If developers assume that mobile devices are always connected to the web, application usefulness increases and user experience is enhanced (Abolfazli et al., 2013), but to use cloud computing it is mandatory that mobile devices have access to fast and seamless internet connection, which is not a reality in most of the time. Thus, mobile cloud computing is limited by slower or interrupted internet connections (Yu and Yuan, 2011). So, it is important to make offline support available in a mobile application. Many storage cloud services can synchronize between offline with online data, like Apple’s iCloud. Hence, the user can have offline access to the data and at same time uses the cloud to synchronize the data between several devices.

Another important concept that can often use the advantages of cloud computing is the ubiquitous or pervasive computing. It is the idea that computers will seamlessly integrate into humanity’s world and into our daily lives to the point that no user input will be required for a computer to understand what it should do (Huang and Mangs, 2008).

The idea of ubiquitous computing was first described in a 1991 visionary paper called “The Computer for the 21st Century”, written by Mark Weiser (Chen and Yen, 2009).
2.3. MOBILE COMPUTING ASPECTS

Back then Weiser (1991) foresaw that the idea of personal computers, laptops, and other machines was only a transitional step toward the real potential of information technology, and that these machines could not make computing an integral and invisible part of our lives.

The true ubiquitous computer is one that takes into account the human world, pushing all computing and the computers themselves to vanish into background, and the technology required to that is divided in three parts: cheap low power computers with convenient displays; software for ubiquitous applications; and network that ties them all together (Weiser, 1991). More than 20 years later of the publication of Weiser’s paper, the computers described by him seems to be represented by modern mobile devices.

The idea of a small device that could run an application that dynamically adapts to environmental conditions would have seemed like science fiction twenty years ago (Huang and Mangs, 2008). However, now with all the sensors embedded on the devices to get contextual information required in ubiquitous computing was made viable and relatively easy, furthermore these devices are getting more affordably and the very mobility of them can make ubiquitous computing a reality now and in even more in the next years.

However, mobile computing are not yet fully ready to the ubiquitous computing. Network problems cannot always make a seamless connection possible, security issues in mobile computing must be fixed, and the great variety of mobile platform can make difficult for two devices to communicate with each other. These will be some of the challenges the next generation of mobile computing will face.

The following sections will cover some aspects of Mobile Software Engineering (MSE) that take into account the constraints and characteristics presented above.

2.3.5 Agile vs. Plan-Based Development

As mobile applications escape the individual developer reality and became more complex there is a need to use some formal development process that address mobile aspects. All aspects such as context, different devices and sensors, and connectivity issues must be present on the formal development process. The user experience in general is especially critical, for example, there is a greater need to create prototypes of the user interface, especially when developing for multiple devices (Wasserman, 2010).

The unique requirements and constraints associated with mobile systems have brought new challenges to mobile software development. As it demands extensive improvements to traditional systems development methodologies in order to fulfill the special needs of this field, it has been shown that Agile methodologies are appropriate methods for the
development of such systems (Rahimian and Ramsin, 2008).

However, many aspects must be evaluated before choosing a development approach. Agile approaches to software development consider design and implementation to be the central activities in the software process. They incorporate other activities, such as requirements elicitation and testing, into design and implementation. Therefore, the requirements and the design are developed together, rather than separately. By contrast, a plan-driven approach to SE identifies separate stages in the software process with outputs associated with each stage. Iteration occurs within activities with formal documents used to communicate between stages of the process. For example, the requirements will evolve and, ultimately, a requirements specification will be produced. This is then an input to the design and implementation process (Sommerville, 2011). An overview of these two kinds of process can be seen on Figure 2.5.

![Plan-Based Development](image1)

**Figure 2.5** Plan-based and agile developments (Sommerville, 2011)

Despite the advantages of agile methods, even with some interesting mobile approaches proposed, like the one proposed by Rahimian and Ramsin (2008), not always it could be a good approach. The need of detailed specification of design before implementation, motivated by the constraints and aspects of mobile development, could make a plan-based development a better option.

The rapid changing in the mobile computing, mainly new versions of platforms and new devices with new features, must also be taken in account in the development process,
2.3. MOBILE COMPUTING ASPECTS

in this case an agile approach can be a good solution, where the software must be quickly updated according with new platform features for example.

There is good advantages in both plan-based and in agile development for a mobile context, the most important point is to address the mobile aspects into the development process. Most of the software projects include practices from plan-driven and agile approaches (Sommerville, 2011), and this seems to be a good direction in mobile development.

2.3.6 Non-Functional Requirements and Architecture

The success of any application, mobile or not, depends on a lengthy list of non-functional qualities. Among those most relevant to mobile applications are performance (efficient use of device resources, responsiveness, scalability), reliability (robustness, connectivity, stability), quality (usability, installability), and security (Wasserman, 2010). The user experience is strongly attached to non-functional requirements and the developers must find the most important of these requirements to the application.

These set of non-functional requirements also impacts directly in the mobile development, since they may affect the overall architecture of a system rather than the individual components. For example, to ensure that performance requirements are met, the architect may have to organize the system to minimize communications between components (Sommerville, 2011).

Mobile software architectures also must be adapted to mobile context, including non-functional requirements into account and other mobile constraints like cloud computing, which is been used extensively by applications that are more integrated with other kinds software, mobile or not.

In general a well adopted architecture pattern in mobile development is the Model-View-Controller (MVC) pattern, most of the mobile SDK have MVC embedded in their development tools and practices. MVC separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other. The Model component manages the system data and associated operations on that data. The View component defines and manages how the data is presented to the user. The Controller component manages user interaction and passes these interactions to the View (Sommerville, 2011). The separation of presentation and interaction from system data can allow developers to work on the user experience more efficiently.

Figure 2.6 shows a conceptual architecture view of the MVC pattern, this view is an abstract view of the system that can be the basis for decomposing high-level requirements.
into more detailed specifications (Sommerville, 2011). However, in mobile development may be interesting use other architecture views such as process view, which shows how, at run-time, the system is composed of interacting processes, this view is useful for making judgments about non-functional system characteristics such as performance and availability (Sommerville, 2011). A developer view can be used to break the software in individual modules, mainly to assist the coding phase of the application.

All these MSE aspects will evolve with the mobile technologies and be more necessary, as new constraints regarding mobile computing emerges. Some of these new constraints will be showed in the next section.

2.4 Mobile Next Generation

The mobile computing rapidly evolved in the past few years, but much more have to be improved in order to face the challenges of the future in this field. Mobile computing is a frontier of hardware and software development that is likely to challenge every facet and resource of our technology sector (Edmondson et al., 2014).

We have already discussed pervasive computing in Section 2.3, and it seems that it will play a fundamental role in the mobile next generation, since innovative mobile and pervasive computing applications, characterized by an extensive use of location and social based features and an ever-increasing interaction with the surrounding environment, are
The applications will face a constantly connection to multiple networks and integration with other devices. The next generation will require specific support functions, such as for seamless session mobility and automatic discovery of nearby resources, in fact, applications will need to deal with highly dynamic environments such as heterogeneous networks, characterized by the overlapping of several different wireless networks (Benincasa et al., 2014).

This seamless connection will be combined with the growing number of the connected and remotely operable devices for the Internet of Things and with the adoption of innovative interfaces such as Google Glass, which will contribute to the creation of new communication paradigms. Next generation applications will need to smartly take advantage of multiple communication links in accordance with their communication semantics, user preferences, and current network state, enabling mobile data offloading when possible. (Benincasa et al., 2014).

