# An investigation into security aspects addressed during the development of enterprise mobile applications

**K. Kemp**

**22731865**

Dissertation submitted in partial fulfilment of the requirements for the degree *Magister Scientiae* in Computer Science at the Potchefstroom Campus of the North-West University

Supervisor: Prof HM Huisman

Co-supervisor: Dr L Drevin

May 2017

NORTH-WEST UNIVERSITY ®
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT

It all starts here ™

# ACKNOWLEDGEMENTS

I firstly want to thank my two supervisors, Prof. Magda Huisman and Dr. Lynette Drevin from the North-West University of Potchefstroom. They were always there when I needed help and guided me to the completion of this study.

I want to thank each and every person that participated in this study for their time and support.

I want to thank my parents. If it weren’t for them, I wouldn’t have this opportunity and I wouldn’t be the man I am today. Thank you for all the time you spent into sculpting me into this person, thank you for always supporting me in everything even though I have made mistakes, and finally thank you for always giving me money when I was broke...

Finally, I want to thank someone, or something that I don’t think gets enough recognition, and that is Google. Google, I want to thank you for always being there, not only in my studies, but also in times of procrastination and doubt. Thank you for all the information you have given me and thank you for your Scholar section when your other information was deemed useless.

# ABSTRACT

The rapid escalation in the use of mobile devices in enterprises has also increased the number of enterprise mobile applications (EMAs) being developed. It seems that security is not comprehensively defined in the software development methodologies (SDMs) of mobile applications. The purpose of the study is to acquire knowledge of whether enterprises that use mobile device architectures have adequate security measures in place regarding information assets and processes when developing mobile applications.

The approach of this study is interpretative in nature. Extensive literature reviews of SDMs and security aspects were done. This was followed by case studies conducted at companies where interviews from experts were mainly used to gather data on the development of EMAs. Theme analysis and cross-case analysis to provide were used to create a framework that may be used as a guideline for developing EMAs by incorporating security aspects.

The findings of the study include that little is revealed in literature regarding security implementations during the development of mobile software and that the methodologies for developing mobile applications are not well described in terms of security processes. This study contributes towards the discipline of secure software development and specifically EMAs by presenting a framework with guidelines to developers to include security when developing EMAs.

**Keywords**

Case studies, enterprise mobile applications, mobile application development, mobile applications, secure development framework, security, software development methodology.

# OPSOMMING

Die vinnige toename in die gebruik van mobiele toestelle in ondernemings het ook ‘n toename in die ontwikkeling van ondernemingsmobiele toepassings (OMTs) teweeg gebring. Dit blyk dat sekuriteit nie volledig omskryf word in die sagteware-ontwikkelingsmetodologieë (SOMs) van mobiele programme nie. Die doel van die studie is om inligting in te samel oor ondernemings wat mobiele toestelargitekture gebruik en te kyk of hulle voldoende sekuriteitsmaatreëls in plek het ten opsigte van inligtingbates en -prosesse tydens die ontwikkeling van mobiele toepassings.

Die studie is interpretatief van aard en uitgebreide literatuuroorsigte van SOMs en veiligheidsaspekte is gedoen, gevolg deur die uitvoer van gevallestudies by maatskappye waar onderhoude met kundiges hoofsaaklik gebruik is om data in te samel oor die ontwikkeling van OMTs. Tema- en kruisgevalontledings is gebruik om 'n raamwerk op te stel wat gebruik kan word as 'n riglyn vir die ontwikkeling van OMTs deur veiligheidsaspekte te inkorporeer.

Die bevinding van die studie is dat daar min inligting in die literatuur geopenbaar is in verband met sekuriteitsimplementering tydens die ontwikkeling van mobiele sagteware en dat die metodes vir die ontwikkeling van mobiele toepassings nie beskrywings van sekuriteitsprosesse insluit nie. Hierdie studie dra by tot die dissipline van beveiligde sagteware-ontwikkeling en spesifiek tot OMT ontwikkeling deur die aanbieding van 'n raamwerk met riglyne vir ontwikkelaars om sekuriteit in te sluit in die ontwikkeling van OMTs.

**Sleutelterme**

Gevallestudies, mobiele toepassing ontwikkeling, mobiele toepassings, ondernemingsmobiele toepassings, sagteware ontwikkelingsmetodologie, sekuriteit, beveiligde ontwikkelings raamwerk.

# TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| AD | Agile Data |
| AES | Advanced Encryption Standard |
| AM | Agile Modelling |
| API | Application Program Interface |
| ASD | Adaptive Software Development |
| BDD | Behaviour Driven Development |
| BYOD | Bring Your Own Device |
| BYON | Bring Your Own Network |
| CDM | Custom Development Method |
| CSP | Cloud Service Provider |
| DMAIC | Define, Measure, Analyse, Improve and Control |
| DMZ | Demilitarised Zone |
| DSDM | Dynamic System Development Methodology |
| EMA | Enterprise Mobile Application |
| FDD | Feature Driven Development |
| FMEA | Failure Mode and Effects Analysis |
| FTA | Fault Tree Analysis |
| GPS | Global Positional System |
| GUI | Graphical User Interface |
| HTML5 | HyperText Markup Language 5 |
| HTTP | HyperText Transfer Protocol |
| HTTPS | HyperText Transfer Protocol Secure |
| ID | Identity Document |
| IDE | Integrated Development Environment |
| IMEI | International Mobile Station Equipment Identity |
| ISD | Information System Development |
| ISO | International Organization for Standardization |
| ISP | Internet Service Provider |
| LD | Lean Software Development |
| LSS | Lean Six Sigma |
| MAD | Mobile Application Development |
| MADM | Mobile Application Development Methodology |
| MAM | Mobile Application Management |
| MASAM | Mobile Application System Agile Methodology |
| MDM | Mobile Device Management |
| MITM | Man-In-The-Middle |

| | |
|---|---|
| NAS | Network Attached Storage |
| NFC | Near Field Communication |
| NIST | National Institute of Standards and Technology |
| OS | Operating System |
| COTS | Commercial Off-The-Shelf |
| OWASP | Open Web Application Security Project |
| PKI | Public Key Infrastructure |
| POPI | Protection of Personal Information |
| RFID | Radio-Frequency Identification |
| RUP | Rational Unified Process |
| SD | Secure Digital |
| SDK | Software Development Kit |
| SDLC | Systems Development Lifecycle |
| SDM | Software Development Methodology |
| SIM | Subscriber Identity Module |
| SIPOC | Suppliers, Inputs, Processes, Outputs and Customers |
| SLeSS | Scrum and Lean Six Sigma |
| SME | Small and Medium Enterprises |
| SMME | Small, Medium and Micro Enterprises |
| SMS | Short Message Service |
| SPEM | Software and Systems Process Engineering Meta-model |
| SQL | Structured Query Language |
| SSL | Secure Sockets Layer |
| TDD | Test Driven Development |
| UbiComp | Ubiquitous Computing |
| UHRFID | Ultra High Radio-Frequency Identification |
| UI | User Interface |
| USB | Universal Serial Bus |
| USP | Unified Software Process |
| VPN | Virtual Private Network |
| XP | eXtreme Programming |

# CHAPTER 1 INTRODUCTION

## 1.1 Introduction

Chapter 1 presents an overview of the study. Firstly, the problem statement is presented. After this a literature review is given that leads to the research question. Next the method of research is discussed and the chapter concludes with the presentation of the structure of the dissertation and a summary of the chapter. Figure 1-1 shows a representation of the chapter.

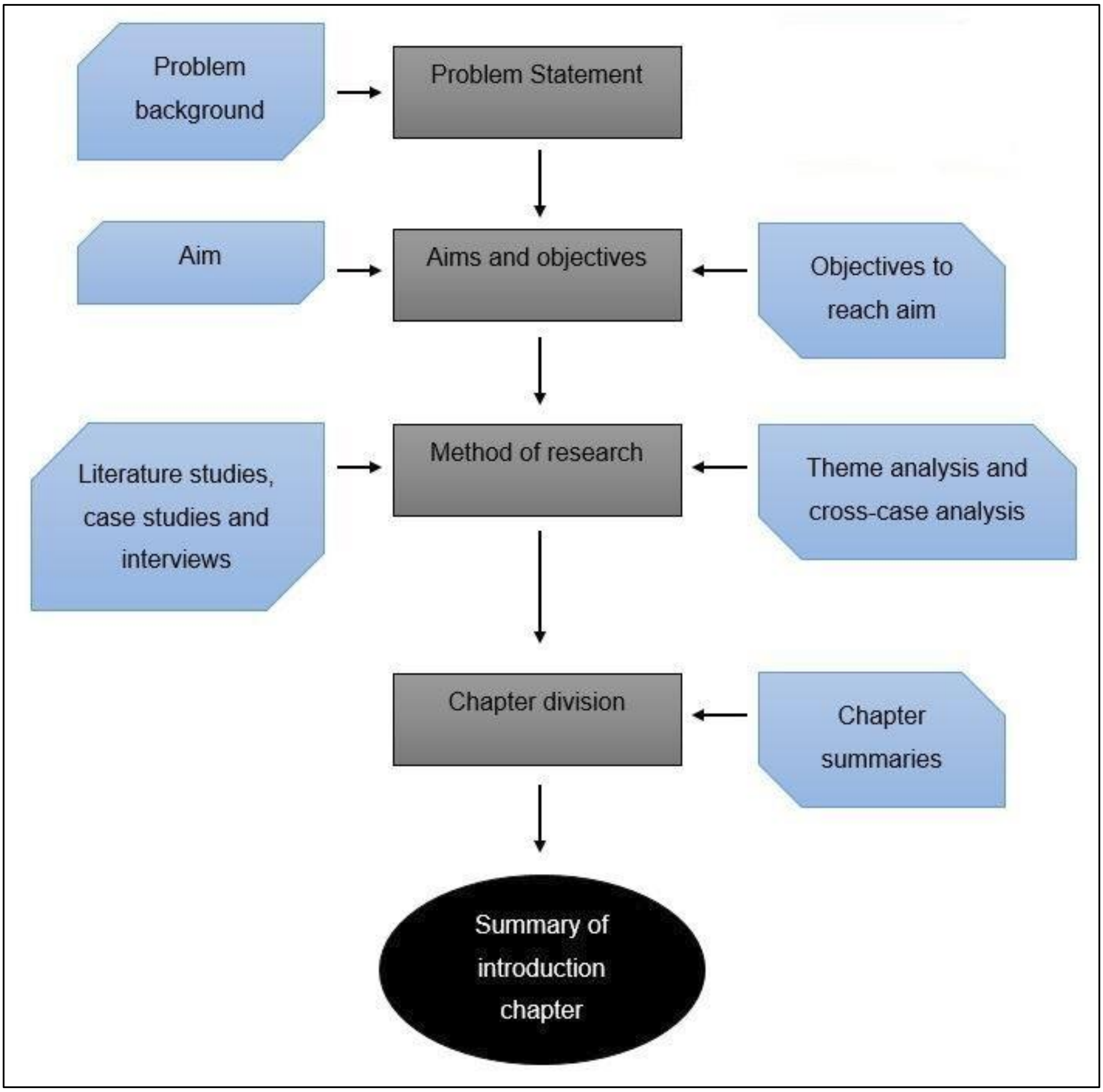


**Figure 1-1: Chapter 1 representation**

## 1.2 Problem statement and background

This section provides background on the problem at hand and gives a clear description thereof. It will conclude with a problem statement as well as a research question for the study.

### 1.2.1 Problem background

Enterprise mobile applications (EMAs) are enterprise software that run on mobile devices (Giessmann, Stanoevska-Slabeva & De Visser, 2012:1364) and improve the interactions between employees and business partners (McAfee, 2006:145) while improving productivity. According to Giessmann *et al.* (2012:1364), EMAs support users in their core business processes.

Jain and Shanbahg (2012:28) state that mobile applications are often developed at a very quick pace, without properly addressing security. With mobile applications being developed at such a pace, the malicious activities surrounding them are also increasing at a fast pace (Ahmad, Francis, Ahmed, Lobodzinski, Audsin & Jiang, 2013:575). Leavitt (2013:17) claims that one or more security flaws were found in 90 percent of tested mobile applications; according to Zhu, Ziong, Ge and Chen (2014:951), this is leading to users being more and more hesitant in adopting mobile applications.

Security in EMAs should have a high priority during development, considering the risk of information being stolen or lost (Gajar, Ghosh & Rai, 2013:63). As a result of the global increase in mobile devices, the use of mobile devices and mobile applications has also increased in enterprises. It should be noted that this integration of mobile devices into the enterprise does not come without risks and challenges. Some of the challenges include: Bring Your Own Device (BYOD) with the associated risks, device and/or data loss, interception of data (which can be threatening to a business that keeps sensitive data on mobile devices), malware (malicious software), vulnerable applications and compromised devices (Delac, Silic & Krolo*,* 2011:1469-1470; Hasan, Dmitriyev, Gómez & Kurzhöfer, 2014:1; Lin, Huang, Wright & Kambourakis*,* 2014:22). Many of these challenges can, and should, be addressed during application development.

The process by which mobile applications and EMAs may be developed is called Mobile Application Development (MAD). This process may be supported by Mobile Application Development Methodologies (MADMs). Current MADMs make use of different aspects of agile Software Development Methodologies (SDMs) (Flora & Chande, 2013:10). These MADMs are all constructed by combining different agile SDMs (Flora & Chande, 2013:12), which do not have any visible security models in their frameworks. The following paragraph gives insight into the lack of security in different SDMs.

Sani, Firdaus, Jeong and Ghani (2013:37) state that the security element is unavailable in the different phases of an agile Software Development Methodology (SDM) called Dynamic System Development. Azham, Ghani and Ithnin (2011:414) agree with this statement by asserting that Scrum (another SDM), along with other agile methods, do not include security practices or implementations. Ghani, Izzaty and Firdaus (2013:1071) also mention Scrum, eXtreme Programming (XP) and Feature Driven Development (FDD) as agile SDMs that do not show elements of security in their various phases.

Some examples of MADMs are: Mobile-D (Abrahamsson, Hanhineva, Hulkko, Ihme, Jäälinoja, Korkala, Koskela, Kyllönen & Salo*,* 2004:174), Scrum and Lean Six Sigma (SLeSS) (Da Cunha, Dantas & Andrade, 2011:283), MASAM (Jeong Lee & Shin, 2008:363) and Scrum, which is used in software development but can be used in the development of different mobile applications (Da Cunha *et al.,* 2011:284-285). MADMs are constructed from different agile SDMs, which show little to no signs of security implementation. Investigation into MADMs separately also show little signs of security implementation (Sathyan & Sadasivan, 2010:1). This lack in security is the catalyst for conducting this study and identifies a gap to be filled.

### 1.2.2 Problem statement

Based on the initial investigation by the author, little is known about the security in the development of EMAs. The MADMs and SDMs used show inadequate evidence of security in the design and implementation.

The question to be answered in this study is whether security aspects are being addressed in the Enterprise Mobile Application (EMA) development process, and if not, what can be done to improve the problem. A framework will be constructed by incorporating different security aspects into the MAD development process as guidance to improve security when developing EMAs. These guidelines are important to address the discrepancy between the small increase in security and the large increase in mobile device usage, as well as the mobile application increase.

Now that the problem and the research question are clear, the aims and objectives of the study can be discussed.

## 1.3 Aims and objectives

Along with the problem statement, research on SDMs and MADMs reviewed showed little to no signs of guidelines provided for security implementation. This gave lead to the research title and the reason for the study.

*“An investigation of the security aspects addressed during the development of enterprise mobile applications.”*

As the title suggests, the purpose of the study is to acquire knowledge on whether enterprises that use mobile device architectures have adequate security measures in place regarding information assets and processes when developing mobile applications.

The aim of this study is to investigate the development of EMAs, together with any security challenges that might occur (during or related to their development). The secondary objectives needed to be satisfied to reach the aim are:

- To develop a framework to guide developers regarding different security aspects that could be included during the development of EMAs.
- To investigate existing MADMs;
- To conduct a literature review on different information security practices and policies in enterprises;
- To determine the different information security aspects and policies used in existing SDMs and MADMs;
- To investigate development practices of EMAs in different enterprises, as well as how and why they address general information security in the development;
- To use the information obtained in this study to suggest improvements for existing development practices; and

Having clarity on the aim of the study, the method of research is now discussed.

## 1.4 Method of research

The research paradigm that underlines this study is interpretivism. Interpretivist studies attempt to identify, discover and clarify (what, how and why) the different elements of a social setting and how they are related and co-dependent (Oates, 2006:292). The social setting in this study was an enterprise and the ‘What’, ‘How’ and ‘Why’ refer to the following:

- **What –**The security aspects and policies surrounding EMAs and the development thereof;
- **How –** How the enterprise develops EMAs and how security is implemented during development; and
- **Why –** Why the specific practices and policies are implemented.

Interpretivism mainly uses qualitative data analysis. Lakshman, Sinha, Biswas, Charles and Arora (2000:371) claim that qualitative research relates to any situation where the research variables are not obvious or when the number of participants is inadequate for statistical analysis. Oates (2006:266) agrees by stating that qualitative data include any data which are not numerical. The data are generally found in interviews, websites, models, different types of organisational documents and through case studies.

In this study, multiple case studies will be done by conducting semi-structured interviews at different enterprises. The data analysis will be done in a qualitative manner by means of theme analysis and cross-case analysis.

The research methodology will be discussed in more detail in Chapter 4. The next section presents the structure of the dissertation.

## 1.5 Chapter division

### Chapter 1: Introduction

This chapter presents insight into the background of SDMs, MADMs and the security thereof. It elaborates on the problem statement and research question, and states the different aims and objectives. Finally, this chapter presents the research method that will be applied and gives a summary of the study’s layout.

### Chapter 2: Systems development methodologies

Chapter 2 is the first of two literature review chapters, discussing various types of SDMs. It starts with the definition of SDMs and proceeds to discuss agile development as well as a few agile SDMs. It then proceeds with a discussion of MAD and how it coincides with agile development. The chapter concludes with a discussion of challenges during MAD, including security as a challenge.

### Chapter 3: Security

This chapter, the second of the two literature review chapters, discusses different security aspects related to enterprises. It presents different security threats to mobile devices and mobile applications, after which it concludes in a discussion on different security implementations during the development of EMAs.

### Chapter 4: Research methodology

This chapter elaborates on the research method/process, paradigm and purpose of the study. It also discusses the manner in which the data were collected via interviews and how it was analysed by means of theme analysis and cross-case analysis.

### Chapter 5: Data collection and analysis

This chapter gives a summary of each case (organisation) where data were collected. It presents the different methods of data collection, as well as the process in which data is analysed via theme analysis and cross-case analysis.

### Chapter 6: Results and discussion

This chapter presents the framework created from the results as well as a discussion of its elements. Each node of the framework is presented and discussed in detail to help guide developers in the development of EMAs and other mobile applications.

**Chapter 7: Conclusion**

This is the final chapter that shows how each objective was reached. The chapter includes the conclusion that was drawn from the study and enforces the motivation for the research. The chapter concludes with the study's limitations, as well as different recommendations for future work.

## 1.6 Summary

This chapter served as an introduction to the study. It presented the problem background, as well as the problem statement and research question. The chapter introduced the aims and objectives, as well as the study's research method. The chapter concluded with a chapter division, presenting a summary of each chapter.

The next chapter presents the first literature review topic, namely Systems Development Methodologies.

# CHAPTER 2 SYSTEMS DEVELOPMENT METHODOLOGIES

Chapter 2 introduces and discusses relevant topics and concepts from a number of different literature sources. These include the definition of SDMs, agile SDMs and MADMs. The chapter presents a summary of SDMs, agile SDMs and MADMs, and indicates how each of these elements complement each other. Figure 2-1 shows a representation of the chapter.

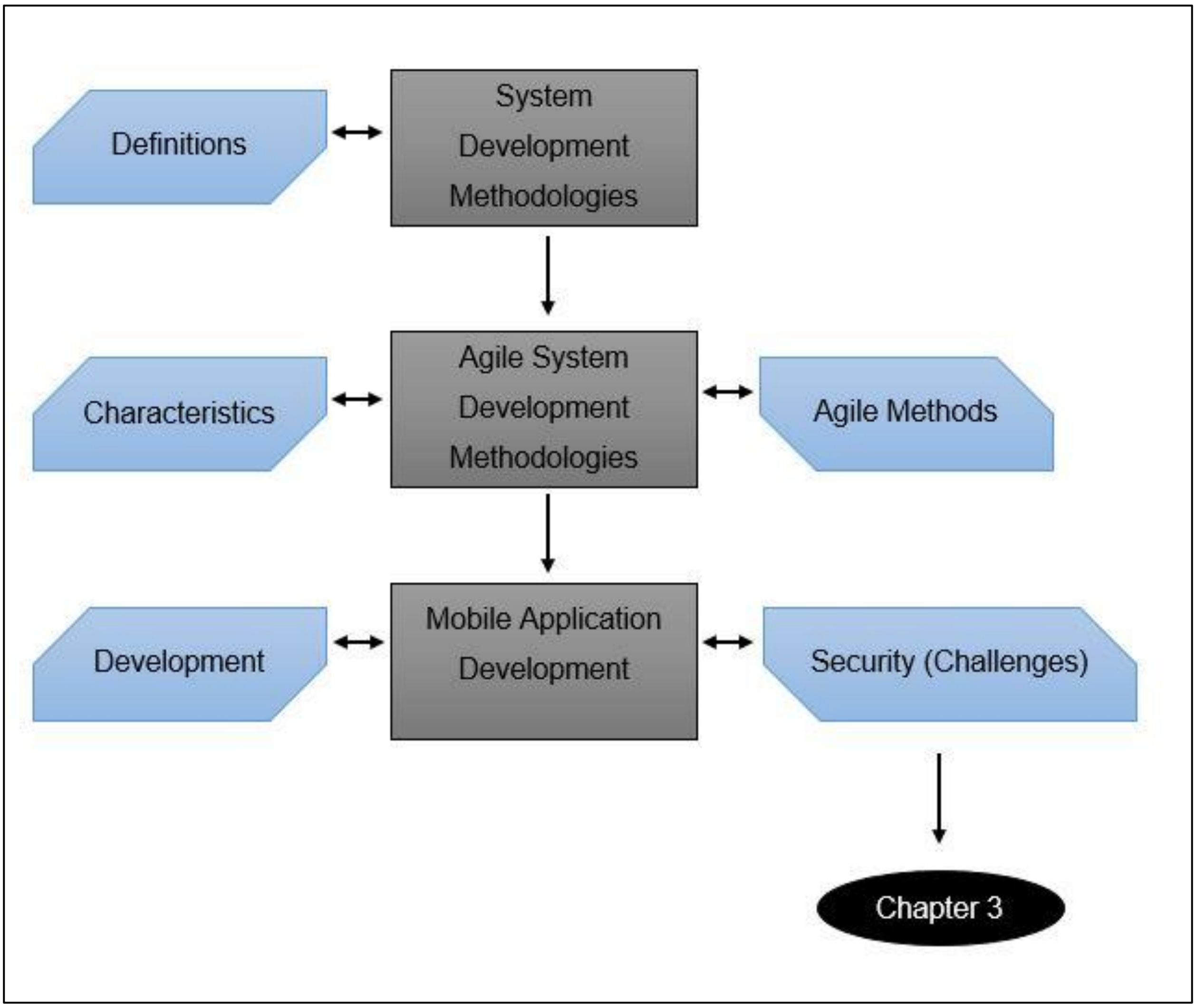

Definitions
System Development Methodologies
Characteristics
Agile System Development Methodologies
Agile Methods
Development
Mobile Application Development
Security (Challenges)
Chapter 3

**Figure 2-1: Chapter 2 representation**

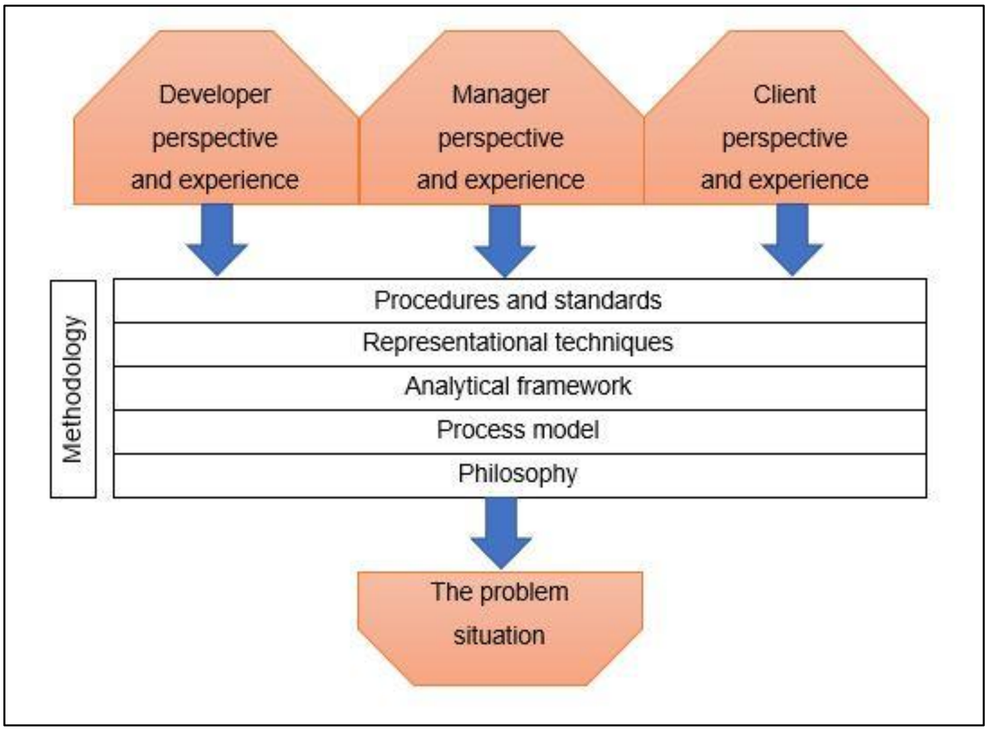
## 2.1 Definition of SDM

This section will elaborate on what a SDM is and how it can be interpreted by different people, by looking at different definitions from literature. According to Iivari, Hirshheim and Klein (2000:193), a SDM is a good approach to follow in solving complex development problems, although the methodology is not easy to use. Mathiassen and Sandberg (2014:62) agree by stating that the software development industry has extensively adopted SDMs, although it remains a very complex process. For these reasons, and because defining a SDM is no easy task, a number of different definitions are presented in Table 2-1, as found in the literature.

**Table 2-1: Different definitions for SDMs**

<table>
<tr><th>Cited from (Author)</th><th>Definition</th></tr>
<tr><td>Susan 1995</td><td>

Susan (1995:1) says a methodology can be seen as more than just a method, but rather a sequential group of elements coming together. These elements are illustrated in Figure 2-2, ordered from the top to the bottom.



Developer perspective and experience
Manager perspective and experience
Client perspective and experience
Methodology
Procedures and standards
Representational techniques
Analytical framework
Process model
Philosophy
The problem situation

**Figure 2-2: Elements of a SDM**

A methodology embodies an analytical framework which is initiated through a set of analytical methods, tools and techniques. The next element is the process model which relays information about the development activities. Standards and procedures, including participation of users and formal meetings, are an important part of the methodology definition, since techniques and outputs have to conform to standards that were pre-defined. Lastly, the philosophical basis is required to justify each element’s need in the methodology, as well as commitment from and between the development team.

</td></tr>
</table>

| **Cited from (Author)** | **Definition** |
|---|---|
| Susan 1995 | Susan (1995:2) describes a SDM as *“an holistic approach: it embodies an analytical framework which is conveyed through inter-subjective representational practices and operationalised through a ‘toolbox’ of analytical methods, tools and techniques”.* |
| Brinkkemper 1996 | Brinkkemper (1996:276) defines the methodology of information systems development as *“the systematic description, explanation and evaluation of all aspects of methodical information systems development”.* |
| Iivari, Hirshheim and Klein 1998 | Iivari *et al.*, (1998:196) defines an Information System Development (ISD) approach as “a set of *guiding principles, fundamental concepts, and principles for the ISD process”.* This idea of an approach agrees with the philosophy statement referred to in Figure 2-2. From the definitions it can also be seen that SDMs contain different processes with different methods which have to be completed in a specific order. This is visible even in the earliest of definitions. It is important to note that this whole collection (philosophy, methods, processes, tools and techniques) are needed to solve a development problem in the optimal way. |
| Huisman and Iivari 2006 | Huisman and Iivari (2006:32) give a more detailed description of a SDM, based on the definition of Iivari *et al.* (1998:196). They claim that a SDM can be seen as a combination of the following:<br>• **Systems development approach,** which can be seen as the philosophical goals on which the SDM is built. These goals include guiding principles, beliefs, fundamental concepts and the SDM principles which drive the developmental decisions and actions.<br>• **Systems development process model,** which can be seen as the different phases of development and the sequence in which they are completed.<br>• **Systems development method,** which is the methodical way of completing one of the phases mentioned in the process model. This includes guidelines, techniques, tools and any activity followed in the method. |

| Cited from (Author) | Definition |
|---|---|
| | - **Systems development technique(s),** which can be seen as different ways activities are performed. The procedures are followed to complete an activity. |
| Avison and Fitzgerald 2006 | Avison and Fitzgerald (2006:568) claims: *"A systems development methodology is a recommended means to achieve the development, or part of the development, of information systems based on a set of rationales and an underlying philosophy that supports, justifies and makes coherent such a recommendation for a particular context. The recommended means usually includes the identification of phases, procedures, tasks, rules, techniques, guidelines, documentation and tools. They might also include recommendations concerning the management and organisation of the approach and the identification and training of the participants".* |
| Ispareh, Ladani, Panahi and Azadani 2010 | Ispareh *et al.* (2010:1) explain that there are different phases that must be carried out when using SDM. According to them, the most general phases are the requirements analysis phase, the conceptual design phase and the implementation phase. |
| Rehman, Rauf and Shahid 2010 | Rehman *et al.* (2010:1) describe a SDM as a strategy which the development team follows. They propose that the strategy consists of different processes, methods, and tools. |
| Ruparelia 2010 | Ruparelia (2010:8) defines a Systems Development Lifecycle (SDLC) as *"a conceptual framework or process that considers the structure of the stages involved in the development of application from its initial feasibility study through to its deployment in the field and maintenance".* Furthermore, Ruparelia (2010:8) states that there are various models that describe several approaches to the SDLC process and that the SDM will describe the steps that are within the process. |
| Conger 2011 | Conger (2011:4) defines a methodology as *"the tenets, tools, philosophy, and so on about how to approach problem analysis and design. Within a life cycle stage, a methodology guides the work via tools and techniques,* |

| Cited from (Author) | Definition |
|---|---|
| | *focusing analysis on a specific aspect of the work".* |
| Consel 2011 | Consel (2011:77) declares that software development relies on a development paradigm to be successful. This paradigm helps to structure, program and compose the different parts of the system. This paradigm is normally incorporated in the form of a programming language, as well as the tools and techniques used to complete development. Consel (2011:77) also claims that programming is a very important part of the development process, although it is not the only important part. |
| Mathiassen and Sandberg 2014 | Mathiassen and Sandberg (2014:62) describe a software process or SDM as a set of parallel and serial activities that, in unison, create a software product of value. |

From Table 2-1 it can be observed that the definition of a SDM has remained mostly unchanged through the years. All of the literature sources consulted and presented in Table 2-1 concur that a SDM consists of tools and techniques that can be used in different ways and in different stages or phases of development. In some of the later definitions it should be noted that the development team and stakeholders are more involved in different aspects of the methodology than in earlier years. None of these definitions make reference to security aspects.

Based on this comparison, a SDM can be regarded as a structured process of development that includes different stages, participants (stakeholders) and methods. Tools and techniques are used in these different stages to fulfil the development need of the user, as well as the philosophical need of the SDM and software. The software should be completed by standards which are accepted by the user and other relevant participants.

For the purpose of this study, agile SDMs and MADMs will be discussed in more detail in the following sections, before discussing security. The next section will elaborate on agile SDMs.

## 2.2 Agile SDMs

Agile SDMs will be discussed in this section due to its integration with MADMs (refer to Sections 2.2.1, 2.2.2 and 2.3.2). Corral, Sillitti and Succi (2013:19) state that many different frameworks used for MAD intersect with agile development.

According to Abrahamsson, Salo and Ronkainen (2002:3), agile development can be defined as software development methods that attempt to offer a solution to the business community's eagerness for lightweight development with faster and more flexible development processes. Another definition of agile SDMs can be interpreted through the work of Jönsson (2013:3): agile SDMs are SDMs adapted from the traditional waterfall model to be more flexible; it is centred on the principles of self-organising and cross-functional development with close collaboration between different development teams. Jönsson (2013:4) also claims that time and speed, along with precision, should be emphasis as part of the methodology.

Agile methods were introduced in the late 1980s in an attempt to compensate for the limitations of traditional SDMs (Rehman *et al.,* 2010:3). According to Tan, Tan and Teo (2008:88), the development requirements and business environment change constantly. As a result, the software solution meeting the user requirements was no longer dependent on whether the completed product satisfies the initial plan, but rather whether it satisfies the user's needs at the time of completion. Agile SDMs address this issue introducing a more incremental and iterative approach (Rehman *et al.,* 2010:3).

In addition to these claims regarding agile SDMs, more recent research from Tripp and Riemenschneider (2014:3993) show that agile SDMs overall produce better software. They also state that teams who develop applications by using agile methods are shown to be more motivated and satisfied. Akbar and Safdar (2015:315) also claim that frequent iterations, faster coding, faster delivery, little documentation, adaptability and the constant changing of requirements in agile SDMs have largely been accepted by the software development world. Ambler (2009:2) agrees by stating that agile development produces better software since development occurs in regular intervals where each interval is reviewed and corrected according to changes in requirements. Ambler (2009:6) provides the following definition for agile development, despite stating that it is difficult to provide a fixed definition:

*"Agile software development is an evolutionary (iterative and incremental) approach which regularly produces high quality software in a cost effective and timely manner via a value driven lifecycle. It is performed in a highly collaborative, disciplined, and self-organising manner with active stakeholder participation to ensure that the team understands and addresses the changing needs of its stakeholders. Agile software development teams provide repeatable results by adopting just the right amount of ceremony for the situation they face".*

Ambler (2009:4) claims that seventeen methodologists (now known as the Agile Alliance) came together and formed the Agile Manifesto in an attempt to confront the challenges that were faced by developers at the time. Something that should be noted is that these seventeen individuals agreed on various issues of agile development, although they came from different backgrounds.

According to the Agile Manifesto, agile SDMs build on four values (Tan *et al.,* 2008:88; Ambler, 2009:4; Fowler & Highsmith, 2001:2):

1. The individuals and interactions between them are above the tools and techniques used;
2. The working software is above the documentation thereof;
3. Collaboration with the user is above contract negotiations; and
4. Responding correctly to change is above following a fixed plan.

In addition to these four values, the Agile Manifesto also lists twelve supporting principles (Beck, Grenning, Martin, Beedle, Highsmith, Mellor, Van Bennekum, Hunt, Schwaber, Cockburn, Jeffries, Sutherland, Cunningham, Kem, Thomas, Fowler & Marick., 2001; Fowler & Highsmith, 2001:3-5):

1. Customer satisfaction through means of quick and continuous delivery is the highest priority;
2. Agile processes use constant change in requirements as a competitive advantage and thus is open to it;
3. Deliver working software as quickly as possible, from as little as a couple of weeks or months;
4. Users and developers have to work together on a daily basis throughout the project lifecycle;
5. Projects have to be completed around motivated individuals; the workers have to be trusted in the work they do and their work environment should support their needs;
6. Face-to-face conversation should be a priority over electronic communications; it is more effective and more efficient;
7. The primary measure of project progress is working software;
8. All participants should be able to maintain an indefinite constant working pace; agile development promotes this;
9. Agility is enhanced by good design as well as attention to technical detail;
10. Simplicity is very important; it can be seen as the art of expanding the amount of work not done;
11. The optimal designs and requirements appear from self-organised teams; and
12. Working in intervals, the team has time to reflect on their work and adjust to do better in the next interval, if necessary.

Ghobadi (2014:2) agrees that these values and principles, along with hard work and dedication, help in supporting developers using agile SDMs. The Manifesto, along with the principles, has helped the growth and acceptance of agile SDMs in different software organisations. Akbar and Safdar (2015:315) state that traditional software development practices have been largely replaced by agile SDMs and that agile approaches have become the preferred and suggested choice of software development approach.

### 2.2.1 Agile characteristics

After discussing the principles, values and the reason for the development of agile SDMs, the characteristics of this type of SDM are presented.

Nerur, Mahapatra and Mangalara (2005) describe the characteristics of agile SDMs as *"short iterative cycles of development driven by product features, periods of reflection and introspection, collaborative decision making, incorporation of rapid feedback and change, and continuous integration of code changes into the system under development".* This description entails all the properties previously mentioned in the definitions. Work from Nurer *et al*. (2005:75) as well as Akbar and Safdar (2015:316), on the characteristics of agile SDMs, are summarised in Table 2-2.

**Table 2-2: Agile SDM characteristics (Adapted from Nerur *et al.*, 2005:75; Akbar & Safdar, 2015:316)**

| **Characteristic** | **Description** |
|---|---|
| **Fundamental assumptions** | High-quality, adaptive software can be developed by small teams using the principles of continuous design improvement and testing based on rapid feedback |
| **Control** | People centric |
| **Management style** | Leadership-and-collaboration |
| **Knowledge management** | Tacit |
| **Role assignment** | Self-organising teams – encourages role interchangeability |
| **Communication** | Informal |
| **Customer's role** | Critical |
| **Project cycle** | Guided by product features |

| Characteristic | Description |
|---|---|
| **Development model** | The evolutionary-delivery model |
| **Desired organisational form/structure** | Organic (flexible and participative encouraging cooperative social action) |
| **Technology** | Favours object-oriented technology |

In addition to the characteristics presented in Table 2-2, Table 2-3 illustrates additional characteristics identified by other authors.

**Table 2-3: Additional characteristics of agile SDMs (Adapted from Cockburn & Highsmith, 2001:131-132; Nerur *et al.,* 2005:74-75; Nerur & Balijepally, 2007:82; Tan et al., 2008:88; Dube & Dixit, 2010:196; Rehman *et al.,* 2010:3)**

| Characteristic | Description |
|---|---|
| **Process model** | Incremental and iterative approach |
| **Phases** | • User stories<br>• Release planning, including iterations of:<br>– Iteration planning<br>– Create unit test<br>– Develop code<br>– Continuous integration<br>– Acceptance test<br>– Small release<br>• System in use |
| **Order of steps** | Steps are carried out incrementally and occur iteratively |
| **Documentation** | The development of the software is more important than the documentation |
| **Management** | • Teams are self-organised<br>• Leadership-collaboration<br>• Manager is a facilitator |
| **User involvement** | • User involvement and collaboration is key |

| Characteristic | Description |
|---|---|
| | - Getting expert user involvement gives developers quick feedback of what the users want in the end |
| **Time and resources** | - The time and budget is fixed<br>- The functionality of the system is based on the time and budget |

Projects that are developed using agile SDMs are broken down into smaller projects (iterations/cycles) where each iteration is formed by several phases (Nerur *et al.*, 2005:75):

- Planning;
- Development;
- Integration;
- Testing; and
- Delivery.

The phases proposed here differ slightly from the phases presented by Dube and Dixit in Table 2-3; the phases used depend on the specific SDM used. At the end of each small project following these phases, the output is a small release which can be presented to the customer. This release can also be re-evaluated to determine if the current state of the release satisfies the needs of the customer. The changes made during this step will be incorporated in the next iteration (Rehman *et al.*, 2010:3).

There are many different agile SDMs. Table 2-4 presents a summary of the different agile SDMs mentioned in different literature sources.