The resource consumption is also an issue that will shape the next mobile generation. Future mobile environments will host multiple applications competing for scarce resources, with various requirements and priorities, raising the very real need to mediate between applications’ resource requests, considering their priorities and preventing excessive resource consumption (Benincasa et al., 2014).

Words like connection, networks, and pervasive, are present in the current mobile paradigm and seems to be mandatory in the next generation. This application mobility raises additional security issues (Benincasa et al., 2014), which will bring some paradigm changes as well. So from now the mobile software engineers must prepare themselves to the changes of paradigms that will follow the evolution of mobile computing. The SE itself will have to embrace this changes and be a fundamental part of the mobile development process, in order to allow the mobile next generation to rise without the problems that the current mobile computing faces.

### 2.5 Summary

In this chapter, it was presented some important mobile computing aspects, such as pervasive computing; user experience; and sensors usage. It was also shown why Software Engineering (SE) must be applied in the mobile context and how that can be accomplished using some traditional SE approaches such as process and architecture design to involve the mobile constraints in the development process. Finally, some characteristics about the
challenges of the mobile next generation was discussed.

Next chapter will present a survey made in order to understand a real life problem and how a mobile application could help within this problem.
3

Mobile Software and Real Life Problems

After all, tomorrow is another day!
—VIVIEN LEIGH (Gone with the Wind)

3.1 Introduction

The main goal of this work is to develop a mobile application using software engineering principles applied in the mobile context in order to solve a real life problem. Thus, a real life problem was chosen to use the main mobile capabilities available, and already discussed in Chapter 2, to try to solve it.

The violence against women was the real world problem chosen. This problem was chosen mainly because it is related to the violence in general, a major problem in many of brazilians cities. A previous unpublished work\(^1\) on this area was made, where an iOS application was developed to send emergency alerts in case the user be in some sort of dangerous situation. This work and the experience obtained in the development process also motivated the selection of a theme related with security and violence prevention.

However, develop a software related to a real world problem like this requires a certain level of knowledge on the domain, which has proved a challenge since violence against women is a very specific domain, non related with any computing discipline and very obscure to us. To develop an application that could help to solve the problem, a data collection study had to be performed in order to understand the violence against women, its circumstances, and how a mobile software solution could help at least to minimize the problem.

\(^1\)https://itunes.apple.com/us/app/savi/id590385285?mt=8
In this chapter, will be presented the data collection technique and how it was applied to collect the information needed to develop the application. Section 3.2 describes all the process used in the data collection phase and how each step of the process was conducted. Section 3.3 shows the results and conclusion obtained with the survey, including the software approach chosen in order to solve the problem. Finally Section 3.4 summarizes the chapter.

3.2 Data Collection

First, a data collection method had to be chosen. A method can be defined as a set of organization principles around which empirical data is collected and analyzed (Easterbrook et al., 2008). Since the data to be collected were composed of very specific information of a theme little known by the time, the method chosen had to allow to identify the population involved with the violence against woman and allow that the knowledge required to develop a software solution for the problem could be acquired.

Two methods were identified that could accomplish this task. The first method was Action Research, which the main goal is to attempt to solve a real world problem while simultaneously studying the experience of solving the problem. This kind of approach is aimed to improve the situation of the object of study (Easterbrook et al., 2008). This method partially attends the needs of the research, but it requests the presence of a problem owner willing to collaborate to both identify a problem, and engage in an effort to solve it (Easterbrook et al., 2008). The problem owner could be a chief police officer or a public defender that works directly with the violence against women, but these actors could not always be accessible for the research.

The second method was the Survey, which is used to identify the characteristics of a population, selecting a representative sample from it and using base-rate questions to analyze the data (Easterbrook et al., 2008).

The precondition for conducting survey research is a clear research question that asks about the nature of the target population (Easterbrook et al., 2008), and since some research questions of this work were directly related with the population, such as: “Can a mobile software solution be effective to help to solve the violence against the woman problem?”; and: “Do women under violence have access to mobile devices?” the Survey method approach was chosen.

Yet, the Survey method has a purpose to produce quantitative or numerical information about the studied population from the answers collected in questionnaires (Kitchenham
3.2. DATA COLLECTION

and Pfleeger, 2008), and some qualitative data could be interesting for the development of the application. Therefore, a mixed approach of survey with some aspects of action research was chosen, employing data collection and analysis techniques associated with both quantitative and qualitative data (Easterbrook et al., 2008). Furthermore the social imprint of the project guided us to that approach, since in SE the blend of technical and human aspects lends itself to combining qualitative and quantitative methods, in order to take advantage of the strengths of both (Seaman, 2008).

In general the data collection method was based on Easterbrook et al. (2008) description of survey, which define some main activities, that will be discussed in the next sections, associated with some qualitative methods.

3.2.1 Setting Objectives

The objectives of a survey are a statement of the survey’s expected outcomes or a question that the survey is intend to answer (Kitchenham and Pfleeger, 2008). Thus, before setting the survey main objectives, the population was established and some research questions were defined.

The population is composed by people directly involved with the violence against women, which is composed by the violence victims, and professionals who work against the violence, such as police chief officers from women’s police stations (DEAM), public defenders, or social workers.

Then, the following research questions were made:

• **How, when and where happens the violence?**
  
The circumstances of the violence are important to understand the problem to better propose a solution and begin to identify the possible contexts to be used in the development process.

• **Can a mobile software be effective to help to solve or minimize the problem of violence against women?**
  
This question verify if a mobile device with an specific application embedded can really help the population with the problem.

• **Does woman under violence or any possible user of the application have access to mobile devices?**
  
This question aims to analyze if the population will be able to use the application, since the users may not have conditions to afford an mobile device.
How a mobile application can help with the problem?

Once a mobile application is shown viable to act on the problem, there must be found the options of how the application can act and choose the best option.

Based on these questions the objectives of the survey were established as follow:

i) Learn about the problem; its causes and consequences.

ii) Understand how the authorities; the victims; and population in general deal with the problem.

iii) Increase the knowledge about the population.

iv) Understand the viability of the use of a mobile application on the problem.

v) Once a mobile application is found viable, make a initial requirements elicitation for the mobile application.

Kitchenham and Pfleeger (2008) also propose that after the objectives have been detailed some aspects can be specified, such as:

- Hypotheses to be tested.

A mobile software solution is viable and can help to prevent future aggressions or minimize the damage caused to the woman; and allow the fast arrest of the aggressor.

- Alternative explanations to be investigated.

Understand what kind of tools the women have to fight the violence; how these tools are available to them; how technology in hardware and software is used; and how the current public system of denunciation works in Brazil. These kinds of information can help to identify gaps where the application could act.

- Resources necessary to achieve the objectives.

In order to achieve the objectives, some location where the population can be found need to be identified and some trips need to be organized to visit these places and apply the survey instruments.
3.2. DATA COLLECTION

3.2.2 Survey Design

In this study, a cross sectional design was used to gather information, in which participants are asked for information at one fixed point of time (Kitchenham and Pfleeger, 2008). To perform the survey, a printed self-administered questionnaire was used and semi-structured interviews were made. Each single interview was made between January and February, and the questionnaires were performed in March. All information were gathered in 2014.