**Table 2-4: Different agile SDMs mentioned in literature**

| Author(s) | SDMs mentioned in literature |
|---|---|
| Cockburn and Highsmith (2001:132) | - Dynamic System Development Methodology (DSDM)<br>- XP<br>- FDD<br>- Scrum<br>- The Crystal Methodologies |
| Boehm (2002:64) | - Adaptive Software Development (ASD)<br>- Agile Modelling (AM)<br>- Crystal Methods |

| Author(s) | SDMs mentioned in literature |
|---|---|
| | • DSDM<br>• XP<br>• FDD<br>• Lean Software Development (LD)<br>• Scrum |
| Paetsch, Eberlein and Maurer (2003:10-11) | • ASD<br>• AM<br>• DSDM<br>• XP<br>• FDD<br>• Scrum<br>• The Crystal Methodologies |
| Chow and Cao (2007:962) | • DSDM<br>• FDD<br>• XP<br>• LD<br>• Scrum<br>• The Crystal Methodologies |
| Ambler (2009:12) | • Agile Data (AD)<br>• AM<br>• XP<br>• FDD<br>• Scrum |
| Mishra and Mishra (2010:223) | • XP |
| Iivari and Iivari (2011:517) | • Extreme Programming<br>• Scrum |
| Ralph (2011:6) | • Extreme Programming<br>• Scrum<br>• Unified Software Process (USP) |
| Vavpotic and Vasilecas (2011:109) | • Crystal Methodologies<br>• Rational Unified Process (RUP)<br>• Scrum<br>• XP<br>• Test Driven Development (TDD)<br>• Oracle's Custom Development Method (CDM) |
| Dyck and Majchrzak (2012:5304-5305) | • RUP<br>• XP<br>• Scrum |
| Singh, Kumar and Bansai (2015:139-140) | • XP<br>• Scrum<br>• FDD |

When looking at the specific SDMs mentioned in Table 2-4, it must be noted that they all differ in some way and have their own specific uses. According to Mathiassen and Sandberg (2014:62), there exist agile models that can be tailored to a project or a user's specific need. This is one of the reasons why agile SDMs are used for MAD, as mentioned at the beginning of Section 2.2. A selection of the agile SDMs mentioned in Table 2-4 will be discussed in more detail in the next section.

### 2.2.2 Agile development methodologies

In this section, the XP, FDD and Scrum SDMs will be discussed. These SDMs will be discussed based on their processes, roles and responsibilities, practices and their scope of use. Work from Chowdhury and Huda (2011:363) claims that these three SDMs are some of the most well-known agile SDMs. More recent literature has also been added to show these SDMs are still relevant.

#### 2.2.2.1 eXtreme Programming (XP)

According to Putra, Yuliawati and Mursanto (2012:137), XP is a widely accepted SDM that is used to improve the quality of software that is being developed. Carvalho and Azevedo (2013:254) agree by stating that XP is widely adopted and offers practices to a variety of business contexts.

##### *2.2.2.1.1 Process*

The XP SDM process consists of five different phases. These phases are explained below (Abrahamsson *et al.,* 2002:19-20):

- **Exploration:** In this phase, the customer(s) write what they wish to be included in the first release on story cards. Each of these cards should describe one feature that they want added to the release. The development team uses the time in this phase to familiarise themselves with the tools and technologies that will be used in the project.
- **Planning:** The story cards are ordered according to importance while an agreement is made about the first release. The development team use this phase to estimate the effort that will be required to complete the first release. A schedule is agreed upon and work is started.
- **Iterations to release:** The schedule is broken down into a number of iterations before the first release. The customer decides which story cards are selected for each iteration. These iterations are performed and implemented. The functionality of the iteration is tested at the end of each iteration.
- **Productionising:** In this phase, additional testing and performance checking is done before the final release to the customer. This phase includes finding any additional changes

that can be made and deciding whether the change will be added to the current release or kept for an update later in the life cycle. If the decision is made to add the change, the iterations of development are shortened and quickened.

- **Maintenance:** During the maintenance phase the system is maintained while it is implemented. This is done with the customer to collaborate on the functionality of the system.

#### *2.2.2.1.2 Roles and responsibilities*

In the XP SDM there are different roles played by different stakeholders:

- **Programmer:** Programmers write and test the system code to ensure that it works. Communication with programmers is key (Beck, 2005:81).
- **Customer:** The customer writes the story cards and determines satisfaction in terms of the requirements. The customer also writes the functional test to see if the story cards are satisfied (Abrahamsson *et al.*, 2002:21).
- **Tester:** Testers help the customers to run the functionality test, and assist programmers with their tests (Beck, 2005:74).
- **Tracker:** Trackers ensure that the schedule is up to date and that time spent on every element of the XP SDM is spent well (Abrahamsson *et al.*, 2002:21).
- **Coach:** The coach is responsible for understanding the XP SDM holistically and guiding the rest of the stakeholders (Abrahamsson *et al.*, 2002:22).
- **Manager:** Different managers of the project make big decisions that lead the project to a success in the end. The managers communicate different responsibilities to different stakeholders and makes sure that the jobs are done correctly (Beck, 2005:76-77).

#### *2.2.2.1.3 Practices*

The practices used in the XP SDM are all taken from different existing methodologies (Beck, 2005:35). The following are some of the practices:

- **Planning game:** Close communication between the customer and the development team should be ensured. The development team estimates the time that it will take to implement the customer story cards and the customer then decides about the scope and the time of releases (Abrahamsson *et al.*, 2002:23; Carvalho & Azevedo, 2013:256).
- **Small releases:** The releases are implemented and tested at least once every three months, but earlier if possible (Carvalho & Azevedo, 2013:256).

- **Simple design:** The focus of the development team is to design the simplest possible system. Complexity in code and development can waste unnecessary time (Abrahamsson *et al.*, 2002:24; Carvalho & Azevedo, 2013:256).
- **Testing:** The development of software and systems is tested. Unit tests have to be implemented before any code is written and run continuously (Carvalho & Azevedo, 2013:256).
- **Pair programming:** Two developers are located at the same computer, taking turns to code. The one developer will actively code, whilst the other will give ideas and help out (Beck, 2005:42; Carvalho & Azevedo, 2013:256).
- **Collective ownership:** Any of the developers can change any part of the code at any time (Abrahamsson *et al.*, 2002:24; Carvalho & Azevedo, 2013:256).
- **Continuous integration:** As soon as a piece of code is ready it is implemented into the system. This allows the system to be built and integrated more than once a day and always be as up to date as possible (Beck, 2005:49; Carvalho & Azevedo, 2013:256).
- **40-hour week:** The maximum of a working week should be 40 hours. If team members are required to work two continuous weeks of overtime, management has to take action to resolve the problem (Abrahamsson *et al.*, 2002:24; Carvalho & Azevedo, 2013:256).
- **On-site customer:** A customer from the client company has to be present with the team at all times (Carvalho & Azevedo, 2013:256).
- **Coding standards:** Programmers have to follow set rules and standards to ensure that different pieces of code do not differ too much and that integration are made easier (Beck, 2005:67; Carvalho & Azevedo, 2013:256).
- **Just rules:** The rules followed by the team can be changed at any time as long as all parties affected agree on the changes (Abrahamsson *et al.*, 2002:25; Carvalho & Azevedo, 2013:256).
- **Open workspace:** A large space with cubicles is preferred and the pair-programmers are located in the centre of the space (Abrahamsson *et al.*, 2002:25; Carvalho & Azevedo, 2013:256).

#### *2.2.2.1.4 Scope of use*

Beck (2005:144) states that the XP SDM as a methodology, is not suitable everywhere and should be used at the appropriate times and for appropriate projects. According to Abrahamsson *et al.* (2002:26), the XP SDM is aimed at small to medium sized teams where the communication between the stakeholders is enabled at all times. Programmers should not be scattered and should be in close proximity to one another. Abrahamsson *et al.* (2002:26) further mention that there should be no resistance to using XP as development methodology – all parties involved should agree on the methodology otherwise it might end in failure.

#### 2.2.2.2 Scrum

In this section the processes, roles and responsibilities, practices and scope in which the Scrum SDM can be used, will be discussed as mentioned in the work of Abrahamsson *et al.* (2002) and (Schwaber, 2004). Other literature will also be mentioned throughout the section.

##### *2.2.2.2.1 Process*

The development life cycle in the Scrum SDM consists of three main phases namely: the pre-game phase (which consists of two sub-phases), the development phase and the post-game phase. The phases are detailed next (Abrahamsson *et al.,* 2002):

**Pre-game phase -** this phase consists of two sub-phases (Abrahamsson *et al.*, 2002:29):

- **Planning:** This sub-phase includes the definition of the system to be developed. A product backlog list containing all the requirements that are currently known by the team is used. These requirements are acquired from the customer, marketing department and the software developers. A choice is made regarding the prioritisation of the requirements, after which the effort of implementing each requirement is estimated. The backlog list is updated constantly as details become more apparent and newer items for implementation come to light. The planning sub-phase also includes an explanation of the team, as well as the tools and technologies (resources) that the team will use.
- **Architecture design:** The product backlog is used in this sub-phase to create a high-level design of the system based on the requirements. If the project is updating or changing an existing system, the enhancements with any problems they might cause are listed in the backlog. A meeting is held where the design is reviewed by the team and the customer.

**Development phase (game phase) -** This phase can be seen as the agile part of the methodology where stakeholders should expect the unexpected and be ready for anything. The different environmental variables and their associated changes are identified in this phase. This is done by different practices during each sprint (explained below) in the development phase. The Scrum SDM aims to control all variables throughout development instead of only at the beginning of the cycle.

Sprints are iterative cycles where the system and its functions are developed and enhanced to create new increments of the system. Each of these sprints includes the following phases (Abrahamsson *et al.*, 2002:30):

- Requirements;
- Analysis;
- Design;
- Evolution; and
- Delivery.

**Post-game phase -** This is the final phase of development and contains the finalisation of the release. This phase starts when an agreement has been made that all requirements have been met by the system. Steps included in this phase are integration, testing and documentation (Abrahamsson *et al.*, 2002:30).

*2.2.2.2.2 Roles and responsibility*

In the Scrum SDM there are three main roles presented by Schwaber (2004:9) and two supplementary roles added by Abrahamsson *et al.* (2002:31). Following is the explanation of the five different roles.

- **Scrum master:** This is a completely new role introduced by the Scrum SDM; the responsibility of this role is to ensure that the project is completed according to the practices, values and rules of the Scrum SDM. The Scrum master communicates with the customer, management and the Scrum team to ensure that the project progresses as planned (Schwaber, 2004:9; Guang-yong, 2011:218).
- **Product owner:** The product owner is responsible for managing and controlling the project and to ensure that the product backlog list is made visible (Guang-yong, 2011:218). The product owner is selected by the Scrum master, management and the customer (Schwaber, 2004:9).
- **Scrum team:** The Scrum team is responsible for developing the functionality of the system as well as managing and organising themselves accordingly (Schwaber, 2004:9) as well as other tasks assigned by the Scrum master (Guang-yong, 2011:218). The Scrum uses the requirements in the product backlog list to create a functioning system. They are involved in effort estimation, reviewing the product backlog list and suggesting redundancies that they believe need to be removed (Abrahamsson et al., 2002:31).
- **Customer:** The customer is involved in giving requirements for the product backlog list for new items or enhancements on existing items.

- **Management:** Management is responsible for the final decisions. They also participate in the setting of requirements and any extra goals for the project.

#### *2.2.2.2.3 Practices*

This section consists of the practices used in the Scrum methodology as mentioned by Abrahamsson *et al.* (2002:30-34) as well as Schwaber and Beedle (2002):

- **Product backlog:** This practice uses the current knowledge of the project and defines everything that is needed in the final product. It is constantly updated and lists all the prioritised business and technical requirements. This practice includes the creation of the product backlog list, as well as maintenance and control thereof (Abrahamsson *et al.,* 2002:32; Schwaber & Beedle, 2002:35).
- **Effort estimation:** This practice is an iterative process of estimating the time and cost of each item on the product backlog list (Schwaber & Beedle, 2002:35).
- **Sprint:** Sprint is the process of adapting to the constantly changing variables in the project. Each sprint lasts about 30 calendar days and has the goal to produce a new executable increment of the system. The Scrum team uses the following practices (Abrahamsson et al., 2002:32):
    - **Sprint planning meeting:** This meeting is done in two phases. The first phase is attended by all the stakeholders to discuss what has to happen in the next sprint. The second phase is only attended by the Scrum master and the Scrum team to discuss how the completed product increment will be implemented (Schwaber & Beedle, 2002:47).
    - **Sprint backlog:** This is the starting point of each individual sprint and is a list of product backlog items selected, for the next sprint, to be implemented. The sprint backlog is stable until the end of the sprint, unlike the product backlog that is dynamic and can change at any time (Abrahamsson *et al.,* 2002:33).
    - **Daily scrum meeting:** These are casual meetings of approximately 15 minutes to keep track of the progress of the Scrum team. During these meetings progress updates are made since the previous meeting, and planning is done for the immediate future (Abrahamsson *et al.,* 2002:33-34).
    - **Sprint review meeting:** On the final day of each sprint the results are presented by the Scrum master and the Scrum team to the rest of the stakeholders. The participants of the meeting assess the progress and requirements, and decide on the work that has to be done in the next Sprint (Schwaber & Beedle, 2002:54).

##### *2.2.2.2.4 Scope of use*

The Scrum SDM is normally suitable for small projects or any project that have small teams. These teams consist of less than 10 engineers (Abrahamsson *et al.*, 2002:36).

#### 2.2.2.3 Feature driven development

In this section the processes, roles and responsibilities, practices and the scope in which FDD can be used will be discussed.

##### *2.2.2.3.1 Process*

The process cycle of the FDD methodology consists of five sequential processes which will be discussed below. All roles mentioned in these processes will be elaborated on in Section 2.2.2.3.2.

- **Develop an overall model:** The domain experts should already know what the scope, the context and the requirements of the system are at the start of the process, based on available documentation. The gathering of the requirements is not included in the FDD cycle and is normally done separately (Abrahamsson *et al.,* 2002:48).
  The domain members use the scope and context of the system to do an initial walkthrough of what should happen. Thereafter, a more detailed walkthrough is done of each domain area. These walkthroughs are used to create object models of each area. Different small groups of developers make their own individual models which are then presented to different stakeholders at a review meeting. The model selected during the review meeting can either be the best model or a composition of different presented models. This model is merged into the overall model before the phase is completed (Goyal, 2007:10).
- **Build a feature list:** The feature list contains all the features for the system being built. This list is created from the walkthroughs, object models and existing requirement documentation (Abrahamsson *et al.,* 2002:48). According to Goyal (2007:11), the team breaks down the whole domain into different areas, referred to as major feature sets. Each of these areas is broken down into activities, referred to as feature sets. In each of these sets, a step contained within an activity is referred to as a feature.
- **Plan by feature:** This process includes the design and creation of a high-level plan. This plan contains feature sets that are ordered by priority and dependencies (this order is decided by the Project Manager, Development Manager and the Chief Programmers (Goyal, 2007:12)) and assigned to the chief programmers. The different classes that were identified in the first process of the cycle are assigned to different individual programmers (Abrahamsson *et al.,* 2002:49).

- **Design and build by feature:** A few features are selected from the feature sets. Feature teams are formed to develop the selected features. After the teams are put together, the design and build by feature phases can start. The length of these iterations varies between a few days and up to two weeks (Abrahamsson *et al.,* 2002:49). To shorten the development time, the feature teams can work on different feature sets at the same time and do not have to do the same work. When one of the teams is done with an iteration, the features are built into the main version of the system and that specific group can start with a new iteration of features until all the features are completed (Goyal, 2007:13). These final two processes are done in iterations and can be regarded as a single phase broken up into two sub-phases.

#### *2.2.2.3.2 Roles and responsibility*

Goyal (2007:5) states that the roles of the FDD SDM can be separated into two different groups: supporting roles and additional roles. This section will elaborate on these two groups of roles.

**Supporting roles**

- **Domain expert:** This role can be fulfilled by any mixture of users, clients, sponsors and business analysts. The role's responsibility is to understand how the different requirements should perform and be implemented into the system. The domain expert transfers this knowledge to the developers to ensure that the system functions as intended (Abrahamsson *et al.,* 2002:52).
- **Domain manager:** This role leads the different domain experts and ensures that any conflict of opinion is resolved. The domain experts have to agree on the requirements and the functions they serve (Abrahamsson *et al.,* 2002:52).
- **Release manager:** The role is responsible for reporting any progress from the chief programmers to the project manager (Goyal, 2007:6).
- **Language guru:** This is a very important role in projects that use a new language for the first time. This person or persons are responsible for having detailed knowledge of a specific language before the project starts in order to assist any developer (Goyal, 2007:6).
- **Build engineer:** This role is responsible to set up, maintain and run the build process at regular intervals (Abrahamsson *et al.,* 2002:52; Goyal, 2007:6).
- **Tool smith:** This role is responsible for the creation of small applications (tools) to help developers with a specific problem or to make their work easier (Abrahamsson *et al.,* 2002:52; Goyal, 2007:6).

- **System admin:** This role is responsible for configuring, managing and also troubleshooting any network and computer in the network that the team works with (Abrahamsson *et al.,* 2002:52; Goyal, 2007:6).

**Additional roles**

- **Tester:** This role is responsible for testing any piece of software that has been developed and confirming that it works correctly. A tester is also responsible for setting up the tests used (Abrahamsson *et al.,* 2002:53).
- **Deployer:** This role is directly involved with the implementation of the new system, as well as the conversion of any existing data to the format the new system uses (Abrahamsson *et al.,* 2002:52; Goyal, 2007:6).
- **Technical writer:** This role prepares and writes user documentation used in and for the project (Goyal, 2007:6).

#### *2.2.2.3.3 Practices*

Abrahamsson *et al.* (2002:53) claim that the FDD SDM consists of a set called “best practices”. Although these practices are nothing new to any existing methodology, the combination of these practices together with the process cycle of the FDD SDM makes for a great and very unique SDM. Following is this set of “best practices”:

- **Domain object modelling:** This explains the domain of a certain problem and results the framework used to add features to the domain.
- **Develop by feature:** Developing through the use of a list of client-valued functions as well as tracking of the progress (Abrahamsson *et al.,* 2002:53). If a single function is regarded as too complex to be created in the time given, it is decomposed into smaller functions (Goyal, 2007:8).
- **Class ownership:** Each class in the system has one representative that takes responsibility of that class (Abrahamsson *et al.,* 2002:53). This person is responsible for any content of the class, as well as ensuring that its integrity is not compromised (Goyal, 2007:8).
- **Feature team:** Small teams are formed to complete a specific job.
- **Inspections:** Inspections are conducted by the user of the well-known defect-detection system to ensure that one or several of the small feature teams do not have any problems (Abrahamsson *et al.,* 2002:53; Goyal, 2007:9).
- **Regular build schedule:** This practice ensures that there is always a functioning system available for demonstrative purposes (Abrahamsson *et al.,* 2002:54).

- **Configuration management:** This practice is used to identify the latest versions of any completed source code file and to provide a means to track those files (Abrahamsson *et al.,* 2002:54).
- **Progress reporting:** Completed work is reported to the essential stakeholders on the project (Goyal, 2007:10).

#### *2.2.2.3.4 Scope of use*

Palmer and Felsing (2002:xxiii), the authors of a book on FDD, state that the methodology should be taken into serious consideration for any software organisation and project. It delivers quality as well as business-critical software and it does so in very good time. Abrahamsson *et al.* (2002:54) further state that it is a very good choice for any type of software development, although it is hard to find any reliable experience reports.

As presented in Table 2-4, there are many other SDMs used in agile development. The XP, FDD and Scrum SDMs were discussed to provide an overview of how agile SDMs work and how even some of the same type of SDMs can differ. It should be mentioned that there is little to no mention of security in any of these agile SDMs, as this will be the focus of the study. The next section will discuss MAD.

## 2.3 Mobile application development (MAD)

According to Kim, Oh and Moon (2013:300), the development of smartphones is starting to make a very large impact on companies related to applications and the development thereof (Holzer & Ondrus, 2009:55). Palmieri, Singh and Cicchetti (2012:179) claim that mobile phones are a necessity in present times, rather than a luxury. Other than making calls, there are many types of applications that can be used to make everyday life easier. Rishi (2012:120) agrees with this by saying that mobile phones today are used as a computer, music player, navigation system, a search engine and much more. These applications or features are gaining popularity rapidly and can be built-in or downloaded separately. Some examples of applications and features include (Palmieri *et al.*, 2012:179):

- Camera applications;
- Music player applications;
- GPS (Global Positioning System) applications;
- Accelerometer applications; and
- Games.

Huy and Van Thanh (2012:907-908) divide the type of applications into two categories: client-side applications and server-side applications. It should be noted that different people categorise applications in different ways and categories.

- Client-side applications **-** These are applications that run stand-alone on a mobile device without any network connection needed. Native applications, platform-based applications and mobile widgets are the three main types of client-side applications (Huy & Van Thanh, 2012:907).
    - **Native:** These applications are normally developed in C and C++ to allow developers to develop as close as possible to the hardware. The applications are compiled to machine code and the platform executes them directly.
    - **Platform-based:** These applications are designed and developed for a specific platform and device (Windows, iOS, Android, etc.) and cannot run on any other device without alterations (Huy & Van Thanh, 2012:908.
    - **Mobile Widgets:** These applications are small task-specific applications that control web content for different reasons.
- Server-side applications - These are applications that need a connection to a web server. They are normally heavily dependent on this server. The two types of server-side applications are Mobile Web applications and HTML5 applications (Huy & Van Thanh, 2012:908).
    - **Mobile Web:** These applications enable functions for processing of information to be initiated remotely from the web server. They consist of three layers: client layer, application layer and the database layer.
    - **HTML5:** These applications are introduced to create a standard which consists of a set of features that can manage all the functions that current technologies are managing. Instead of having different applications, HTML5 now manages all of them by itself (Huy & Van Thanh, 2012:908).

In an article written by Joorabchi, Mesbah and Kruchten (2013:15), applications are divided into three different categories: native, web-based and hybrid. This is very similar to the categorisation of Huy and Thanh (2012:907-908), with the addition of hybrid applications. Joorabchi *et al.* (2013:15) briefly explains these categories:

- **Native applications:** These applications have to be adapted for different devices and normally run on the device’s OS.
- **Web-based applications:** These applications run inside the browser of a mobile phone and thus require the phone to have a browser application installed.

- **Hybrid applications:** These applications are web-based applications that are wrapped in native development. In other words, it is a web-based application that contains different features of native applications.

Farago (2012) and Rishi (2012:122), claim that there are many categories in which applications can be divided. They use the top five categories to talk about the growth over the period from 2011 to 2012. The five top categories as well as their growth are as follows:

- Photo and video: 89%;
- Music: 72%;
- Productivity: 66 %;
- Social networking: 54%; and
- Entertainment: 40%.

Furthermore, Rishi (2012:120) states that applications can be consumer-based as well as enterprise-based. The differences are discussed next:

**Consumer mobile applications:** In an article on mobile applications, Hundermark (2015) states that consumer applications are just everyday applications used by everyday people. Kuusinen and Mikkonen (2013:535) agree in saying that consumer applications are basically facilities that are provided to users on their mobile phones.

**Enterprise mobile applications (EMAs):** Being the main focus of this study, EMAs will be discussed in more detail than consumer mobile applications. According to Giessmann *et al.* (2012:1364), these types of applications are much like computer-based enterprise software, but run on mobile phone devices rather than on a computer. McAfee (2006:145) declares that organisations adopt these types of applications to help improve the interactions between different employees and business partners. Giessmann *et al.* (2012:1364) define EMAs as *"applications that are designed for and are operated on mobile devices and which facilitate business users within core and/or support processes of their enterprise".*

Al Bar, Mohamed, Akhtar and Abuhashish (2011:60) claim that when EMAs are used correctly, the capabilities are almost endless: solutions needed to achieve prosperity for the business are provided nonstop, decision making is enhanced, and decisions can be made on real-time data at any time and any place. This enhanced decision making also increases general productivity, which leads to business processes moving along at a quicker pace.

Radia, Zhang, Tatpamula & Madisetti, (2012:843) add to this by stating that a successful EMA should be measured by the following results and noted to what extent they have been achieved:

- **Business transformation:** Business transformation is reached by automating processes and achieving a point where human interaction with the system is as low as possible.
- **Efficiency:** Process automation results in cost reductions and gains in productivity, indicating that efficiency has been achieved or has been improved.
- **Effectiveness:** This result can only be given definition by stating goals and objectives, and how close they are to be reached. Effectiveness depends on how the enterprise perceives success in goals and objectives.

Al Bar *et al.* (2011:62) emphasise the following business benefits of using EMAs:

- Business processes are improved;
- Productivity is improved;
- Employees, students and management are continuously improved;
- Reporting and system visibility is improved;
- Real-time access for support technicians is provided (this helps with time-sensitive problems);
- Field support is improved;
- Operational costs are reduced;
- Service status is more visible to customers;
- Return on investment expense is reduced; and
- Real-time, two-way access to customers, inventory, and most other work-related information is provided.

Unhelkar and Murugesan (2010:33) agree by stating that one of the prime benefits of using EMAs is that most business processes can be done in real-time wherever employees might be, which enhances user and employee satisfaction. Employee locations, time of use and other real-time data are recorded to support productivity.

Along with this benefit mentioned, Unhelkar and Murugesan (2010:34) also offer the classification of different EMAs. Figure 2-3 presents a summary of these classifications. Examining the pyramid from top to bottom, the more complex type of applications are mentioned first and thus the lower type applications possess an increased mobility.

**M-Collaboration** supports internal collaboration as well as collaboration with other enterprises of concern. Collaboration of mostly all stakeholders makes these EMAs complex.

**M-Operation** has no interaction with customers, as it provides support to operational aspects of the enterprise. Examples include inventory management, human resource management, supplier management, and information of each aspect (Unhelkar & Murugesan, 2010:35).

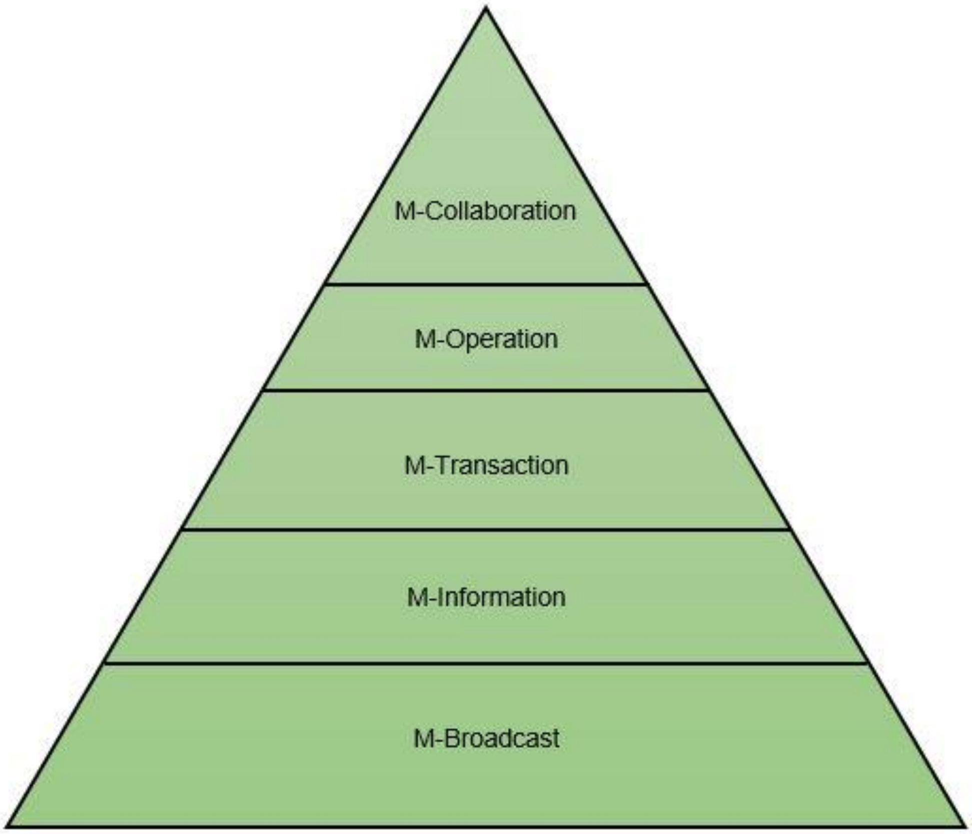


**Figure 2-3: Categorisation of EMAs**

**M-Transaction** is used for all types of transactions. Users can place orders for, buy and sell goods by using these EMAs, depending on the enterprise.

**M-Information** is used to retrieve requested information to a single user.

**M-Broadcast** is used to broadcast information to groups of users. This is the least complex type of EMA (Unhelkar & Murugesan, 2010:34).

It is important to mention that Al Bar *et al.* (2011:61) state that security and the fear of having an insecure infrastructure, is one of the highest ranking reasons for enterprises not deploying mobile applications. It can be seen from the literature that applications can be categorised in many different ways. It all depends on how the user views the application and the different qualities contained in the application. The rest of the section will elaborate on the development of applications, the agile qualities of MAD and different challenges discussed in literature with the focus on security challenges.

### 2.3.1 Development of mobile applications

The development of mobile applications and services was controlled and managed, in the most part, by the mobile network operators and phone manufacturers. This has changed recently when smartphones started flooding the market (Holzer & Ondrus, 2009:55). The growth in mobile application development started when the iPhone AppStore was opened in 2008 (Wasserman, 2010:397). Recently the sales of smartphones have almost doubled from 149 million in 2010 to 297 million in 2011. This increase in sales has a close relationship to the large variety of applications that users have access to (Kim et al., 2013:300).

An application can be any piece of software on a smartphone that holds a purpose to a certain individual (Huy & Thanh, 2012:907). Most small needs of individuals can be satisfied by designing and developing an application for it. These applications have to perform in a satisfactorily manner, have efficient performance and quick response time with limited resources and high availability. Applications have to be developed in the least amount of time and be made available at the lowest cost possible, if it has a chance of succeeding in a market of millions (Corral *et al*., 2013:19). Furthermore, Abrahamsson (2005:20) asserts that developers have to come up with ways to get applications implemented efficiently, with the limited specifications mobile devices have. Mobile phone users, in many cases, expect their phones to have the same processing power as their desktop computers, and the mobile applications to have the same specifications as computer software (Flora & Chande, 2013:9). This makes the job of developers abundantly more difficult.

### 2.3.2 MADMs

This section will cover different MADMs that may be used for MAD. The methodologies selected are Mobile-D, MASAM and SLeSS. These methodologies have been selected based on their regular and independent mention in literature by Flora and Chande (2013), Abrahamsson *et al.* (2004), Nosseir, Flood, Harrison and Ibrahim (2013), Khalid, Zahra and Khan (2014) and Spataru (2010). These methodologies are all combinations of different existing agile SDMs.

#### 2.3.2.1 Mobile-D

Mobile-D was one of the first MADMs that realised the need for agility in MAD. This was as a result of the quick growing mobile technology and the growth in the use of mobile devices and applications in the corporate world (Abrahamsson *et al.,* 2004:174). Techniques from three different agile SDMs are used in Mobile-D: XP, Crystal Methodologies and RUP (Flora & Chande, 2013:11).

Abrahamsson *et al.* (2004:174) claim that Mobile-D was developed with the intention to address some of the challenges experienced in MAD. Examples of these challenges include:

- The limited capabilities and the rapid evolution of any terminal device;
- The various protocols, standards and network technologies used in different devices and device manufacturers;

- The need to operate on many different platforms without making much changes;
- Different needs that device users have; and
- A very strict time to market requirement.

According to Spataru (2010:4), Mobile-D aims to deliver a completed product in about ten weeks and has a team comprising a maximum of ten developers. Abrahamsson *et al.* (2004:175) further states that throughout the development lifecycle of Mobile-D, the following important elements are what make up the different practices of the SDM:

1. Phasing and placing;
2. Architecture line;
3. Mobile test-driven development;
4. Continuous integration;
5. Pair programming;
6. Metrics;
7. Agile software process improvement;
8. Off-site customer; and
9. User-centered focus.

If these principles are compared to some of the characteristics of the SDMs explained in Section 2.2.2, it can already be seen that Mobile-D resembles traditional agile SDMs.

VTT Electronics (2006) and Spataru (2010:22), give a summary of the Mobile-D lifecycle. This summary is given below:

**Phase 1 - Explore**

The purpose of this phase is to plan and establish the start of the project. There is no real timeframe attached to this phase and it may overlap with part of the second phase. It is a very important phase to establish a base for the project to build from (VTT Electronics, 2006). The following stages are contained in this phase:

- Stakeholder establishment;
- Scope definition; and
- Project establishment.

**Phase 2 - Initialise (0 iteration)**

The purpose of this phase is to make sure that the development team is ready for any issue that might arise, to ensure project success (VTT Electronics, 2006). The following stages are contained in this phase:

- Project setup;
- Planning day in 0 iteration; and
- Release day in 0 day iteration.

**Phase 3 - Productionise**

This phase uses an iterative and incremental development cycle to implement the functionality into the product from any requirement listed by the user; it can be referred to as the development phase (VTT Electronics, 2006). The following stages are contained in this phase:

- Planning day;
- Working day; and
- Release day.

**Phase 4 - Stabilise**

In some larger systems, the project is broken down into smaller, more manageable projects or subsystems. In this phase the subsystems are integrated and the finished product is verified as fit for purpose (VTT Electronics, 2006). The following stages are contained in this phase:

- Planning day;
- Working day;
- Documentation wrap-up; and
- Release day.

**Phase 5 - System Test and Fix**

This phase ensures that the system works correctly and that it includes all the functionality required by the user (VTT Electronics, 2006). Requirements could be used here to make sure all functionality was implemented. The following stages are contained in this phase:

- System test;
- Planning day;
- Working day; and
- Release day.

#### Phase 6 - Evolve

Spataru (2010:22) adds a final phase which deals with the continuous feedback from users. The following stages are contained in this phase:

- Data analysis;
- System testing;
- Planning day;
- Working day; and
- Release day.

Nosseir *et al.* (2012:282) state that the Mobile-D MADM is worth mentioning and suggests it for development use.

### 2.3.2.2 MASAM (Mobile Application System Agile Methodology)

Flora and Chande (2013:14) claim that MASAM is a combination of the XP, RUP and SPEM (Software and Systems Process Engineering Meta-model) SDMs. It was designed for creating mobile software in a quick and successful manner, using an agile approach. It is Graphical User Interface (GUI)-based and uses the different agile approaches for rapid development, while utilising domain knowledge (Jeong *et al.,* 2008:363).

#### *2.3.2.2.1 Process assets*

According to Flora and Chande (2013:14), MASAM contains three process assets which are described according to SPEM. These process assets with examples are given below:

**Role**: This asset defines different skills and responsibilities of one or more individuals. Different roles used in MASAM are: Planner, Manager, User Interface (UI) designer, Developer, Development team, Initial development team, Tester and User.

**Task**: This asset can be seen as a unit of work that can be assigned to a specific role. These tasks can have a timespan varying between a few hours and a few days. Different tasks that can be assigned in MASAM are: Product summary, Initial planning, User definition, Initial analysis, Select resources, Select process, Establish environment, Write story card, UI design, Define architecture, Planning, Iteration plan, Face-to-face meeting, Incremental design, TDD, Refactoring, Release plan, Feedback, Pattern manage, Pair programming, Integration, Acceptance test and User test figure.

**Work Product:** This asset refers to any input and output of tasks. These work products include: Product summary, Project planner, UI sample, UI model, UI pattern, Architecture pattern, Application pattern, Story card, Task card, Architecture model, Component model and Test case.

*2.3.2.2.2 Process*

The process assets are designed to help out with the standard process of development in MASAM. This process consists of four phases (Jeong *et al.,* 2008:364).

**Development preparation phase**

Conversing with clients and users are required throughout the entire MASAM process, In the Development preparation phase this client conversing is essential to the success of the project (Jeong *et al.,* 2008:364). During this phase the needs and wants of the customer is acquired and the preparation for development is done. Jeong *et al.* (2008:364) believe that the future of the project should be a shared vision between the company and the customer.

The activities and their respective tasks are shown in Table 2-5.

**Table 2-5: Development preparation phase activities (Adapted from Flora & Chande, 2013:15; Jeong *et al.,* 2008:364)**

| **Activity** | **Task** |
|---|---|
| Grasping product | Product summary |
| | Pre-planning |
| Product concept sharing | User definition |
| | Initial product analysis |
| Project setup | Development process coordination |
| | Project resource coordination |
| | Pre-study |

**Embodiment phase**

In this phase the requirements of the customer are represented with some form of prototype. The requirements are incrementally implemented into pieces of code, growing the prototype as the life cycle continues. Domain knowledge is utilised during this phase to make development faster. Domain knowledge can be seen as the recycling of old code, code patterns and even GUI patterns (Jeong *et al.,* 2008:363). Some clients do not know exactly what they want so a pattern similar to what they described can be located and used to set up the prototype for presentation purposes. All domain knowledge is stored and recycled with every project (Jeong *et al*., 2008:365).

The activities and their respective tasks are shown in Table 2-6.

**Table 2-6: Embodiment phase activities (Adapted from Flora & Chande, 2013:15; Jeong *et al.,* 2008:365)**

| **Activity** | **Task** |
|---|---|
| User need understanding | Story card workshop |
| | UI design |
| Architecting | Non-functional requirement analysis |
| | Architecture definition |
| | Pattern management |

### Product developing phase

In previous phases the needs and requirements of the customer is acquired, but in many cases the requirements are not clear. For this reason, these requirements are implemented in code when development starts and small releases are given to the customer to ensure that the project is on the right track. Requirements can be added in this phase as well (Jeong *et al.,* 2008:365). Jeong *et al.* (2008:365) claim that the number of these releases is determined by the development team and the releases are coded incrementally in iterations.

The activities and their respective tasks are shown in Table 2-7 (Flora & Chande, 2013:15).

**Table 2-7: Product developing phase activities**

| **Activity** | **Task** |
|---|---|
| Implementation and preparation | Environment setup |
| | Development planning |
| Release cycle | Release planning |
| | Iteration cycle |

| Activity | Task |
|---|---|
| | Release |

**Commercialisation phase**

During this phase the development company confirms that the application satisfies the application policy of any country it might be implemented in. The application undergoes different tests to confirm this and to confirm that the application is simple and easy enough to use. According to Jeong *et al.* (2008:365), ease of use is the most important success factor of MASAM.

The activities and their respective tasks are shown in Table 2-8.

**Table 2-8: Commercialisation phase activities (Adapted from Flora & Chande, 2013:15; Jeong *et al.,* 2008:365)**

| Activity | Task |
|---|---|
| System test | Acceptance test |
| | User test |
| Product selling | Launching test |
| | Product launch |

#### 2.3.2.3 SLeSS (Scrum and Lean Six Sigma)

The SLeSS MADM was proposed by Da Cunha *et al.* (2011:283). They state that it is "an integration approach of Scrum and Lean Six Sigma used in real projects of developing embedded software customisations for mobile phones. This approach enables the achievement of performance and quality targets, progressively improving the development process and the outcome of projects". Flora and Chande (2013:12) agree by claiming that SLeSS is a combination of two SDMs, Scrum and Lean Six Sigma (LSS), and is used for the development of software customisations for mobile phones. Section 2.2.2.2 already introduced much of the Scrum terminology, and as such it will not be repeated in this section.

In the execution of SLeSS, Scrum is firstly executed by itself, followed by the implementation of LSS as a quality framework. For the Scrum part of SLeSS, the following characteristics are presented and recommendations made (Flora & Chande, 2013:16):

- **Size of Sprint:** Between one and two weeks.
- **Size of team:** Between four and nine people.
- **Sprint backlog:** Include process improvements and customised activities; the client and the development team identify any problems or issues in a Sprint which can be resolved by team members.
- **LSS:** The Scrum Master and the Product Owner should have a complete understanding of the techniques, management processes and development processes of LSS.

Once Scrum is settled in the project and the organisation, LSS is applied to complement the established Scrum. LSS is represented by the Define, Measure, Analyse, Improve and Control (DMAIC) phases (Flora & Chande, 2013:16). Table 2-9 shows the phases and which backlog items are addressed in each phase.

**Table 2-9: DMAIC phases in LSS part of SLeSS (Adapted from Da Cunha *et al.,* 2011:290; Flora & Chande, 2013:16)**

| **Phase** | **Backlog Item** |
|---|---|
| Define | Contracts |
| | Performance indicators |
| | Initial analysis |
| | Initial data collection |
| Measure | SIPOC (Suppliers, Inputs, Processes, Outputs and Customers) |
| | Process map |
| | Diagram of cause and effect |
| | Matrix of cause and effect |
| | Impact effort matrix |
| | Initial capability |
| | Measurement and inspection system |
| | Data collection |

| **Phase** | **Backlog Item** |
|---|---|
| Analyse | FTA (Fault Tree Analysis) |
| | FMEA (Failure Mode and Effect Analysis) |
| | Critical inputs |
| Improve | Action plan |
| | SIPOC |
| | Process map |
| | Analysis of results |
| Control | Control plan |
| | Project closure |

The practice of SLeSS strives to obtain higher outcomes such as the following:

- Better and quicker adaption to change in requirements;
- Ensuring that deadlines are met;
- Ensuring that overtime is kept to a minimum and that no time is wasted; and
- Delivering more versions with fewer defects.

#### 2.3.2.4 Critical comparison

Throughout the years MAD has adopted the agile manner of development. The specific SDMs used in MAD might not be the same as other agile SDMs used for traditional software development, but they still contain parts and characteristics from normal agile SDMS.

As depicted in Table 2-10, the MADMs all contain or are compiled from other agile SDMs. This allows the MADM to have characteristics from other SDMs, but to be built for a specific type of application or software.

**Table 2-10: MADMs created from other SDMs (Adapted from Flora & Chande, 2013:11-12)**

| MADM | Year of development | SDMs used to create MADM |
|---|---|---|
| Mobile-D | 2004 | XP, Crystal Methodologies and RUP |
| RaPiD 7 | 2005 | AM |
| Hybrid Methodology Design | 2007 | ASD and NPD |
| MASAM | 2008 | XP, RUP and SPEM |
| SLeSS | 2011 | Scrum and Lean six Sigma |

#### 2.3.2.5 Conclusion to MADMs

MADMs coincide well with agile development, as presented in Table 2-10. When conducting MAD, the projects are normally shorter than other software projects; thus agile development is a good choice. This is affirmed by Flora and Chande (2013:9) who claim that agile SDMs are normally used in cases where deadlines are tight. Even though some traditional agile SDMs such as Scrum and XP are used for MAD, most MAD SDMs were designed and developed for MAD exclusively. Mathiassen and Sandberg (2014:62) state that different agile methods exist which can be tailored to specific project or user needs; this aspect supports MADMs very well since these methodologies are normally combinations of traditional agile SDMs.

Finding the correct SDM for a development is not always easy, especially since the act of developing applications also hold other challenges (Wasserman, 2010:398). In the next section, security as a challenge will be discussed. Similar to the discussion on agile SDMs in Section 2.2.2, it should be noted that there is little to no mention of security in any of these MADMs.

### 2.3.3 Security as a challenge in MAD

As the need for mobile applications grows, so does the complexity and the problems behind the development. Designing, developing, testing and deploying code for a variety of mobile platforms can be difficult with the range of different languages and project structures or frameworks (Castro-Castilla, 2014:4).

Mentioned in the introduction of Section 2.3, mobile applications can be regarded as consumer applications (Kuusinen & Mikkonen, 2013:535) or EMAs. Future reference to mobile applications in this study will include both of these application types, unless specified otherwise.

Wasserman (2010:397) states that most mobile applications are relatively small and are developed by one or two developers. In these situations tracking and documenting of the development process almost always falls away due to the small team size and the time that it takes to develop the application. Jain and Shanbahg (2012:28) state that mobile applications are often developed at a very quick pace, without security being properly addressed. With mobile applications being developed at such a pace, the malicious activities surrounding them are also increasing at a fast pace (Ahmad *et al*., 2013:575). Leavitt (2013:17) claims that one or more security flaws were found in 90 percent of tested mobile applications (the research did not specify whether these applications were consumer applications or EMAs); according to Zhu *et al.* (2014:951), this is leading to users being more and more hesitant when adopting applications.

Along with these security challenges in MAD, the development of EMAs and the integration of mobile applications into the enterprise come with their own set of challenges (Hasan *et al.,* 2014:1). Nyambo, Tarimo and Yonah (2014:29) concur by stating that the security level of EMAs is not increasing along with the increase in their use. Hasan *et al.* (2014:1) assert that higher security may be reached by setting up restrictions and policies, but this could lead to a lower than expected acceptance of the applications. They also state that security should be addressed during development, even if only partially.

SDMs (agile or otherwise) and MADMs show little to no sign of security implementation in their phases (refer to Sections 2.2.2 and 2.3.2.5). Sani *et al.* (2013:37) agree with this by declaring that the security element is unavailable in the different phases of an agile SDM called Dynamic System Development (mentioned in Table 2-4). Azham *et al.* (2011:414) also agrees with the statement by asserting that Scrum (another SDM mentioned in Table 2-4) and other agile methods do not include security practices or implementations. Ghani *et al.* (2013:1071) mention Scrum, XP and FDD as agile SDMs that do not show elements of security in their various phases.

Whitman and Mattord (2011:23) state that security should be considered in all phases of development, whether it is in the SDM guidelines or not. They also mention that if the project is security-based or is in need of more security than normal projects, the phases in the SDM used can be adapted to suit the security needs. Futcher (2011) proposed an integrated risk-based approach to software development by incorporating security into the SDLC. This shows that security may differ depending on the specific project and thus having specific security aspects and implementations in the SDMs guidelines can result in over-doing the security of a project. Therefore it is the developers’ responsibilities to implement and consider security instead of the

SDM itself. Gregory (2003) agrees by adding that it is always good to have a security representative present at approval meetings.

A conclusion can now be reached that security in EMAs is partly accomplished by restrictions and policies set up for application use and by addressing security in part during development (even though different SDMs, including agile SDMs and MADMs, do not guide the developers to security implementations). A balance should be established between enterprise security and mobile security, and thus the study focuses on these two aspects.

## 2.4 Summary of SDM literature

The definitions provided at the beginning of this chapter presented an overview of what a SDM is. The chapter further discussed agile SDMs and MADMs. Agile characteristics were mentioned along with three different agile SDMs to show that even SDMs of the same type can differ in specifics. MADMs were discussed to show how mobile applications could be developed. It was emphasised that MADMs can be regarded as agile SDMs tailored to meet a specific developer need. This is confirmed by Flora and Chande (2013:10) who state that agile is the way to go for MAD, because of the constant changes in user requirements and tight deadlines, and confirmed in Section 2.3.2.5.

In conclusion, it should be clear that throughout all the research into different types of SDMs there was little to no mention of security aspects. Chapter 3 will elaborate more on security, especially on EMA security and the security implemented during development.

# CHAPTER 3 SECURITY

Chapter 3 introduces and discusses relevant topics and concepts regarding information security. These include concepts such as Ubiquitous Computing (UbiComp), Bring Your Own Device (BYOD), malware and other general security threats. Furthermore it presents insight into different security perspectives from literature. Figure 3-1 shows a representation of the chapter.

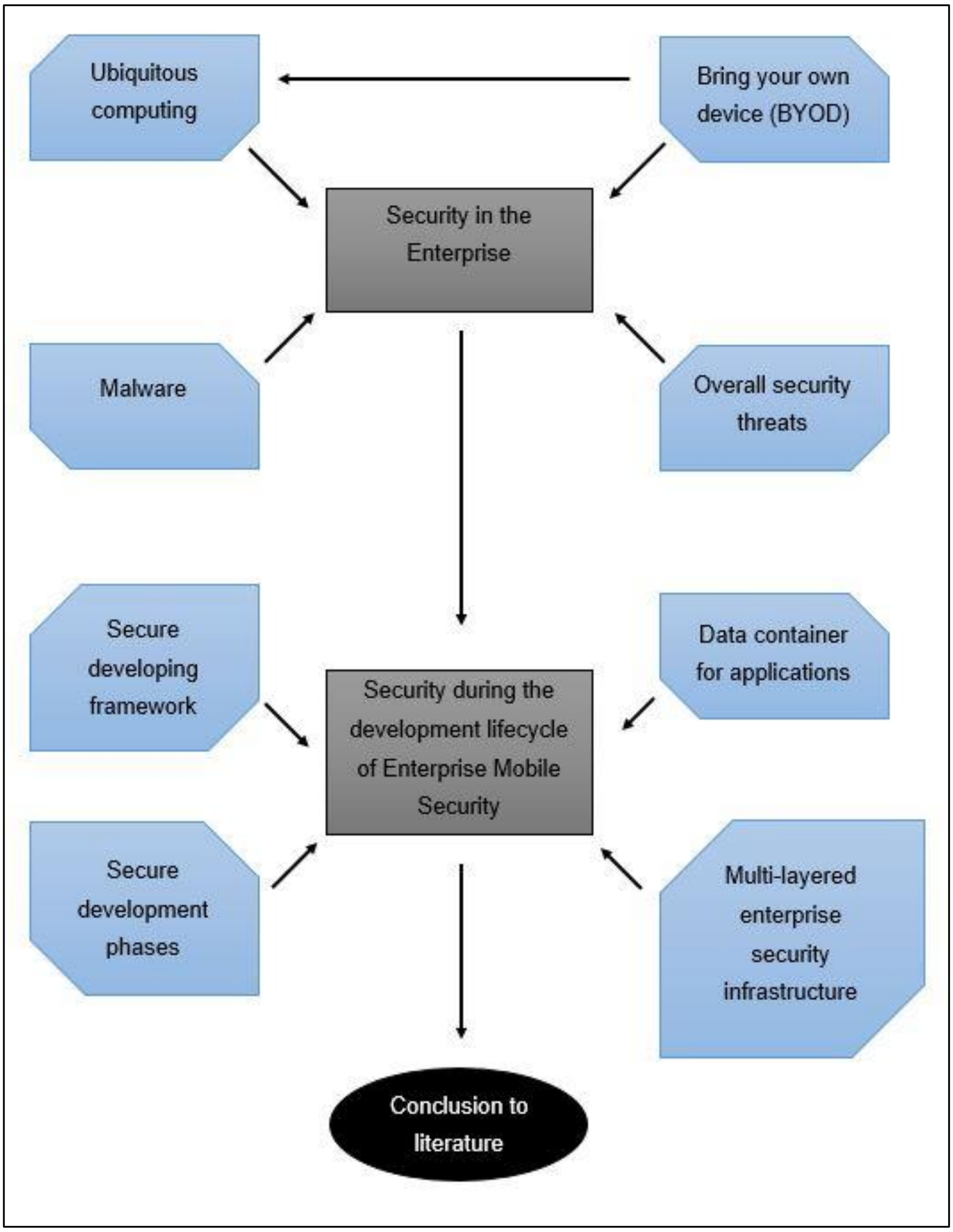

Ubiquitous computing
Bring your own device (BYOD)
Security in the Enterprise
Malware
Overall security threats
Secure developing framework
Data container for applications
Security during the development lifecycle of Enterprise Mobile Security
Secure development phases
Multi-layered enterprise security infrastructure
Conclusion to literature

**Figure 3-1: Chapter 3 representation**

## 3.1 Security in the enterprise

As seen in Section 2.3.3, security in MAD has become more of a challenge than most realise, because of the increase in EMAs and insignificant increase in security. This section will elaborate on different security issues encountered in enterprises and some of the policies and restrictions that are set up to counteract the security threats. Security during MAD will also be discussed by looking at different security frameworks used to develop EMAs as well as consumer applications.

Before discussing enterprise security, UbiComp and malicious software (malware) will be discussed. UbiComp include the sub-topics of Wi-Fi and BYOD. These topics have an impact on enterprise security.

### 3.1.1 UbiComp

Park, Choi, Eom and Chung (2014:1374) define UbiComp as having access to information and communications technology systems, anywhere and at any point in time. A ubiquitous system basically allows users to surround themselves with different computing devices that can support their personal lifestyle.

Tang *et al.* (2013:419) define UbiComp as a user-based computing standard where users can appreciate an ever changing mobile situation. It provides users with different real-time services which can be accessed from any location.

Mark Weiser, the visionary behind UbiComp, said that the vision he has for computing is that the thought of computing should take into account the human world, but leave computers in the background (Weiser, 1991:78). Rogers (2006:404), continues to say that this vision that Mark had, had a focus on the term “calm”. This is explained best by her words in her article on UbiComp: *“His picture of calm technology portrayed a world of serenity, comfort and awareness, where we are kept perpetually informed of what is happening around us, what is going to happen and what has just happened. Information would appear in the center of our attention when needed and effortlessly disappear into the periphery of our attention when not”*.

If we look at technology today and how it is being used, this vision is not far from becoming a reality. As mentioned in section 2.3, smart-devices have become a necessity more than a luxury. Waking up in the morning is soon followed by checking schedules, emails, calendars and more. Smart-devices are currently used for all these activities and more (Park *et al*., 2014:1374), including being an alarm to wake up.

Having all this different information on one device can compromise the user as well as the device. Many high-end applications require an Internet connection to function or to download data. Downloading large files can potentially be very expensive, and thus people may use Wi-Fi.

Free Wi-Fi access points can be accessed at many places; people often connect to any such point to avoid high data costs. This section will elaborate on the threats of connecting to unknown Wi-Fi access points.

#### 3.1.1.1 Using smartphones for UbiComp

The three devices envisioned by Weiser to be used in UbiComp is a tab, pad and board (Park *et al*., 2014:1374).

- **Tab:** A window which is extended from a computer screen and displays what is happening on the screen which it extends from.
- **Pad:** The interface for data and command transmissions that should handle all the devices that make up the UbiComp system.
- **Board:** A large screen-type device for mostly sharing information with a large group of people (an example would be an electronic black board in a classroom).

The three components work together with a base station which houses all the data collected. The data have to get to the base station in some way, thus sensory devices are set up and these collect data and sent it to the base station. The architecture can be imagined as presented in Figure 3-2.

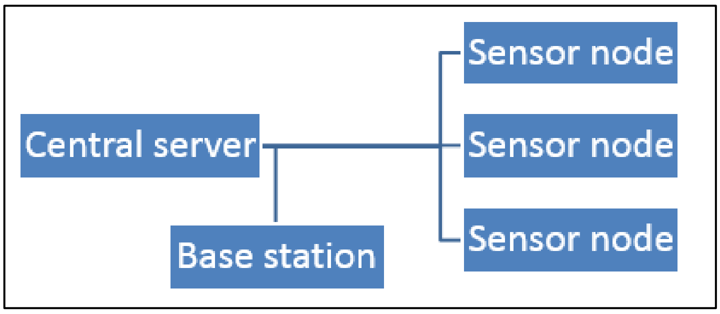


**Figure 3-2: Ubiquitous system architecture**

By means of illustration, a smartphone or tablet will be used as example to explain Figure 3-2 and the different devices used in the envisioned system. A smartphone and tablet already have a screen; this screen can extend information from the Internet or a cloud service. The devices also have different interfaces which can help with recording and transmitting of data. Thus, they already act as the tab and pad mentioned in the vision (Park *et al*., 2014:1376). These devices also have different *sensory nodes* built into them, like cameras, accelerometers, gyroscopes, GPS, and more. These act as the different sensor nodes needed in the setup. The *base station* and *central server* can be the smart-device's internal memory (because of advancements in technology making them so large), or a cloud service (which is discussed in Section 3.1.2.2.4) which is linked to the device (Park *et al*., 2014:1376).

Qiu, Zhang, Ming, Chen, Qin and Yang (2013:518) claim that UbiComp systems have been widely adopted in various different areas of work. This adoption has forced designers of these systems to focus more on security when designing UbiComp systems. According to Park *et al.* (2014:1376), Wi-Fi is one of the largest security problems in UbiComp, since everything is being connected via Wi-Fi networks; Wi-Fi will be discussed next.

#### 3.1.1.2 Wi-Fi in UbiComp

This section will elaborate on compromised access points, why they can be dangerous for users and their information, as well as Man-In-The-Middle (MITM) attacks used to steal information. The basic explanation of a MITM attack is a hacker or malicious person who intercepts data in transit. According to Park *et al.* (2014:1378), this type of attack is very difficult to accomplish on a wired connection (to the Internet), because of Internet Service Provider's (ISP) interference and security. On a wireless system it is a much easier, as depicted in Figure 3-3 (Park *et al.*, 2014:1379).

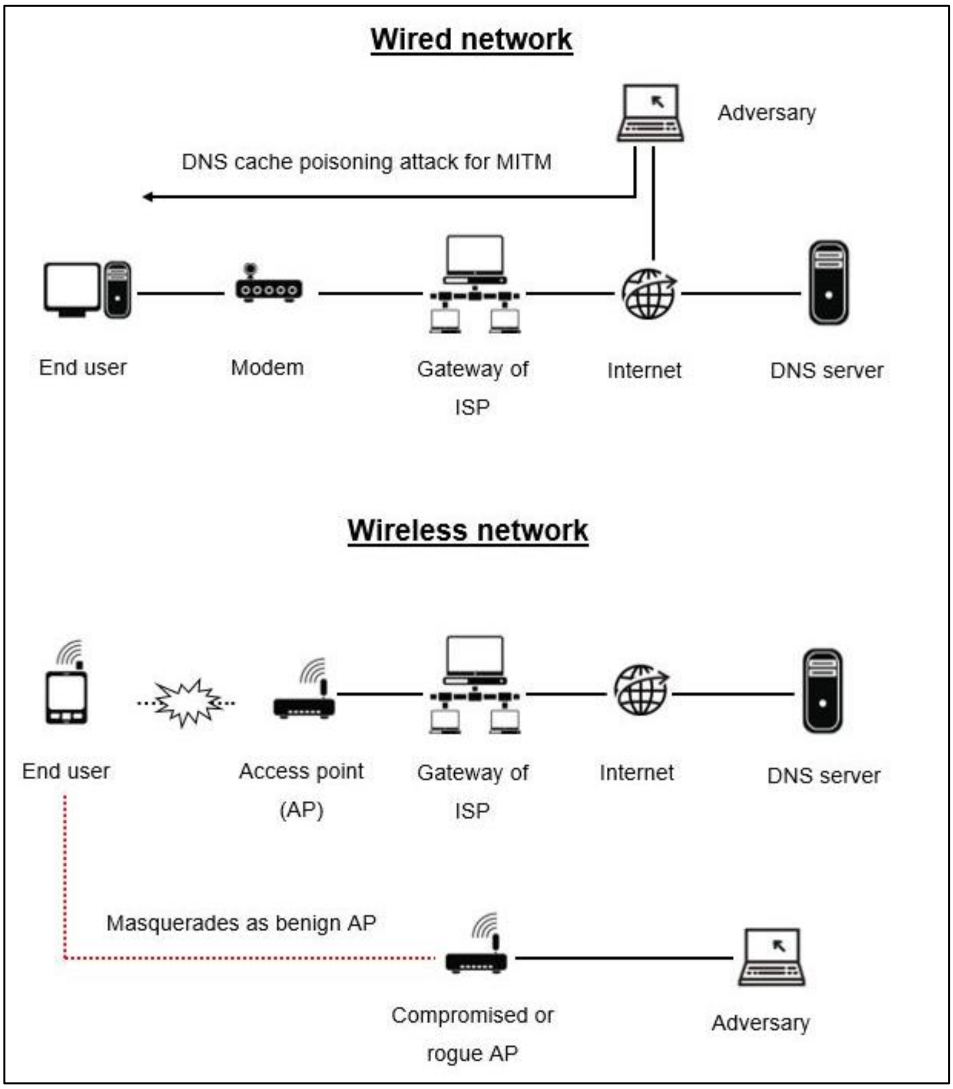


**Figure 3-3: Wired network setup vs. wireless network setup**

It can be seen in Figure 3-3 that the adversary (hacker) has to go through many different channels to get to the end-user. With wireless networks, if the adversary employs a dummy or compromised wireless modem, he gains direct access to the end-user's device. This makes it much easier to intercept any data transmitted by the user to and from the Internet (Park *et* al., 2014:1379).

Consider the following scenario: a user is using an UbiComp setup with a smartphone, a free cloud service and a computer at home (for extra services and links to the smartphone). The user uses a compromised access point to gain access to free Wi-Fi with his smartphone, while doing online banking (or any other service that needs authorisation). The hacker intercepts the data (authorisation details) sent to the banking site and gains access to some parts of the smartphone (depending on permission, which will be discussed in the next section on malicious software). The hacker now has the user's banking details as well as other details saved on the smartphone. These details might include:

- Identification details;
- Other authorisation details;
- Home address; and
- Family information.

All this information could be given to a random stranger just by using an unknown Wi-Fi access point. This might be a threat to any mobile phone user, but could pose more of a threat if the phone, being connected to the unsafe access point, was a work phone. Qiu *et al.* (2013:519) claim that one of the easiest and most effective ways to implement security in UbiComp systems, or even Wi-Fi networks, is to use an encryption system when transferring data. Authentication standards can be added to improve security even more. The next topic to be discussed is BYOD.

### 3.1.2 Bring your own device

BYOD is a trend that has emerged (in 2011 (Leavitt, 2013:16) in many different organisations and enterprises, in which these businesses allow their employees to bring their personal devices (laptops, mobile phones and tablets) to work (Lin *et al.,* 2014:22). Leavitt (2013:17) declares that 75 percent of workers in developing countries use their personal devices at work and 44 percent of workers in developed countries use their personal devices at work. He further states that although BYOD can raise productivity and morale in employees, BYOD can cause security problems for employers.

Armando, Merlo and Verderame (2014:247) assert that this can be a large security risk, since critical infrastructure assets are being controlled and monitored by means of mobile devices.

They further state that although the BYOD paradigm has become increasingly popular, different security concerns continue to arise and general acceptance has not yet been acquired.

#### 3.1.2.1 Potential threats of BYOD

There are different potential threats when engaging in the BYOD paradigm. This section will discuss a few of these threats.

- Employees often use weak or no passwords, leaving these devices that could contain important information about the organisation vulnerable (Leavitt, 2013:17);
- Although it opens the device to more freedom with different applications, the rooting or jailbreaking of devices could compromise the Operating System (OS) level security;
- BYOD environments could be easy to hack, because the employees use the devices on both secure networks and unsecure networks outside of work;
- Vulnerabilities could allow hackers to weaken or damage mobile devices used in the paradigm;
- Passwords, credentials, and any kind of personal information can be compromised;
- Corporate data and important information can be compromised if a personal device controlling that data are compromised;
- Malware could be installed on compromised devices to cause different kinds of problems (Leavitt, 2013:17; Ali, Qureshi & Abbasi, 2015:56);
- Microphones and cameras could be hacked to be used as eavesdropping equipment (Leavitt, 2013:17);
- Employees might lose equipment which could compromise any data on the device if the device was not protected by strong passwords or if the data was not adequately encrypted (Leavitt, 2013:18; Ali *et al.,* 2015:57).
- Restricted applications can be accessed by using privilege escalation attacks in order to gain access to data by interacting with the EMAs via different Inter Process Communication actions (Ali *et al.*, 2015:57).

Although all these risks exist, there are ways to avoid security threats to an extent. Some of these security policies are discussed in the following section.

#### 3.1.2.2 Security practices for BYOD

Although the risks in Section 3.1.2.1 are real, there are ways to get around them. Many of these risks are based on mistakes made by human users, and as such, many of the risks can be avoided by engaging in policies to improve security. These policies are discussed next.

#### *3.1.2.2.1 Base BYOD security requirements*

The requirements given in this section are base security requirements that all enterprises that are using BYOD architecture should have in place (Ali *et al.*, 2015:57).

- The enterprise must ensure that data are always encrypted, both in storage and in during communication of the data. This avoids unauthorised access to the data.
- Access to data must never be granted before the verification of requesting party. Different employees should have different levels of data access and the system should be set up around this hierarchy.
- The architecture must ensure that enterprise data and employees' personal data are kept separate. This guarantees that security implementations are only carried out on enterprise data.
- IT administrators should be able to block devices that do not comply with different implemented policies. These policies include, but are not limited to, password strength and complexity, secure enterprise data removal, monitoring and remote logging of data.
- The architecture must be automated to the point of efficiency and as simple as possible. It should also be as cost effective as it is secure.

These requirements, as mentioned, are only initial security implementations and the following sections will elaborate on other ways security can be improved.

#### *3.1.2.2.2 Mobile device management (MDM)*

MDM refers to the control and management of the devices being used in the organisation by specialists from the IT department (Pogar, Gligora & Davidovi*,* 2013:749; Ali *et al.*, 2015:58). The MDM practice allows companies using the BYOD paradigm to inventory, monitor, manage, secure, and also apply different policies to employee- and corporate-owned mobile devices that are used in the workplace (Leavitt, 2013:18). These policies could include anything such as (Gajar *et al.*, 2013:67):

- Specific encryption policies;
- Disabling certain sensors on the devices; and
- Mandatory complicated/complex passwords.

All approaches that follow MDM work through a management application that is uploaded to all the devices in the BYOD paradigm (Pogar et al., 2013:749). This application receives commands from a central server which is commanded by the IT department or a specific IT specialist (Leavitt, 2013:18). Furthermore, the IT departments are enabled to distribute different applications to the devices, lock different applications and data, wipe data and also enforce certain settings on the devices. (Gajar *et al.*, 2013:67). In some cases, MDM restricts the actions that can be performed by employees on the devices to increase security, and this could reduce productivity (Leavitt, 2013:18).

#### *3.1.2.2.3 Mobile application management (MAM)*

According to Leavitt (2013:18), the focus of MAM is to manage and limit the access that users have to mobile applications and to protect the limited programs and data that the applications use. Pogar *et al*. (2013:749), asserts that in MAM, IT specialists prepare solutions for managing applications and data which will be installed on the different devices. They further define MAM as "a capability of managing applications on devices distanced from the central location".

According to Pogar *et al*., (2013:749), the applications are stored in a safe place (it can be seen as a box which is closed off from the rest of the world), and from there the data can be acquired. Leavitt (2013:18) stresses that only secured applications will be able to access the organisational network. One way of doing this is to establish an organisational application-store where secure applications can be acquired, instead of acquiring these applications from public application-stores.

MAM is based on the following principles, according to Pogar *et al*., (2013:749):

- Application management is done by users (humans) and not the hardware of software. Emphasis is placed on the organisations applications and data.
- Employees are given different roles and rules, and the organisational application management should be organised according to the execution thereof. The application management needs to be flexible enough to take into consideration any future change in the roles or rules.
- Public application-stores may be used, under control of and through recommendations by the IT administrators. This should be done whilst still following predefined rules set by BYOD and always keeping the organisation's data in mind.
- Living by the following phrase: "Configure once, apply everywhere".
- Having a full view of all activities of all employees at all times.

These principles and rules may change from organisation to organisation, but the basic principle always remains the same.

#### *3.1.2.2.4 Cloud storage*

In some cases organisations have adopted the use of cloud storage (Leavitt, 2013:18). This provides employees with mobile access to different applications and data from their organisation, while also providing security to the data through the use of encryption (Leavitt, 2013:19). The employees can gain access to the cloud via an authentication process that uses any physical form of security, such as tokens, certificates, smart cards or Short Message Service (SMS) messages (text messages), in combination with a username and password (Gajar *et al.*, 2013:68).

Cloud storage is a good way of securing information and data, but the data and applications remain only secured whilst on the cloud (Leavitt, 2013:19). Once an employee downloads the data from the cloud to their device and unencrypts the data, the safety and security of the downloaded data rests in the hands of the employee using the device. In this case, MAM and MDM can help keep the data more secure. More information on cloud security and security threats while using cloud services will be discussed in Section 3.1.4.3.

#### 3.1.2.3 Conclusion to UbiComp and BYOD

With regard to BYOD in general, it is simple to see that most risks are related to humans. In addition to this, when working with the human aspect of security, there are few ways to ensure that security will not be compromised. Thus it is very important to implement rules and roles to different people in the organisation. This will enable the tracking of the person(s) responsible in the event that security is breached.

In conclusion, humans have a large impact on security and therefore policies are put into place. Humans make mistakes, and thus the threat of malware to users of mobile phones has to be discussed. The next section will elaborate on malware and the threats they may pose.

### 3.1.3 Malware

Malware or malicious software can be seen as any type of program or piece of code that was written with malicious intent. According to Armando *et al.* (2014:248), malware is one of the main threats to mobile devices. Felt, Finifter, Chin, Hanna and Wagner (2011:3) agree by stating that even though mobile malware was only seen as a concept at first, it is now becoming a large threat. Different types of malware can exploit vulnerabilities in mobile applications and corrupt or compromise the integrity of data on the device. Lin *et al.* (2014:22) state that malware can easily be hidden in applications that seem innocent.

Seo, Gupta, Sallam, Bertinno and Yim (2014:43) assert that although application-stores have given users an easy way to acquire different applications, they have also made the distribution of malware very easy. They go further to state that although official application-stores (such as Google Play Store and Apple’s AppStore) have a process to screen incoming applications for malware, it is a very moderate screening process and malware authors can easily bypass the screening. Other unofficial application-stores do not offer security of this kind, and should be avoided as far as possible. One of the most popular techniques used by malware authors today is to take a genuine application’s source code, inject it with malware and then rewrap it to look authentic (Seo *et al.,* 2014:43; Zhou & Jiang, 2012:97). The next sections will provide more detail on malware.

#### 3.1.3.1 Types of malware (What)

According to Felt *et al.* (2011:4), there are three types of threats that these malware pose. These are:

- **Malware:** This type of threat gains access to a user’s device to steal information, damage the device in some way or even just annoy the user. The malware is installed by the user, not knowing that the application has malicious software injected into it, or it could be installed by remote access through vulnerabilities the device might have. Malware include: Trojans, worms, viruses and botnets.
- **Personal spyware:** The purpose of this threat is to steal information. It collects information, such as location, message histories, call histories, images, calendar entries and more, over a specific time-period. Personal spyware is normally installed directly to the phone, and thus the coder or author of the spyware needs physical access to the device. After installation, the spyware sends the specified information to the author or coder of the spyware.
- **Grayware:** With grayware, the intent is not always malicious. Legitimate companies use grayware to spy on their customers to collect information for marketing, different statistical profiling or analysis. Companies can decide whether their clients know about the spying or not. Grayware, as can be deduced from the name, sits on the border of the law though. It can be used for good, but in the wrong hands it can be used to get information from users who have knowledge about it, but do not really know what the information will be used for. If clients find out that they are being spied on and the companies did not inform them, or the information is used for something other than a specific contract implies, it is punishable by fines.

These are the different types of threats that malware pose. Malware can also be categorised in different types (classes) of malware. The following are different classes of malware:

- **Virus:** This class of malware enters a device, via hardware or software, with the intent to multiply and cause different malicious tasks on the device (Penning, Hoffman & Nikolai, 2014:182). A virus requires human intervention (for example, using the infected files) in order to multiply (Peng, Yu & Yang, 2014:927).
- **Worm:** This class of malware have many different types of malicious intent. They gain access to a device without the user’s knowledge and spread through self-replication.

Worms do not always have to attach itself to a specific file, but use files to travel throughout the device and cause a variety of devastating effects.

- **Trojan horse:** This class of malware includes software that appears to be legitimate, but poses some harmful effect when opened or used. This malware is named after the large wooden horse used to penetrate the gates of Troy. It is normally used in mobile devices to record different forms of communications (Penning *et al.*, 2014:182).
- **Spyware:** This class of malware is generally used for information collection. Generally, spyware collect information for advertising purposes for a third party (Peng *et al.*, 2014:927). However, if used maliciously, it can steal important information from users, which can lead to identity theft or bank accounts being emptied (Penning *et al.*, 2014:182). This class of malware poses a large privacy risk.
- **Backdoor:** This program is installed on a device or a specific application and provides access to the device or application via different network connections. In the mobile world these connections can be anything from Bluetooth to Wi-Fi, and even the Internet (Peng *et al.*, 2014:927).
- **Rootkit:** This class of malware can be harmless, but can also be the cause of huge losses and unwanted access to devices. Rootkits are hidden amongst different files, processes and network links (Peng *et al.*, 2014:927). It attacks the OS itself and thus has deep intrusion into the device and not only some applications.
- **Botnet:** Installed on a group of different devices, this malware class can be used to remotely control infected devices. This malware class is generally used to initiate attacks over different networks, on a large scale (Peng *et al.*, 2014:928).

Just as these different threats and types of malware exist, so do security solutions exist that help detect and prevent the malicious effects. Felt *et al.* (2011:4) and Penning *et al.* (2014:184) provide security measures that have been set up by the creators of smartphone OSs. These measures are discussed below:

- **Markets:** The application-stores (also referred to as markets) are different for Android and iOS, although the idea remains the same. Android has an official application-store, which gives users a slightly better security screening for applications when compared to other application-stores. Users can find other locations to download applications, but the security would be much lower. Apple’s application-store and iOS also have a screening process for applications, and increase security further by allowing users of iPhones or iPads to only install applications from the store and from no other locations. This security measures can be circumvented by the user through jailbreaking an iOS device or rooting an Android device. Jailbreaking allows users to install applications from more locations than just the

Apple application-store, but diminishes the security of the OS on the device. Rooting does not allow the user much more freedom since Android users can already allow applications from unofficial application-stores, but it diminishes the security of the device and OS. Penning *et al.* (2014:184) claim that the markets can be perceived as the bouncers of the channel between applications and devices. The Google Play Store even has a malware detection system named Bouncer, which scans all applications before they become available for download.

- **Permissions:** The Android and iOS software protects the user by asking permission before allowing some applications to access certain information. If a user has not given privileges to an application to use certain information and the application tries to use it, an alert could be given to the user. Without permissions, applications would be able to use any information without the user knowing it (Felt *et al.,* 2011:4).
- **Signature-based detection:** The mobile application markets use signature-based detection systems, among others. It analyses different known malware patterns and results, and uses this analysis to detect different malware types which might be attached or hidden in an application. New and unknown malware might get past this type of detection system.
- **Built-in security:** This refers to security built into the device itself by the manufacturers on OS level. It helps to detect variations in application patterns in order to detect malware. Different manufacturers have different security implementations, for example, Samsung uses KNOX, a strategy which involves secure booting, integrity measurement in various ways and OS-level (Kernel) enhancements to security (Penning *et al.*, 2014:184).
- **Security awareness training:** This is important in any type of enterprise with mobile paradigms set up. Security awareness training should focus on teaching employees (or any mobile device user) the different malicious activities that exist and giving them knowledge about various ways to prevent them.

Looking at these different types and characteristics of malware it can be observed that if any of these malware gets installed onto a device running an EMA, the company using this device can suffer intense losses of data and integrity.

#### 3.1.3.2 Why malware is developed (Why)

Now that the ‘what’ has been discussed (refer to Section 3.1.3.1), it is time to address the ’why’. The reasons why people use malicious software and the incentives will be discussed in this section. Felt *et al.* (2011:5) categorised malware according to the different motivations or incentives for the malware type. The following are the different incentives of the authors of malware:

- **Amusement:** The author writes the malware for the sole purpose of his or her own amusement (Felt *et al.,* 2011:5). There is no other motivation behind the malware.
- **Selling the information retrieved:** The OS of different smartphones has an API (Application Program Interface) which can be queried to retrieve information about the user. Examples of information are: location of the user at the current time, lists of different applications on the device, contact lists and even the unique IMEI (International Mobile Station Equipment Identity) number. This information can be important to different groups of people for different reasons, and thus it can be sold to them for a price (Felt *et al.,* 2011:5; Peng *et al.*, 2014:928).
- **Stealing user credentials:** Many people use their smartphones for shopping, emailing and even banking. All these activities require passwords or some kind of authentication. Some users save their authentication information (whether used on the mobile phone or for another type of authentication in their everyday life) to a document on their mobile phones to serve as a password manager. The information saved is not always limited to passwords and authentication information, and may include credit card credentials and identification information. If malware target this type of information, it could be used for identification fraud and other crimes (Felt *et al.,* 2011:6). Peng *et al.* (2014:928) state that bank account details can be stolen by means of Trojan horses and this can result in wealth being lost/stolen.

Penning *et al.* (2014:183) agree with these citations by stating that mobile devices are data-centred devices and that many people store sensitive and important data on their devices. If an attacker could access this data, they could sell it or use it to gain access to various accounts or places. Attackers can use the following methods to access data or information on mobile devices:

- **Premium-rate calls and SMSs:** These types of calls and SMSs can be costly to a user, but are necessary (or wanted) at times for things such as stock quotes, technical (or other) support and even adult services. A user’s device and associated account can be abused by installing malware on the device that charge the premium-rate calls and SMSs to the device with the installed malware, but send the service to another user (Felt *et al.,* 2011:6; Peng *et al.*, 2014:928).
- **SMS spam:** Spamming via mass SMS is used for advertising as well as spreading phishing links. Many of these commercial spammers are motivated to use malware for their spamming needs since spam is considered illegal in many countries. It is also common to use some form of compromised device to deliver the malware or initiate the mass spam (Felt *et al.,* 2011:7).

- **Optimising search engines:** Many different websites rely on search engines to direct traffic towards them. Search engines rank websites according to the relevance of the search terms and how many people click the website after entering the search term. Since it is a well-known fact that websites on the second and subsequent pages of Google seldom gets seen or visited, malware can be used to raise a specific website in terms of rankings, thus putting it higher on the search engine's 'found' list. This will direct more traffic toward the website (Felt *et al.,* 2011:7).
- **Ransom:** Malware can be used to enforce blackmail demands. For example, an application can steal information from a device and post it online, and then demand a certain amount of money before the stolen data are removed. Another example is when malware add password protection to information or applications on a device, and then demand a ransom that should be paid before the author removes the password protection (Felt *et al.,* 2011:7). Penning *et al.* (2014:183) affirm that financial gain is one of the top motivations for cybercriminals and that malware can easily be used for various tasks for making or stealing wealth.
- **Access to private (unauthorised) networks**: Penning *et al.* (2014:183) claim that with BYOD (discussed in Section 3.1.2) becoming such a popular paradigm with enterprises in present times, networks are easier to compromise. The BYOD paradigm works with a large network which connects most (if not all) enterprise mobile devices. This means that the compromise of a single device can lead to the compromise of the entire network; thus security implementations for such situations was discussed in Section 3.1.2.

The 'why' of malware was discussed; it can be seen that there are many reasons for people to use malware. These people can be someone who just wants a little amusement based on the suffering of other people, or even an organisation that needs free advertising in different ways.

Some of these reasons might not look like motives to be concerned about, but when looking at points like the theft of user credentials, selling of information (no matter what information) and also attain access to private networks, these are things that companies running EMAs have to look out for.

The next part of this discussion will deal with the 'how' of malware.

#### 3.1.3.3 How malware is installed or activated (How)

This section will discuss how malware is installed, how the software can be activated, how the data are transmitted and how malware can use permissions to its own benefit, as seen in a survey done by Zhou and Jiang (2012:97-104).

#### *3.1.3.3.1 Installation*

Installation of malware can be categorised into three techniques: repackaging, update attack and drive-by download. According to Zhou and Jiang (2012:97), these techniques are not exclusive and different variants of the techniques might be used.