3.2.3 Developing the Survey Instrument

In Kitchenham and Pfleeger (2008), the guidelines instrument used is a self-administrated questionnaire suggested to be made with close questions focused on quantitative and statistic data. However, one of the objectives of the survey is to learn about the problem and gather some possible requirements for the application. Thus, some qualitative data and a personal opinion about the subject is important in order to achieve all objectives, turning the survey in fact into a mixed-approach.

Hence, it was developed two different survey instruments for two different groups inside the population: a questionnaire to be performed with women victim of violence who are denouncing or already denounced the aggressor to the authorities; and a semi-structured interview destined to professionals who work to fight the problem in some way.

- Questionnaires

The questionnaire was composed of sixteen questions and had the goal to get a better knowledge of the problem from a direct point of view of the victim, providing some statistic data regarding the profile of the victims and their access to technology with close questions and getting the personal opinion of them with open questions. These sixteen questions can be divided in two groups: Personal questions; and questions about technology.

The personal questions are composed of six questions: two closed; three closed with open fields for justification; and one open question. This group of questions aims to get information about the relation of the woman with the denouncing system, trying to identify some gaps in the system that can difficult the process of fighting the violence against woman and where the application could act to improve the situation.
The questions about technology are composed of ten questions: Five closed questions; three closed with open field for justification; and two open questions. This part of the questionnaire is intended to gather information about the access of the victims of violence of technology in general and to get their personal opinion about the use of technology in the fight against the violence.

**Interviews**

The focus of the interviews is to get some qualitative data from a point of view of professionals who work directly with the violence against the woman. The interview was made in a semi-structures format, which is a mixture of open-ended and specific questions, designed to elicit not only the information foreseen, but also unexpected types of information (Seaman, 2008).

With the interview was expected to get some information or some point of view that was not covered into the main idea of the project. The interview guide has seven base questions, divided into more specific and more open-ended questions. These questions can allow to understand what is the role of the interviewee in the problem, get a overview knowledge of the profile of the victims (the first group of the selected population), find the viability of a mobile application to solve or minimize the problem, directly ask for suggestions about how the application could help, and get some unpredicted kind of information.

### 3.2.4 Evaluating Survey Instruments

Once the instruments have been developed it was needed to evaluate them. Kitchenham and Pfleeger (2008) call this process pre-testing, and present several goals: to check that the questions are understandable; to assess the likely response rate and the effectiveness of the follow-up procedures; to evaluate the reliability and validity of the instrument; and to ensure that our data analysis techniques match our expected responses.

To evaluate the instrument, Kitchenham and Pfleeger (2008) discuss about assembling a focus group with people representing either those who will use the results of the survey or those who will be asked to complete the survey or make a pilot study with a small sample of the population. However, due the difficult to reach any of the members of the population, it was not possible to pre-test the instruments with someone of the group that will complete the survey.

To evaluate the instruments, mainly the self-administrated questionnaires, a pre-testing was made with some students of the Institute of Psychology at Federal University
3.2. DATA COLLECTION

of Bahia (UFBA). The objectives of the survey was explained to them and they were asked to evaluate the questions of the questionnaire and the interview guide, and then make suggestions based on the goals listed above.

A second evaluation of the questionnaire was made possible during some of the interviews, since they were performed before the questionnaires, where the interviewees also evaluated the questionnaires, giving a point of view from inside the population.

The final version of the questionnaire can be observed in the Appendix A.

3.2.5 Obtaining the data

Find accessible participants of the population to conduct the survey was a little challenger and first some locations to perform the questionnaires and the interviews had to be identified. Three locations were selected to perform the survey: Specialized Station of Attendance to the Woman (DEAM); Nucleus for Interdisciplinary Study About the Woman (NEIM); and Specialized Nucleus on Defense of the Woman Victim of Violence.

All these places are located in the city of Salvador-Bahia, and were selected because all are respected institutions specialized in the fight of the violence against the women and other women studies.

First, the interviews were made, starting at the Specialized Nucleus on Defense of the Woman Victim of Violence, where was performed the interview with a public defender and coordinator of the nucleus. Then a professor at University Federal of Bahia and member of the Maria da Penha Law Observatory was interviewed at NEIM. Finally, a social worker and the chief police officer of the DEAM, were interviewed at DEAM.

All the interviewees are specialists on the problem, dealing every day with the violence against women. Their participation on this project was essential and made possible some discoveries and conclusions.

The questionnaire was conducted at the DEAM (which is a police station that only take care of cases of domestic aggression against women) with women that were there to denounce some kind of aggression. Five women answered the questions in one afternoon. This number was smaller then was expected (about thirty questionnaires were printed). For some security reasons and intern protocols, the questionnaires only were allowed to be performed at DEAM in the last days of the survey schedule, so only one afternoon was available to that.
3.2.6 Analyzing the data

With the data gathered from the questionnaires and from the interviews, it was analyzed according to the objectives. The questionnaires were analyzed measuring the quantitative data of the closed questions, and grouping the qualitative data into the aspects described by the participants in the justification fields and open questions.

Although the interviews had a guide with some questions, the dynamics of some interviews make hard to group the data in some question. Thus, in order to analyze the data from interviews, it was established some topics regarding the objectives of the interview: profile of the woman who denounce and who do not; viability of a mobile application to solve or minimize the problem; and possible software requirements. The information of the four interviews were compiled into these topics.

The final results were compiled into the main objectives of the survey and a mobile software solution was proposed based on the results. The next section presents the data analyzed; results and the solution proposed.

3.3 Results

The next sub section presents the data analysis process and the information gathered with the survey instruments. Moreover will be presented some issues found in the process, the main findings, and the mobile application scope based on the data analysis.

3.3.1 Questionnaires Analysis

- **Age Group**
  
  i) How old are you?
  
  The goal of this question is to have an idea about the range of the violence regarding the age of the victims. We have five different answers: Eighteen; twenty-six; twenty-seven; thirty-nine; and fifty-four years old.
  
  These values can show that the violence is present in different group of ages. In this case, we have a range from eighteen to fifty-four years old. These answers gave us the idea of the possible users of the application, according to the group of ages.

- **Denouncement System**
  
  This set of questions aims at learning about the current denouncement system and how women interact with it.
ii) Is it the first time you denounce violence?
This is a closed question, its objective is to have an idea if recurrent aggressions are denounced.

All women answered “Yes” to this question, and it may indicate that once the violence is denounced it does not happen again or it happens more rarely.

iii) Have you ever ceased to denounce violence for any reason, or know someone who has left?
This is a closed question with justification field and tries to evaluate if women trust the denouncement system, and why they do or not.

Two women answered “Yes” and three answered “No”. The two women who answered “Yes” pointed that fear of threats from the aggressor and the exposition were factors that inhibited the denouncement. This can indicate that the fear of the repercussions of the denouncement may overlap the trust of the system.

iv) Do you think the current denouncement and support system offered by the government are enough?
This is a closed question with a justification field, its objective is to measure how much confidence women have in the system.

Only one woman answered “No” to this question, she pointed that the system “is too slow”. This answer was not very clear since it may refer to the delay on the initial screening, where the woman wants in a room to have her first orientation about the denouncement process, or it may refer to the whole denouncement process. The other four women answered “Yes” and it was pointed that government shows some concerns regarding the violent against women and that the denouncement makes that the violence does not happen again, showing results by solving most of the cases. These answers indicate that the denouncement system is efficient, besides its flaws.

v) Do you feel safe about denouncing through the current model?
This is a closed question with a justification field. It measures how safe the woman feel once she makes the denounce.