- **Repackaging:** Repackaging is one of the most popular techniques used to distribute or install malware (refer to Section 3.1.3). The idea is to take an existing application and inject it with a malicious piece of software, by disassembling the application and encapsulating the malware inside it. The application is then repackaged to look benevolent, with no harmful intent. The application is then downloaded from the specific application-store, and the malware is spread (Zhou & Jiang, 2012:97). This technique uses the application to carry the entire load of the malware, but this may expose the presence of the malicious code.
- **Update attack:** This technique makes detection more difficult. The technique remains the same by encapsulating malicious code into popular applications or software, but instead of putting the entire payload into the application, only updated code is added to the application. This means that when the application runs, the malware is downloaded from another source. This ensures that applications scanning for malware do not identify the malware as easily as having the entirety of malicious code embedded into the application (Zhou & Jiang, 2012:98).
- **Drive-by download:** This technique adds to the previous two techniques, but are related to marketing. It relies on the interest of users by using feature-rich and interesting applications within which to include the malware, thus downloads of the application should be more than other, more standard, applications (Zhou & Jiang, 2012:99). This technique exploits the naivety of users and the fact that people download applications based on the look of the application and simply to have more applications.

These three techniques are very similar and may look like the same technique wrapped in different wrapping-paper. The first technique takes any application and includes the malicious code into it. The second technique tends to be sneakier and does not put the malware itself into the application, but instead puts a link to download the malware after the application itself has been downloaded and run. The third technique shows that the malware can be marketed better by making it more interesting.

Now that the installation has been done, the malware has to be activated. The next section will explain how this happens.

###### *3.1.3.3.2 Activation*

The installation of malware has been discussed, but malware inserted into different applications are harmless if it is not activated. According to Zhou and Jiang (2012:100), the malware is normally activated through different OS events. Different devices might use different event triggers, but in most cases these are the main events used to trigger the malicious code (Zhou & Jiang, 2012:100). They give examples of events on an Android device as the following:

- BOOT (completion of boot);
- CALL (phone events);
- PKG (activities including different Android packages);
- SMS (SMS/MMS);
- Universal Serial Bus (USB) storage activities;
- BATT (activities including power/battery);
- NET (network activity);
- MAIN (activity in main); and
- SYS (any system events).

After the malware has been activated, it has to complete some malicious activity. The next section will discuss different activities that malware will engage in.

###### *3.1.3.3.3 Malicious activities*

This section is similar to Section 3.1.3.2 in discussing different reasons of malware distribution. The discussion will include privilege escalation in devices, devices being used as remote controls, malware using devices to charge different things off the user's finances and finally, stealing information from users for different reasons.

- **Privilege escalation:** Privilege escalation can be seen as exploiting a flaw in an OS or application to gain higher access to a device. Zhou and Jiang (2012:101) claim that smartphones and other mobile devices (focusing on Android) have different libraries and packages that can be exploited to gain higher access to the device itself. The intent of the malware would be to exploit different libraries and to gain the highest access to the device as possible; this makes the device easier to exploit.
- **Remote control:** Many different malware types turn devices into remote control devices, which can be controlled by the malware author (Zhou & Jiang, 2012:101). With this capability, the malware author cannot only use the device from a remote area, but also see all activities that the device is used for and manipulate those activities for their own benefit.
- **Financial charge:** One example of financial charge can be ransom, mentioned in Section 3.1.3.2. Another example mentioned by Zhou and Jiang (2012:101) is to purposely have the

malware subscribe the user of the device in different premium-rate services that have high financial charges. At times this can be seen as amusement to the author, instead of real malicious intent from the author.

- **Information collection:** This type of malware is used in many cases to steal credentials and other information from users (Zhou & Jiang, 2012:103). Regardless of whether the information will be sold or whether the malware authors have other malevolent intentions, information is stolen from users by embedding different malware types in applications and waiting patiently for the user to download.

As the activities of malware have been discussed, the question of permissions might arise. How do some of these malware get permission to do what they do? The next section will briefly discuss this.

#### *3.1.3.3.4 Uses of permission*

Zhou and Jiang (2012:103) discuss permissions and different ways in which malware can exploit them. When a user downloads an application from an official application-store, he/she is normally asked to allow permissions to different applications, or told what permissions the OS will allow for the application. The user can then decide what permissions he/she will allow by choosing whether he/she wants to download the application); thus if the application needs permissions that the user feels unsafe with, he/she can find other applications that will fulfil his/her needs. On the other hand, if the required permission does not look like something that malware would use, the user should have no problem with allowing permissions.

Authors normally write malware in two different ways. In the first way pertains to the repackaging of the application, when permissions are added by the author to the list presented to the user when prompted to allow permissions or told what permissions the OS will grant. The second way is to hide the permissions that the application will use from the user. The way is normally used when the author does not want the user to know what permissions the application will use. However, if the user keeps track of permissions and notices that the application is using permissions that were not listed, this could be a problem. This could cause suspicion in the user and result in the application and the malware being deleted.

The following is a list of some permissions that can be exploited by Android malware, given by Zhou and Jiang (2012:103-104), while discussing their survey:

- INTERNET;
- READ_PHONE_STATE;
- ACCESS_NETWORK_STATE;
- WRITE_EXTERNAL_STORAGE;

- READ_SMS;
- WRITE_SMS;
- RECEIVE_SMS;
- SEND_SMS;
- RECEIVE_BOOT_COMPLETED; and
- CHANGE_WIFI_STATE.

It is important to know what permissions applications ask for and what permissions they will actually use. Many applications ask for a complete list of permissions, while they only use a few of them. These applications should be avoided, because they open the device up to malware and intrusions.

As can be seen from this ‘how’ section, it could be easy to mistake a harmless application for a malicious application, without knowing the difference at all. The safety of the user is mostly up to themselves, and different applications should always be taken into consideration before just settling for one. Different permissions should be checked as well as alternative features of different similar applications. Official application-stores should also be used in preference to unofficial stores.

Overall malware is a threat that companies have to take into account when working with mobile devices that run EMAs and use the company’s data (or even other companies’ data). When developing EMAs they also have to be taken into account, needing the knowledge of what to protect against.

### 3.1.4 Overall mobile security threats

This section will discuss different mobile threats that users need to be aware of, as well as some solutions and prevention techniques. The majority of these threats and solutions come from Sathyan and Sadasivan (2010:3-4), who propose a collaborative approach to security in enterprises (this will be discussed in Section 3.2.4).

#### 3.1.4.1 Major threats

The following are some of the major threats that enterprises using mobile devices might encounter (Sathyan & Sadasivan, 2010:3):

- **BlueBugging:** Allowing the user to access any activity on the user’s phone via Bluetooth.
- **BlueJacking:** Sending of an unwelcome message across Bluetooth.
- **BlueSnarfing:** Gaining unapproved access to the victim’s data over Bluetooth.
- **Cryptanalytics:** Gaining access to encrypted data without having the information needed to unencrypt the data.

- **Device cloning:** Cloning the entire mobile device, and by doing so any charges made on the clone are charged to the cloned device; all data and information is also cloned.
- **Falsifications of content:** Taking any content and falsifying it by making unwanted changes and by doing so corrupting the original.
- **Eavesdropping:** Secretly listening in on a conversation without consent.
- **MITM attack:** Eavesdropping, but instead of just listening, the messages are sent toward the attacker and relayed to the person that they were meant for, without the two parties realising; the conversation is managed by the attacker.

#### *3.1.4.1.1 Solutions*

The following are some of the solutions against the threats mentioned in Section 3.1.4.1, proposed by Sathyan and Sadasivan (2010:3-4):

- **Mutual authentication:** Authentication system where both parties, communicating with each other, have to give consent before a communication is opened.
- **Digital signature:** A signature is used to authenticate document transfer between employees to ensure that the document received has not been altered in any way; this is often done by a public key infrastructure (encryption/decryption).
- **One time password:** A new password is generated for each session that requires authentication; this password has a limited lifetime to ensure that security is at its highest.
- **Out-of-band authentication:** Authentication is received via an outside channel (any channel other than the one the user is using); for instance, if the user has to authenticate ownership of an account, an email with an authentication code will be sent (the email is the outside channel).
- **Digital image:** Data can be encrypted into images with steganography and can only be unencrypted with the correct password or authorisation techniques.
- **Shared secret (encryption methods):** A closed Public Key Infrastructure (PKI) in which only certain parties are allowed access to keys and the infrastructure, and not the entire organisation.
- **Poison pill:** A message is sent to a device to completely wipe all data; this is generally done when a device is stolen or lost, puts the data in jeopardy.
- **Device tracking:** The GPS coordinates of a stolen or lost device can be used to track it.
- **File encryption:** Any data on the device (memory cards and local) can be encrypted and stored externally.
- **Multi-factor authentication:** This form of authentication involves one or more additional levels of security. For instance, if a user requires a username and password to log into an application, an additional form of previously selected authentication can be used to ensure

that the user is authenticated (an image or photo of the user) (Jain & Stanbhag, 2012:32; Sathyan & Sadasivan, 2010:4).

- **Service provider validation:** Every mobile device has a transmission signal which can be seen as a fingerprint, which is unique regardless of the changes made to the device; this can be tracked by the service provider (Sathyan & Sadasivan, 2010:4).

These threats are mainly vulnerabilities revealed by Sathyan and Sadasivan. Section 3.2.4.3 contains a table (Table 3-1) giving a summary of these threats and solutions. The next section presents a view on security threats in a cloud environment.

#### 3.1.4.2 Threats in a mobile cloud environment

In a survey done by Jana and Bandyopadhyay (2014:3), a summary of threats is given. This is presented in Figure 3-4.

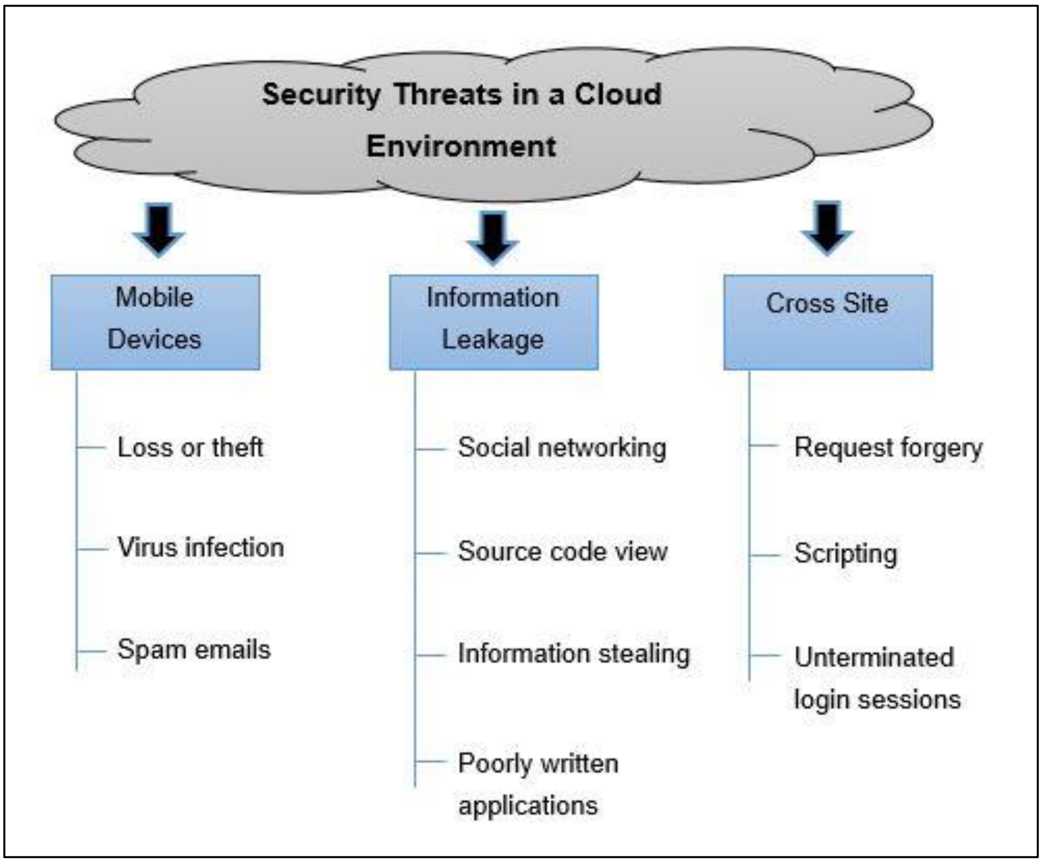


**Figure 3-4: Security threats in a cloud environment**

Along with these threats, Jana and Bandyopadhyay (2014:4) give the following threats:

- **Flaws in cryptography:** Cryptanalytics (refer to Section 3.1.4.1) is the hacking of encrypted files. Cryptography is the act of encrypting data and should be done in the correct manner with focus on the correct areas. Jana and Bandyopadhyay (2014:4) mention that many cryptologists encrypt using a high security algorithm, thus the focus should not be on the algorithm but rather on the selected keys. In many cases, the algorithm is rendered useless without the key, thus strong encryption/decryption keys should be used.

- **APIs and interfaces that might be insecure:** Cloud Service Providers (CSPs) monitor and manage the performance of cloud services, and therefore they provide a set of APIs that may be used by the client. These APIs act as guidelines in using the cloud services in the correct way, and not following the guidelines and setting up the cloud service in the correct manner may result in a lack of security.
- **Multitenancy:** In normal enterprise setups employees each have their own logical cloud space, which allow them to store data in a secure manner. Central data (which all employees use) is generally stored on a physical server and not on a cloud, but in cases where the data are stored on a cloud, multitenancy is reached. When multiple users connect to and use resources from the same physical cloud, it results in different security threats (Jana & Bandyopadhyay, 2014:4).

#### 3.1.4.3 General security threats/solutions

This section will reveal general security threats and solutions which might not apply to one singular condition, but might help in improve security surrounding the development of EMAs in general. Nyambo *et al.*, (2014:33) state that it is always important to have security software installed on a mobile device, whether the applications are secure or not.

A security manager is a security utility which allows the device to filter unsecure data from secure data (secure/unsecure data differs in different enterprises). This is accompanied by a utility (user awareness manager) which allows the management of personal user data and enterprise data (Radia *et al.*, 2012:847). The separation of these two types of data automatically increases security where enterprise data might be mistaken for personal data and might be downloaded or uploaded to an unsecure platform.

Radia *et al.* (2012:852) claims that knowing the following information is very important:

- Who is accessing what application, and what the application is doing;
- Who is controlling and authorising access and how it is done; and
- How the privacy of accessed information is maintained.

Even though cryptography and encryption have been mentioned, Yun and Xiao-hui (2009:386) claim that the importance of data encryption cannot be disregarded and that encryption infrastructures should be setup and managed to be as secure as possible. They mention that any encryption infrastructure resembling a PKI should be adequate.

### 3.1.5 Conclusion to security in enterprise

For any mobile device user, all these threats might seem small and the security to prevent them might seem like too much effort. This might be true in some cases where the user does not have any important or sensitive information on the mobile devices, but in the case of enterprise security it is not true. Depending on the enterprise, threats might even be larger than initially thought; security will require stricter policies and employees might have to follow more rules.

It is important to know when the security is necessary, but it is always good to keep security in mind. In most cases policies and rules can prevent users from putting their devices in jeopardy, but in some cases the security has to be built into the applications. Section 2.3.3 briefly mentioned security in mobile applications; the next section will elaborate on the security during the development of these mobile applications, more specifically EMAs.

## 3.2 Security during the development lifecycle of EMAs

In this section the core of the study will be discussed, namely security aspects addressed during the development of EMAs. These aspects, together with other data received from interviews, will be summarised and used to create a framework. This framework will be used to help and guide developers in creating more secure applications for use in enterprises. The next section starts with a brief explanation of a security framework created by Hasan *et al.* (2014:1-3), followed by different researchers' perspectives on security implementations in enterprises.

### 3.2.1 Secure developing framework

It is stated that mobile devices have higher exposure to threats as a result of the many different interfaces, namely: Secure Digital (SD) cards, Bluetooth, USB connections and Wi-Fi. The National Institute of Standards and Technology (NIST) listed these high-level threats (Hasan *et al.,* 2014:1):

- Physical security lack in mobile devices;
- Untrusted devices being used;
- Untrusted networks being used;
- Untrusted applications being used;
- Interaction with different devices/systems;
- Untrusted content being used; and
- Location-based services being used.

Jain and Stanbhag (2012:30) add the following to the list of high-level threats:

- Device loss or theft;
- Interception of and tampering with data;

- Malware;
- Vulnerable applications;
- Compromised devices;
- Social engineering; and
- Exploitations of web browsers and vulnerability in the OS.

Unhelkar and Murugesan (2010:38) also add to these two sets of threats by asserting that any type of communication via wireless and mobile communication networks are more susceptible to attacks than physically wired networks. Frameworks with priorities focused on security can prevent attacks on high-level threat areas, even in wireless and mobile networks.

Before presenting the framework, it should be mentioned that it was originally adapted from the Service Oriented Architecture Decision Model framework. It provides architectural decisions to be used to support EMA development. The framework takes into account three different perspectives, namely (Hasan *et al.,* 2014:2):

- **Business perspective:** The data are classified into different security levels by the enterprise.
- **Technical perspective:** The countermeasures needed for the different security levels and threats are applied.
- **User perspective:** This perspective works together with the previous perspectives. A high level of restrictions can result in high-level security, but this can result in low user acceptance. As mentioned in Section 2.3.3 a balance should be managed. The user has to accept the security countermeasures applied.

Along with these three perspectives, two models are used to complete the framework. These models are the guidance model and the decision model, and will be discussed next.

#### 3.2.1.1 Guidance model

The guidance model is based on the business perspective, where the security requirements are acquired for the different types of data. This guides the enterprise and developers to create a concept model of security for the EMA being developed. It also helps them to check if the needed security requirements are, or will be, fulfilled (Hasan *et al.,* 2014:2).

The model consists of two components, which are the following:

1. A list of mobile security threats or issues, the likelihood of each of them occurring and the harm that they might cause to the system.
2. A list of countermeasures that might be applied, their known uses in previous applications and the acceptance rate of the countermeasure of previous users.

The two listed components are established during the risk analysis process - SP 800-30 that is provided by NIST. The enterprise is informed that a specific threat exists and that a countermeasure is needed (Hasan *et al.*, 2014:2).

#### 3.2.1.2 Decision model

The decision model helps with the decision making in terms of countermeasures. It presents the security concept of the EMA and is created in the tailoring step. After the guidance model presents the security threats and possible countermeasures, the enterprise tailors the choices made. Thus in the tailoring step the enterprise might remove irrelevant security issues or add new security issues. Along with tailoring the choices, the enterprise might add additional security countermeasures to each issue, to ensure certain issues are more secure. After this is done, the decision model sends the choices back to the guidance model to ensure that all security requirements are fulfilled and user acceptance is high enough (Hasan *et al.*, 2014:3). The decision model is put on a “decision loop”, thus if the requirements and an acceptable user acceptance are not acquired, the whole process loops back to a previous step.

#### 3.2.1.3 Conclusion

To conclude, Figure 3-5 (Hasan *et al.*, 2014:3) shows a workflow of the secure developing framework’s process.

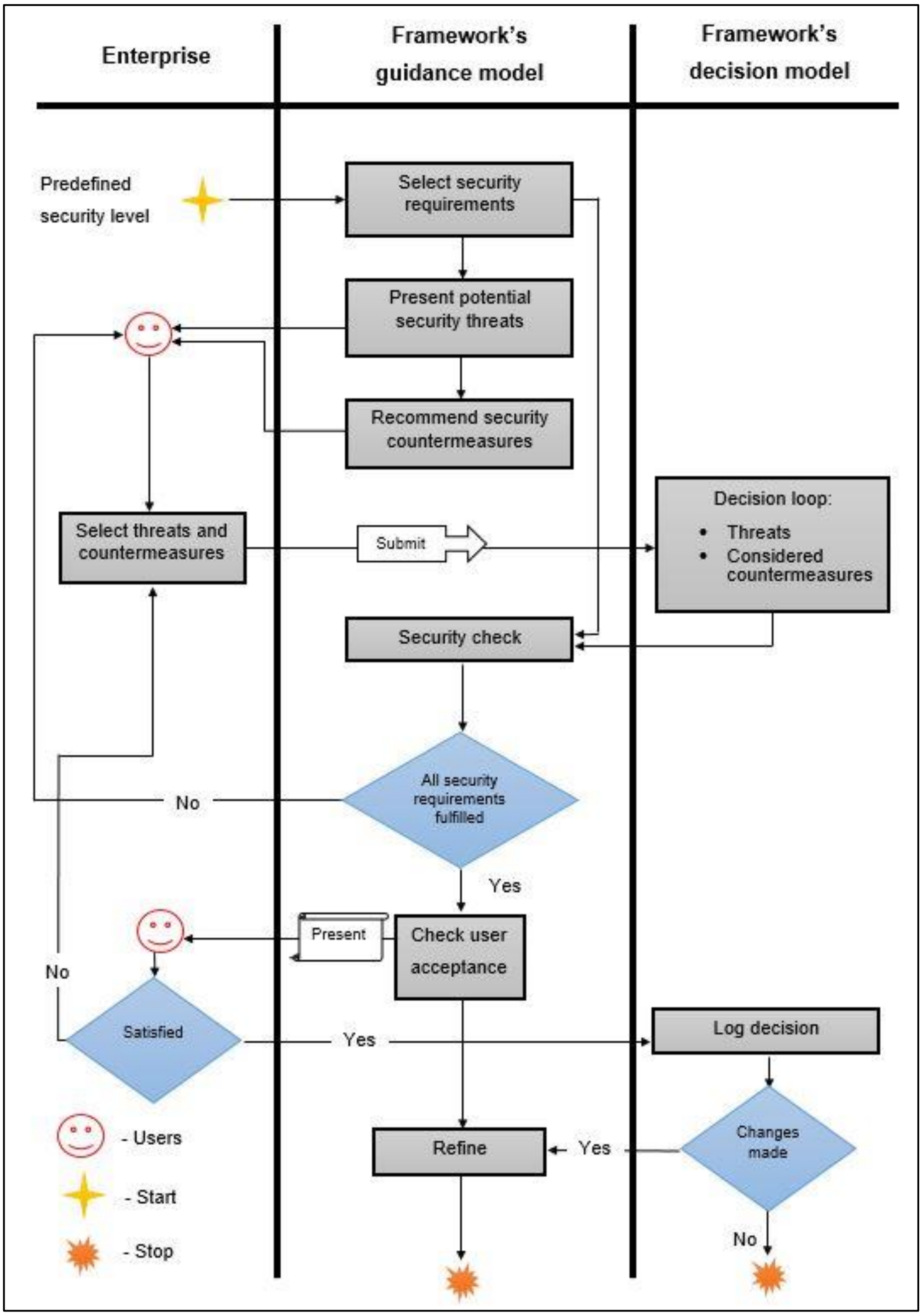

Enterprise
Framework's guidance model
Framework's decision model
Predefined security level
Select security requirements
Present potential security threats
Recommend security countermeasures
Select threats and countermeasures
Submit
Decision loop:
• Threats
• Considered countermeasures
Security check
No
All security requirements fulfilled
Yes
Present
Check user acceptance
No
Satisfied
Yes
Log decision
Refine
Yes
Changes made
No
- Users
- Start
- Stop

**Figure 3-5: Workflow of the secure developing framework's process**

It should be clear from looking at the different checkpoints in the figure (*Select security requirements*, *Recommend security countermeasures*, *Security check* with a decision loop, *Select threats and countermeasures* and more) and refinements in this framework that security is assessed during the entire development lifecycle and not only during development.

In the following sections, the different views with which people see security and how it should be implemented will be discussed, as acquired from various literature sources.

### 3.2.2 Implementing security in different phases in development

In this section, Jain and Stanbhag's (2012:31-33) perspective on secure mobile application development will be discussed.

Jain and Stanbhag (2012:32) mention a few issues to take into consideration when designing mobile applications. These aspects, in no specific order, are:

- Access and privilege control;
- Data encryption (with all data, whether it be data in transit or in local storage); and
- Strong password policies, accompanied by strong account lock-out policies.

It should be noted that one of the most important things when it comes to development is communication between project team members (Lennon, 2015:1). Jain and Stanbhag's (2012:31-33) perspective on secure mobile application development revolves around a circular secure development approach which consists of three main phases, namely:

- Secure application design;
- Security awareness for developers; and
- Security assessment.

#### 3.2.2.1 Secure application design

In this phase threats are modelled and adequate security controls are assessed. Depending on the application, different appropriate security controls are incorporated to fill the security gaps identified during the threat assessment. It is important to note that the applicability and platform support of security controls should be properly evaluated before implementation.

##### *3.2.2.1.1 Multi-factor authentication*

Multi-factor authentication means that more than one layer of security is used. Instead of having traditional passwords, personal Identity Document (ID) numbers and/or secret questions (that are all defenceless against guessing, social engineering and brute-force attacks), two or more independent security layers of security are implemented in addition to the existing ones (Jain & Stanbhag, 2012:32).

##### *3.2.2.1.2 Digital signatures*

Data security can benefit from PKI for authentication and data integrity (Jain & Stanbhag, 2012:32). This is normally used inside the enterprise when transferring data between employees to prevent MITM attacks (refer to Section 3.1.2.1).

##### *3.2.2.1.3 Data encryption and transport*

As mentioned before data encryption is important for all data, whether in transit or in rest in storage. Certain applications exist that can encrypt data that is in transit. Data which are in rest can be encrypted by OSs such as iOS, Android and Blackberry OS; they provide encryption libraries to be used (Jain & Stanbhag, 2012:32).

#### 3.2.2.2 Security awareness for developers

In order to secure application development, all developers should undergo developer training and be made aware of secure coding practices. This training should be administered before the coding phase and should cover any and all coding errors that might be a risk and end in vulnerabilities.

According to Jain and Stanbhag (2012:32), a good place to start with these errors and threat risks are the Open Web Application Security Project (OWASP) list of security risks. The list follows (Jain & Stanbhag, 2012:30):

- Insecure data storage;
- Weak server-side controls;
- Insufficient transport-layer protection;
- Client-side injection;
- Poor authorisation and authentication;
- Improper session handling;
- Security decisions via untrusted input;
- Side-Channel data leakage;
- Broken cryptography; and
- Sensitive information disclosure.

In addition to this list, Jain and Stanbhag (2012:32-33) add more security guidelines to follow while developing. These are discussed in the following subsections.

##### *3.2.2.2.1 Perform secure logging and error handling*

Commented code has to be logged for different debugging purposes; al logs also have to be logged to the global log. Poor exception handling has to be avoided, as this can result in the disclosure of sensitive information (Jain & Stanbhag, 2012:32).

##### *3.2.2.2.2 Follow the principle of least privilege*

The permission model, provided by the mobile device's OS, has to be correctly implemented. In addition to this, the principle of least privilege needs to be followed to ensure isolation of data (Jain & Stanbhag, 2012:32).

##### *3.2.2.2.3 Validate input data*

Input validation and duplicate validation is very important on the server side (Jain & Stanbhag, 2012:32). This ensures that only the required data can be entered into the application, reducing the chances of backdoors and unwanted entry into the application. It might be a good idea to implement security controls for the different validations (Jain & Stanbhag, 2012:33).

##### *3.2.2.2.4 Implement secure data storage*

Sensitive data should not be stored on a client device, but rather be kept on the server where it can be retrieved by the client devices. Standard encryption algorithms with strong keys should be used to encrypt sensitive data instead of self-coded encryption algorithms (Jain & Stanbhag, 2012:33).

##### *3.2.2.2.5 Avoid insecure mobile OS features*

Features such as cut, copy, paste and auto-complete are all insecure features that can be exploited to retrieve sensitive data from the application. They should be disabled for security purposes (Jain & Stanbhag, 2012:33).

#### 3.2.2.3 Security assessment

All applications have to undergo an extensive security assessment before they are released into production. This is done to ensure that all threats have been covered and no, or the least possible, security risks remain. This assessment may uncover gaps in the design and can contribute to the improvement of different security policies incorporated in the design and development (Jain & Stanbhag, 2012:33).

#### 3.2.2.4 Conclusion implementing security in different phases in development

In conclusion to Jain and Stanbhag's (2012:30-33) view on security during development, Figure 3-6 provides an overview of their work.

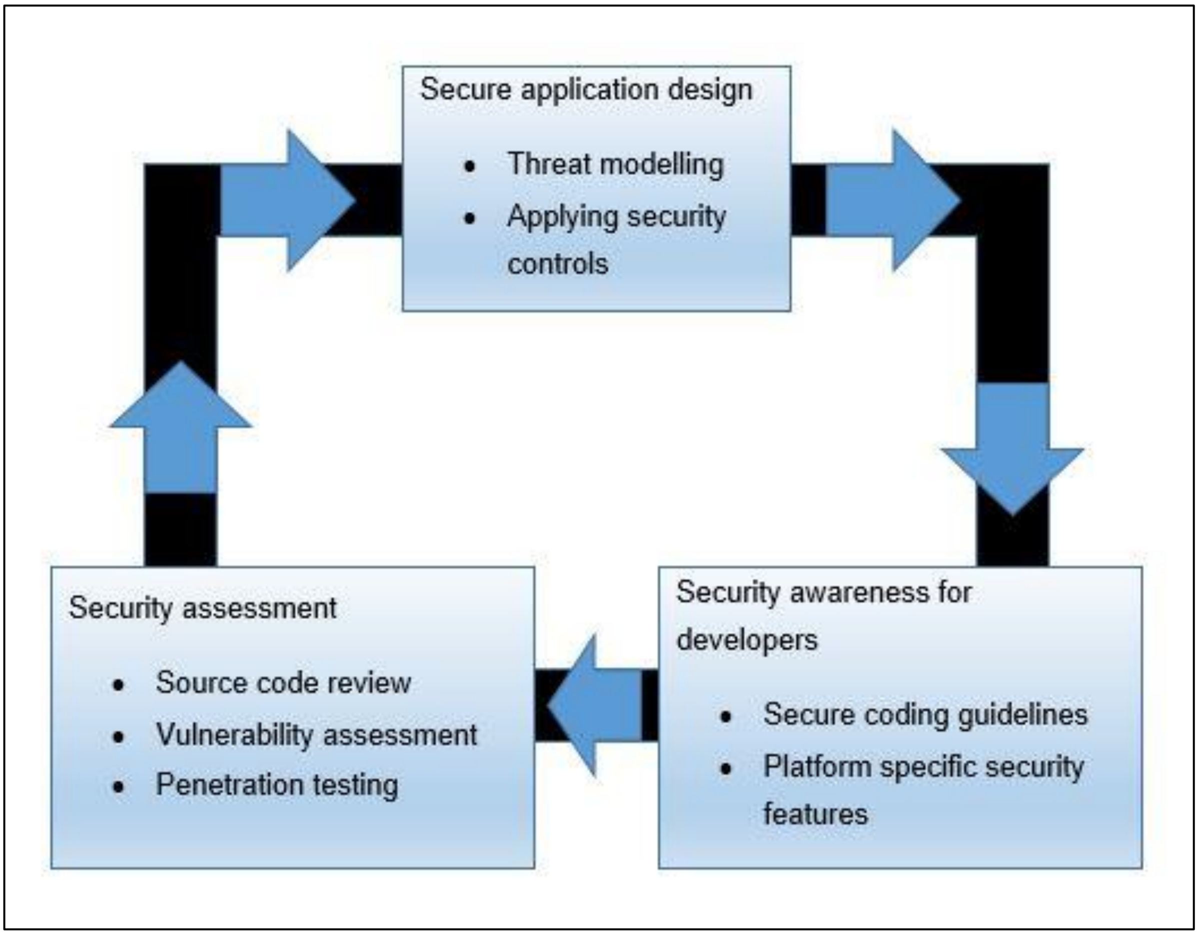

Secure application design
• Threat modelling
• Applying security controls
Security awareness for developers
• Secure coding guidelines
• Platform specific security features
Security assessment
• Source code review
• Vulnerability assessment
• Penetration testing

**Figure 3-6: An overview of secure MAD approach**

This model shows that development policies are always a smart choice when used in combination with proper preparation. Many security threats exist as a result of coding standards that are not followed, threats that were not assessed correctly and lazy developers. This can be prevented if the correct steps are followed.

By looking at the framework presented by Hasan *et al.* (2014:3) (refer to Section 3.2.1), as well as the model by Jain and Stanbhag (2012:30-33), it can already be deduced that security is not only development or preparation, but both in synchronisation with each other. The next section will consist of a short discussion about a virtual data container for EMAs, to protect the applications from outside harm and data loss.

### 3.2.3 Data container for applications

The work of Jaramillo, Smart, Furht and Agarwal (2013:2) is based on the concept of BYOD where employees can use their personal mobile devices at work, both for work and personal ventures. The aim is to allow employees to check their personal information without being too restricted or having to use authentication methods. This can be a problem if the enterprise data needs to be secured. Some form of separation between the user data and enterprise data have to be introduced, and this is where virtual containers come in (Jaramillo *et al.*, 2013:2).

The basic idea is to let EMAs that are using/containing enterprise data run within some kind of virtual container (sandbox), so that it is safe from any outside tampering. This is done by "containerising" the data at application level, by putting a specific layer of security/protection around the EMA. This layer separates enterprise data from personal employee data (Jaramillo *et al.*, 2013:2). Vaquero, Rodero-Merino and Buyyam (2011:46) claim that some cloud services give users this type of security, allowing their applications to run in a container-like environment in order to provide the data with more security.

#### 3.2.3.1 Container architecture

Figure 3-7 shows the system architecture of the virtual container. The container is run by a remote client via a web connection. It runs through a secure proxy with authentication and authorisation services, to ensure that the mobile device that connects to it is secure. When this secure session is started with the device, it enters a demilitarised zone (DMZ) behind a firewall, which is basically a safe zone (Jaramillo *et al.*, 2013:4).

Inside this safe zone applications can be browsed and downloaded, in a secure way, from the application-store. Some of these applications need backend (support) services which are made available via the application service, also inside the safe zone. Along with the application service, a security service resides inside the safe zone. This provides different applications in the store with authentication and authorisation tasks (Jaramillo *et al.*, 2013:4). This safe zone connects to a pre-existing intranet, which also contains different applications, security systems and developer support tools.

#### 3.2.3.2 Conclusion to data container for applications

A secure area was created for applications (and their data) to reside in instead of building the security into the applications during development. This benefits employees and raises user acceptance, because less policies around personal data and information are introduced. Personal data and enterprise data can now be separated along with their security.

This type of security can be a huge asset to enterprises when different security levels are required for different levels of data. For instance, personal data (lowest level of needed security) and enterprise data (different types of high-level priority data) can be accessed without increasing device or application security. If the container is up to standard and is set up for the specific enterprise and the type of threats they face, this can allow employees to access their personal data without endangering enterprise security. In this case enterprise security will be encrypted in such a way that can only be accessed when opened in the container area (Jamarillo *et al.*, 2013:4).

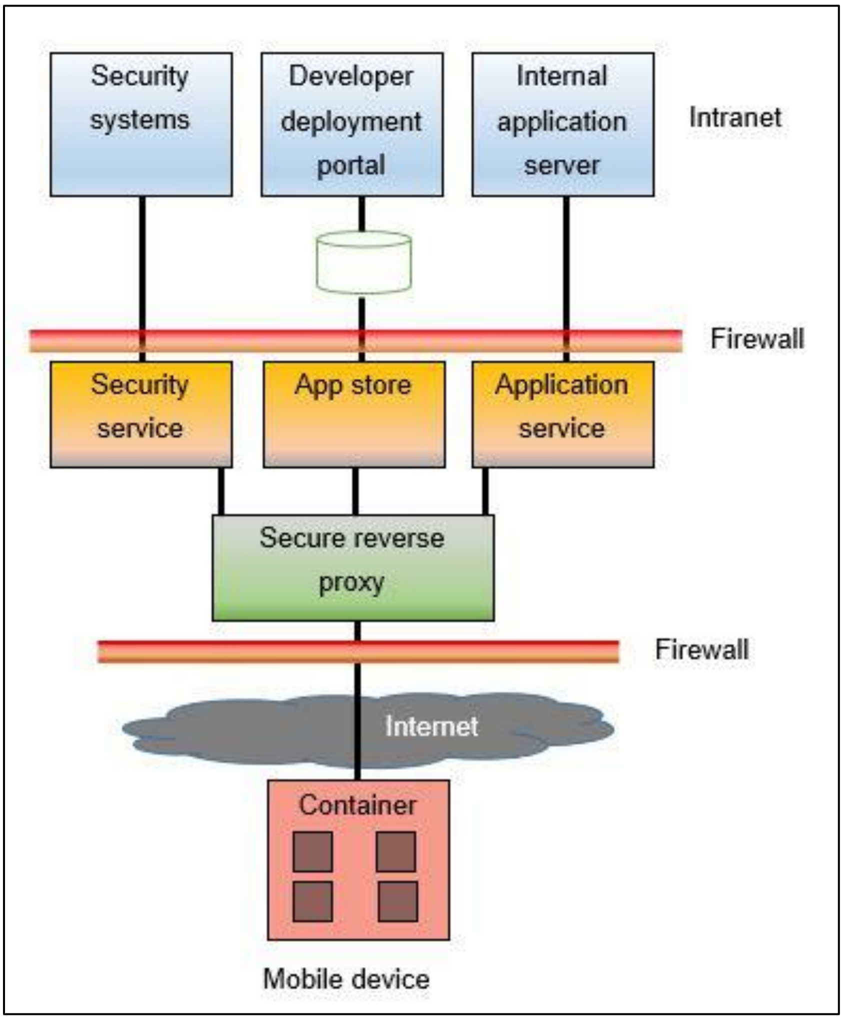


**Figure 3-7: Container architecture**

Different topics on security show that different security measures are used for different threats and levels of risk. Overall security in the enterprise use policies and rules to be followed by employees, where the development team normally uses different frameworks to be followed in order to get the highest possible security for applications. As mentioned in Section 2.3.3 a balance between application security (this includes security during development) and security policies in the enterprise needs to be established. The next section elaborates on a security infrastructure, for mobile enterprises security.

### 3.2.4 Multi-layered collaborative security infrastructure for the enterprise

Sathyan and Sadasivan (2010:1) state that hacking methodologies have evolved as a result of the evolution of the mobile industry throughout the years. This results in a variety of security threats that cannot be addressed by one specific solution. They propose a collaborative approach which ensures privacy and security to both customer and enterprise data. Shen, Lin and Rohm (2009:9) agree, but also state that technology alone cannot provide complete security. Therefore an infrastructure or architecture should be used, which involves different technological advancements; human contribution should not be overlooked.

#### 3.2.4.1 Key components for balance

Brown (2014:36) claims that getting the correct balance for security can be a difficult task; it involves ensuring that the necessary (and desired) technologies are included in the enterprise infrastructure, but also keeping all devices and data as secure as possible, even from new and unknown threats. The balance, mentioned before, has two key components which have to be addressed for maximum security, and they are (Sathyan & Sadasivan, 2010:1):

- **Enterprise system security:** This component ensures that the server framework used in the enterprise, as well as the mobile network infrastructure support security in transfer and handling of data, different connection aspects and communication.
- **Device and application security:** This component ensures that all mobile devices and applications are secure from outside threats and attacks. This can include authentication and authorisation of devices, device content security and ensure that data contained on the devices are secure via encryption and data transfer policies.

These two components, combined with specific domain security needs, will be discussed in the following sections.

#### 3.2.4.2 Enterprise system security

In most cases enterprises work with an infrastructure and system framework that supports multiple platforms e.g. Android, iOS and BlackBerry. Since BlackBerry is not used as much in present times, it can be replaced by Windows Mobile OS. BlackBerry was the leading enterprise deployment platform in 2010 because of its encryption and security advantages (Sathyan & Sadasivan, 2010:1).

When one specific platform is not the focus of the enterprise and multiple platforms are used, commercial off-the-shelf (COTS) products are used to create a collaborative enterprise system security infrastructure (Sathyan & Sadasivan, 2010:2). Many of these products are ready to address the different security requirements an enterprise encounters. In some cases enterprises develop their own security infrastructure, but it is recommended to rather use COTS products and integrate them with the rest of the infrastructure, since they are already made for the job.