Only one woman answered “No” and she did not justify the answer. The other four answered “Yes” and it was pointed that the denouncement intimidates the
aggressor, making him under suspect of anything that happens to the woman. Also was pointed that the idea of a system specialized on the women protection brings some security. These answers can confirm the efficiency of the system on resolve the problem, since the aggressor feels intimidated and the woman feels more secured.

vi) How do you think this system could be better and more efficient?
This is an open question, the objective is to get the personal opinion of the woman about how the system could be better. This answer may point out some directions that the application could take (not yet covered by the other questions) in order to improve the system.

Only one woman answered this question with some information. There were another two “No comments” and “I don’t know” answers. The woman who gave some suggestions said the women have to take courage and that she system should engage more in actions against the aggressors.

This answer gave us an insight about the measures against the aggressors, that could be one direction the app should take, act directly on the aggressor.

• Access to Technology

In this set were made questions that could get information about the access of computer technologies of women victims of violence. All these questions are closed, just in order to get some quantitative data.

vii) Do you have a computer at home?
All five women answered “Yes” to that question. This indicates that they have the basic access to technology in their own homes.

viii) Do you have internet connection?
Only one woman answered “No” to that question, the other four answered “Yes, with a wireless network available”.

ix) Do you own a smartphone or a tablet?
It had three “Yes, with a 3G or 4G plan enabled” and two “No” answers. It is interest to associate the two “No” answers with the age of the participants. They were the oldest women to complete the survey with thirty-nine and fifty-four years old. This information could indicate some kind of generation characteristics, which
must be taken into account when developing the application, since we have a large
range of ages inside this group of women. It also indicates that who has a mobile
device also has a mobile internet connection available.

The next two question only needed to be answered if this question was answered
positively, in this case only three women.

x) How often do you use the internet on your smartphone and/or tablet?
All three women answered “Several times a day”. This indicates that they are used
to consume data from the internet in a mobile context.

xi) What kind of mobile applications do you use more frequently in your
smartphone and/or tablet?
In this question was presented a list of kinds of mobile application and was asked
to the participant to rate each category from 0 to 5, where 0 is never used and 5
always used. The objective was to know if the participants make use of mobile
applications on their devices or just use the basic functions.

Following the objective of the question, we inferred that they marked the kinds
of application they use, and concluded that they have the habit to use mobile
applications. This question revealed some problems regarding its validity, that will
be discussed later in this chapter.

• Use of technology to solve the problem

xii) Do you think the use of technology (computers, cell phones and other
devices) is capable of effectively help to prevent and fight the violence against
women?
This is a closed question with a justification field. Two participants answered “No”
without justified the answer. One answered “No”, pointing that the use of this kind
of technology cannot predict the act of violence, since the violence comes without
be announced. The other two women answered “Yes” and said that the technology
can be used for the women who is afraid to denounce the violence can have some
channel of communication.

Despite three participants do not believe that computer technology could help with
the problem, we could get an insight about one possible functionality which is
some how enable a channel of communication to the women who want to denounce
but is afraid of the repercussions.
xiii) Do you are aware of some endeavor to fight the violence against women that use some kind of technology?
This is a closed question with justification field. Its objective is to get the knowledge of some software or hardware technology that is been used to solve the problem.

Only one participant answered “Yes” and mentioned television and internet as technologies used. This question did not bring any interest conclusion that could help with the project, perhaps this question was poorly formulated regarding the context of the expected answer.

xiv) The city of Salvador is studying to implement in 2014 the use of a device (panic button) in order to help to prevent the violence against women who are under protective measure granted by the justice. Do you think this is a valid measure?
This is a closed question with a justification field. This question gives one exemple of how technology can be used in the context of the problem (in this case only for women under protective mesure, where several criminal charges and restrictions are applied to the aggressor in order to protect the woman) and asks for the opinion of the participants about the measure cited.

Two participants answered “No” without justification. The other three answered “Yes” with some opinions which pointed that this kind of mesure intimidates the aggressor, who not always keeps distance from the woman, even under protective mesure.

xv) How do you think that computer programs and/or applications for smartphones and tablets can assist on fighting the violence against women?
This is an open question to get the direct opinion of the participants.

One participant answered “No comments”, other said that this kind of measure cannot help at all. The other three participants gave similar answers, saying that this kind of technology can mainly help with information and communication, that could increase the number of denounces.

• General comments

xvi) Do you have some comment about this study or about the questionnaire?
Open question to know some general comments of the participant. Only one
gave some comments, saying that only the law Maria da Penha works, which is a Brazilian law that exclusively deals with domestic violence against woman. This shows that the denouement system is relatively efficient, at least from the point of view of the participants of the survey.

3.3.2 Interview Analysis

In order to analyze the qualitative data of the interviews it was opted to group all the gathered information into topics based in the main objectives of the interviews. This choice was made because the interviews had different approaches from each other according the answers given by the interviewees and followed different courses, leaving the path established by the guide. Thus the interviews were analyzed as the following topics:

• Profile of the Women Who Denounce and Who Do Not.

The violence is present in all social classes, but women who denounce are mostly low income and with low level of education. In general, women with more resources deal with the problem in other ways, where the repercussion can be minimized. Women who do not denounce generally are trapped in the so called cycle of violence, which involves cultural subjects, fear of threats, protection of the family, or lack of knowledge about the denouncing system. The consequences of the denounce impacts the woman, her family, and the aggressor. Once these consequences are accepted by the woman, the cycle of violence is broken and the denouncement is made.

The system is efficient in solve most cases and many women go into protective measure once the denouncement is made, where several actions are made in order to protect the woman, and it was pointed that many times the aggressor stands down when it gets to that point. However, when the aggressor does not respect this measure, he usually goes for the murder or to brutally assault the victim.

The interviews showed that the violence is not attached to a social profile, it happens in all layers of the society. However, the denouncements using official channels are usually made by women who do not have resources to try to solve the problem in another way. Also, it was shown that despite the efficiency of the denouncement system in solve most cases, in some cases the aggressor keeps to attack the victim, even under protective measure.
Some efforts could be made to encourage the woman to denounce the violence, or to better protect women under protective measure.

• **Viability of a Mobile Application to Solve or Minimize the Problem**

There was a concern about the viability of the project, if some software solution, including a mobile one, could help with the problem in an effective way. All interviewees confirmed that software solutions can be effective in many ways and some of them even said that some endeavors involving different knowledge fields are very welcome to fight against the problem. The Maria da Penha law itself was pointed as a very mutable law, that can be easily adapted in order to improve the women protection, that includes possible changes in the law that makes the use of a mobile application or other kind of technology viable. Next topic will discuss some software themes and possible directions the project could take.

• **Possible software approaches**

Using the information of the interviews it was detected three possible directions the project could take. At this point, no requirement or official scope of the software was determined. Here it is listed and then later selected the software approach in order to solve or minimize the problem.

**Information**

A software that allows access of women to information about the violence and how to combat it in a simple and accessible way. It would also allow the exchange of information among women who have suffered violence or not. Thus she would feel in a safe and welcoming environment and can be able to better deal with the problem using the information available in the system. The software should also educate and inform the society in general about this problem.

**Prevention**

It has been found that software to prevent an attack is difficult to achieve because the aggression can not be predicted. But prevention can be done in other ways, such as to prevent the aggressor to repeat the violence or through educational approaches, like mentioned in the last topic.