The following can be seen as categories in a mobile security infrastructure or platform:

- Digital rights management;
- Storage security management;

- Application security management;
- Device security management;
- Certificate management;
- ID management;
- Encryption management; and
- Secure access protocol management.

If one of these categories have a specific need which is unique to the enterprise, custom development could help. However, most categories can be addressed by COTS products (which in some cases address more than one category) (Sathyan & Sadasivan, 2010:2).

#### 3.2.4.3 Device and application security

In this infrastructure the device and application framework consists of the following:

- Communication and transaction security;
- Authentication security;
- Subscriber Identity Module (SIM) card security;
- Device software security;
- Account security;
- Device's physical security; and
- End user security.

If this framework is not set up in the correct manner and device and application security is not up to standard, this might lead to loss, privacy intrusion and data integrity issues for personal and enterprise data. The enterprise might lead an economic loss; overall it can end in the enterprise losing a competitive edge (Wei, Gomez, Neamtiu & Faloutsos, 2012:253).

In Section 2.4.4 different security threats and their solutions were discussed as Sathyan and Sadasivan (2010:3-4) elaborated on them. These are used in a summary of security threats and there solutions given in Table 3-1. Solutions that are explained at one threat or vulnerability will only be mentioned if it applies to another threat or vulnerability.

**Table 3-1: Solutions to security threats (Adapted from Sathyan & Sadasivan, 2010:3-4)**

| **Threats and vulnerabilities** | **Defining terms** | **Solutions** |
|---|---|---|
| **Cryptanalytics** | Analysing restricted encrypted data without permission and where | • **One-time password:** Password that is only valid for a short time period;<br>• **Shared secret:** Type of encryption key |

| Threats and vulnerabilities | Defining terms | Solutions |
|---|---|---|
| | decryption information is generally needed. | that involves only the two parties included in the communication;<br>- **Memory encryption:** Memory cards or external storage devices can be encrypted to keep files secure; and<br>- **Digital images:** Method of encryption where the data are encrypted into an image and an encryption key is needed to retrieve the data. |
| **Device cloning** | Completely cloning a device, including its settings, information, calls, call logs, purchases, etc. | - **Data encryption**: Works very well with this type of security breach, because the data stays encrypted even if the device is cloned; and<br>- **Validation from the service provider**: Includes the checking for differences in the device fingerprint and the cloned device. |
| **Eavesdropping** | Listening to transactions/conversations without user permission. | - Digital image;<br>- Shared secret;<br>- One-time password; and<br>- Encryption of data in transit. |
| **MITM attack** | Involves eavesdropping, where the attacker makes the different parties believe something which is not true by altering the message and relaying it. | |
| **Falsification of content** | Using any original content and changing or adding to it. | - Digital image;<br>- Shared secret;<br>- One-time password; and<br>- Encryption of data in transit and when |

| **Threats and vulnerabilities** | **Defining terms** | **Solutions** |
|---|---|---|
| | | stored. |
| **Inadequate authentication** | When an account or data are not equipped with secure enough authorisation and authentication techniques. | • **Mutual authentication:** Requires two parties in a communication and involves both parties entering some type of authentication before the communication is initiated; and<br>• **Out-of-band authentication:** Authentication technique via any other channel than which the one individual is connecting to;<br>• One-time password; and<br>• Shared secret. |
| **Lost device** | Any mobile device which might have been stolen or misplaced. Losing a device results in all security being out of the original owner's control, at that time. | • **Poison pill**: A trigger message is sent to the device to destroy the device or even specific data on the device.<br>• **Data encryption:** Does not leave the data or device destroyed, but it does make the data useless to the new owner, if he/she does not own the decryption data.<br>• Tracking of the device;<br>• Service provider validation; and<br>• Memory encryption. |
| **BlueBugging** | Bluetooth attack which allows the attacker to perform different kinds of activities on the victim's device. | • One-time password;<br>• Shared secret;<br>• Tracking of device;<br>• Data encryption; and<br>• Bluetooth authentication/authorisation across devices which connect via |
| **BlueJacking** | Bluetooth attack where | |

| **Threats and vulnerabilities** | **Defining terms** | **Solutions** |
|---|---|---|
| | unwanted messages are transferred across a Bluetooth connection. | Bluetooth. |
| **BlueSnarfing** | Bluetooth attack which gives the attacker access to unauthorized data on the victim’s device. | |

Sathyan and Sadasivan (2010:3-4) claim that it is important for a collaborative approach to have multiple solutions to deal with threats in an enterprise. Table 3-1 shows different solutions to different threats in order to have the highest feasible security for every possible scenario. Brown (2014:36) also claims that more is always better with security solutions, especially with new and unknown vulnerabilities constantly arising.

#### 3.2.4.4 Conclusion to multi-layered collaborative security infrastructure

Sathyan and Sadasivan (2010:4) mention that different enterprises exist and that the domain (type) of enterprise can change the level of security needed. They state that the most critical domain is the banking/financial domain and thus has its own predefined security policies that need to be adhered to. This can apply to the health domain, as well as many other domains, it all depends on the risk and consequences of data getting in the wrong hands. In conclusion to their work, they summarise the security ecosystem of an enterprise in Figure 3-8 (Sathyan & Sadasivan, 2010:5). This figure shows that Enterprise mobile security needs to include a range of different security aspects and factors.

## 3.3 Summary to security literature

This section serves as conclusion to both literature reviews, namely Software Development Methodologies (refer to Section 2) and Security (refer to Section 3). In conclusion to the literature reviews it can be noted that in both normal SDMs and MADMs, security is barely mentioned. However, frameworks exist that can be followed with a MADM to create a secure EMA. Full security cannot be introduced only by creating an application that is secure though, and security should be balanced between enterprise system infrastructure and application control (which involves the development of the secure application). Thus, it is necessary to have secure applications and policies to be followed by employees.

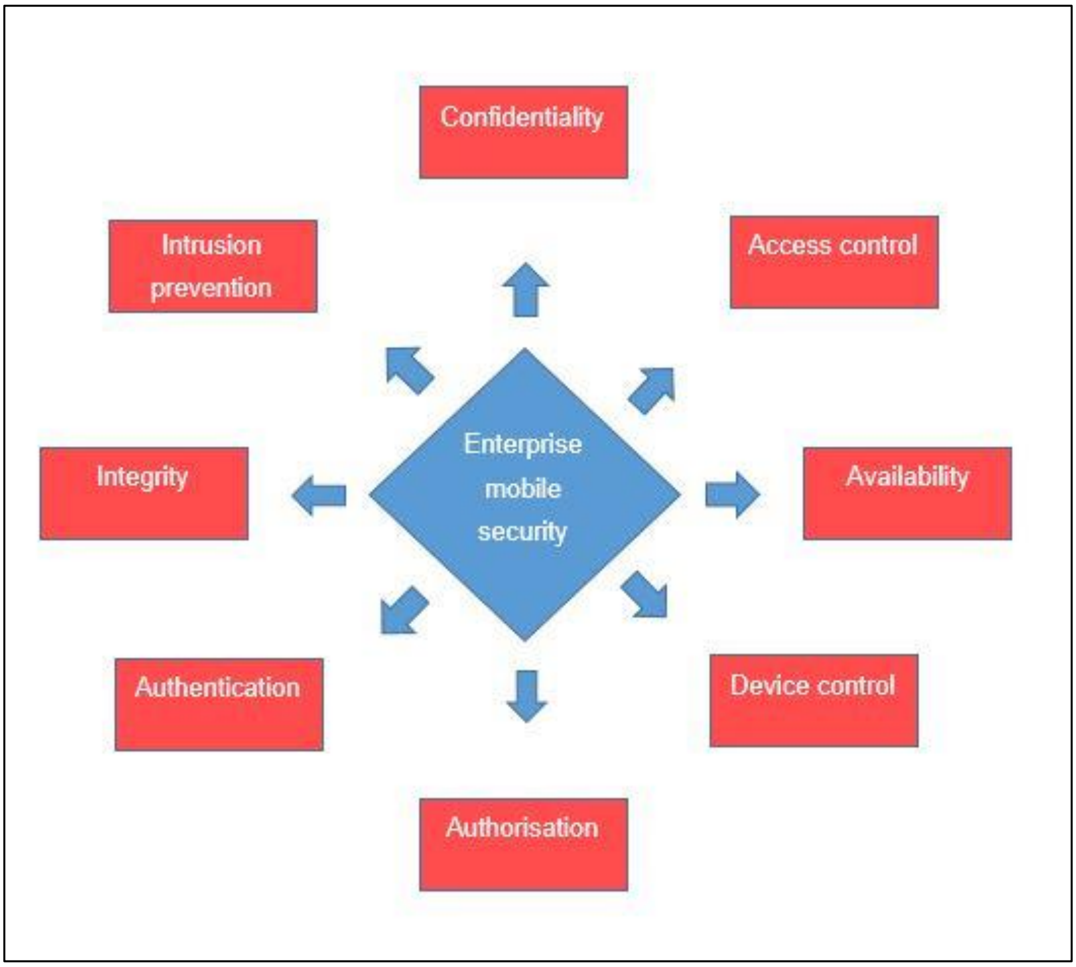


**Figure 3-8: Enterprise security ecosystem**

In Section 3.2 different perspectives on enterprise mobile security were discussed. These four perspectives or methods of implementing security all differ, but none of them give an encompassing security solution. It can hardly ever be said that security has fully been implemented, thus it is important to have as many different security perspectives as needed. When implementing security, different precautions should be taken whilst balancing security and user access/acceptance.

As seen in various literature sources (refer to Section 2.3.3), security is an element which does not make much of an appearance in various agile SDMs. In order to investigate this further, case studies were conducted at several companies where data were gathered via interviews. The following chapters elaborate more on this. The next chapter presents the research methodology for this study.

# CHAPTER 4 RESEARCH METHODOLOGY

## 4.1 Introduction

This chapter elaborates on the research methodology used during the study, as well as a discussion regarding the different methods of data collection and analysis used during the study. Figure 4-1 shows a representation of the chapter.

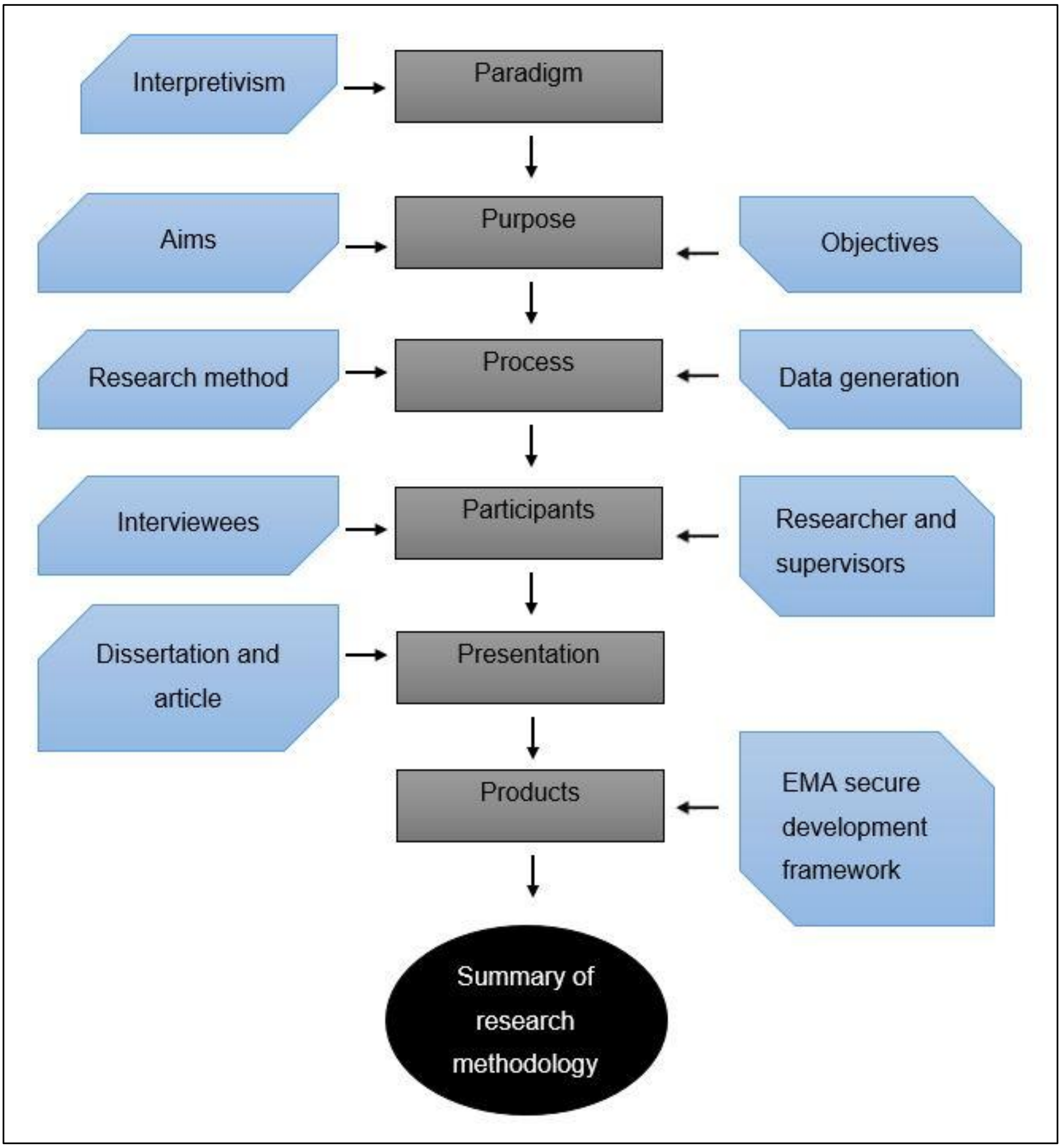


**Figure 4-1: Chapter 4 representation**

Research methodologies can be seen as guidelines for conducting research. Each methodology follows its own epistemology, which can be described as being the philosophy of knowledge or how knowledge is gathered (Krauss, 2005:758). Krauss (2005:759) and Dudovskiy (2011) assert that a research methodology can be seen as three separate aspects working together, namely:

- **Epistemology**, which is how the researcher comes to know the knowledge;
- **Ontology**, which is the philosophy or nature of reality, what the researcher comes to know; and
- **Methodology**, which is the particular set of methods followed, according to specific principles, to obtain the knowledge.

Figure 4-2 presents a depiction of how methodologies work. The three aspects work together to form one final product which can be seen as the research methodology.

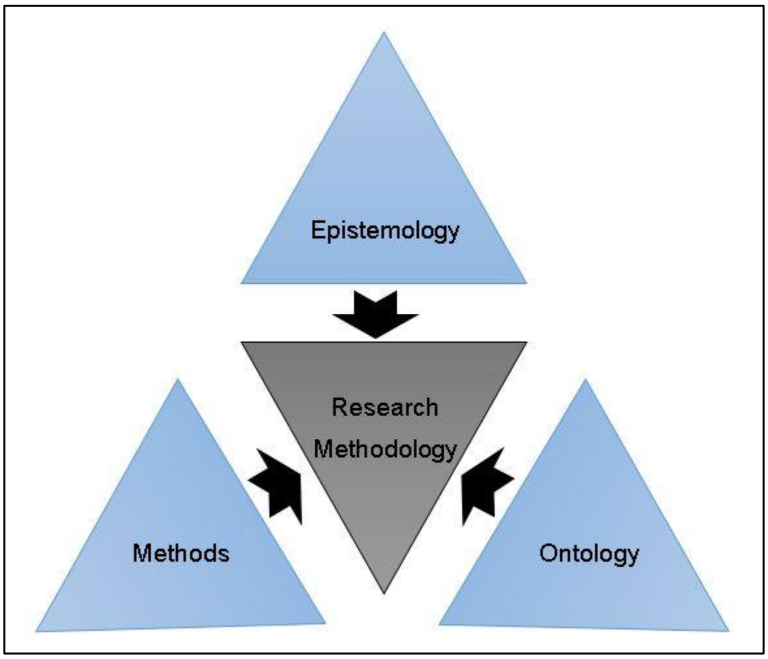


**Figure 4-2: Depiction of research methodologies (Adapted from Krauss,2005:758-759 and Dudovskiy, 2011)**

The research methodology for this study is discussed following the six P's of research as Oates (2006:11) established them. The six P's are (Oates, 2006:11-13):

- **Paradigm:** A model or a shared perspective about the study;
- **Purpose:** The topic at hand and reason for conducting the study;
- **Process:** The activities undertaken during the study as well as the sequence thereof;
- **Participants:** All the persons directly involved in the research;
- **Presentation:** The way in which the research is presented and explained to others; and
- **Products:** The contributions to the field, outcomes of the research.

These elements will be discussed in more detail as the chapter progresses. The first element discussed will be the paradigm.

## 4.2 Paradigm

There are different paradigms for conducting research; they will be discussed before elaborating on the one used in this study.

Positivism can be regarded as a scientific method of research (Creswell, 2013:7; Oates, 2006:283), which is commonly carried out in a quantitative manner. Positivism is normally used to determine effects or outcomes, and to identify and assess the causes that influence the different outcomes. This is generally done via experiments or statistical analysis (Creswell, 2013:8).

In Interpretivism, researchers believe that individuals seek an understanding of the world around them as they produce their own personal meanings of experiences. This type of research is generally conducted in a qualitative manner (Creswell, 2013:8). Oates (2006:292) states that this type of research does not try to prove or refute a hypothesis. She agrees with Creswell (2013:8) in saying that it attempts to identify, explore and explain the different elements of a specific social setting.

The Social Critical research is a transformative paradigm (Creswell & Clark, 2013:9). In this type of research the researcher does not only try to understand (what) and explain (why) a phenomenon, but he/she also pursues the empowerment of various groups of people through change (Oates, 2006:297). Creswell (2013:9) agrees by stating that the research has an agenda to oppose some type of oppression (through economic, political or cultural authorities (Oates, 2006:296)) that might reform the lives of participants, the institutions that individuals work for or areas they live in and even the life of the researcher.

The research paradigm that underlines this study is interpretivism. Interpretivistic studies attempt to identify, discover and clarify (what, how and why) the different elements of a social setting and how they are related and co-dependent (Oates, 2006:292). The social setting in this study was an enterprise and the ‘what’, ‘how’ and ‘why’ are as follows:

- **What –**The security aspects and policies surrounding EMAs and the development thereof;
- **How –** The manner in which the enterprise develops EMAs and implement security during development; and
- **Why –** The reasons for implementing the specific practices and policies.

Interpretivism mainly uses qualitative data analysis. Lakshman *et al.* (2000:371) claim that qualitative research relates to any situation where the research variables are not obvious or when the number of participants is inadequate for statistical analysis. Oates (2006:266) agrees by stating that qualitative data include any data that is not numerical. The data are generally found in interviews, websites, models, different types of organisational documents and through case studies. In this study qualitative data were collected while conducting case studies.

In this particular study case studies were done by conducting interviews at different enterprises (case studies and interviews will be discussed in Sections 4.4.1.1 and 4.4.2.1 respectively). The data analysis was done in a qualitative manner by means of theme analysis and cross-case analysis.

Looking at the 'what', 'how' and 'why' regarding the elements of security in EMAs, the purpose comes to light. The purpose is the next P on the list (refer to Section 4.1).

## 4.3 Purpose

The research was initiated with a problem statement, which read as follows:

*"Based on the initial investigation by the author, little is known about the security in the development of EMAs".*

Along with this statement, research on SDMs and MADMs reviewed showed little to no signs of guidelines provided for security implementation. This gave lead to the research title and the reason for the study.

*"An investigation of the security aspects addressed during the development of enterprise mobile applications."*

As seen in the title, the purpose of the research is to acquire knowledge of whether enterprises that use mobile device architectures have adequate security measures in place regarding information assets and processes when developing mobile applications. Thus the reason for interpretivism becomes clear, seeing that a social setting is studied along with different associated elements.

As mentioned in Chapter 1, the aim of this study is to investigate the development EMAs, together with any security challenges that might occur (during or related to their development). The objectives, needed to reach the aim, are also repeated from Chapter 1 for recollection purposes:

- To investigate existing MADMs;
- To conduct a literature review on different information security practices and policies in enterprises;
- To determine the different information security aspects and policies used in existing SDMs and MADMs;

- To investigate development practices of EMAs in different enterprises as well as how and why they address general information security in the development;
- To use the information obtained in this study to suggest improvements for existing development practices; and
- To develop a framework to guide developers regarding different security aspects that could be included during the development of EMAs.

A sequence of activities was carried out to satisfy these objectives. The process in which these activities were completed is discussed as the next on the list of P's (refer to Section 4.1).

## 4.4 Process

This section elaborates on the sequence of activities carried out to complete the study. Figure 4-3 presents a model of the research process used in this study. The sections following present a description of the elements depicted in the model, focused on in this chapter.

The literature review and topics reviewed was mentioned in Chapter 1 along with the research question. The focus of this section is on the method of research, the data generation methods and data analysis methods used; these methods will be discussed next.

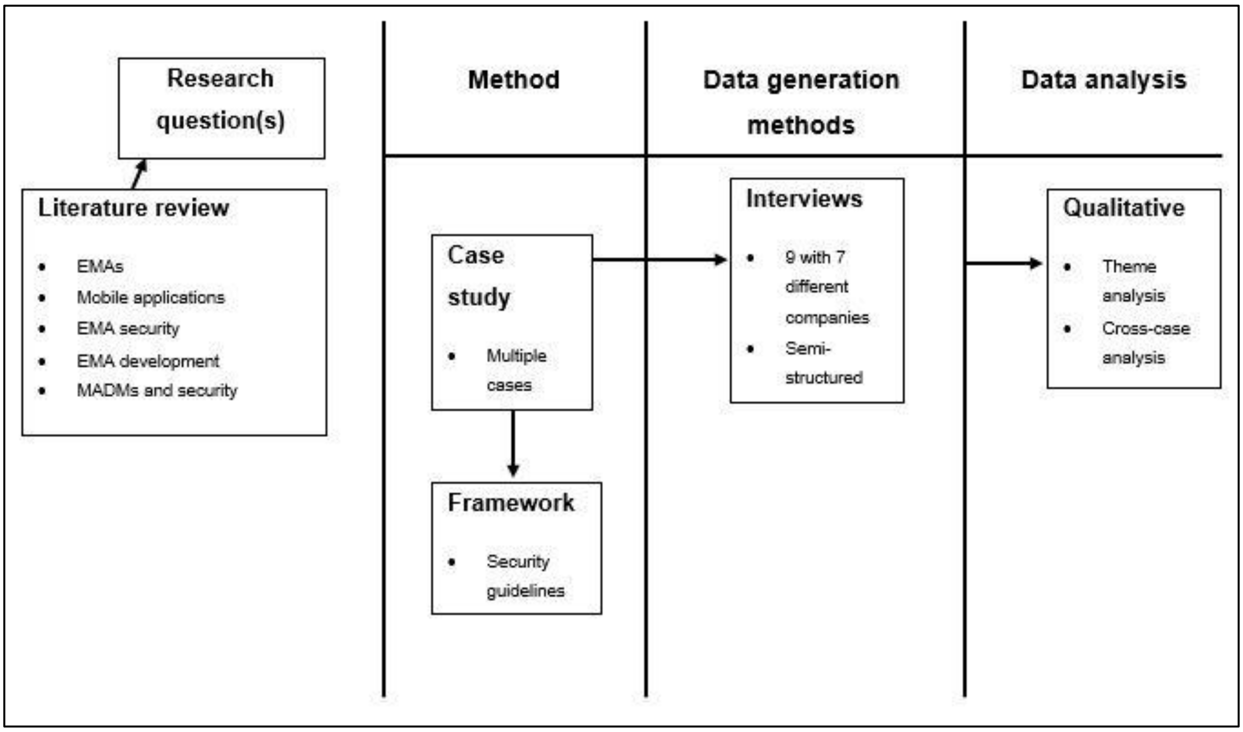


**Figure 4-3: Representation of the research process (Adapted from Oates, 2006:33)**

### 4.4.1 Research method/strategy

The research strategy used was a case study, combined with interviews as a data generation technique.

#### 4.4.1.1 Case study

According to Oates (2006:35), a case study is a strategy that focuses on one specific instance of the investigation. Examples of the instance being investigated are: enterprises, departments of an organisation, a system, a specific recorded discussion, a person or persons of interest to the study and even a decision of some sort. Yin (2009:17) defines a case study as the illumination of a decision or decisions. The clarification of these decisions will enlighten the researcher as to why they were taken, how they were implemented and which result was acquired. These decisions mentioned can result from individuals, processes, enterprises, programs, institutions as well as any type of event.

In this particular study, the instances or decisions are the security aspects addressed during, and around, the development of EMAs. According to Yin (2009:53), when a study such as this contains more than one case, it is categorised as a multiple-case case study. Seven separate case studies were conducted in this study, each focusing on a different enterprise to investigate different security policies, frameworks, practices and implementations set in place. According to Noor (2008:1602), a case study is not intended as a study of a whole organisation, but instead to focus on a specific area or issue. The data found in literature and the data acquired from the case studies were used to create a framework to improve current security practices for the development of mobile applications.

Yin (2009:10) and Rowley (2002:17) claim that in the instance of a case study, the result should be an answer to the questions ‘why’ and ‘how’. The multiple cases in this study will investigate the ‘what’ (along with ‘how’) different policies, practices and implementations the enterprises use with regard to development and security, and also ‘why’ they use these. The ‘why’ and ‘how’ will be revealed via data analysis of the interview data of each case.

Siggelkow (2007:20) states that the size of the sample does matter, but if the sample is small and the cases are strong, the size of the sample starts to matter less. Noor (2008:1602-1603) asserts that case studies are especially beneficial, and can add abundance of information, when it comes to understanding a particular situation in great depth. As mentioned, an in-depth interview was done in each case to acquire data on each enterprise along with a follow up discussion/call or email. Where possible, the premises and offices of the enterprises were visited to obtain a more comprehensive insight into what they do and how they do it. The next section will elaborate on how data were generated via interviews.

### 4.4.2 Data generation

As mentioned briefly in Section 4.4.1, the data generation technique used was a series of semi-structured interviews.

#### 4.4.2.1 Interviews

Oates (2006:186) defines an interview as a conversation with an unspoken purpose, with a set of underlying assumptions that do not generally apply to normal conversations. Rowley (2014:16) claims that interviews work better in situations where the total amount of participants is limited. A lot of information can be obtained from interviews with a small number of participants, while questionnaires have to be completed if a larger number of people are questioned in order to get the correct of information.

Oates (2006:187) states that interviews are best suited for situations where:

- Detailed information has to be obtained;
- Complex or open-ended questions need answers;
- Emotions, experiences or feelings, which cannot be perceived from a questionnaire, need to be explored; and
- Sensitive or privileged information need to be asked, and participants might feel more at ease with a human being than a pre-defined list of questions.

All of these situations were present in this study, and thus interviews were best suited.

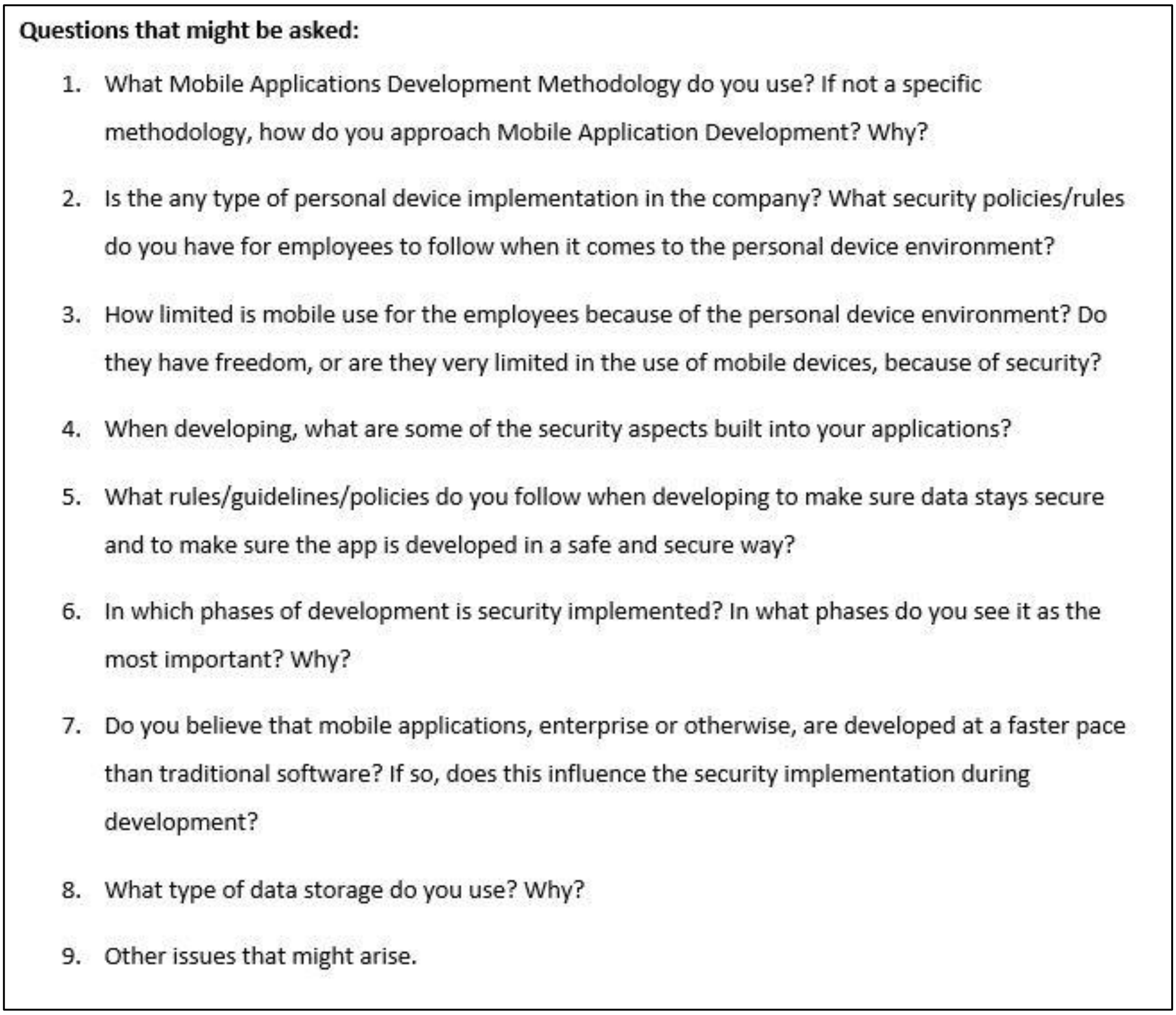
Creswell (2013:190) claims that the researcher conducts face-to-face, telephone or focus group interviews in qualitative interviews. These interviews are unstructured or semi-structured and open-ended. In this study semi-structured, face-to-face interviews were done, with the exception of three telephonic interviews using the Skype application.

In this particular study, nine interviews were done with seven different companies (purposive and theory-/criteria-based sampling). Two of the companies provided an additional person to help with data collection. Different criteria were used to select these organisations, they are listed below:

- Organisations of different sizes were selected. Two organisations were small (1 to 50 employees), two were medium (up to 1000 employees), two were large (up to 5000 employees) and one was very large (anything above 5000 employees). This categorisation is self-defined since the Small and Medium Enterprises (SME) and Small, Medium and Micro Enterprises (SMME) categorisations did not coincide well with the sizes of the different organisations interviewed. According to the National Small Business Act (102 of 1995), see Appendix A, the SME and SMME categorisations only look at small and medium type organisations and this study was conducted using organisations that can be classified as large and even very large. It can also be seen in Appendix A that the highest number of employees that a medium organisation can have is 200; thereafter an organisation is categorised as large. One of the organisations interviewed in this study had 50 000 employees, and as such cannot be categorised along with an organisation of 1000 employees.

- The business sectors of the organisations should differ to some extent.
- The organisation should be in the development of EMAs or be using EMAs in the organisation.
- The interviewees should be knowledgeable about the software development process as well as some security policies and practices.

Background reviews were done on the different organisations after before approaching them to ensure that they could provide sufficient knowledge on the different topics for this study. Dates were agreed on; the interviews were all conducted between the 5th of July 2016 and the 22nd of August 2016.



**Questions that might be asked:**

1. What Mobile Applications Development Methodology do you use? If not a specific methodology, how do you approach Mobile Application Development? Why?
2. Is the any type of personal device implementation in the company? What security policies/rules do you have for employees to follow when it comes to the personal device environment?
3. How limited is mobile use for the employees because of the personal device environment? Do they have freedom, or are they very limited in the use of mobile devices, because of security?
4. When developing, what are some of the security aspects built into your applications?
5. What rules/guidelines/policies do you follow when developing to make sure data stays secure and to make sure the app is developed in a safe and secure way?
6. In which phases of development is security implemented? In what phases do you see it as the most important? Why?
7. Do you believe that mobile applications, enterprise or otherwise, are developed at a faster pace than traditional software? If so, does this influence the security implementation during development?
8. What type of data storage do you use? Why?
9. Other issues that might arise.

**Figure 4-4: Questions asked during semi-structured interviews**

Six of the interviews were conducted in person, while the other three interviews were conducted over Skype. The interviews conducted in person took places within the offices on the organisation's premise, allowing better insight into the organisation's operations. Some of the interviewees were very busy, but agreed to do the interviews after hours via Skype. The interviews were recorded with consent and transcribed. The transcriptions total to 71 pages of data to be analysed. The data from the transcriptions were used to generate usable data during theme and cross-case analysis. The data from these two separate analysis techniques were used to develop the final framework (presented in Chapter 6). The questions asked during the interviews are depicted in Figure 4-4.

It should be mentioned that because the interviews were semi-structured, not all of the questions were asked to all the interviewees. In the event that other issues arose while conducting the interviews, these issues were discussed and noted. These questions were asked to satisfy at least a minimum of one research objective. Table 4-1 lists the different objectives that the questions addressed.

**Table 4-1: Objectives that are satisfied by questions**

| Question | Objective(s) |
|---|---|
| 1. What MAD Methodology do you use? If not a specific methodology, how do you approach MAD? Why? | Reveals more about development practices in the enterprise and how they approach it.<br>(Research objective: 1, 4) |
| 2. Is there any type of personal device implementation in the company? What security policies/rules do you have for employees to follow when it comes to the personal device environment? | Reveals more about different personal device policies; this links to security policies on data and data loss.<br>(Research objective: 2) |
| 3. How limited is mobile use for the employees because of the personal device environment? Do they have freedom, or are they very limited in the use of mobile devices, because of security? | Reveals how secure the devices are vs. how much freedom the employees have when using personal devices, this links directly to data security.<br>(Research objective: 2) |
| 4. When developing, what are some of the security aspects built into your applications? | Reveals security aspects and practices addressed during the development of EMAs or development in general.<br>(Research objective: 3, 4) |
| 5. What rules/guidelines/policies do you follow when developing to make sure data stays secure and to make sure the app is developed in a safe and secure way? | Reveals information about policies and implementations on data security during development and in general.<br>(Research objective: 2, 4) |
| 6. In which phases of development is security implemented? In what phases do you see it as the most important? Why? | Reveals the development practices employed in the enterprise, as well as the security implementation during different phases.<br>(Research objective: 1, 3, 4) |

| Question | Objective(s) |
|---|---|
| 7. Do you believe that mobile applications, enterprise or otherwise, are developed at a faster pace than traditional software? If so, does this influence the security implementation during development? | Reveals how the interviewees see the development of mobile applications or EMAs vs. the development of traditional software; this links to the security during the development of EMAs or mobile applications.<br>(Research objective: 1, 3, 4) |
| 8. What type of data storage do you use? Why? | Reveals information about data policies as well as security policies on data.<br>(Research objective: 3, 4) |

These questions and the objectives were used in the fulfilment of objective 5 and 6, as well as well as the completion of the study.

The next section will elaborate on the next P, participants (refer to Section 4.1).

### 4.5 Participants

The next P in the list is participants. In general, participants in research are:

- Everyone directly involved in the research, which include interviewees, respondents to questionnaires and anyone else supplying documents and information;
- The researcher;
- Colleagues of the researcher;
- Academics who read or review the paper, as well as students that might learn from it; and
- Anyone who might use or benefit from the research.

In this particular study, the participants are:

- The researcher;
- Researching supervisors; and
- Interviewees from selected organisations:
  - **Case 1:** *Person 1* has an M.Sc in Computer and Electronic Engineering degree, an Honours B.Sc in Computer Science and Information Systems, as well as an Honours B.Sc in Computer Science and Computer Engineering. He acts as one of the Directors of the company and as a Technology Engineer. *Person 2* has a B.Sc in Computer Science and acts as a Director and as a Product Developer in the company.

- **Case 2:** This person has an Honours B.Sc Information Technology degree and acts as a Senior Web and Application Developer.
- **Case 3:** This person has a B.Sc Computer Science and Business Accounting degree and acts as the owner of the company (development house).
- **Case 4:** *Person 1* has a degree, as well as an MBA. He has 29 years of experience and works in the financial crime control and fraud section of the company. *Person 2* has a B.Sc Computer Science degree, as well as an MBA and acts as a Lead Security Architect in the company.
- **Case 5:** This person is Microsoft Certified, as well as a certified Scrum Master. He is a Lead Developer in the company, working on web and mobile applications, as well as line of business software.
- **Case 6:** This person has an Honours B.Sc in Computer Science and Information Technology degree and acts as a Software Developer (doing work in security as well) within the company.
- **Case 7:** This person has a degree in Economics and Informatics, as well as an Honours B.Sc in Computer Science and Information Systems degree. He is a consultant at the company where he has been for four years, and does system integration and business process automation.

The following section will discuss the next P, being presentation (refer to Section 4.1).

## 4.6 Presentation

The study is presented in dissertation format and contributes a new security framework (presented in Chapter 6). An article, titled: *Security implementations during the development of enterprise mobile applications,* was also written and submitted to the Information and Computer Security Journal.

The study was presented orally at a student conference at the North West University where the study was completed. This conference was conducted on University grounds (NWU Potchefstroom Campus) on the 11th of November 2016.

The following section will discuss the next P, products (refer to Section 4.1).

## 4.7 Products

The final P is the product. As mentioned in the objectives in Section 4.3, the study will conclude with the design and creation of a secure development framework for EMAs. This framework will include different policies, guidelines, standards and rules to be followed when developing EMAs in order to result in secure EMAs.

## 4.8 Summary

In this chapter the method of research, focusing on the interpretive paradigm, along with the purpose of the study was discussed. It elaborated on data collection methods, looked at case studies and interviews, as well as data analysis methods such as theme analysis and cross-case analysis, all being conducted in a qualitative manner. The chapter concluded with an explanation of the different participants in the study and the various products that originated from the research.

The next chapter presents the data collection and analysis for this study.

# CHAPTER 5 DATA COLLECTION AND ANALYSIS

## 5.1 Introduction

This chapter discusses the way in which data were collected, as well as the methods that were used to do data analysis. As mentioned in Chapter 4, the data were collected via semi-structured interviews and data analysis was done via qualitative methods. The methods used were theme analysis and cross-case analysis. The results and the interpretations of the study will be represented in this chapter. Figure 5-1 shows a representation of the chapter.