**Institutional**

It was also mentioned another way to combat violence, that would be an indirect way, by improving the current system used by agencies working on this issue (like
DEAM and the Office of Public Defenders) and possibly make a direct communication between the agencies involved, making the denouncement process quicker. It was pointed out that the current software used is inefficient and insecure.

3.3.3 Threats to Validity

As just five women victim of violence participated on the survey, the data obtained through questionnaires may be questionable. A bigger sample of this group of population could have delivered some more solid data to be analyzed. However, the data from the professional’s interview could confirm the some information collected from the questionnaires, giving it more reliability since the reliability of data from interview can be considered high, taking into account the expertise of the professionals interviewed.

Another threat found was the format of the question six of the questionnaire (see Appendix A). It seemed not to be a very good choice, since it showed itself not to be very clear about how the participant should proceed. The three participants just marked some options. This problem made the answers unclear about what the participants meant.

3.3.4 Findings

In this sub-section, it will be discussed the main findings of the survey by answering the research questions proposed at the beginning of the study.

- **How, when and where happens the violence?**

  The violence happens in an unpredictable way, making hard to prevent it, even if the aggressor had shown signs of violence before. It could happen anywhere at any time in every social layer. The violence could also be psychological or oral aggressions. Attacks to women under protective measure or that already denounced are not common but when it happens it tends to be violent and possibly followed by murder.

- **Can a mobile software be effective to help to solve or minimize the problem of violence against women?**

  Software solutions in general were pointed to be effective to help with the problem, including in a mobile context. Thus, a mobile application can be effective to help to minimize the problem. To really solve it, it must be applied some other measures which involves education of the society and even a cultural change.
• The woman under violence or any possible user of the application has access to mobile devices?

The data analysis has shown that mobile devices are accessible to all possible users, given the questionnaires answers and the interviews. An initiative in the state of Paraíba in Brazil developed a mobile application that works like a panic button to alert the police to attacks, and fifty mobile devices were acquired by the government and will be borrowed to women who denounced violence and are under risk (G1PB, 2014). This is an example of how even women or possible any user who does not have enough income to afford a mobile device could use of the software.

• How a mobile application can help with the problem?

Given the data analysis and the software approaches listed, it was decided to take the prevention approach, in order to improve the protection of women who are under protective measure or that has denounced the aggressor and are under threat or other risks. With this kind of software more women may be encouraged to take the necessary steps to stop the violence and provide extra protection to the women that already denounced.

3.4 Summary

In this chapter, it was presented the real world problem of violence against women and the data collection method used to gather information about the problem. It also shows the steps made in order to analyze the data and the final results.

Next chapter presents Artemis, a mobile application designed to help to combat the violence against women based on the information gathered by the survey presented in this chapter.
Artemis - A Mobile Application to Combat Violence Against Women

I know you’re out there. I can feel you now. I know that you’re afraid.

You’re afraid of us.

—NEO (The Matrix)

4.1 Introduction

Chapter 2 presented some characteristics of mobile development and mobile software engineering. It showed how mobile development is different and how software engineering is necessary in order to develop better mobile application as they grow more complex and mission critic.

This chapter will present Artemis, a mobile application that aims to minimize the violence against women, problem presented in Chapter 3, following the information obtained in the survey.

Artemis is the greek goddess of hunt and protector of women, she and her priestesses have rejected all constraints imposed to women by men. Although Artemis is neither very complex nor a big project regarding its size, it has a mission critic scope since it deals with the security and protection of its users. Thus, it was written following some of the aspects from Chapter 2 that were suitable to the project.

This chapter is organized as follows: Section 4.2 shows the process created to apply the aspects of mobile computing in the SE. Section 4.3 presents the function and non-functional requirements of the application, just as its functionalities. Section 4.4
show an architecture overview of Artemis and Section 4.5 presents its components and modules in detail. Section 4.6 introduces the technologies used in the development process. Section 4.7 gives some details about the implementation of the APP. Section 4.8 demonstrates the main functionalities of Artemis and Section 4.9 summarizes the chapter.

### 4.2 The Development Process

In order to follow the characteristics and constraints of mobile development, it was developed a process based on the work of Nosseir et al. (2012), where a spiral mobile development process focused on usability was proposed. The process developed for this work is also based on non-functional requirements such as usability, but also takes into account important aspects such as context.

The spiral design of the process gives it a plan-based characteristics where some planning is made. However, the implementation stage adopts a more agile approach, where the requirements are separated into tasks and these tasks are selected dynamically in each iteration within this stage.

Each iteration belongs to a determined stage of the process and is divided into four phases: Determine Requirements; Design; Develop and Test; and Plan Next Iteration, each phase with its own set of steps. The whole process can be divided into three stages, as shown in Figure 4.1.

![Figure 4.1 Spiral process proposed](image-url)
4.2. THE DEVELOPMENT PROCESS

• **Stage 1 - Initial Project Setup** This stage represents the first iteration of the process and includes the following steps:

**Step 1.1 - Define all possible contexts**
The first step of the whole process is to define the contexts of the application, answering questions such as who is going to use the APP, how, and where. The context is important in the requirements elicitation and even in the architecture.

**Step 1.2 - Define usability attributes**
This step is responsible for choosing what usability attributes are relevant for the application. Nosseir *et al.* (2012) defines seven main usability attributes that can be used to assess the usability of mobile applications: effectiveness; efficiency; satisfaction; learnability; memorability; error; and cognitive load.

**Step 1.3 - Define system requirements**
All function and non-functional system requirements are defined and requirement decisions are made. Context information must be taken into account in order to generate functional and non-functional requirements. A special attention must be paid to the non-functional requirements, that can define the behavior of a mobile application.

**Step 1.4 - Create site-map**
The first step of the design phase is to create a site-map of the application in order to define how the information in the APP will interact. Site-maps are described in Chapter 2.

**Step 1.5 - Create wireframe with all screens**
The last step of this stage and iteration is to create a low-fidelity wireframe to have the first impression of the application’s screens and the interaction between them. The site-map must be used as a base to build the wireframe.

• **Stage 2 - Architecture and Prototype**
This stage represents the second iteration of the process which is responsible for the definition of a functional prototype and architecture. This stage prepares the project for the implementation stage.
Step 2.1 - Define tasks from the requirements
With the whole set of requirements defined, they are broken into tasks and all tasks are associated with an application’s screen, in order to allow the development of an entire screen and test its usability.

Step 2.2 - Define the system architecture
An architecture is made using the MVC pattern. The MVC pattern allows to separate the interface from the logic, which makes simpler to test functionality and usability separated. Also, it makes possible to change the interface with few or none impact to the logic.

Step 2.3 - Create a functional prototype
A functional prototype is made based on the wireframe. This prototype will be used in the first usability test.

Step 2.4 - Usability test using the defined attributes
In this step, in order make the first usability test using the prototype, some metrics are defined from the usability attributes.

Step 2.5 - Define next iteration screens and tasks to develop
At the end of this stage the architecture is created. The prototype is tested and validated, and the tasks are defined. In this step, the first implementation iteration is planned by choosing which screens will be developed.

Stage 3 - Implementation
This stage represents all other iterations, the total number of iterations depends on the size of the project and the tasks chosen to be developed. This stage applies a more agile concept.