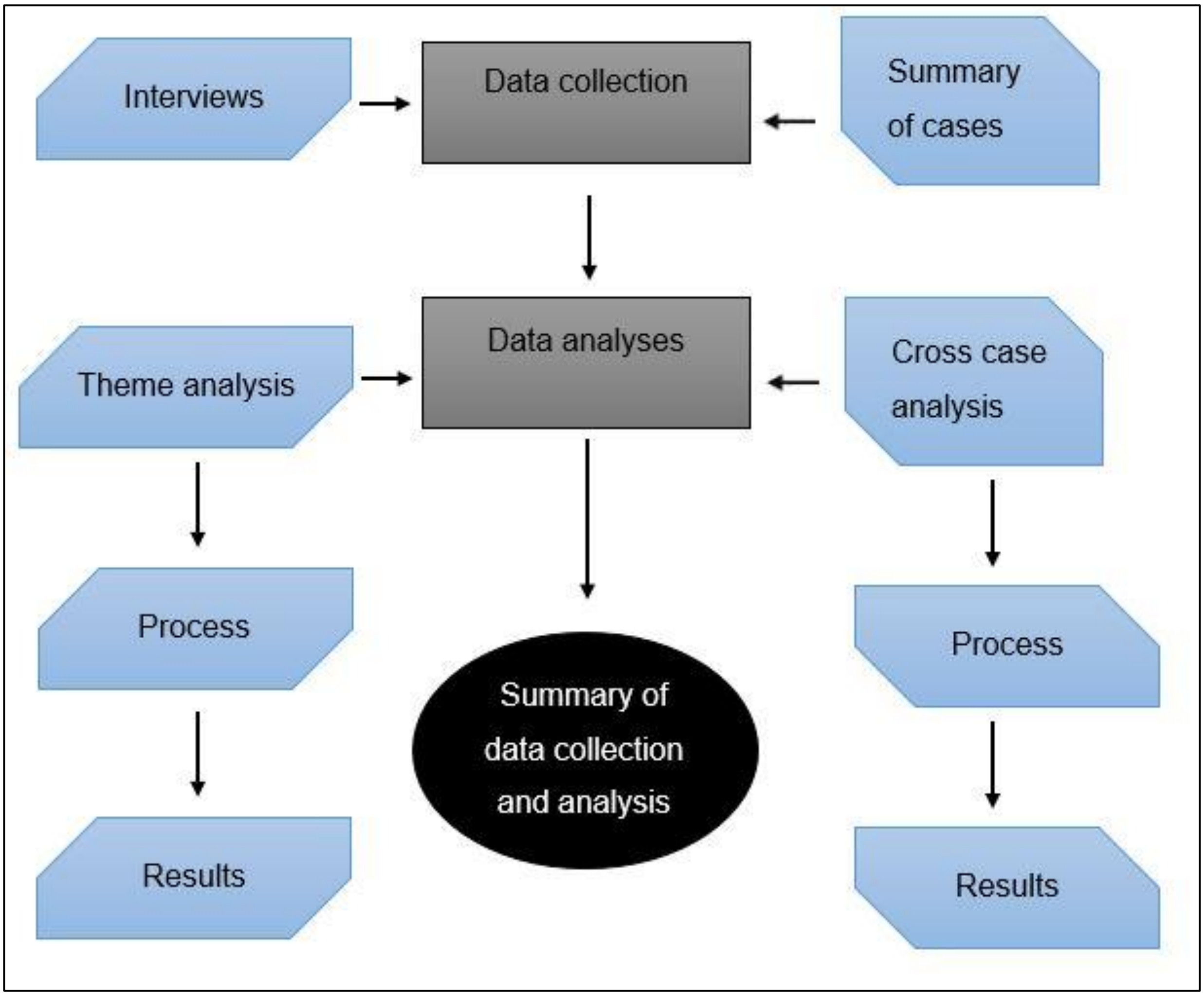


**Figure 5-1: Chapter 5 representation**

## 5.2 Data collection

As mentioned in Section 4.4.4.1, interviews were done at seven different companies. This section will elaborate on the different companies that were interviewed and how the interviews were conducted. The semi-structured interviews were scheduled with persons in each company that were knowledgeable on their software development processes. The interview agenda, sent to each of the interviewees before the interviews, included a list of possible questions that they might expect during the interview. This agenda helped the interviewees prepare beforehand, which made the interviews less time consuming. Figure 4-4 depicts the list of questions.

Each question relates to the objectives of this study (refer to Section 4.3). There was also opportunity to address issues that arose from the discussions. Each of the cases will be presented in the following sections.

### 5.2.1 Case 1

This is a small, growing, company consisting of three people that are doing national and international work in security and open source technology projects for different companies. The company started in 2014 with two freelance developers. They are currently busy with an International Organization for Standardization (ISO) standard of their own and have been working on license plate recognition and other systems working with Radio Frequency Identification (RFID), Ultra High Radio Frequency Identification (UHRFID) and Near Field Communication (NFC). They do not use EMAs themselves, but have developed such applications for their clients in the past and have experience in the field.

### 5.2.2 Case 2

This is a large agricultural company with a focus on grain products and other agricultural products, and the servicing thereof. The company recently started doing mobile application development to help with their own and their clients' needs. Although they only started with the mobile development aspects recently, they have been doing development for longer and have been one of the leading agricultural companies in South Africa for a very long time. The applications that they develop are used mostly by their clients for use in the field, as well as for different market research purposes. They use these applications on-site as well, but they act as *ad hoc* applications for different processes and are not used as software themselves that improve business processes.

### 5.2.3 Case 3

This organisation provides IT solutions to different types of companies in the form of mobile applications and other traditional software. They are a small development house, with only thirteen employees. They have done work in many different areas including: financial, accounting, payroll, IT and others. Being a development house, they do not use EMAs inside their own business, but have experience in developing them for clients.

### 5.2.4 Case 4

This is a very large organisation; one of the top six largest financial institutions in South-Africa, that does business internationally as well. The organisation has been in South-Africa for 154 years and started to build the franchise in the rest of Africa in the 1990s. The development that they do is focussed on themselves (using EMAs to improve their own business processes) and to improve the everyday lives of their clients. As a bank, they are largely focused on security and will always take their client's data and the safety thereof as top priority.

### 5.2.5 Case 5

This is a medium development organisation which provides solutions to different customers around South-Africa. They are a large development house and do development for a diverse range of businesses. They have background in many areas, from development for non-profit organisations to higher priority work for financial institutions. They are customer centric and will ensure that the customer is secure and happy. They have experience in the development of EMAs, but do not have the use for EMAs inside their own business. Although they do not use EMAs, they have started implementing the use of personal devices.

### 5.2.6 Case 6

This is a small to medium development organisation with just over 50 employees. They help to solve sophisticated IT problems for different financial institutions. The organisation work with different software and mobile aspects of development and because they work in the financial sector, they always ensure the utmost security and client well-being. They have a type of personal device implementation for their own business and business processes, but do not make use of EMAs. They do however have experience in the development thereof and since they are doing work exclusively for financial institutions, they have a large focus on security and the safety of data.

### 5.2.7 Case 7

This company is a leading African corporate and investment bank and the investment arm of a larger financial group. They do work for their clients in terms of financial advice, funding, trading, corporate banking and principal investing solutions. They also have a development team who develops solutions for these different parts of their business. The company has a presence all around Africa, with branches in the UK, India, China and the Middle East. They have a type of personal device implementation used to help with business processes, but being a bank, they make sure that their data stays as secure as possible. The development in EMAs that the company do is mostly done for their clients.

Table 5-1 provides a summary of the different cases.

Table 5-1: Summary of different cases in the study

| Case | Organisation size | Business sector | Uses EMAs | Develops EMAs |
|---|---|---|---|---|
| **Case 1** | Small (3 employees) | Information technology and systems | No | Yes |
| **Case 2** | Large (1001 – 5000 employees) | Agricultural | No | Yes |
| **Case 3** | Small (13 employees) | Information systems (Development house) | No | Yes |
| **Case 4** | Very large (50 000 employees) | Financial | Yes | Yes |
| **Case 5** | Medium (201 – 500 employees) | Information systems (development house) | No | Yes |
| **Case 6** | Small – medium (Just above 50 employees) | Financial and information systems | No | Yes |
| **Case 7** | Large (1001 – 5000 employees) | Financial | Yes | Yes |

Having done a background summary of each company interviewed for the study, further analysis can now be done into their business and business processes.

## 5.3 Analysis

The full analysis process involved the transcription of the interview conversations, theme analysis of the data and a cross-case analysis between the different companies. This will be explained in more detail in the following section.

### 5.3.1 Theme analysis

Theme analysis or inductive content analysis (Elo & Kyngäs, 2008:109), is a process of analysis which includes the following steps: open coding, creating categories and abstraction.

Open coding is the first step of theme analysis. It consists of reading through the material (interview transcriptions in this case) and making notes and headings which describe certain topics and aspects that arise. The material is read through as many times as needed to make sufficient headings and notes (Elo & Kyngäs, 2008:109). Thereafter, the headings and notes are used to generate categories (themes).

After the open coding has been done, the lists of different themes are grouped into different upper headings. This grouping is done to reduce the number of similar themes. These themes are not just headings of similar data though; the themes are used to categorise data that "belong" to the group. As such, two sets of data under the same heading can be different but belong to the same group, because it can be used to interpret the same thing. This helps to describe different phenomena and to increase the understanding thereof (Elo & Kyngäs, 2008:111).

Abstraction is the final step and can be seen as formulating a general description of the research topic, through means of themes. The themes, generated in the second step, are now split into main and subthemes. The subthemes are different related events that create one main theme. The main themes are several themes that make up parts of the research topic when put together (Elo & Kyngäs, 2008:111).

The process followed in this study is described in the next section.

### 5.3.2 The process

Firstly, the interview recordings (which were done with permission from the interviewees) were transcribed to have it in a textual format. Afterwards, the transcriptions were analysed by attentively reading through them and coding each piece of information that seemed important at the time with relation to the research topics. This step was iterated a number of times to ensure that all the important information was coded correctly.

These codes were used to compile an Excel spreadsheet where the codes were categorised into different themes, along with the data that belong to each theme. This spreadsheet was analysed and a second, more summarised spreadsheet containing the themes and the data from the first one was produced. After this a final spreadsheet was created where the themes were categorised even further by adding sub-themes to the main ones. This made the themes more manageable for the theme analysis. These themes were then used to do a theme analysis which is displayed in Section 5.3.3. After completion of the theme analysis, the cross-case analysis was done to obtain propositions that were used to create the framework. These propositions are given in Section 5.3.4.

### 5.3.3 Themes used for categorisation

As mentioned in Section 5.3.1, different themes were used to categorise the data obtained from interviews. This section will list the different themes (as well as their sub-themes) and give a short description of each.

- **Type of work –** This theme was used to categorise the types of work that the company does. This theme was elaborated on in Section 5.2 and will be referred to again in the theme analysis in Section 5.3.4. It comprises two sub-themes: specific and coding.
- **Type of client** – This theme was used to categorise data on the specific client(s) that the company caters for.
- **SDM** – This theme was used to categorise all data on development methodologies that the company uses. It contains sub-themes such as: agile, security, planning, testing and mobile versus traditional.
- **Practices –** This theme was used to categorise different practices that the company uses to do different activities. It was mostly focused on security and personal device practices; thus the sub-themes are: security and BYOD.
- **Storage –** This theme was used to categorise data on different storage practices. The sub-themes are: physical, cloud and hybrid.
- **Security –** This theme was used to categorise data on security and security practices. The sub-themes are: specific, development and data.

Now that the themes have been explained, the analysis is presented.

### 5.3.4 Theme analysis

This section displays the results from the theme analysis for each case separately. For each case, the themes are used to categorise the different data on the case. This section presents a summary of the company with regard to the different themes that were identified in the theme analysis.

#### 5.3.4.1 Case 1

This was the first case where two different people were interviewed. P1 or P2 is used before any quote to display which person made the statement.

**Type of work**

The work that the company do mainly pertains to the server-side, hardware and software, as well as the security aspects thereof. The work that the company is currently busy with focuses on RFID, UHRFID and NFC. They use these types of technology in parallel with their development and other projects. They focus mostly on security aspects of the work that they do, but security always depends on the project they are busy with. The different languages that they use to develop are: C++ and Java, where mobile development is done with Android, which is also Java-based.

### Type of client

Most of the company's clients are long term clients and have been with the company since they started. They also work with international companies. The company is still very small and relatively new, thus they are still busy building a client base. The clients that they mostly do work for are clients in need of technology solutions in the area of hardware, software and security.

### SDM

This company does not use any formal SDM since they are small, but mention that they do follow agile methods of development (P1: "We try to do more agile. We are not yet at the point that we can use a formal methodology, we are too small"). They do development in iterative cycles or sprints (no mention was made of length of sprints). Being a company specialising in security, they make sure to always develop in a secure manner and that security is well addressed throughout the development process (P1: "Because our development is always in regards to security we implement security continuously"). No specific phases or area of development should be considered as more or less important when it comes to security; it should depend on the application, client and data. They develop in containerised environments and use container technology to ensure that development is done quickly, efficiently and securely.

Another important part of development and security is planning, thus they have a very rigorous planning phase (P1: "We have quite a sturdy planning phase"). They involve their clients in the development as well, seeing that they do agile development. They further use Behaviour Driven Development (BDD), which is an adaptation of Test Driven Development, seeing that the client perspective and happiness is very important (P1: "There is a new flavour of it called Behaviour Driven Development"). User stories are a big implementation during their planning and help with the prioritisation of features and development. Along with the rigorous planning phase, they also welcome friendly, but harsh, criticisms seeing that peer reviews are important.

They use a test framework along with different methods of testing, depending on the application written. The test framework includes bug fixing and regression bug tests to ensure that introduced changes do not open up another bug or error (P1: "It is typically a regression bug that arises because of something that changed, but tests are done to pick up these bugs before production"). In testing, similar to planning, peer reviews are very important; peer testing is done to ensure that everyone has the same idea of the development being done. According to the interviewees, there is no difference in the development when looking at traditional software vs. mobile software. It does not depend on the type of development, but more on the specific software being developed. All software needs full attention to detail when being developed.

**Practices**

When it comes to general development practices, they use practices such as source control, version control (which help with testing and bug fixing) and containerised environments (along with Docker technology integrating with containers). The containers help with most development processes to ensure that everything runs smoothly and securely (P1: "The container is an abstracted environment for the process to run in with its necessities"). It increases security of data, decreases maintenance of IDE (Integrated Development Environment) setups (P1: "The IDE does not need constant maintenance"), increases security in general during development as well as the ease of development.

There is no data involved in the containers so no chance of data loss (P1: "There is no encapsulated data"). Data are supplied to the processes through the mounting of files and folders, and via network access. Containers do not take much processing power and the work is scalable (much of the same job can be done in parallel), thus it increases production and development speed. Something mentioned during the interview was that permissions are sequentially added to applications during development, when needed. It is bad practice to just add all permissions to applications, because it can open the application to outside malicious activity (P1: "No, we are very strict about that, we do not add permissions that are not necessary").

The company do not have any real personal device implementation since they are very small. They do use personal devices in the company, but not for business processes; thus it is not really seen as a BYOD implementation. They state that in South Africa it is not really done, except in very large companies that need the implementation (P2: "Not a very large policy in South Africa"). Companies that do use such an implementation do not always use it in its complete sense either, and sometimes only use selected aspects of the BYOD policy (P2: "Companies that do implement the policy do not always implement the entire policy and they might not have the necessities to implement it correctly"). Depending on the company and its needs, company devices for each employee might be a better choice.

**Storage**

The company uses a hybrid storage system, but most of the development data are stored on a physical Network Attached Storage (NAS) (P1: "A NAS is a Network Attached Storage, it is a computer with many hard drives which we push our data towards"). This system is used because of the sensitive nature of client data, and it reduces the chance of data loss and data leakage. For less sensitive data such as documentation and paperwork, they use cloud storage like Google Drive (P1: "Google Drive specifically for documentation and sending documents to clients").

**Security**

When looking at general security implementations, they use: Secure Sockets Layer (SSL) Certificates (as well as their own variation of this type of security (P1: "The standard we are developing is based on X 509")), HyperText Transfer Protocol Secure (HTTPS), encryption, digital certificates, digital signatures, containerised environments and the SDK and API being as secure as possible. They are involved with ISO standards and are assisting with the publication of new standards (P1: "We are also involved with ISO and do many tests to decide when to publish in the next version of ISO"). In development of any software, authentication and communication are always done properly along with the general implementations mentioned above (P1: "We will not publish something that does not have proper authentication. The communication also has to be of high quality").

The security of the application depends on the client, development requirements, existing systems (and their security), and use cases. Therefore, depending on what the project asks for, they will add the proper extra security measures. Since they work inside containers and that data are stored centrally in physical storage, data loss or leakage has a very small chance of occurring. Other measures are taken outside the containers to make sure data are secure where it resides (for instance encryption mentioned before).

Appendix B presents a table which contains the themes from which this case was analysed.

#### 5.3.4.2 Case 2

P1 is used before any quote to display the statements made by the person.

**Type of work**

The type of work that this company do is mainly focused on the agricultural (farming) sector. The applications that they develop are mostly for market research (of agricultural products) and applications to help workers in the field. They are fairly new to MAD, but in present times it integrates well with the work they do. They use languages such as C# and C#-based web development.

### Type of client

The company’s clients are generally farmers and people working in the agricultural sector. Their clients are not too concerned with security since the data that they use is not of very a sensitive nature.

### SDM

No mention is made of a formal SDM, but they do use agile methods of development. Iterative development is done with sprints lasting generally two weeks (but the length can differ when their schedule is tight) (P1: “At some point we were working in cycles of two weeks, but because deadlines were heavily pushed we had to shorten our cycles”). A time is declared for a sprint and they work towards the end of the sprint. Everything is planned before the sprint starts. At the end everything is analysed and new sprints are planned.

When developing, they use a framework that includes its own level of security; this is done with an application template (P1: “A skeleton app (template) will be used that already has security implemented”). The template that they use for new applications already has security built into it, thus the basic security is addressed in every application. Along with the template, they also add other levels of security such as authentication, encryption, tokens and HTTPS. They say that their clients are not too concerned with security since the data not being sensitive and internal applications are developed around Windows accounts for authentication. Data loss and leakage is reduced by not storing any data on the applications. This is elaborated on later in the Storage section (P1: “So we do not store data there that is of sensitive nature”).

They do not mention specific testing methods, but they do state that it is an important part of their development. They see mobile development as something new to the industry. Their perception is that people see applications as something that have to be developed as fast as possible to get the application published and running, before the competition steals the idea. They do not see it as a security risk in their line of work, because their data are not of a sensitive nature. They do mention that security should be addressed throughout development and no specific phase or area of development should be isolated (P1: “If you do not get it sorted out from the start, you will have to backtrack and fix problems”).

### Practices

Authentication, encryption, tokens and HTTPS were all mentioned in the previous section on SDMs; these are all practices that they implement to increase security in their applications as well as around the offices and during development. They use Windows account logins for employees; this helps with everyday authentication (P1: “On top of our services we also authorise with AD (Windows) accounts”). These authentications do not only go through

Microsoft security protocols, but also through their own security protocols that are added to the authentication process. One of these protocols state that timers are set on some of the login processes. Each process call included in the login process has a general time that it should take. These times get checked against the timers and if the times differ too much, the login is cancelled because malicious activity might be present. These process calls have HTTPS, when in need of high security, where other calls are done with HyperText Transfer Protocol (HTTP). The company is moving towards having HTTPS on everything.

Encryption is done with Advanced Encryption Standard (AES) 64; when data are very sensitive they go the extra mile with token encryption (P1: “The tokens are encrypted with AES”). They are trying to implement token encryption more and more to improve general security. When it comes to data security practices, time sensitivity is added to some data to ensure that it becomes useless at a certain time, if or when it gets lost. They are new to using MAD practices, thus some aspects are still being implemented, but they are moving in that direction.

### Storage

There is no real hybrid storage system for their offices. All storage is cloud-based (P1: “Everything is cloud-based”). They have databases which are stored on physical storage (P1: “We have a local database here”), but it is still accessible via Wi-Fi and wireless connections; this puts them in the same category as cloud storage. Data are not stored on the applications and thus the data are used and updated to and from the storage point (P1: “We fetch it from the servers”). Cloud storage is perfect for what they want to do, seeing that the applications have to be able to access data from anywhere, and that data are not of a sensitive nature.

### Security

Specific security practices are: Windows accounts for their own employees, sensitive data are kept off of applications and stored on-site, the sensitive data can only be accessed with the correct level of authentication, process calls are done with HTTP and HTTPS, tokens (as well as other data) are encrypted with AES 64 (P1: “Yes, HTTPS, and also the outside of application storage of data”) and applications are written with the correct permissions, so they cannot be used as backdoors by hackers (P1: “So by default the permissions will be off and we add them as we proceed”). The applications that have to access data have to go through an authentication process, the service (process) calls have to be accepted and the encrypted tokens have to be accepted, thus they feel they have enough security for accessing data via applications.

During development, the Software Development Kit (SDK ) that they use has its own levels of security; thus along with the other practices, they feel safe. They do not see security as a priority, seeing that the data that they work with is not very sensitive. They do see security as important throughout the development lifecycle. There is no specific phase or area where it should be more or less important. Although no phases are isolated as being more important when implementing security, they do mention that auditing and testing are both a very important part of development in general (P1: “We get testers and also auditors”).

Data are another area where security is at question. Since data are not stored on the applications, the only security that they have is the storage where data are at rest. Data are not of a sensitive nature, so not much effort is put into extra storage security. Data are encrypted, and the applications used to unencrypt the data have the encryption keys encoded into them (P1: “So the key, that is compiled into the code, is needed”), so very little (to no) chance of unencrypting the data without the applications and a valid login.

Appendix C presents a table which contains the themes from which this case was analysed.

#### 5.3.4.3 Case 3

P1 is used before any quote to display the statements made by the person.

**Type of work**

This company is a solution server; it serves solutions to many different types of clients. They have backgrounds in various sectors and enjoy doing new developments when given the opportunity. They do not see themselves as a development house but rather a mentorship house, looking at the human aspect of things and how people fit into development. The passion that a person has for development is very important to them, and they believe that the more the person learns, the better work he will be able to do; thus they try to teach via development. They do not mention any specific languages that they work in, but being a development house, they use the language necessary for the project.

**Type of client**

They have many different clients since the company is a provider of development solutions. Clients range from financial, accounting, logistics and labour companies, as well as smaller companies in need of problem solving.

**SDM**

The company is a very people-driven development house, who regards development as people developing themselves: the more the person develops, the more the person will be able to do for you. The people developing for the company are people with a passion for what they do, who do not just see it as a job; this results in high quality products (P1: “A person that really wants to develop versus a guy that has a degree and thinks he is this big asset”).

No mention was made of a specific formal SDM, but they do use different agile methods of development. In their perspective, agile is used as a guideline for flexibility and adaptability. They feel that a formal rule system is not necessary if the development is meant to be flexible and adaptable (P1: “You are flexible, constantly on a feedback cycle that you have to adapt to. So how can agile really be a set of rules?”). They try to give good, regular feedback to clients to ensure that the product is what the client wants and of the best quality. They work in iterative cycles, or sprints, to be able to give regular feedback and to be as flexible as possible.

They have specific development principles, including separation of concerns, intent and no duplication, as well as different best practices they follow while developing. The best practices are mentioned to come from XP (refer to Section 2.2.2.1). They do not really look at security during development, except if it is really necessary and the client needs or wants it. When security is considered, they do perceive it as important throughout the development lifecycle; no specific element should be isolated as more or less important (P1: “I think it is important continuously. I do not believe in isolating any element”).

They do not do exclusively apply MAD, but see mobile and traditional software development as the same. The only difference is in terms of what the client wants and needs, not how the development practices differ. They are of the opinion that the methodology should not change just because the type of software changes, it depends on the software. They believe that the happier the customer, the better for the company. They consider speed as a factor with mobile application development, but also with traditional software. However, they mention that speed should not leave openings for errors. Quality is the most important aspect of software.

**Practices**

They use a Microsoft environment and framework along with their own development practices and principles. Best practices along with their 'separation of concerns', 'intent' and 'no duplication' policies also help them to develop efficiently. They see development (and any type of job they do or thing they work with) as a tool to help people improve themselves, and thus the work that they do is a practice to improve on people and on themselves (P1: “By improving the person, it can have an exponential impact”). Different security practices that they incorporate are SSL certificates and secure server login and authentication. For the company, security is more of a “do as needed” practice (P1: “Until security is really a problem, we do not really focus on it”).

**Storage**

Storage depends on the data being stored, but because of the nature of their company, cloud storage makes more sense (P1: “Practically cloud just makes more sense to me”). It provides easier access and less trouble. They use a secure server for data storage, depending on the sensitivity of the data. It also uses wireless technology for the most part, since this is regarded as more practical. They do have a type of a hybrid storage system, referring to both physical and cloud storage; all storage is connected wirelessly. They use two different types of cloud storage; physical storage that requires wireless connections to act as an internal cloud as well as a cloud system on the Internet.

**Security**

As mentioned in the SDM section, the company see security as a “do as needed” practice. Depending on the specific project, basic security practices are important. They have not really worked on applications that require high security, but they do write all their applications with basic security: authentication, secure data policies, SLL certificates and additional security measures if the data are of a more sensitive nature (P1: “We will do the basics such as having a secure server and we have our SSL certificates, etc.”). It all depends on the clients and their requirements: if they need security, it will be added. In their opinion, when needed and if the client requests, security is important throughout the development lifecycle. Looking at data security, data storage is a big thing when working with mobile development, but they feel that storage of data and the type of encryption depends on the client and what they would need for the data (P1: “What encryption is used and where data is stored, etc.”).

Appendix D presents a table which contains the themes from which this case was analysed.

#### 5.3.4.4 Case 4

This was the second case where two people were interviewed. P1 and P2 are used before any quote to display which person stated what.

**Type of work**

The company is a financial institution (bank) that do development for themselves as well as for their clients (internal and external facing), whilst focusing on data safety and security. The applications are of a banking nature, helping clients to make payments and utilise their funds in the safest and most secure way. Client data safety is important, but it is also important that the clients feel safe and know that their data are safe (P1: “To give extra peace of mind not only to us, but our clients”). The general languages that they use for development are: Angular JS, Angular Node JS, Java-based languages and at times HTML5.

**Type of client**

As a bank, the company has clients both in South Africa and around the world. There is no limit to the type of client that they may have: large companies, international companies and individuals. Their client base is also very large, with millions of transactions per day and tens of thousands of employees.

**SDM**

The company is a very large organisation and thus they use formal SDMs; rather than using a specific SDM, they use characteristics of different formal agile SDMs (P1: "Using different characteristics from different agile methodologies"; P2: "So the move to agile is important in present times with a formal agile development framework"). They see their development as traditional way, but incorporating agile methods.

They work in intervals and sprints of one to two weeks. To them an important part of the agile development is being very client-based, and ensuring that the client is happy. Thus, development is flexible and adaptable to be able to respond to the client's requirements when they change at any time. In bigger development projects, a hybrid approach to development (using both characteristics from traditional development and agile development) might be taken. They are of the opinion that the waterfall approach is too slow in general, and thus use an agile approach (P2: "look at the waterfall approach in our current environment, it is just too slow"). They also believe that using development tools are very important in moving toward agile development.

A very important aspect in all applications that they develop is authentication. During development the highest possible API and SDK updates are used to ensure that the basic security implementations are added to the applications. They mention that coding standards are used to ensure that development is done securely and that everyone is on the same page. Development is always done securely, whether they are developing an internal or external facing application (P1: "Whether it is an app for us to use here, or an app for clients, requirements are met to the fullest and to make sure security is also met to the extent the app needs it"). Security is planned during the planning phase to ensure that security is addressed in each phase and that nothing is left out. The company considers its clients and the safety of their data as the most important aspect; as such, security is regarded as important during the entire development lifecycle and should be assessed and addressed also during the production stage.

Data are kept secure during development, by adding control access levels to data and ensuring that only the people that need the data can access it. A data protection framework (P2: "Part of your protection of data, your POPI Act, Protection of Personal Information") is set up to ensure that data remains secure when accessed. The steps in this framework:

1. Get ownership of data;
2. Identify and classify data;

3. Integrity of data;
4. Controls; and
5. Authenticate, audit and login.

The company tries to keep data as static as possible. When the data needs to be accessed, it is accessed and consumed from a central point to ensure that data is not leaked. If data needs to be moved around, the correct steps are taken and the data are encrypted with a high level of encryption to ensure that the data will remain inaccessible by someone from the outside, in the event of a data leakage.

They see planning as a very important step of development which includes the planning of security (as mentioned before) as well as threat and risk assessment. The basic process includes a planning session to identify what needs to happen, combined with a risk/threat analysis. After this, security is built up as the lifecycle continues and extra security measures are added as needed. Security is assessed and addressed in each development phase, thus security is always important no matter which phase of development you are currently in (P1: “Every step or phase in development security is assessed and dealt with accordingly”; P2: “Security must be present at all phases”).

Testing, risk management and peer reviews are also planned during the planning phase, seeing that these elements are very important. Data security is important to them, and thus a very rigorous testing process is conducted on all applications to remove all errors and bugs (P1: “A good testing framework is used to make sure everything works as they should”; P2: “To do testing and code reviews during the development lifecycle”). Different levels of testing are used, and if a specific test is not satisfactory in some way, another test is developed and run. In the end, all tests have to be passed for the development lifecycle to continue.

They develop software other than mobile software, but mention that mobile and traditional software do not have to have different effort values. It does not depend on the type of software that is written, but more on the specific software and what it should be able to do. Mobile applications, in some cases, might be released quicker than traditional software. The reason for this is that mobile applications in general are developed toward their features, and getting a baseline application (with fewer features, adding more features as the lifecycle progresses) is much quicker than developing a whole piece of traditional software. In the end, it all depends on the specific piece of software.

**Practices**

Their security practices include biometrics security systems such as voice recognition (for employees and clients), fingerprint scanning, retina scanning and more. Different methods of security are used to help recognise and identify different clients. Some of these methods are voice recognition, username and password authentication, Fido (cryptography and extra authentication), "Big Data" along with machine learning techniques, and more. For employees sending different types of data to each other, they use email services like WorxMail, which is setup in a secure way to ensure that all data are monitored and that no malicious activity is sent or received via email (P1: "We use a mail software called WorxMail, which is configured to a standard of security we deem fit for the bank, which is a very high level of course"). This ensures that data sent via email is transmitted safely and reaches its intended destination without external tampering while in transit. There are also other practices implemented to make sure that data stays secure (P1: "Different protocols and policies are in place to make sure data stays secure"; P2: "We have basic controls and we have the extended controls").

Device and data auditing are conducted to ensure that the correct people are accountable for different reasons. Device locking and remote wipes are also set in place to ensure that company data can stay out of danger when devices are lost or data leaked. Employees are trusted not to move data around too much; this is monitored to ensure that data stays in secure environments. The company believes that not only data and people have to be kept secure and need rules and guidelines. As such, containerised areas are used to ensure that applications run in the most secure way possible, even after development and security has been built into the applications (P2: "You will need to come through via a fully containerised application on your endpoint that will communicate securely").

Looking at personal device implementations, the company has BYOD as well as Bring Your Own Network (BYON) implementations to an extent (P1: "We do use policies like BYOD as well as BYON"). The work related devices or networks that use enterprise data are setup securely and frameworks are used to help keep sensitive and enterprise data safe and secure. For the remote possibility that data can be leaked or lost, policies are put in place to assess the risk of the data lost (data loss prevention) and rebuild the data. They also use remote wiping for devices that are lost or stolen. They have different ways of keeping data secure when using personal devices versus using company devices; some of these policies include data segregation, multi-factor authentication, containerisation and MDM. The BYOD implementation (P1: "Employees can bring their own devices from home and use them here for different reasons") is not complete and the company is still moving towards a more complete implementation. They recognise that BYOD implementation is necessary in a company of their size, because personal device usage can be unavoidable at times (P2: "It is almost impossible to stop. So this means that we have to respond in ensuring those facilities are used securely"). Since devices have to be set up and configured correctly, they cannot be defended more; security depends on the people using the data and devices. Thus policies and guidelines have to be given to the employees that use the devices. There is tension between how much freedom employees are given in boundaries of trust versus how much control can be exercised. This depends on the type of data and the type of applications, as well as the people using the applications and accessing the data (P1: "We make sure that data being used to develop is kept at different data access levels, depending on the sensitivity of the data"; P2: "Depending on the sensitivity of the data as well as the use-case, we will come up with a set of controls").

**Storage**

Both physical and cloud-based storage are used, and thus a hybrid system is put into place. Data segregation is applied to different data types and classifications (the sensitivity of the specific piece of data) to ensure that the correct data are kept properly secure. Enterprise data and personal data are also segregated. They try to leverage more modern types of storage, but data sensitivity remains a problem; until this is addressed they will always have a hybrid system. Physical data storage is used for sensitive data as well as data that do not have to move around a lot. Less sensitive data that moves around more are kept in cloud-based storage or some type of volatile storage. Saying this, it is more of a guideline than a rule, it all depends on the specific piece of data (P1: "This is not a rule saying that all sensitive data is kept on physical and all non-sensitive data is kept on the cloud, it can be seen more as a guideline, because exceptions are always there").

Data encryption is also very important for any data in transit, no matter what sensitivity. Data are stored globally, with a lot of different divisions being established. For this reason, data are well monitored to ensure that it remains secure regardless of it being in transit or at rest in all storage areas. Data are not hosted in any country where the data are in danger or where the country is opposed to the Protection of Personal Information (POPI) Act. There is a drive towards cloud, but because certain data can be too sensitive to be stored on the cloud service, there is a privacy office in charge of permitting what data are allowed on the cloud and what data are not (P2: "We have a privacy office that looks at what we are permitted").

**Security**

Specific security practices that are mentioned are as follows: different biometric mechanisms, the three A's (authentication, authorisation and administration), security controls regarding auditing and login, data protection and zonal protection, and Fido for extra security regarding cryptography and authentication. Regardless of what applications are developed, authentication is important; in a bank all security is important (P1: "Security in overall, in a bank is important").

During development in general, testing and code reviews are also of utmost importance and helps to ensure that everything is developed efficiently and secure. Looking at development security, threat and risk levels, authentication and data access can all differ depending on the application or users. The SDK ensures that security is implemented to the fullest extent to which the application requires it, regardless of whether the application is internal or external (P1: "Security is also met to the extent the app needs it"). This is done after a risk/threat assessment is done to ensure that all the additional security measures for which the SDK does not account are added.

The application as well as its security is put through rigorous testing processes. Data has to stay secure during development; this is done by assigning access levels to different data and employees, as well as ensuring that data stays as static as possible and only moves around when really necessary. Data are encrypted with RSA encryption and token encryption is used with certain data, depending on the sensitivity (P1: "When data is very sensitive we might even use extra token security, where a token is sent with the data, encrypted with RSA"; P2: "So if it is sensitive it must be encrypted"). One of the policies put into place for employees is that data (enterprise data) should not leave the company premise. Different measures are taken when data are lost, including remote device access (which includes alteration of lost devices as well as wipes) and data recreation. Other data security policies include digital rights management, containerisation, multi-factor authentication and more. The POPI Act assists with this.

Private and public key policies are put into place when it comes to encryption and the movement of data to help with the need-to-know basis of certain information. The data protection framework is put into place to ensure that only employees that have access to certain information can access that information and no one else (P2: "So we have a data protection framework where data is consumed"; P2: "There is a need-to-know basis, so if you do not need it, then you will not have it"). As mentioned before, non-sensitive data that move around via emails are also monitored and kept safe by the WorxMail setup. The Storage section states that different classifications of data are stored on different types of storage media to help with security. Different storage methods have different levels of security and wireless storage generally has a lower level of security. In terms of international storage, data cannot be kept in one place and has to move around. As a result, other security measures are taken, for instance data encryption and data validation (P1: "Data is always monitored"; P1: "If I send data overseas you can make sure that the right amount of key data (encryption key) is sent and received").

Another measure to ensure that data that are not at rest or in the location it should be in is kept secure is a time-sensitivity implementation to make data only available for a certain amount of time after being accessed before it expires, and has to be accessed again. All in all different protocols and policies are in place to ensure that data stays secure no matter where it is or where it is moving to. The bank considers the different sensitivity levels of data and act upon the data depending on this sensitivity, thus different policies are put into place for different situations (P2: “We do not treat all objects (data) with the same importance or apply the same security policy to it”).

Appendixes E and F presents tables which contains the themes from which this case was analysed.

#### 5.3.4.5 Case 5

P1 is used before any quote to display the statements made by the person.

**Type of work**

This company is a medium sized development house, doing several kinds of work for different companies. Specific types of development mentioned include web applications, mobile applications and line of business software. Client relations are very important to the company, thus they ensure that software is up to standard and meeting client requirements (P1: “I am also responsible for client relationships, making sure the architecture is right and all that”). Being a development house, they use different types of languages for different jobs.

**Type of client**

They build applications for different types of clients, but mention banks as some of the larger companies that they work for and non-profit organisations as some of the smaller companies that they do work for.

**SDM**

They mention formal SDMs such as Scrum and Kanban (P1: “For certain projects we use Kanban and some projects we use Scrum”) for use in some projects, but in general they just use different characteristics of agile development (P1: “In general we always use the agile methodology”). They work in sprints of one to two weeks, depending on the priority of the project. They see client relations as a priority, and as such work as adaptable and flexible as possible. During development, security is important from the start and should always be a priority (P1: “I think right from the word go, building the architecture with security in mind”). Depending on the client and the specific project the security requirements can differ, but it should always be seen as priority when needed.

**Practices**

Different general practices that they use during development include Outlook, Sharepoint and other Microsoft Suite products, Trello, Jira and other tools with mobile counterparts. Outgoing communications can be accessed via mobile devices. There is a type of BYOD setup (P1: “You can bring your own device, and I think we can set them up with VPNs and so on”), but this is not highly secured since the data are not regarded as extra sensitive. The data are run through a secure Virtual Private Network (VPN) and other minor security details are configured for the BYOD setup. MDM is not implemented, and as such no application restrictions or data restrictions on personal devices are in place (P1: “With us it is more open, I can just download any mail application”). They mentioned that larger companies that they work for do implement such policies.

**Storage**

The company used to only have physical storage servers, but now run a hybrid system with both cloud and physical storage (P1: “We use both physical and cloud”). They like the idea of having a one-stop-shop for information and data, thus having data stored centrally and using it from there. This increases general security around data storage since data are kept static and not moved around a lot. Each employee receives one terabyte of data for personal data storage and sharing, and security on that storage depends on the individuals, in addition to the normal security measures taken by the company for the cloud service (P1: “Drive on the cloud where they can store and share data with other employees”). They mainly use cloud storage, but physical storage is used in some cases.

**Security**

Since they work for many different companies, they see a lot of different implementations where security is not always top priority. For instance, when the client is a bank, security is always well prioritised above the rest. Common precautions are in place for applications developed to prevent things like hackers, data extractions, Structured Query Language (SQL) injections, cross site scripting, forgery and other common malicious activities (P1: “So pretty much using Microsoft frameworks that will prevent those kinds of things, Cross Site Scripting, Forgery and all those things”).

Along with the prevention of malicious activities, they believe that it is important to think of and act upon security throughout the whole development life cycle from the start if security is a requirement in the application. Authentication and authorisation are important and are used when working with systems and data. Data encryption on the other hand is more of an afterthought and it depends on the applications and the data (P1: “Encryption I think it is more of an after-thought”). When transmitting data, it also depends on the data and the applications and the users that have access to the data, before thinking of encryption.

Appendix G presents a table which contains the themes from which this case was analysed.

#### 5.3.4.6 Case 6

P1 is used before any quote to display the statements made by the person.

**Type of work**

This organisation specialises in software development for banks and financial institutions and focuses on security aspects thereof. They are exposed to mobile application development in the form of EMAs.

**Type of client**

The organisation works for banks and financial institutions, so their main clients are financial institutions. This already puts a priority on security and client-based needs, because financial institutions are very client-based and mainly work with client data and funds.

**SDM**

They use agile and waterfall development methods, as well as different variations of these methods, depending on the specific project. No formal methodologies are used, but they do use different characteristics from agile and waterfall methods (P1: “Agile and waterfall basically, are the main ones that they use, and then of course there are some variations of those”). Applications are always developed in a secure way and protection against basic attacks is built into all applications; extra security measures are added when necessary. APIs used for development have security filtering to help with secure development. They mention that security should always be important in applications, but that they work with several companies and see that clients and some developers neglect to mention security and then want to add it at a later time (P1: “But a lot of them develop their apps and then do security later on”).