Step 3.1 - Create functional screen interface
The first step of the implementation stage is to develop the interface elements of the screen.

Step 3.2 - Usability test using the attributes defined
Again an usability test is made. However, this time the test is applied to the developed screen. The same tests used in the prototype can be reused, or some new tests
4.3 Requirements Set

The requirements shown in this section can be grouped into two priority levels: high and medium. A high priority requirement must be implemented and the application cannot exist without it. Medium means that the application can exist without that requirement but it is highly recommend that it be implemented. All requirements were based on the results of the survey, presented in Chapter 3.

4.3.1 Functionalities

The application have two different kinds of users, the woman under protective measure and the contacts associated to her. The application does not support the registration of these users, they will be registered by the organization that will control the system, probably some government institution linked to the combat of the violence against women. However, the application supports an authentication system that, given an username and password, will authenticate the user. Then, the user will be specific sent to an environment inside of Artemis dedicated to the kind of user that logged in. In this dissertation, when it is needed to refer to the users of the application, the woman under protective measure will be called "main user" and each one of her contacts will be called "contact".

Besides the authentication system, Artemis has three main functionalities that was absorbed into the requirements.

- Proximity Detection
When the woman enters into protective measure, the aggressor denounced will be demanded to use a hardware (see section 4.5) that will able the application to detect his approach. Hence, the main functionality of the application is to detect this approximation and warn the user and a set of pre-registered contacts about it. They will be warned by some notification inside and outside of the app. Then, the location of the woman and her approximate distance to the aggressor will be showed in a map.

- **Send Emergency Alerts**
  The main user of the app, the woman under protective measure, will be able to send an emergency alert to all contacts. The contacts will receive the alert and will be able to see the location of the woman who send it in a map, like how is done in the Proximity Detection functionality.

- **Contact Emergency Numbers**
  The woman under protective measure will be able to contact directly one of her contacts by calling or SMS. Emergency numbers also will be available in the app, like the police number and the exclusive national number to denounce violence against woman. In turn, each contact will be able to contact the woman who are associated to them and also the emergencies numbers cited before.

4.3.2  **Functional Requirements**

Sommerville (2011) specifies functional requirements as statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations.

The functional requirements of Artemis have basically two kind of priorities. Requirements associated to the Proximity Detection functionality have high priority. The other two functionalities were defined as medium priority. The Proximity Detection functionality was set with high priority because none of the solutions found to combat the violence against women had an approach like this, so the priority was to develop this functionality.

- **FR 01 – Detect aggressor approximation**
  The application must detect when the aggressor come closer in a given distance. The detection will occur by a hardware, such as an iBeacon device (this technology will be addressed later in this chapter), worn by the aggressor.
4.3. REQUIREMENTS SET

Priority: High

• FR 02 – Warn the main user when the aggressor is detected

When the aggressor is detected the application must warn the main user with a notification followed by a sonorous warning, even if the application is closed or it is in background. The main user must be directed to the map with the aggressor’s location if tapped on the notification.

Priority: High

• FR 03 - Warn the contacts when the aggressor is detected

When the aggressor is detected the application must warn the contacts with a background push notification followed by a sonorous warning. The contact must be directed to the map with the main user location and aggressor’s location if tapped on the notification.

Priority: High

• FR 04 - Warn the main user when the aggressor leave the detection zone

When the aggressor leave the detection zone the application must warn the user with a notification followed by a sonorous warn, even if the application is in background.

Priority: High

• FR 05 - Warn the contacts when the aggressor leave the detection zone

When the aggressor leave the detection zone the application must warn the contacts with a notification inside the application. In this case no background push notifications must be made.

Priority: High

• FR 06 - Show aggressor’s location in a map to the main user

When the aggressor is detected the application must allow the user to see the possible aggressor location and her own location in the map. Due hardware limitations, the aggressors position in the map is just estimated to give the main user an idea of his location.

Priority: High

• FR 07 - Track aggressor and the user
When the aggressor is detected the application must allow the contacts to see the estimated aggressor’s location and the main user location in a map.

**Priority:** High

- **FR 08 - Start Emergency Alert**
  The application must allow the main user to send an emergency alert to the contacts which will allow them to track her location in a map.

  **Priority:** Medium

- **FR 09 - Stop Emergency Alert**
  The application must allow the main user to stop an emergency alert, stopping the tracking.

  **Priority:** Medium

- **FR 10 - Warn the contacts when the user start an Emergency Alert**
  If an emergency alert is sent by the main user, the application must warn the contacts with a background push notification and a notification inside the application. The contact must be directed to the map with the main user location if tapped on the notification.

  **Priority:** Medium

- **FR 11 - Authenticate user**
  The application must authenticate the user with password and user name. After the authentication be made, the user must be redirected to the specific screen of each kind of user.

  **Priority:** High

- **FR 12 - Emergency Numbers**
  The application must allow the user to fast reach its contacts or the main user (depending on the kind of user is logged in: main user or contact) in order to send a SMS or to make a phone call. The police emergency number and the emergency number for denouncing violence against women will be available as well.

  **Priority:** High
4.3.3 Non-Functional Requirements

According Sommerville (2011) non-functional requirements are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. In this case, the set of non-functional requirements is composed by some constraints regarding the mobile development as shown on Chapter 2.

- **NFR 01 - Usability**

  The application must be easy and instinctive to use. With a few steps and with no or little training the user must be able use the application’s functionalities. This requirement is very important for the user experience and for the Artemis’ context.

- **NFR 02 - Scalability**

  Technology changes must be easy to make in the application, on the server side and on the hardware used to locate the aggressor. Scalability is also important because mobile computing is always evolving, and new technologies are integrated within devices and platforms in relative small amount of time.

- **NFR 03 - Background capabilities**

  The main functionalities such as tracking and the warn notifications must work, even if the application is closed or in background mode.

- **NFR 04 - Portability**

  The application architecture and components must be developed in a way that the portability for another mobile platform can be performed without much effort.

- **NFR 05 - Security.**

  The access of the application and user data must be controlled and protected.

4.4 Architecture Overview

In order to have a better separation between the Graphical User Interface (GUI) and the logic, a MVC approach was chosen. This separation is important in order to develop the usability of the application. The scalability non-functional requirement was taken into account as well, the architecture modules allows changes of technologies without impact all application logic.
Also, most of the mobile SDKs follow the MVC pattern. Thus, the architecture proposed can be applied into the most important mobile platforms, attending the portability non-functional requirement. A server component was included into the architecture in order to manage the server side of the application.

In this case the MVC behaves like a layered architecture, where each component only communicates with other component in the layer below or above. Figure 4.2 shows a conceptual view of the architecture.

![Artemis' architecture conceptual view](image)

Figure 4.2 Artemis’ architecture conceptual view

### 4.5 Architecture Components

This section discusses each architecture component and its modules. Figure 4.3 shows a development view of the architecture, containing all modules and the interaction between them.
4.5. ARCHITECTURE COMPONENTS

Figure 4.3  Artemis’ architecture development view
• **View**

The View component includes all the GUI items, such as screens, text boxes, labels, tables, etc. This component only communicates with the controller component, and is not aware of the logic of the application, the View just sends user events to the Controller and receives possible GUI updates from it. In a mobile context the View generally will be composed by the native GUI components of the mobile platform, or customized versions of them.

• **Controller**

The Controller component is responsible for the interaction between the GUI and the logic of the application and to update its items by managing inputs from View and Model components and sending the results back to the GUI located in the View component. This component has three modules, each one controlling its respective view, which is a set of GUI elements representing a screen in the View component.

**Contact Controller**

The Contact Controller module is responsible for manage the contact screen logic, which is the screen of the application destined to a contact user.

**User Controller**

This module represents the interaction between the main user screen and its respective logic.

**Login Controller**

The Login Controller module deals with the authentication logic and the login screen. This controller is also responsible to set which kind of user is using the application, the main user (woman under protective measure) or a contact.

• **Model**

The Model component manages the application data and the communication with the Server. This component provides all data used in the application and send it to the controller, which is responsible for applying the system logic using the data. The Model also send requests to the Server component and returns the responses to the Controller. The Model component is composed by the following modules:
### 4.5. ARCHITECTURE COMPONENTS

**Artemis Server Handler**

This module manages the communication with the server responsible for controlling all interaction between main users and the contacts. The Controller uses this module to send emergency alerts and proximation notifications to the server.

**Beacon Handler**

The Beacon Handler manages the interaction of the application with the hardware used to detect proximation. The module detects the hardware and send a response to the User Controller module, warning about the proximation.

**Location Handler**

This module manages the GPS status and location monitoring of the main user. It is used by the User Controller module to get the current location of the main user, which is sent to the server in order to share the location with the contacts.

**WebService Handler**

Responsible for any requests made to the web service API. This module is used by Login Controller module to authenticate the user and is also used by User Controller module in order to send background push notifications to the contacts.

**Server**

The Server component is an abstraction of the server side of the application. This module is responsible for managing the server used to send information between main users and contacts and for managing the web service API. All requisitions made to the Server component are made by the Model.

**Artemis Server**

This module is responsible for the interaction between the data sent by the main user and the contacts. It makes possible the direct communication between devices and the information shared among them. Basically it manages the emergency alerts and the proximity detection information traffic.
WebService API and Push Server

This module is responsible for all web requisitions. It manages the user authentication and any changes in the database regarding the user data. It is also responsible for background push notifications, which is made by contacting the Push Server module. The Push Server is different for each mobile platform, it makes remote background notifications possible. It is responsibility of the WebService API to send the request to the right push server with the right parameters, depending on the platform used to send and to receive the notification. The push server used by Apple for iOS application is called APNS (Apple Push Notification Service), for exemple.

4.6 Technologies

The following technologies were used to build the application:

- **Cocoa Touch**

  It is Apple’s framework created to develop iOS applications, it is based on the Cocoa framework used to write Mac applications, but it has the focus on touch based interfaces. Cocoa Touch is composed by a series of frameworks and libraries, like UIKit, which is responsible for the iOS GUI. Cocoa touch is written in Objective-C.

- **Objective-C**

  It is the language used to develop the application. It is a superset of C language and provides object-oriented capabilities based on the Smalltalk programming language. Objective-C is one of the languages supported by Cocoa Touch. Apple recently announced the new Swift language also supported by the framework. It was chosen mainly because of the expertise of the developers on the language.

- **Python**

  It is the language used to develop the server used by the Artemis Server module. Python was chosen for this task because it is a well spread language with plenty of material available, but what really motivates its use for developing the server

---

2. [https://www.python.org](https://www.python.org)
was the Twisted\footnote{https://twistedmatrix.com/trac/} event-based engine, which makes simpler to write socket based applications that use TCP, UDP, SSH or other protocols.

- **PHP\footnote{https://php.net}**

  PHP is a popular general-purpose scripting language that is especially suited to web development. It was used to develop the API used by the WebService API module. It was chosen also because of the expertise of the developers with the language and because of the good performance of the language for HTTP requests, that can be easily made.

- **MySQL\footnote{http://www.mysql.com}**

  It is the data base used to persist data used by Artemis, mainly user data. It was chosen to be the project database because of its good integration with PHP and for its good documentation and material available.

- **iBeacon Hardware**

  iBeacon was the technology chosen to detect the presence of the aggressor. It is basically a low-energy bluetooth with an ID that can communicate with mobile applications using the embedded bluetooth 4.0 in the devices. It was chosen because it is a relative cheap hardware and it is multi platform, iOS and Android applications can talk with an iBeacon hardware.

The architecture was proposed in a way that this set of technologies could be changed, so Artemis can be portable to other mobile platforms and use other technologies.

### 4.7 Implementation

The first choice regarding the implementation was to develop using native code of each mobile platform. That choice was made because the application is relatively simple and it strongly relies on functionalities that can be implemented different in each platform, such as the communication with iBeacon hardware and the different kinds of push servers used in background notifications. Also a cross-platform approach would sacrifice the user experience, and the application must be as usable as possible, since usability is a
mandatory non-functional requirement. The issues regarding cross-platform approaches were shown in Chapter 2.

Although the Android platform could be a better choice to start the development of Artemis, since it is a cheaper hardware more available to the possible users of the application, it was chosen to develop in the iOS platform. This choice was made because of the expertise of the developers in the platform and because of the available hardware to make tests. However the architecture was made thinking on a multi platform context.

Artemis has about 1600 lines of code, according cloc tool\(^6\), and was developed in a period of about two months by a single developer, Table 4.1 shows the code statistics of the project.

The usability and functionality tests were made using an iPhone 5 running iOS 7.1 simulating a main user, and a second generation iPod Touch running iOS 6.1 simulating a contact user. The iBeacon hardware was simulated by a Macbook computer using a low-energy bluetooth 4.0 USB dongle.

### 4.8 Artemis in action

This section aims at showing Artemis operating its functionalities in order to demonstrate its use. The demonstrations follow the test scenario presented in Section 4.7, with an iPhone 5 used to represent a main user and a iPod Touch used to represent a contact.

**Main Screen**

Artemis only have one screen, all functionalities can be accessed from it. This decision was made according to the usability metris used to develop the application. This screen was projected to be easy to use and fast to learn the functionalities, according NFR 01 -

---

\(^6\)http://cloc.sourceforge.net

---

58
4.8. ARTEMIS IN ACTION

Figure 4.4 Artemis main screen - main user

Figure 4.5 Artemis main screen - contact user
Usability. Figure 4.4 shows the screen from a main user perspective and Figure 4.5 shows the same screen if a contact user is logged on.

The application functionalities are accessed as follows:

• **Main Screen - main user**

  From this screen the main user can send an emergency alert to the contacts by pressing the round red button, automatically receive notifications about proximity detection, and make emergency calls to the contacts or to emergency numbers using the rectangular buttons at the top and at the bottom of the screen.

• **Main Screen - contact user**

  The contact user basically can only trigger the Emergency Numbers functionality, in this case the round red button is used to contact the main user, instead of send an emergency alert. The Emergency Alert and the Proximity Detection are triggered by the main user and the contact automatically receives updates about it.

The two main functionalities will be demonstrated in this chapter.

**Proximity Detection**

This functionality is always triggered by the application every time the aggressor gets close of the main user. According *FR 01 – Detect aggressor approximation, FR 02 – Warn the main user when the aggressor is detected* and according *FR 03 - Warn the contacts when the aggressor is detected*, a notification is sent locally to main user and by background push to the contacts. Figure 4.6 shows the proximity being detected and the main user being warned. Figure 4.7 shows a contact receiving a background push notification.