When it comes to mobile applications versus traditional software, many people rush their mobile applications, because it looks and feels faster. Since they work on bank systems and bank applications, the mobile features have to roll out concurrently with the web applications and other systems (P1: “Often the bank launches a new feature and then it needs to be ready on the web and the app at the same time”). Accordingly, the mobile application will take the same amount of time and no less effort than the computer software equivalent's features. Quality is very important.

**Practices**

Different coding practices are put in place, but the developer mentions that people tend to neglect those types of practices if they need to get something out as fast as possible. This is a common practice not only at their company, but all around the IT community (P1: “Coding practices are nice to have but many people do not do it”). They further ensure that people who work with extra sensitive information go on training courses to ensure that they can code securely and work with the data in a secure way, and that the data are also audited by external companies (P1: "Audited by an external company"). The company has a type of personal phone usage, but no mention is made of BYOD policies and implementations. All employees have company laptops that they work on.

**Storage**

They do not mention any specific details on storage, but mention that the company has a hybrid storage system. They have file sharing enabled for specific data that is not too sensitive in nature. When sensitive data are moved, encryption is added to the data to make it more secure (P1: "Physical servers, file shares and then there are a lot of them using cloud storage").

**Security**

Security aspects that they consider include authentication, single sign-on solutions, internal and external auditing and encryption. This is combined with the basic security aspects built into SDKs and APIs (P1: "There is security filtering on their APIs"). They believe that security is always important during development and that no specific area or phase should be singled out. Their business is focused on financial institutions; as such security is always important when looking at client funds (P1: "There are a lot of security stuff I do for banks and other financial institutions").

Appendix H presents a table which contains the themes from which this case was analysed.

#### 5.3.4.7 Case 7

P1 is used before any quote to display the statements made by the person.

**Type of work**

This company works in the financial sector (a bank) and the development that they do is mostly focused on internal facing applications for employees to use and banking applications for their clients. The different languages they use include, but are not limited to: Json, Javascript, C# and HTML5. The languages that they use depend on the type of application or software being developed.

**Type of client**

They have many different clients because they are a bank and cater to all different types of clients. There is no real limit to who a client can be. They also have international offices doing work with international clients and companies.

### SDM

In the past they used a waterfall method of development, but regard this as being too slow; they now use agile development (P1: “Agile development methodology”; “The problem that they have with the waterfall method is that is it just too slow”). They do not mention a specific formal SDM, but mention a few characteristics of an agile approach. They try to be as flexible as possible and work in iterative cycles (sprints) of generally two weeks. Since planning is very important, they have a daily stand-up meeting policy where they talk about the specific project and give an executive summary. Demonstrations are given to the client either weekly or bi-weekly, depending on the project, to present project progress and to give an indication of how the product will look. The clients’ input is very important, seeing that the bank is very client-based and clients and their data are priority to them.

They state the importance of segregating internal and external facing applications and recognise that they will differ in some ways (P1: “If it is an internal application, it is very basic”; “When it is externally used, I would say that developers focus on functionality”). Internal applications have lower security because the people using them have specific authorisation; that is sufficient for the data running via those applications. External facing applications have higher security since the data in banks belong to the customer, making the risk factor high.

It is mentioned that there is not a specific part of the development process where security should be regarded as more or less important, and that the importance of security depends on the different application types (P1: “I do not think there is a specific area”). The company sees development as a practice and therefore the type of development should not dictate a difference in the level of effort. All types of development should be considered equal, since security depends more on the type of software being developed (P1: “It really depends on the complexity of the application being developed”). Mobile and traditional development are regarded as different in the sense that traditional software is generally developed to an endpoint and then rolled out into production, where mobile applications are written to a baseline and features are added incrementally (P1: “Basics have to be in place for the first sprint and then functionality is added by iteration through more sprints”). The time may differ depending on the specific application, but the level of effort should differ.

### Practices

The company looks at two different types of development for which they have different practices: internal and external development. For internal development they do not consider too many practices since the applications will be on the network and facing internally; the applications are secured by company authentication (P1: “They feel that if you are authenticated on their network, they are happy with the security”). External facing applications or client facing applications involve more practices. Security teams need to assess risks and the audience that the application will be facing. Depending on the information gathered, different security protocols and practices are put into place for the applications (P1: “There is a whole security team that looks at technology”; “but it depends on the audience”).

Development tools are used to accelerate development and make it easier for developers. These tools are also used to help make development more secure, meaning that development is done in a more secure way and that the application is secured when development is completed. Employees are issued with company iPads and laptops which connect to the true company network where enterprise data are kept secure from outside networks; these devices are configured to connect securely, whilst the access to the network is managed and monitored closely. They do not have a BYOD implementation, but they have a guest network to which mobile devices can connect (P1: “They have a guest network that you can connect any device”). The implementation of BYOD is not a necessity for the company and implementing it incompletely could result in security problems. The access to the guest network is secured by a PIN code, which is not very secure. However, since sensitive files are segregated from the open network, it is near impossible to access files from the network. The guest network is used to access Internet, emails and specific applications.

### Storage

Physical storage is mostly used, because cloud storage is considered as a big risk for investment or any other banks (P1: “Cloud is obviously a large security risk for a bank”). Cloud systems are used, but they have private cloud initiatives where the cloud is not completely hosted on the Internet, but rather on the intranet (P1: “They talk more of a private cloud than a true cloud that is entirely on the Internet”). So it is cloud-based, but hosted locally. The private cloud can be seen as physical storage, but employees connect wirelessly.

### Security

Banks always have top-notch security and are very paranoid about data loss and their client's’ happiness (P1: “Investment banks, any bank really, is very paranoid when it comes to security”). Security configured laptops and iPads are issued to employees, with only have the necessary applications installed. Another level of security is the use of a DMZ, which basically puts the application in a secure area where it can work outside of the public domain. It puts the application in a specific part of the network so that it only has access there and nowhere else. A DMZ is a good security protocol, because even if the person gaining access has malicious intent, he/she only has access to that specific area of the network.

In general, security is fully configured in such a way that anyone who really wants to gain access to or break into sensitive data will have to make use of social engineering. Access is not just given to anyone; even employees are having difficulties to connect to something simple such as the VPN to their workstations (P1: “It is the biggest process just for me to connect to my workstation via the VPN”; “They guard the privileged areas very, very, very strictly”). The correct procedures and channels need to be followed to ensure that employees are granted access to the network. This involves having to sign legal documents, supported by a good justification.

During development data are kept safe by assigning access permissions to different developers; this is done by assigning accounts to individuals as an additional level of authentication and security (P1: “They have to create an account for you on the application itself or alternatively a service account can be created”). This account has different levels of security access; depending on the role, the account has more or less access to data and also other dependencies. Security is addressed throughout the entire development lifecycle; they do not see a specific area of development were security is more important (P1: “I do not think there is a specific area”). They have a guest network (mentioned before) that increases security for enterprise data by letting outside people connect to a specific portion of the network, away from sensitive data. Different access levels are given to different employees and the different networks all have authentication protection to ensure that only authorised personnel can connect. They have a private cloud system for some data, which acts as a local intranet that only the employees can access.

Appendix I presents a table which contains the themes from which this case was analysed.

These sections presented each case by listing the main themes that emerged from the interview data. The next section will elaborate on the cross-case analysis.

### 5.3.5 Cross-case analysis

This section provides different propositions derived from the information that was obtained from the theme analysis. These propositions will be used to create the framework that will guide developers in the implementation of security aspects during the development of EMAs and other mobile applications.

#### 5.3.5.1 Propositions

This section will provide an explanation of how the propositions were generated. The Seaman (1999:568) method was used as a baseline for the proposition generation, with the following steps:

1. The first case was reviewed and a list of propositions (short phrases describing a fact regarding the case) was compiled.

2. The second case was taken and another list of propositions was compiled before the two lists were compared to each other.
3. When a fact from the second list agreed with a fact from the first list, the proposition would stay as is, otherwise the proposition would be revised into a new proposition.
4. These steps were repeated for each case until all seven cases were compared to each other.
5. C1-C7 is used to display the case in which the fact is true.
6. P1-P2 is used to display which person in each case stated the fact.

The propositions are listed next in their respective categories: SDM, Practices, Storage and Security.

### 1. SDM

This section presents the propositions regarding SDMs.

**Proposition 1.1 (Revised)**

Formal SDMs are not generally used, but rather different characteristics of agile development (C1P1, C2P1, C3P1, C5P1, C6P1, C7P1). It all depends on the size of the project (C6P1). For larger companies it is important to move towards formal SDMs (C4P1); not necessarily a specific SDM, but rather a mixture of different methodologies (C4P1P2, C5P1).

**Proposition 1.2 (Revised)**

In present times the waterfall approach is too slow (C4P2, C7P1). Where very large projects are concerned, a hybrid approach between agile and traditional development might be pursued (C4P1P2).

**Proposition 1.3 (Revised)**

Development tools are a large part of agile development (C4P2, C7P1). These tools help with ease of development (C5P1, C7P1), as well as the security of the application (C7P1).

**Proposition 1.4**

Agile development should be as flexible and adaptable as possible (C3P1, C4P1, C5P1, C7P1).

**Proposition 1.5 (Revised)**

Agile development is done in iterations, called sprints (C1P1, C2P1, C3P1, C4P1, C5P1, C7P1). This helps with regular feedback to the clients (C3P1, C4P2, C5P1, C7P1).

**Proposition 1.6 (Revised)**

The length of sprints are generally one (C4P1, C5P1) to two weeks (C2P1, C4P1, C5P1, C7P1), but can differ depending on the workload (C2P1, C5P1, C7P1).

**Proposition 1.7 (Revised)**

Planning is a very important aspect (C1P1, C2P1, C4P1, C7P1) and is important to ensure that security is going to be addressed during the lifecycle (C4P1P2). Planning of each sprint is done separately (C2P1) and can be iterative as well.

**Proposition 1.8**

Clients and their thoughts on a project are very important in agile development (C1P1, C3P1, C4P1, C7P1).

**Proposition 1.9**

User stories are a large part of the planning phase (C1P1, C4P2).

**Proposition 1.10 (Revised)**

Peer reviews are good for any project (C1P1, C4P2), combined with friendly criticism (C1P1).

**Proposition 1.11 (Revised)**

Testing is important (C1P1, C2P1, C4P1P2) and good testing frameworks help to develop errorless and bug free applications (C1P1, C4P1). No test should be failed by any application it was meant for (C4P1).

**Proposition 1.12**

Peer testing can help to identify faulty issues that other tests did not find (C1P1, C4P2).

**Proposition 1.13 (Revised)**

Speed can be a factor with mobile development (C2P1, C3P1, C4P2, C6P1, C7P1), seeing that companies want to get applications out as fast as possible (C2P1). This can be conceived as a security risk (C2P1, C6P1), but should not leave openings for errors or security risks (C3P1, C6P1).

**Proposition 1.14 (Revised)**

There should not be any difference between mobile software and traditional software (C1P1, C3P1, C4P2, C6P1, C7P1), except in the way that they are developed (C1P1, C7P1) and what the client needs (C3). Both software types need the same amount of attention to detail (C1P1, C4P2, C6P1, C7P1). It all depends on the specific software (C1P1, C3P1, C4P2, C6P1, C7P1).

**Proposition 1.15**

Quality is very important in all development (C1P1, C3P1, C6P1).

**Proposition 1.16 (Revised)**

Auditing is an important step of development (C2P1, C4P2, C6P1), as well as knowing who to hold accountable for what (C4P2).

**Proposition 1.17**

Best practices help to improve product quality in software development (C3P1, C6P1).

**Proposition 1.18**

Planning practices can include aspects such as:

- Daily stand-up meetings (C7P1);
- Demonstrations (C7P1);
- User stories (C1P1, C4P2);
- Requirement discussions (C1P1);
- Sprint planning (C2P1);
- Security planning (C4P1);
- Threat and risk assessment (C4, C7P1); and
- Testing planning (C4P1).

The next section will focus on practices.

## 2. Practices

This section presents the propositions regarding practices.

**Proposition 2.1**

Source control and version control are good practices to take into account while developing software (C1P1).

**Proposition 2.2 (Revised)**

Small companies do not have any use for BYOD implementations and policies (C1P2). Very large companies do have a use for it, because it makes many things easier; the need for BYOD can become unavoidable (C4P1).

**Proposition 2.3 (Revised)**

Companies using BYOD implementations do not necessarily use the entire implementation, but rather only use parts thereof (C1P2, C4P2, C5P1, C6P1, C7P1). It can be unavoidable for some companies to adopt the policy (C4P1). Adopting the policy incompletely and in the incorrect way can result in security flaws (C7P1).

**Proposition 2.4 (Revised)**

Devices in a BYOD setup have to be made secure (C4P1P2, C5P1) to keep enterprise data safe, but the people also have to be given the correct guidelines and policies for using the devices (C4P2).

**Proposition 2.5**

BYOD device security depends on the type of data being used, thus freedom is given with regard to what data are being used (C4P2, C5P1).

**Proposition 2.6**

Instead of BYOD, many companies choose to use a company device policy instead where things can be managed and monitored more closely (C7P1);

**Proposition 2.7**

Different BYOD practices to keep devices and data safe:

- Secure setups (C4P1P2);
- Personal and enterprise data segregation (C4P1P2);
- Multi-factor authentication (C4P2);
- Containerisation (C4P2);
- Application restrictions (C5P1); and
- MDM (C4P2, C5P1).

The next section will focus on storage.

## 3. Storage

This section presents the propositions regarding storage.

**Proposition 3.1**

The type of data storage depends on the data being stored (C3P1, C4P1P2, C5P1).

**Proposition 3.2**

Data travelling via email should be monitored and kept secure while in transit (C4P1).

**Proposition 3.3 (Revised)**

Data storage is segregated between physical and cloud depending on data types and classifications (C4P2, C5P1), for example, personal versus enterprise (C5P1). This can be done by a privacy office (C4P2).

**Proposition 3.4 (Revised)**

Physical storage is good for sensitive data (C1P1, C4P1, C7P1) that do not move around a lot (C4P1) and decreases the chance of data loss (C1P1).

**Proposition 3.5**

Physical storage can be used in combination with wireless technology to create an internal cloud that is removed from the Internet; this increases ease of use (C3P1, C7P1).

**Proposition 3.6 (Revised)**

Data loss and data leakage can be decreased by not storing data on applications (C2P1, C4P2, C5P1), as well as using remote access to wipe or track lost devices (C4P2).

**Proposition 3.7 (Revised)**

Cloud storage is good for less sensitive data (C1P1, C2P1, C3P1, C4P1, C7P1), and is perfect for companies that do not work with highly sensitive data (it gives ease of access) (C2P1, C3P1, C7P1).

**Proposition 3.8 (Revised)**

Cloud storage (modern storage methods) are leveraged above other methods (C4P2, C5P1), but hybrid systems will be used until sensitivity is not a problem anymore (C4P2).

**Proposition 3.9**

Data are safer when it is static and used from a central point (C4P2, C5P1). Data should not be stored on devices (C2P1, C4P2, C5P1).

**Proposition 3.10**

Data stored internationally should not be stored in countries that make the data insecure because of different privacy related Acts and any other reason. (C4P2).

**Proposition 3.11**

Data have to remain secure during development (C4P1, C6P1, C7P1).

**Proposition 3.12 (Revised)**

Data should be kept as static as possible (C4P2, C5P1), but when data have to move around measures have to be taken (C4P2).

The next section will focus on security.

## 4. Security

This section presents the propositions regarding security.

**Proposition 4.1 (Revised)**

When security has to be implemented (C3P1, C5P1), it is important throughout the entire development lifecycle (C1P1, C2P1, C3P1, C4P1P2, C5P1, C6P1, C7P1). No phase should be isolated (C2P1, C3P1, C4P1P2, C6P1, C7P1) and security should be assessed and addressed during each phase (C4P1P2). Security is even important after development, in the production phase (C4P1).

**Proposition 4.2 (Revised)**

Security may differ when looking at internal applications versus external applications (C7P1), but it does not always (C4P2).

**Proposition 4.3 (Revised)**

Containerised environments can help to increase the speed and security of development (C1P1) and the security of applications and data used inside the environment (C4P2).

**Proposition 4.4**

In the development of any application, authentication should be done properly (C1P1, C4P1, C6P1).

**Proposition 4.5 (Revised)**

Security depends on the client (C1P1, C2P1, C4P1P2, C5P1, C7P1), data (C1P1P2, C2P1, C4), requirements (C1P1, C3P1), existing systems and use cases (C1P1). Thus security can be seen as a “do as needed” practice (C3P1), but basic security practices are always important (C3P1, C4P1, C5P1, C6P1) and security should always be done to the fullest extent that is required by the client or application (C4P1P2, C5P1, C6P1).

**Proposition 4.6**

Risk assessment is used to determine the level of security of an application (C4P1P2 C7P1).

**Proposition 4.7**

When a company works with data of little to no sensitivity, security is not always done to the fullest extent (C2P1).

**Proposition 4.8 (Revised)**

When working with banking clients, the client and data are the most important aspects, and thus security always has to be implemented to the fullest (C4P1P2, C5P1, C6P1, C7P1).

**Proposition 4.9 (Revised)**

Encryption is used to secure data wherever it may reside (C1P1, C2P1, C3P1, C4P1P2, C5P1) or in transit (C4P1P2, C5P1, C6P1) if needed (C5P1), and decryption keys should be kept in a safe place (C2P1).

**Proposition 4.10**

Systems should be made so secure that social engineering needs to take place in order to obtain the information needed to break into something (C7P1).

**Proposition 4.11**

Different security implementations that companies have:

- SSL certificates (C1P1, C3P1);
- HTTPS (C1P1, C2P1);
- Encryption (C1P1, C2P1, C3P1, C4P1P2, C5P1, C6P1);
- Tokens with extra encryption (C2P1, C4P1);
- Digital certificates (C1P1, C4P2);
- Digital signatures (C1P1, C4P2);
- Containerised environments (C1P1, C4P2) and DMZs (C7P1);
- Secured APIs and SDKs for secure development (C1P1, C2P1, C4P1, C6P1, C7P1);
- Authentication on applications (C1P1, C2P1, C4P1P2, C5P1, C6P1, C7P1)
- Authorisation (C4P2, C5P1, C7P1);
- Multi-factor authentication (C4P2);
- Certain data can be made time sensitive to decrease threat risk when data are leaked or lost (C2P1, C4P2);
- Being authenticated as an employee on company premises (Login accounts for employees or consultants) (C2P1, C3P1, C4P2, C6P1, C7P1);
- Access control on data (C2P1, C4P1P2, C7P1);
- Data are kept secure during development (C4P1, C6P1);
- Coding standards / Best practices (C4P1, C6P1);
- Application permissions are coded safely (C1P1, C2P1);
- Biometrics (voice recognition, fingerprint scanning, retina scanning) (C4P1);
- Fido (C4P1);
- Secure email service (WorxMail) (C4P1);
- Remote wiping, locking and accessing of devices (C4P2);
- Encryption key policies (public and private key systems) (C4P1);
- Data protection frameworks (C4P2);

- Microsoft protection framework (C5P1); and
- Securely configured devices (C7P1).

## 5.4 Summary

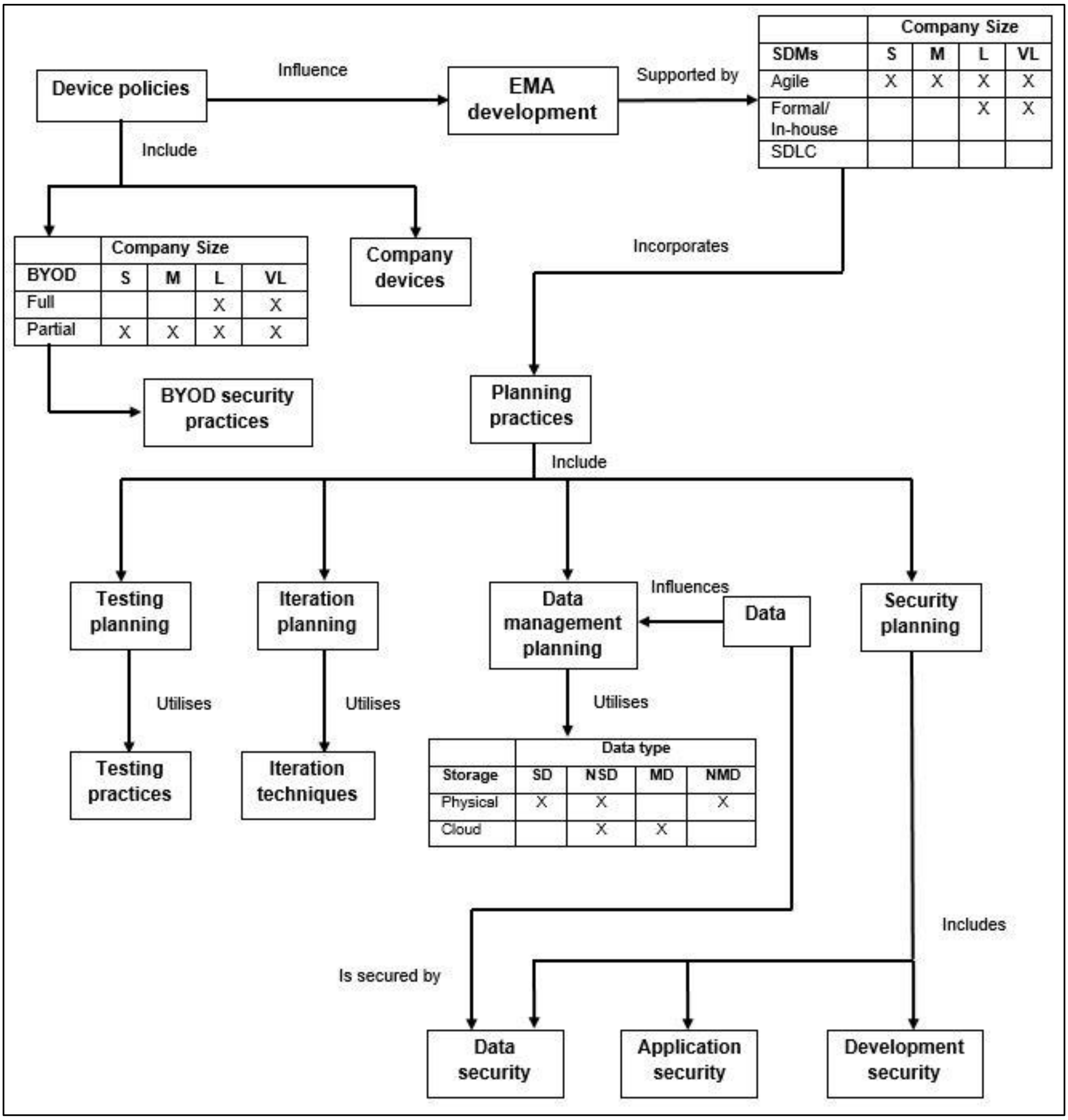
This chapter elaborated on the process in which data were collected and analysed. The data obtained from the theme analysis was presented after which it was used to make propositions in the cross-case analysis. These propositions focused on the themes *SDM*, *Practices*, *Storage* and *Security* that were used to construct the guiding framework.

The next chapter presents the final results, as a framework. It also presents a discussion of the results.

# CHAPTER 6 EMA SECURE DEVELOPMENT FRAMEWORK

## 6.1 Introduction

In this chapter the results obtained from the theme analysis and the cross-case analysis (propositions) are used to construct the EMA secure development framework. This framework is a product aimed at guiding developers in developing more secure EMAs or other mobile applications. It should be mentioned that it is only a guideline and derived from this study's results. The framework is presented in Figure 6-1.



|  | Company Size |  |  |  |
|---|---|---|---|---|
| SDMs | S | M | L | VL |
| Agile | X | X | X | X |
| Formal/ In-house |  |  | X | X |
| SDLC |  |  |  |  |

|  | Company Size |  |  |  |
|---|---|---|---|---|
| BYOD | S | M | L | VL |
| Full |  |  | X | X |
| Partial | X | X | X | X |

|  | Data type |  |  |  |
|---|---|---|---|---|
| Storage | SD | NSD | MD | NMD |
| Physical | X | X |  | X |
| Cloud |  | X | X |  |

Figure 6-1: EMA secure development framework

The framework starts with the *EMA development* node, which is the activity that guides the framework. Each node from there influences one or more other nodes, until all the nodes result in a comprehensive guideline to which developers can adhere to in order to increase security during EMA development. The results follow by presenting the node as well as an explanation thereof.

**1. EMA development**

Looking at EMA development in general, the following aspects should be kept in mind:

- Quality is always important, thus developers should adhere to coding standards and best practices;
- Speed is a factor when working on any type of mobile application, but quality is more important; speed should never reduce the integrity of the application;
- During development auditing is important to ensure that the correct people are accountable and the correct people operate the correct data; and
- Containerised environments can be used to increase development time, efficiency and security.

*EMA development* is influenced by *Device policies* and supported by *SDMs*.

**2. Device policies**

Device policies can pertain to two different implementations, although both may be implemented at the same time: *BYOD* and *Company devices*.

**2.1. BYOD**

The BYOD policy is dependent on the company size. Table 6-1 gives a summary of this.

**Table 6-1: BYOD policy dependent summary**

| | **Company size** | | | |
|---|---|---|---|---|
| **BYOD** | **S** | **M** | **L** | **VL** |
| **Full** | | | X | X |
| **Partial** | X | X | X | X |

When considering the implementations of a BYOD policy, two different types should be considered: full implementation or partial implementation. A full implementation is normally only done in very large companies (large companies may consider a full implementation if it is really needed), since it takes time and money to set up correctly. Any size company can do a partial implementation, but it is of utmost importance that a partial implementation should be done correctly and that every possible security risk is considered.

### 2.1.1. BYOD security practices

It should be mentioned that no matter which BYOD policy is implemented, it can be a security risk. The following security measures can be taken to increase security:

- Set up the devices and restrictions according to the data type and classification - sensitive data will have stricter device policies, whilst less sensitive data will give employees more freedom with devices;
- Ensure that personal data and enterprise data are segregated at all times;
- Implement multi-factor authentication as a good backup to ensure that data stays secure;
- Set up a containerised environment for devices to run securely; this will ensure that data on the devices are also more secure;
- Set up devices with application restrictions so that only certain applications may run on them (these applications can even be restricted to in-house application stores only); and
- Implement a MDM policy to ensure that the devices are monitored and managed correctly.

## 2.2. Company devices

An alternative or additional policy to BYOD is to provide all employees with company devices and setting them up so that only company data can run on them. This increases security and makes the risk of data leakage or loss much less, since device policies are more restrictive than personal device policies such as BYOD.

## 3. SDMs

As mentioned under the *EMA development* node, *SDMs* support the development. The SDM or type of SDM used also depends on the company size (similar to BYOD policy implementations). Table 6-2 shows a summary of the dependants.

It is clear from Table 6-2 that any company can make use of agile development, or different characteristics of the agile development methodology. Formal SDMs are normally only used by very large companies (it may also be considered by large companies if the need arises) because of the time and money required to implement it correctly ensure that all employees are familiar with the SDM. Large and very large companies may also use different formal SDMs and tailor them to their specific needs to create an in-house SDM that they use. The waterfall methodology (SDLC) is also mentioned in the summary, because it is still used. However, in present times it is regarded as too slow for the development needs of many different companies and thus, the movement toward agile development.

Table 6-2: SDM use dependent summary

| | Company size | | | |
|---|---|---|---|---|
| **SDMs** | **S** | **M** | **L** | **VL** |
| **Agile** | X | X | X | X |
| **Formal / In-house** | | | X | X |
| **SDLC (Waterfall)** | | | | |

When looking at agile development, there are a few aspects to take into mind:

- Development tools are a bonus when developing in an agile manner and help with development efficiency, speed and even in some cases, security;
- Development should be done in a flexible and adaptable way;
- Development should be done in an iterative manner and feedback should be regularly given to the client (this allows the client to change requirements if they change their mind about an aspect of the application or project); and
- In agile development it is always good to remember that the client comes first and that their thoughts and ideas are very important.

Agile SDMs and agile development in general incorporate different Planning practices.

### 4. Planning practices

Development starts with *Planning practices*. This is where different development choices are made and different aspects of development is planned. This node, along with its children nodes, is very important since planning is one of the most important things when it comes to development. If planning is not done rigorously, different problems might occur during the rest of development; thus having a good base to work from is always important.

The planning phase include four very important parts: Testing planning, Iteration planning, *Data management planning* and *Security planning*. These nodes and their children nodes will be discussed next.

### 4.1. Testing planning

Testing is a very important step when looking at development and especially security. Having a lack of the necessary tests can result in different security deficiencies and even the application not working as a whole. Testing utilises *Testing practices*, discussed next.

#### 4.1.1. Testing practices

When planning tests, the following practices can be kept in mind to make testing more efficient:

- Having a good testing framework that suits the company needs is important; in some cases different frameworks can be used for different projects to suit the specific needs of the project;
- Testers from outside the company can be incorporated to get an outside view which might result in a bigger variety of tests being run in the end;
- No test should be failed; when a test is setup and failed by an application, the deficiency should be fixed and another test should be set up to ensure that it works correctly; and
- Peer reviewing and code testing is a good practice to implement because the peers could realise something a specific tester did not.

The next planning practice is *Iteration planning*.

### 4.2. Iteration planning

Agile development is done in an iterative manner. These different iterations have to be planned in advance to ensure that the correct activities are completed during each iteration. This planning and the phase itself utilise different techniques, discussed next.

#### 4.2.1. Iteration techniques

Depending on the SDM, the characteristic taken from the iterations may differ in length, activities, techniques and terminology. Selected iteration techniques are discussed next:

- Daily stand-up meetings help to ensure that everyone is on the same page during the development phase and during each iteration;
- In some cases prototypes or demonstrations can be presented to the clients to ensure that they know where development is at present and to provide them with the opportunity to make requirement changes if they want to;
- User stories/cards are used to prioritise different development activities and to ensure that too much work is not taken on at the same time;
- Requirement discussions, with the clients present, are done to make sure everyone knows what exactly has to happen (this technique is a variation of user stories); and

- Threat and risk assessment can be done in this planning phase to ensure that that security is taken into consideration before any development are started (this activity is normally done in the data management phase and the security planning to ensure that data and development is secure and classified correctly; performing these assessments several times cannot hurt the project).

The next planning phase is *Data management planning*.

### 4.3. Data management planning

Data management is influenced by *Data*. Depending on the type of data and classification thereof data can be stored differently. In very large (and in some cases large) companies, privacy offices may be set up to help with data classification and placement. Data stored internationally should not be stored in countries where the data are automatically made insecure by privacy related Acts or other laws. Data stored on applications and roaming devices can lose integrity and be made insecure, thus data should be kept as static as possible and consumed from the point where it resides. If data needs to move around, the correct security measures should be taken (encryption is always a good consideration). Data management utilises different storage methods to ensure that all data resides where it is at its safest.

#### 4.3.1. Storage

There are two storage methods generally used by companies: physical storage and cloud-based storage. Each storage method can be used depending on the type and classification of data. Table 6-3 summarises the different types and their respective storage places.

**Table 6-3: Data classification storage summary**

| | Data classification | | | |
|---|---|---|---|---|
| **Storage** | **Sensitive data** | **Non-sensitive data** | **Moving data** | **Non-moving data** |
| **Physical** | X | X | | X |
| **Cloud** | | X | X | |

Table 6-3 clearly shows the best places to store the different classifications of data. In this table, sensitivity has a higher priority than movement of data. Thus, if the data that moves around frequently, the priority will still be to try and keep it as safe as possible; if that is not possible other ways of security can be considered:

- Sensitive data are generally stored in physical storage and should be kept as secure as possible;
- Less sensitive data can be stored anywhere that has ease of access;
- Data that moves around frequently should be stored in volatile storage (cloud-based); and
- Data that does not move around frequently is generally stored in physical storage.

Physical storage is generally better for sensitive data and data that does not move around frequently. It also reduces the chance of data loss or leakage. Cloud storage is better for less sensitive data and data that moves around frequently. It has a much higher ease of access than physical storage. It should be mentioned that there are not many cases where only one type of storage is used. Since companies normally work with some data which are sensitive and other data which are less sensitive, a combination of the two storage methods can be implemented to have segregated physical storage and cloud storage. Cloud storage is generally leveraged above physical storage because of the ease of access, but until sensitivity is not a problem a combination will always be used.

Another storage system that may be considered is the transformation of physical storage into an internal cloud. This means physical storage is adapted in such a way that it can be connected to wirelessly. Thus, static storage is available on-site, which is safer than having the data stored on the Internet, and the user can connect to it with ease via wireless technology on the intranet.

The final planning practice that will be discussed is *Security planning*.

### 4.4. Security planning

Before looking at security planning, there are a few things that have to be made clear:

- Basic security measures are always important;
- Sometimes security can be “do-as-needed” and thus in some cases security is not implemented to a large extent;
- When security needs to be implemented, it should always be implemented to its full extent; and
- Looking at system security there should be no technical flaws; the system security should be so tight that social engineering needs to take place for anyone to get into the system.

Keeping all of this in mind, it should be mentioned that security is dependent on many different issues. The level of security required depends on the following:

- The need for security;
- The client(s) and their requirements;
- The development requirements, even if the client does not mention it;

- The type, sensitivity and classification of the data that needs protection;
- The security of existing systems that might be integrated with the new one; and
- Whether the application being developed is externally (client) or internally (employee) facing.

The different types of security will be discussed next.

### 4.4.1. Development security

The following aspects regarding development security should be kept in mind during development:

- Risk assessment is always important in order to find the correct security level of an application;
- Authentication is always important in all applications being developed;
- Security may differ depending on the type of application, so the type of application being developed should always be considered;
- Security should be taken into account in all the phases of development; and
- Security should be assessed and addressed in each phase separately, even in the production phase.

A few security practices will now be mentioned that may be used to increase security during development:

- Security filtered APIs and SDKs should be used to develop software;
- Data used during development should be kept secure at all times;
- Employees working with development data should be authenticated, thus access control should always be implemented;
- Biometrics are always a good way to ensure that access control on development data are at its highest (this also makes auditing easier); and
- Containerised environments can help with security during development.

Developing in a secure manner may help with security, but this does not ensure that every aspect of security is taken care of. The application being developed also needs its own security. This is discussed next.

### 4.4.2. Application security

As mentioned in the previous section, the application being developed needs its own set of security implementations. The following are some of these implementations that may be added to different applications to make them more secure:

- SSL certificates;
- HTTPS;
- Authentication (multiple layers if necessary);
- Safe application permissions;
- Biometrics (if devices allow); and
- In some cases decryption keys can be coded into the applications to ensure that only the devices that run the applications can decrypt the specific data.

The final part of security that has to be taken into consideration is *Data security*.

#### 4.4.3. Data security

As mentioned in the *Security planning* section, security can depend on different aspects. The main aspects that data security depends on are the clients' decisions on the security that has to be implemented on their (the clients') data and the classification of the data. Thus it is important to assess the data risk beforehand. These are a few different ideas that may help increase general data security:

- If data are sensitive of nature, security is a must;
- Encryption is important no matter where the data resides or is transmitted to;
- Digital rights management;
- Access control;
- Data used for development should always be kept safe and secure;
- Certain data can be made time sensitive to ensure that it stays secure;
- Authentication and authorisation on data (as many levels as needed);
- Biometrics if possible;
- Secure email services;
- Remote device access (wiping and locking);
- Good encryption key policies; and
- Data protection frameworks.

#### 5. Data

*Data* influences the *Data management planning* and also gets secured by *Data security* and *Secure practices*. These nodes have already been discussed.

## 6.2 Summary

This chapter presented a framework as a guideline for developers to develop secure mobile applications, including EMAs or other mobile applications, to incorporate security. It discussed different device policies that might be taken into account, different SDM options, different planning phases and techniques. It finally discussed different security implementations to take into account during development. It should be mentioned again that this is only a guideline as deduced from this data and can be adapted to suit the needs of the specific project.

# CHAPTER 7 CONCLUSION

## 7.1 Introduction

In this chapter the purpose of the research will be restated. The research aims and objectives will also be stated along with how they were fulfilled. The research contributions will be presented and the chapter will conclude with limitations of this study and recommendations for future work. Figure 7-1 shows a representation of the chapter.

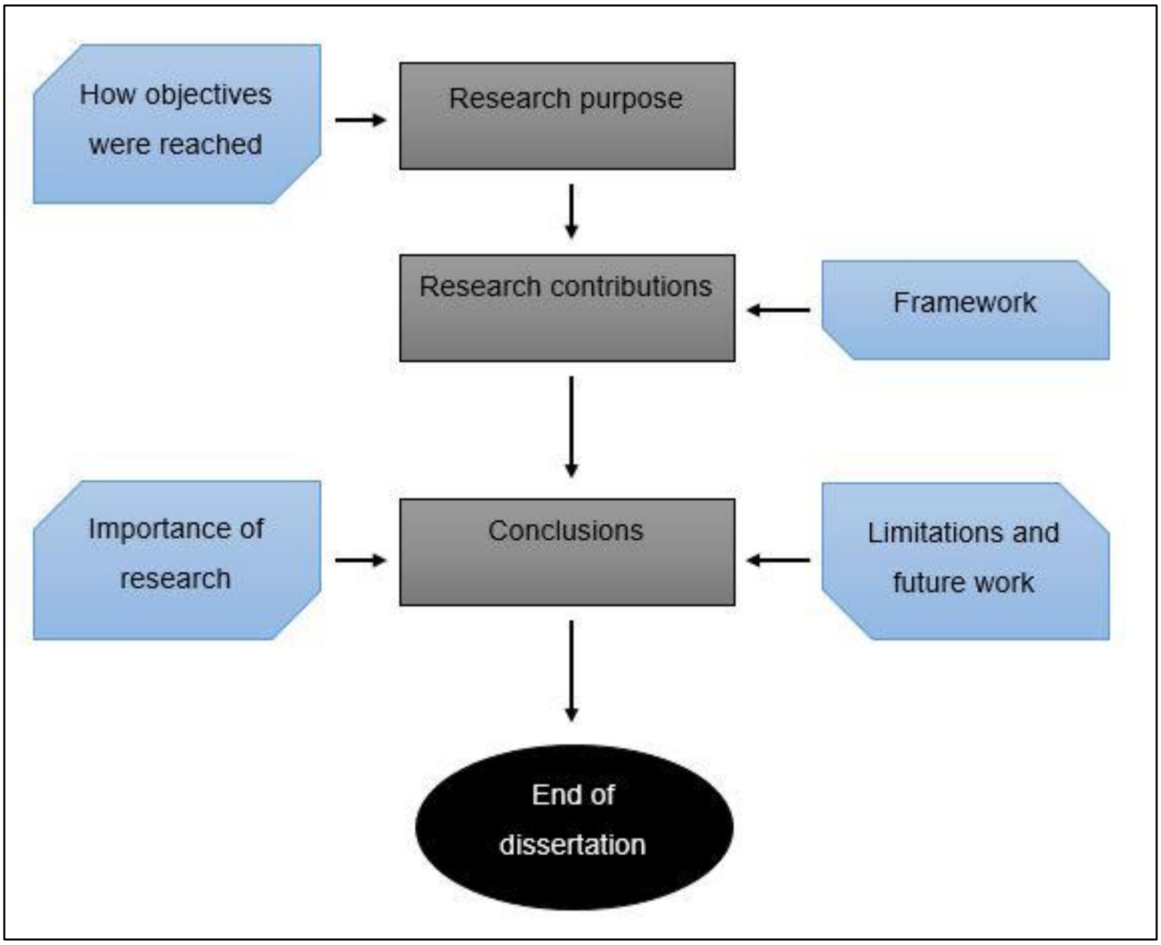


**Figure 7-1: Chapter 7 representation**

## 7.2 Research purpose

The purpose of the study was to acquire knowledge on whether enterprises that are using mobile device architectures have adequate security measures in place regarding information assets and processes when developing mobile applications.