Once a proximity is detected, both main user and contacts can access a map with the main user location and the aggressor’s location, following the *FR 06 - Show aggressor’s location in a map to the main user* and *FR 07 - Track aggressor and the user*. Figure 4.8 shows the proximity being detected and the main user being warned, and Figure 4.9 shows a contact receiving a background push notification. The main user location is represented by the red pin, and the aggressor’s location is represented by the red circle. Because of hardware limitations the application only is aware of the distance of the aggressor, the direction is not known by Artemis. Thus, a circle with the radius equals to the distance from the aggressor is drawn in order to estimate the location.
4.8. ARTEMIS IN ACTION

**Figure 4.6** Proximity detected - main user

**Figure 4.7** Proximity detected - contact user
Figure 4.8  Main user and aggressor’s location - main user

Figure 4.9  Main user and aggressor’s location - contact user
4.8. ARTEMIS IN ACTION

**Figure 4.10** Main user starting an emergency alert

**Figure 4.11** Contact receiving a push notification about the emergency alert
Send Emergency Alerts

An emergency alert is started when the main user taps the round red button and confirms the emergency alert, according to FR 08 - Start Emergency Alert. Then a background push notification is sent to the contacts warning about the emergency alert, according to FR 10 - Warn the contacts when the user start an Emergency Alert. Figure 4.10 shows an emergency alert being started, and Figure 4.11 shows a contact receiving a push notification about the emergency alert.

Once the emergency alert is started, a contact can access the main user location, as seen on Figure 4.12. Figure 4.13 shows the screen of the main user after an emergency alert was started.

4.9 Summary

In this chapter, it was presented Artemis, a mobile application created to combat the violence against women. It was presented the process used in order to develop the application, following the mobile constraints presented in Chapter 2. It was also presented the application requirements set, its architecture and used technologies, and a demonstration of Artemis’ main functionalities. Next chapter is going to present the conclusion of this dissertation.
Figure 4.13 Main user screen after starting an emergency alert
Conclusion

I hope I can make it across the border. I hope to see my friend and shake his hand. I hope the Pacific is as blue as it has been in my dreams. I hope.

—RED (The Shawshank Redemption)

The mobile computing has proven itself to be more than just a trend or another computing paradigm. It is changing how the users interact with technology and software, and how their lives are impacted by it. The future of this paradigm is holding important changes in the way software engineering faces the mobile development.

In order to guarantee the software quality and allow the users to have the experience that the mobile next generation will provide, the software engineering must evolve its techniques and tools to fit the new constraints of the mobile computing.

Chapter 2 presented the aspects and constraints of mobile development and how SE can use some of its techniques in a mobile context. Chapter 3 presented the real life problem of the violence against women and how a mobile application can help to combat it. A survey was presented in order to understand the problem and the possible role of a mobile application in this context. Chapter 4 presented Artemis, a mobile software proposed to improve the security of women under protective measure. It was shown its architecture, set of requirements and a demonstration of its operation. It was also shown the process for mobile development used to build Artemis.

5.1 Research Contribution

The main contributions of this research are described as follows:

- Overview on MSE and mobile computing
A brief overview regarding the use of software engineering in the mobile context was made in this dissertation. Therefore, it was presented the main aspects of mobile development and its applicability in the software engineering techniques.

• **Survey to solve real life problems**

Through this study it was possible to identify how to use data collection techniques in a multi-disciplinary context in order to develop a software to work on real life problems. The problem presented was the violence against women.

• **Artemis mobile application**

To use mobile software in order to help with the problem of violence against women it was developed Artemis. The main goal of the mobile APP is to improve the security of women who already denounce any kind of aggression and are at protective measure. We believe that the initial system is scalable and can be implemented by any city/state in Brazil.

## 5.2 Future Work

The application developed in this work is still a prototype. Thus, many enhancements must be implemented in Artemis, as well some defects must be fixed in order to turn the application fully operational and ready to be used. Also, some important concepts of mobile computing were left out of this work. These aspects of future work are described as follows:

• **Study on security and mobile networks**

These are two of the four core areas of mobile computing discussed in Chapter 2. The study performed only approached mobile development and MSE. However, the mobile security and networks are important as well and have a dominant role in the mobile next generation. Therefore, a detailed study on these two areas must be performed.

• **Proximity detection hardware improvements**

The hardware used in this work, in order to make possible to Artemis to detect the aggressor proximity, was an iBeacon hardware. iBeacons was introduced in Chapter 4. However, this kind of hardware has some limitations regarding the range and precision. Thus, a research must be conducted in order to improve the use of iBeacon hardware or to find another kind of technology to replace it.
• **Further evaluation**

The initial study made in this work aimed at find out more information about the problem presented and how a mobile application could be used to solve it. Now, since the application was developed, a further evaluation must be made to see how effective it is if applied in a real life scenario. Therefore, others empirical studies must be performed in order to evaluate the application.


IDC (2014a). Android and iOS Continue to Dominate the Worldwide Smartphone Market with Android Shipments Just Shy of 800 Million in 2013, According to IDC.

IDC (2014b). Worldwide Smartphone Shipments Top One Billion Units for the First Time, According to IDC.


Appendix
A.1 Form for Survey

1 - How old are you?

2 - It is the first time you denounce violence?
[ ] Yes. [ ] No.

3 - Have you ever ceased to denounce violence for any reason, or know someone who has left?
[ ] Yes. [ ] No.

In case you answered “Yes”, what were the reasons?

4 - Do you think the current denouncement and support system offered by the government are enough?
[ ] Yes. [ ] No.

Why? Describe some reasons.
5 - Do you feel safe about denouncing through the current model?
[ ] Yes. [ ] No.

Why? Describe some reasons.

6 - How do you think this system could be better and more efficient?

7 - Do you have a computer at home?
[ ] Yes. [ ] No.

8 - Do you have internet connection?
[ ] Yes, with wifi network.
[ ] Yes, without wifi network.
[ ] No.

9 - Do you own a smartphone or a tablet?
[ ] Yes, with 3G or 4G plan enable.
[ ] Yes, without 3G or 4G plan enable.
[ ] No.

The questions 10 e 11 only must be answered if the question 9 was answered affirmatively.

10 - How often do you use the internet on your smartphone and/or tablet?
[ ] Several times a day.
[ ] At least once a day.
[ ] Sporadically.
[ ] Almost never.
11 - What kind of mobile applications do you use more frequently in your smartphone and/or tablet? Mark in relation to frequency of use 0-5, where 5 is always used and never used 0.

[ ] Social Networks (Facebook, Instagram, WhatsApp, etc).
[ ] Games.
[ ] Utilities (Calendar, Calendar, Music Player, etc).
[ ] Services (Banking, shopping, etc).

12 - Do you think the use of technology (computers, cell phones and other devices) is capable of effectively help to prevent and fight the violence against women?

[ ] Yes. [ ] No.

Why? Describe some reasons.

13 - Do you are aware of any endeavor to fight the violence against women that use some kind of technology?

[ ] Yes. [ ] No.

In case the answer is "Yes" name the tools

14 - The city of Salvador is studying to implement in 2014 the use of a device (panic button) in order to help to prevent the violence against women who are under protective measure granted by the justice. Do you thing this is a valid measure?

[ ] Yes. [ ] No.

Why? Describe some reasons.
15 - How do you think that computer programs and/or applications for smartphones and tablets can assist on fighting the violence against women?


16 - Do you have some comment about this study or about the questionnaire?