The aim of this study is to investigate the development of EMAs, together with any security challenges that might occur (during or related to their development). The objectives, needed to reach the research aim, are listed below, as well as how they were fulfilled.

**1. To investigate existing MADMs**

This objective was fulfilled in Chapter 2. An in-depth literature review of different SDMs (including normal SDMs, agile SDMs and MADMs) as well as their uses was done. It highlighted the fact that mobile application development is mainly done in an agile manner and that certain agile SDMs are specifically tailored for the development of mobile applications. These specifically tailored SDMs are called MADMs and are made up of existing agile SDMs. Further research into agile SDMs and MADMs showed little to no signs of security implementations during their development lifecycles.

**2. To conduct a literature review on different information security practices and policies in enterprises**

Chapter 3 was used to elaborate on these topics and thus presents the fulfilment of this objective. Different literature sources on security implementations and policies used in enterprises were reviewed. Chapter 3 also reviewed different threats that might be encountered in enterprises and solutions to some of the threats. It proceeded to discuss different mobile security threats as well as their impact on enterprises. The conclusion that was reached is that security is a combination of different implementations and that a single implementation in a distinct area of development is not enough. The chapter concluded with different research perspectives on application security and how applications can be kept secure in different ways.

**3. To determine the different information security aspects and policies used in existing SDMs and MADMs**

To fulfil this objective more literature had to be reviewed and added to Chapter 2. Reviewing more data on agile SDMs and MADMs, it was concluded that existing SDMs (agile or otherwise) as well as existing MADMs had little to no signs of security implementation. The different agile SDMs looked at were: XP, FDD and Scrum. After this MADMs were also examined; the different methodologies investigated were Mobile-D, MASAM and SLeSS.

**4. To investigate development practices of EMAs in different enterprises as well as how and why they address general information security in the development**

This objective was fulfilled by conducting case studies, via semi-structured interviews, with different companies in different business sectors. The interviewees had to be knowledgeable on the software development processes of the organisation, as well as the security implementations during development. The data obtained from the case studies was analysed via theme analysis and cross-case analysis, after which the new data were used to construct an EMA secure development framework.

In many cases the answers from the companies did not differ much, but in some of the cases the answers were very different. One of the reasons for this was the different business sectors as well as the different company sizes. In all cases the companies agreed that security, when needed, is important throughout the entire development lifecycle and should not be isolated to a single, or even a few, development phases. Even though some of the specific development practices (planning, testing storage, etc.) differed, all the companies agreed that agile development is the path to follow when doing MAD. The waterfall methodology has become too slow for present development needs, but a hybrid between waterfall and agile may still be used when the project size (very large projects) call for it.

The investigation into device policies lead to an easy answer: smaller companies do not really have a use for personal device policies while larger companies could have a use for it, and even in some cases a need.

The storage topic highlighted the fact that most companies make use of physical storage as well as cloud storage. While cloud storage is leveraged above physical storage, because of the ease of use, both methods will always be used if sensitivity of data is a problem.

The investigation into security leads to the conclusion that there is no single answer. Security is influenced by many different factors, including the client, the type of application being developed, the data that will be used by the application, whether the application will be used internally or externally, the company it is developed for and also whether there is a real need for security.

**5. To use the information obtained from objective 1-4 to suggest improvements for existing development practices**

This objective was fulfilled along with objective 6. It is a difficult job to give specific improvements for each situation without elaborating and writing a complete new chapter (Chapter 6 was used for this). There are different elements that influence choices made by developers, they are mentioned here:

- The size of the enterprise developing the mobile application as well as the size of the company it is developed for (depending on the size of the company, the type, size and security needs of the application might differ);
- The type of mobile application and what it will be used for (standalone applications will have other requirements than applications that need server and web access);
- Whether the mobile application is used internally or externally (security, as well as other elements, might differ depending on whether the application is used by employees or clients from the outside);

- The business sector in which the application will be used (each business sector has different needs, and thus the requirements will different depending on the sector);
- The type, classification and sensitivity of the data that will be used by the mobile application (this influences the storage options of the data as well as how much it will move around; sensitivity is a large influencing factor);
- The SDM (agile or otherwise) or MADM used to develop the mobile application and what phases and steps are followed to obtain a final product (different choices on development practices will also have to be made when the decision is made not to use a formal methodology);
- The size of the project (larger projects may still be done with hybrid of the waterfall method and agile development);
- Device policies in the enterprise (looking at personal devices vs. company devices, or even a combination thereof);
- The different threats that the device/application might have to face (both the application and the device it will run on have to be secured against outside threats);
- The need for security in general (whether the application is really in need of security carries a large influence); and
- The client thoughts, needs and requirements (in agile development the client is one of the most important elements, and what they request is very important).

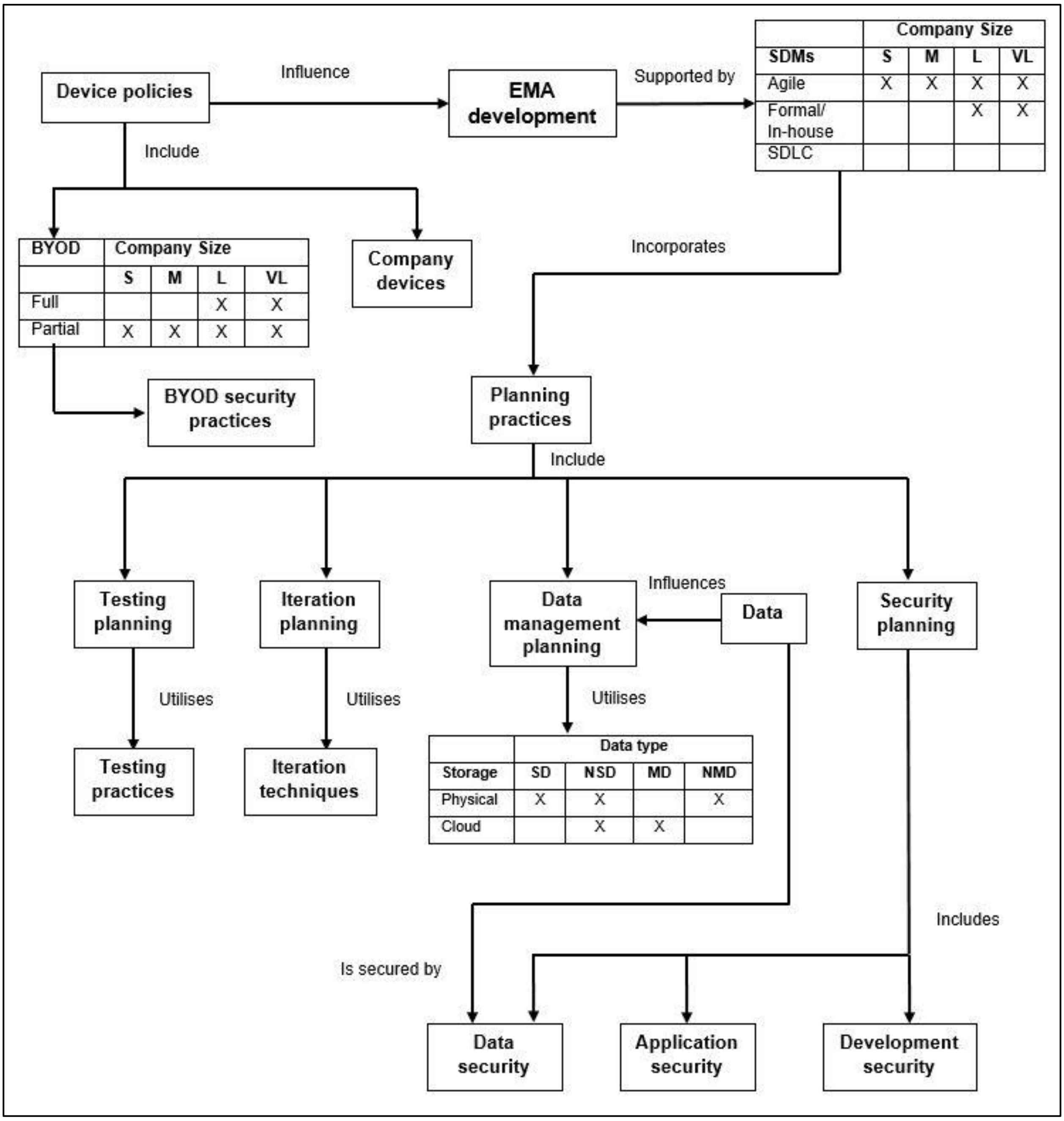
**6. To develop a framework to guide developers regarding different security aspects that could be included during development of EMAs**

After all the other objectives were satisfied, objective 5 and 6 were fulfilled by constructing an EMA secure development framework. This framework may be used as a guideline for developers with the development of more secure mobile applications, including EMAs.

The framework as contribution of this study is presented in Section 7.3.

## 7.3 Research contributions

The previous chapter presented the discussion and the construction of the secure development framework. Figure 7.2 shows the framework again.



| | Company Size | | | |
|---|---|---|---|---|
| SDMs | S | M | L | VL |
| Agile | X | X | X | X |
| Formal/ In-house | | | X | X |
| SDLC | | | | |

| BYOD | Company Size | | | |
|---|---|---|---|---|
| | S | M | L | VL |
| Full | | | X | X |
| Partial | X | X | X | X |

| | Data type | | | |
|---|---|---|---|---|
| Storage | SD | NSD | MD | NMD |
| Physical | X | X | | X |
| Cloud | | X | X | |

**Figure 7-2: Secure development framework as research contribution**

The main research contribution of this study is a secure development framework for EMAs. The framework may act as a guideline for developers in the development of mobile applications as well as EMAs. It may allow them more insight into different device policies in enterprises, different choices of development methodologies and also different data storage options. The framework may also guide developers in more secure ways of development, ensuring that data stay secure in different ways and that applications being developed are more secure.

## 7.4 Conclusion

In this particular study a single conclusion cannot be reached. The different choices in mobile application development are influenced by many different elements. As the framework in Chapter 6 clearly shows, different parts of development are influenced by different elements. For example, the size of the company has an influence on both the device policy (or policies) and the SDM used; the type of data has an influence on the data management and the storage of the data; and security has many different dependents that were all mentioned in the discussion of the framework nodes. Thus the framework helps developers to make the correct choices in their unique development situations.

In conclusion the literature review in Chapter 2 showed that different types of SDMs showed little to no signs of security implementation. This lead to further research into security threats, solutions and policies presented in Chapter 3, which further validated the findings of Chapter 2. This validation showed a gap in the development practices of EMAs. The rest of the study was conducted to address this gap, and thus a framework was constructed to help guide developers in the development of EMAs and other mobile applications.

## 7.5 Importance of this research

The research is important because mobile device usage is rapidly increasing in present times. This results in a lack of security implementations during the development of EMAs as well in the EMA itself. Security deficiencies may have a large negative effect on different enterprises, depending on the sensitivity of data and the type of work that they do. Thus this research study and others of its type are important to address the increase in security during the development of EMAs as well as the use of them in the enterprise.

As a result of some limitations in this particular study, other studies might be done in the same area to address gaps and create solutions for them. These limitations are discussed next.

## 7.6 Limitations

This particular study had different limitations that may be taken into account when future work is done in the area. They are as follow:

- The study was done in South-Africa and not internationally; the study might be repeated outside of South-Africa in other settings;
- The results cannot be generalised for all business sectors, because only a few sectors were included in the study; and
- There was a limitation on funds that made travelling in order to interview different people difficult.

Recommendations for future work are now discussed.

## 7.7 Recommendations for future work

While this study contains a framework to guide developers and proposes different security and development practices to be implemented, the scope of the research may still be extended. Some of the areas that might be investigated further will be discussed next.

### Specific development phases

Although the framework that resulted from this study gave an indication that security has to be assessed and furthermore addressed in each of the development phases, more detail might be given into the specific security implementations in each of the phases.

### Individual SDMs

The research mentioned that different SDMs do not always include security implementations during their development. The SDMs could each be taken and studied individually to identify the best areas to implement security.

# BIBLIOGRAPHY

Acts.  See South Africa.

Abrahamsson, P.  2005.  Mobile software development - the business opportunity of today.  (In. Proceedings of the International Conference on Software Development, p. 20-23).

Abrahamsson, P., Hanhineva, P., Hulkko, H., Ihme, T., Jäälinoja, J., Korkala, M., Koskela, J., Kyllönen, P. & Salo, O.  2004.  Mobile-D: an agile approach for mobile application development.  International Journal of Service Industry Management:174-175.

Abrahamsson, P., Salo, O. & Ronkainen, J.  2002.  Agile software development methods.  VTT Publications, 478(1):1-110.

Ahmad, Z., Francis, L., Ahmed, T., Lobodzinski, C., Audsin, D. & Jiang, P.  2013.  Enhancing the security of mobile applications by using TEE and (U)SIM.  (In.  2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing organised by.  p. 575-582).

Akbar, R. & Safdar, S.  2015.  A short review of global software development (GSD) and latest software development trends.  (In.  2015 International Conference on Computer, Communication and Control Technology (I4CT 2015), p. 314-317).

Al Bar, A., Mohamed, E., Akhtar, M.K. & Abuhashish, F.  2011.  A preliminary review of implementing enterprise mobile applications in ERP environment.  International Journal of Engineering & Technology IJET-IJENS, 11(4):60-65.

Ali, S., Qureshi, M.N. & Abbasi, A.G.  2015.  Analysis of BYOD security frameworks.  (In.  2015 Conference on Information Assurance and Cyber Security (CIACS), p. 56-61).

Ambler, S.W.  2009.  The agile scaling model (ASM): Adapting agile methods for complex environments.  IBM Rational:1-35.

Armando, A., Merlo, A. & Verderame, L.  2014.  Security considerations related to the use of mobile devices in the operation of critical infrastructures.  International Journal of Critical Infrastructure Protection, 7(4):247-256.

Avison, D.E. & Fitzgerald, G.  2006.  Information systems development: methodologies, techniques and tools.  4th Edition.  Maidenhead, UK: McGraw-Hill Education.

Azham, Z., Ghani, I. & Ithnin, N.  2011.  Security backlog in Scrum security practices. Paper presented at the 2011 5th Malaysian Conference in Software Engineering (MySEC).

Beck, K.  2005.  Extreme programming explained: embrace change.  2nd Edition.  Upper Saddle River, NJ: Pearson Education, Inc.

Beck, K., Grenning, J., Martin, R.C., Beedle, M., Highsmith, J., Mellor, S., Van Bennekum, A., Hunt, A., Schwaber, K., Cockburn, A., Jeffries, R., Sutherland, J., Cunningham, W., Kem, J., Thomas, D., Fowler, M. & Marick, B.  2001.  Manifesto for agile software development.  http://agilemanifesto.org/principles.html  Date of access: 7 February 2016.

Boehm, B.  2002.  Get ready for agile methods, with care.  Computer Software Development, 35(1):64-69.

Brinkkemper, S.  1996.  Method engineering: engineering of information systems development methods and tools.  Information and Software Technology, 38(1):275-280.

Brown, D. 2014. Balancing out security. ITNOW, Sep2014, 56(3):36-37.

Carvalho, F. & Azevedo, L.G. 2013. Service agile development using XP. (In. 2013 IEEE 7th International Symposium on Service Oriented System Engineering (SOSE), p. 254-259).

Castro-Castilla, A. 2014. Multiplatform and mobile app development in scheme with gambit/scheme spheres. (In. 2014 Proceedings of ILC on 8th International Lisp Conference, p. 4-5).

Chow, T. & Cao, D. 2007. A survey study of critical success factors in agile software projects. The Journal of Systems and Software, 81(1):961-971.

Chowdhury, A.F. & Huda, M.N. 2011. Comparison between adaptive software development and feature driven development. (In. 2011 International Conference on Computer Science and Network Technology (ICCSNT), p. 363-367).

Cockburn, A. & Highsmith, J. 2001. Agile software development the people factor. Software Management:131-133.

Conger, S. 2011. Software development life cycles and methodologies: Fixing the old and adopting the new. International Journal of Information Technologies and Systems Approach, 4(1):1-22.

Consel, C. 2011. DiaSuite: A paradigm-oriented software development approach. (In. Proceedings of the 20th ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation, p. 77-78).

Corral, L., Sillitti, A. & Succi, G. 2013. Software development processes for mobile systems: is agile really taking over the business? (In. 2013 1st International Workshop on the Engineering of Mobile-Enabled Systems (MOBS), p. 19-24).

Creswell, J.W. 2013. Research design: Qualitative, quantitative and mixed methods approaches. 4th. Thousand Oaks: Sage Publications.

Creswell, J.W. & Clark, V.L.P. 2007. Designing and conducting mixed methods research. Australian and New Zealand Journal of Public Health, 31(4):388-389.

Da Cunha, T.F.V., Dantas, V.L.L. & Andrade, R.M.C. 2011. SLeSS: a scrum and lean six sigma integration approach for the development of software customization for mobile phone. (In. 2011 25th Brazilian Symposium on Software Engineering (SBES), p. 283-292).

Delac, G., Silic, M. & Krolo, J. 2011. Emerging security threats for mobile platforms. (In. MIPRO, 2011 Proceedings of the 34th International Convention, p. 1468-1473).

Dube, R.R. & Dixit, S.K. 2010. Process-oriented complete requirement engineering cycle for generic projects. (In. International Conference and Workshop on Emerging Trends in Technology, p. 194-197).

Dudovskiy, J. 2011. Positivism. http://research-methodology.net/research-philosophy/positivism/ Date of access: April 7 2016.

Dyck, S. & Majchrzak, T.A. 2012. Identifying common characteristics in fundamental, integrated, and agile software development methodologies. (In. International Conference on System Sciences, p. 5299-5308).

VTT Electronics. 2006. Portals of agile software development methodologies. http://agile.vtt.fi/mobiled.html Date of access: February 8 2016.

Elo, S. & Kyngäs, H. 2008. The qualitative content analysis process. Journal of Advanced Nursing, 62(1):107-115.

Farago, P. 2012. Mobile app growth led by video sharing: Youtube in the crosshairs? http://www.flurry.com/bid/84831/Mobile-App-Growth-Led-by-Video-Sharing-YouTube-in-the-Crosshairs#.VSzRMvmUde8 Date of access: 8 February 2016.

Felt, A.P., Finifter, M., Chin, E., Hanna, S. & Wagner, D. 2011. A survey of mobile malware in the wild. (In. Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, p. 3-14).

Flora, H.K. & Chande, S.V. 2013. A review and analysis on mobile application development processes using agile methodologies. International Journal of Research in Computer Science, 3(4):9-18.

Fowler, M. & Highsmith, J. 2001. The agile manifesto. Software Development: The Lifecycle Starts Here:1-7.

Futcher, L.A. 2011. An integrated risk-based approach to support IT graduate students in secure software development. Port Elizabeth: NMMU. (Thesis-Ph.D.).

Gajar, P.K., Ghosh, A. & Rai, S. 2013. Bring your own device (BYOD): Security risks and mitigating strategies. Journal of Global Research in Computer Science, 4(4):62-70.

Ghani, I., Izzaty, N. & Firdaus, A. 2013. Role-based eXtreme Programming (XP) for secure software development. Paper presented at the Special Issue-Agile Symposium, Malaysia.

Ghobadi, S. 2014. Perceived barriers to effective knowledge sharing in agile software teams. Information Systems Journal:1-36.

Giessmann, A., Stanoevska-Slabeva, K. & de Visser, B. 2012. Mobile enterprise applications - Current state and future directions. (In. Hawaii International Conference on System Sciences organised by. p. 1363-1372).

Goyal, S. 2007. Agile techniques for project management and software engineering. (In. Major Seminar on Feature Driven Development, p. 1-19).

Gregory, P.H. 2003. Security in the software development lifecycle. http://searchsecurity.techtarget.com/tip/Security-in-the-software-development-life-cycle Date of access: 18 October 2016.

Guang-yong, H. 2011. Study and practice of import Scrum agile software development. (In. 2011 IEEE 3rd International Conference on Communication, Software and Networks (ICCSN), p. 217-220).

Hasan, B., Dmitriyev, V., Gómez, J.M. & Kurzhöfer, J. 2014. A framework along with guideline for designing secure mobile enterprise applications. (In. 2014 International Carnahan Conference on Security Technology (ICCST), p. 1-6).

Holzer, A. & Ondrus, J. 2009. Trends in mobile application development. Mobile Wireless Middleware, Operating Systems, and Applications-Workshop. 1 ed. Berlin: Springer. p. 55-64.

Huisman, H. & Iivari, J. 2006. Deployment of systems development methodologies: Perceptual congruence between IS managers and systems developers. Information & Management, 43(1):29-49.

Hundermark, L. 2015. Enterprise mobile apps should be as simple and functional as their customer equivalents. http://memeburn.com/2015/03/enterprise-mobile-apps-should-be-as-simple-and-functional-as-their-consumer-equivalents/ Date of access: February 8 2016.

Huy, N.P. & Van Thanh, D. 2012. Developing apps for mobile phones. (In. 2012 7th International Conference on Computing and Convergence Technology (ICCCT), p. 907-912).

Iivari, J., Hirshheim, R. & Klein, H.K. 1998. A dynamic framework for classifying information systems development methodologies and approaches. Information Systems Research, 9(3):164-193.

Iivari, J., Hirshheim, R. & Klein, H.K. 2000. A dynamic framework for classifying information systems development methodologies and approaches. Journal of Management Information Systems, 17(2):179-218.

Iivari, J. & Iivari, N. 2011. The relationship between organizational culture and the deployment of agile methods. Information and Software Technology, 53(5):509-520.

Ispareh, M., Ladani, B.T., Panahi, S.S. & Azadani, Z.N. 2010. Toward a software development methodology for anonymity applications. (In. Proceedings of the 2010 EDBT/ICDT Workshops, p. 1-6).

Jain, A.K. & Shanbahg, D. 2012. Addressing security and privacy risks in mobile applications. IT Professional, 14(5):28-33.

Jana, D. & Bandyopadhyay, D. 2014. Management of security and privacy issues of application development in mobile cloud environment: A survey. (In. International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), p. 1-6).

Jaramillo, D., Smart, R., Furht, B. & Agarwal, A. 2013. A secure extensible container for hybrid mobile applications. (In. 2013 Proceedings of IEEE Southeastcon, p. 1-5).

Jeong, Y., Lee, J. & Shin, G. 2008. Development process of mobile applications SW based on agile methodology. (In. 10th International Conference on Advanced Communication Technology organised by. p. 362-366).

Jönsson, M. 2013. Agile system development: An investigation of the challenges and possibilities of using Scrum. UMEA University.

Joorabchi, M.E., Mesbah, A. & Kruchten, P. 2013. Real challenges in mobile app development. (In. IEEE International Symposium on Empirical Software Engineering and Measurement, p. 15-24).

Khalid, A., Zahra, S. & Khan, M.F. 2014. Suitability and contribution of agile methods in mobile software development. International Journal of Modern Education and Computer Science, 2(1):56-62.

Kim, H., Oh, J.S. & Moon, H.N. 2013. Development of smart mobile app assessment model. (In. 2013 16th Conference on Network-Based Information Systems (NBiS), p. 300-304).

Krauss, S.E. 2005. Research paradigms and meaning making: A primer. The Qualitative Report, 10(4):758-770.

Kuusinen, K. & Mikkonen, T. 2013. Designing user experience for mobile apps: Long-term product owner perspective. Asia-Pacific Software Engineering, 1(1):535-540.

Lakshman, M., Sinha, L., Biswas, M., Charles, M. & Arora, N.K. 2000. Quantitative vs qualitative research methods. India Journal of Pedeatrics, 67(5):369-377.

Leavitt, N. 2013. Today's mobile security requires a new approach. IEEE Computer Society: Computer, 46(11):16-19.

Lennon, R. 2015. Communicating perceived inadequacies during enterprise application development. (In. 2015 IEEE International Professional Communications Conference (IPCC), p. 1-3 ).

Lin, Y., Huang, C., Wright, M. & Kambourakis, G. 2014. Mobile application security. IEEE Computer Society:21-23.

Mathiassen, L. & Sandberg, A.B. 2014. Process mass customization in a global software firm. Software, 31(6):62-69.

McAfee, A. 2006. Mastering the three worlds of information technology. Harvard Business Review, 84(11):141-150.

Mishra, D. & Mishra, A. 2010. Managing requirements in market-driven software project agile methods view. Technical Gazette, 17(2):223-229.

Nerur, S. & Balijepally, V. 2007. Theoretical reflections of agile development methodologies. Communications of the ACM, 50(3):79-83.

Nerur, S., Mahapatra, R. & Mangalara, G. 2005. Challenges of migrating to agile methodologies. Communications of the ACM, 48(5):73-78.

Noor, K.B.M. 2008. Case study: a strategic research methodology. American Journal of Applied Science, 5(11):1602-1604.

Nosseir, A., Flood, D., Harrison, R. & Ibrahim, O. 2012. Mobile development process spiral. (In. 2012 Seventh International Conference on Computer Engineering & Systems (ICCES), p. 281-286).

Nyambo, D., Tarimo, C. & Yonah, Z.O. 2014. Security frameworks in the converged web and mobile applications: A review. (In. Pan African International Conference on Science, Computing and Telecommunications (2014), p. 29-34).

Oates, B.J. 2006. Researching information systems and computing. 1st Edition. Thousand Oaks, CA: Sage Publications.

Paetsch, F., Eberlein, A. & Maurer, F. 2003. Requirements engineering and agile software development. (In. 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, p. 308-313).

Palmer, S.R. & Felsing, J.M. 2002. A practical guide to feature-driven development. 1st Edition. Upper Saddle River, NJ: Prentice-Hall.

Palmieri, M., Singh, I. & Cicchetti, A. 2012. Comparison of cross-platform mobile development tools. (In. 2012 16th International Conference on Intelligence in Next Generation Networks, p. 179-186).

Park, M.W., Choi, Y., H., Eom, J.H. & Chung, T.M. 2014. Dangerous Wi-Fi access point: attacks to benign smartphone applications. Personal and Ubiquitous Computing, 18(6):1373-1386.

Peng, S., Yu, S. & Yang, A. 2014. Smartphone malware and its propagation modeling: A survey. IEEE Communications Surveys and Tutorials, 16(2):925-941.

Penning, N., Hoffman, M. & Nikolai, J. 2014. Mobile malware security challenges and cloud-based detection. (In. 2014 International Conference on Collaboration Technologies and Systems, p. 181-188).

Pogar, I., Gligora, M. & Davidovi, V. 2013. BYOD: A challenge for the future digital generation. MIPRO 13:748-752.

Putra, I.P.E.S., Yuliawati, A. & Mursanto, P. 2012. Industrial extreme programming practice's implementation in rational unified process on agile development theme. (In. 2012 International Conference on Advanced Computer Science and Information Systems (ICACSIS), p. 137-142).

Qiu, M., Zhang, L., Ming, Z., Chen, Z., Qin, X. & Yang, L.T. 2013. Security-aware optimization for ubiquitous computing systems with SEAT graph approach. Journal of Computer and System Sciences, 79(5):518-529.

Radia, N., Zhang, Y., Tatipamula, M. & Madisetti, V.K. 2012. Next-generation applications on cellular networks: Trends, challenges, and solutions. (In. Proceedings of the IEEE, p. 841-854).

Ralph, P. 2011. Introducing an empirical model of design. (In. The 6th Mediterranean Conference on Information Systems, p. 1-12).

Rehman, I.U., Rauf, A. & Shahid, A.A. 2010. Scope management in agile versus traditional software development methods. (In. Proceedings of the 2010 National Software Engineering Conference, p. 1-6).

Rishi, M. 2012. Innovations around mobile applications: Scope for Indian developers. Journal of Technology Management for Growing Economics, 3(2):119-136.

Rogers, Y. 2006. Moving on from Weiser's vision of calm computing: Engaging UbiComp experiences. (In. 2006 Proceedings of the 8th International Conference on UbiComp 2006, p. 404-421).

Rowley, J. 2002. Using case studies in research. Management Research News, 25(1):16-27.

Rowley, J. 2014. Designing and using research questionnaires. Management Research Review, 37(3):308-330.

Ruparelia, N.B. 2010. Software development lifecycle models. ACM SIGSOFT Software Engineering Notes, 35(3):8-13.

Sani, A., Firdaus, A., Jeong, S.R. & Ghani, I. 2013. A review on software development security engineering using dynamic systems method (DSDM). International Journal of Computer Applications, 69(25):37-44.

Sathyan, J. & Sadasivan, M. 2010. Multi-layered collaborative approach to address enterprise mobile security challenges. (In. 2010 IEEE 2nd Workshop on Collaborative Security Technologies, CoSec 2010 organised by. p. 1-6).

Schwaber, K. 2004. Agile project management with scrum. 1st Edition. Washington, WA: Microsoft Press.

Schwaber, K. & Beedle, M. 2002. Agile software development with scrum. 1st Edition. Upper Saddle River, NJ: Prentice-Hall.

Seaman, C.B. 1999. Qualitative methods in empirical studies of software engineering. IEEE Transactions on Software Engineering, 25(4):557-572.

Seo, S., Gupta, A., Sallam, A.M., Bertinno, E. & Yim, K. 2014. Detecting mobile malware threats to homeland security through static analysis. Journal of Network and Computer Applications, 38(1):43-53.

Shen, Y., Lin, F. & Rohm, C.E.T. 2009. A framework for enterprise security architecture and its application in information security incident management. Communications of the IIMA, 9(4):9-20.

Siggelkow, N. 2007. Persuasion with case studies. Academy of Management Journal, 50(1):20-24.

Singh, S., Kumar, N. & Bansai, V. 2015. Adoption of agile methodology in software industry. International Journal of Scientific & Engineering Research, 6(5):198-142.

South Africa. 1996. National Small Business Act 102.

Spataru, A.C. 2010. Agile development methods for mobile applications. University of Edinburgh:1-59.

Susan, G. 1995. The role of methodology in IT-related organizational change. (In. Proceedings of BCS Specialist Group on IS Methodologies, p. 1-14).

Tan, C., Tan, W. & Teo, H. 2008. Training students to be agile information systems developers: a pedagogical approach. (In. Proceedings of the 2008 ACM SIGMIS CPR Conference on Computer Personnel Doctoral Consortium and Research, p. 88-96).

Tang, F., You, I., Tang, C. & Guo, M. 2013. An efficient classification approach for large-scale mobile ubiquitous computing. Information Sciences, 232(1):419-436.

Tripp, J.F. & Riemenschneider, C.K. 2014. Toward an understanding of job satisfaction on agile teams: Agile development as work redesign. (In. 2014 47th Hawaii International Conference on Systems Science, p. 3993-4002).

Unhelkar, B. & Murugesan, S. 2010. The enterprise mobile applications development framework. IT Professional, 12(3):33-39.

Vaquero, L.M., Rodero-Merino, L. & Buyyam, R. 2011. Dynamically scaling applications in the cloud. ACM SIGCOMM Computer Communication Review, 41(1):45-52.

Vavpotic, D. & Vasilecas, O. 2011. An approach for assessment of software development methodologies suitability. Electronics and Electrical Engineering, 114(8):107-110.

Wasserman, A.I. 2010. Software engineering issues for mobile application development. (In. Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research organised by. p. 397-400).

Wei, X., Gomez, L., Neamtiu, I. & Faloutsos, M. 2012. Malicious android applications in the enterprise: What do they do and how do we fix it? (In. 2012 IEEE 28th International Conference on Data Engineering Workshops (ICDEW), p. 251-254).

Weiser, M. 1991. The computer for the 21 century. Scientific American, 265(3):78-89.

Whitman, M.E. & Mattord, H.J. 2011. Principles of information security. 4th Edition. Canada: Cengage Learning.

Yin, R.K.  2009.  Case study research design and methods.  4th Edition.  Thousand Oaks, CA: Sage Publications.

Yun, D. & Xaio-hui, C.  2009.  A study on the security technology of enterprise mobile information system.  (In.  2009 International Conference on Computational Intelligence and Security, p. 385-391).

Zhou, Y. & Jiang, X.  2012.  Dissecting android malware: Characterization and evolution.  (In.  2012 IEEE Symposium on Security and Privacy, p. 95-109).

Zhu, H., Ziong, H., Ge, Y. & Chen, E.  2014.  Mobile app recommendations with security and privacy awareness.  (In.  Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 951-960).

# APPENDIX A

Small business classification according to the National Small Business Act.

**Table A-1: The National Small Business Act (102 of 1995) classifications of different small to medium businesses (Adapted from South Africa, 1996)**

| Sector or sub-sectors in accordance with the Standard Industrial Classification | Size or class | Total full-time equivalent of paid employees<br>Less than: |
|---|---|---|
| **Agriculture** | Medium<br>Small<br>Very small<br>Micro | 100<br>50<br>10<br>5 |
| **Mining and Quarrying** | Medium<br>Small<br>Very small<br>Micro | 200<br>50<br>20<br>5 |
| **Manufacturing** | Medium<br>Small<br>Very small<br>Micro | 200<br>50<br>20<br>5 |
| **Electricity, Gas and Water** | Medium<br>Small<br>Very small<br>Micro | 200<br>50<br>20<br>5 |
| **Construction** | Medium<br>Small<br>Very small<br>Micro | 200<br>50<br>20<br>5 |
| **Retail and Motor Trade and Repair Services** | Medium<br>Small<br>Very small<br>Micro | 100<br>50<br>10<br>5 |

| **Sector or sub-sectors in accordance with the Standard Industrial Classification** | **Size or class** | **Total full-time equivalent of paid employees**<br>**Less than:** |
|---|---|---|
| **Wholesale Trade, Commercial Agents and Allied Services** | Medium<br>Small<br>Very small<br>Micro | 100<br>50<br>10<br>5 |
| **Catering, Accommodation and other Trade** | Medium<br>Small<br>Very small<br>Micro | 100<br>50<br>10<br>5 |
| **Transport, Storage and Communications** | Medium<br>Small<br>Very small<br>Micro | 100<br>50<br>10<br>5 |
| **Finance and Business Services** | Medium<br>Small<br>Very small<br>Micro | 100<br>50<br>10<br>5 |
| **Community, Social and Personal Services** | Medium<br>Small<br>Very small<br>Micro | 100<br>50<br>10<br>5 |

# APPENDIX B

**Table A-2: Case 1 themes, subthemes and data**

| Themes and subthemes | Interview 1+2: Case 1 |
|---|---|
| **Type of work** | |
| Specific | |
| | The type of work they do is between server-side, hardware and software and the security aspects of the different things they work on. |
| | Licence plate recognition system was written for a security company. (Hardware and software components) |
| | Specialise in RFID and NFC (Also work with UHRFID with longer distance) (Card reader or chip reader systems) (Hardware and software components) |
| | NFC and RFID are also used in mobile payment systems (Hardware and software components) |
| | Can use UHRFID to securely identify different cars driving past a specific area, at high speeds and longer distances. |
| Coding | |
| | Different languages used are: Java, C++ and Android (mostly done in Java) |
| **SDM** | |
| Agile | |
| | Work in an agile manner even though being a very small company. They use different agile techniques and agile policies, but no formal methodology is used, because they are so small. Mentions sprints |

| Themes and subthemes | Interview 1+2: Case 1 |
|---|---|
| | and development cycles or iterations. |
| | The mobile development they do is also done in agile, also without a formal methodology. |
| | The Setup of the IDE is done quickly via containers and this also ensures that settings and work is done in uniform and that everyone doesn’t do their own thing. Makes integration easier at the end. This also ensures that security is implemented in the correct areas from the start and then extras can be added where necessary. |
| Security | |
| | Always develop in a secure way |
| Planning | |
| | The planning phase is very important to them. They make sure that planning is done to the fullest. |
| | The stakeholders (includes the client) and their perspectives on the application being developed are very important to them, and they like to keep their clients happy. BDD integrates well with this, using the end user perspective to help finish the product. |
| | User stories are used along with the planning phase to make sure the correct things are done and to help that planning is done correctly and thoroughly. |
| | Along with their rigorous planning phase, friendly but harsh criticisms are welcomed and received with open arms and an open mind. |
| Testing | |

| Themes and subthemes | Interview 1+2: Case 1 |
|---|---|
| | They do Test Driven and BDD. |
| | Testing and making sure the applications is working to the best of its ability is also very important and thus they use different testing methods along with a test framework they have. This is basically the Test Driven Development, but integrates with the BDD (seeing that they are the same thing executed in slightly different manners. |
| | Bug fixes are done correctly along with the test framework to make sure that nothing breaks when something is fixed. Regression bugs are eliminated because of the test framework they use, no bugs will arise because another one was fixed. |
| | Peer reviews and getting different opinions on their work is important to them, this helps with the development process and also with smaller technical details. |
| Mobile vs Traditional | |
| | Doesn't see mobile apps as being less of software than traditional software and give the same amount of attention to detail to both the types of software. It doesn’t depend on the type of software but rather the specific piece of software, attention to detail must always be top. |
| **Type of client** | |
| | Clients are long-term clients. Try to work with the same clients for a long period of time. |
| | They have worked, and still work with some international companies including security companies. |
| | Clients that they started with are still with them, which means the companies are happy with the delivery of service even after years. |

| Themes and subthemes | Interview 1+2: Case 1 |
|---|---|
| | Still building a client base but already have a few large companies and international companies they do work for. |
| | The clients they work for need security to be a high priority, so they work hard to make security one of the top priorities. |
| **Storage** | |
| Physical | |
| | NAS (Physical, onsite storage) for data storage and retrieval. |
| | Containers are also used as a type of real-time storage during development, but no real data are stored there, just used during development for different processes and so forth. An area for secure development. |
| Cloud | |
| | Google drive (cloud storage) is used for different documentation purposes and for client paperwork etc. |
| Hybrid | |
| **Practices** | |
| | They have practices like source control, version control and containerised environments, along with Docker technology that integrated with the containers. Different containers can run on a Docker and this helps with the run of different processes, it also helps with security during development. The container is basically sandboxed and this keeps it and whatever is running inside secure. |

| Themes and subthemes | Interview 1+2: Case 1 |
|---|---|
| | Containers also help with the IDE setup, to help do it efficiently and IDE maintenance is also lessened. |
| Security | |
| | There is no data involved in the containers so no chance of data loss, etc. Data are put into the processes via mounting of files and folders and via network access. Containers also make the jobs they have to do easier in the speed category, doesn't take much processing power and the work is scalable (many of the same job can be done in parallel. |
| BYOD (Personal Device) | |
| | No Bring Your Own Device implementation or policies, but do use personal devices for personal things. When work data are used via mobile device security measures are taken.  (Mentions that in South Africa B.Y.O.D isn't a huge thing and not many companies have the resources or the know how to pull it off correctly) |
| | Companies that do use the B.Y.O.D implementation don't always do it correctly or by the book, but rather take certain policies from it and use half of them to run an implementation that works for them. |
| | Many people feel that it's better in the end to just buy company devices instead of letting everyone use personal devices, because of company devices being more configurable by the company itself, and thus lessons data loss. |
| | This all depends on the company or the type of company in the end. |

| Themes and subthemes | Interview 1+2: Case 1 |
|---|---|
| **Security** | |
| Specific | |
| | Other security practices are things like: SSL certificates, HTTPS, encryption, Digital certificates, digital signatures. |
| | They are currently working on a security ISO standard for an international company. |
| | Permissions are never added to an application if the permission isn't needed by the application. The SDK helps with permission controls while developing |
| Development | |
| | When the app needs it, authentication will always be done to the fullest. Communication should also be running smoothly. But that’s in final products, in demos security might be skipped to reach maximum development efficiency. |
| | The security of an app depends on the client, requirements, use cases and even the already existing systems that they use. |
| | They involve security throughout their development lifecycle and security is always one of their top priorities. |
| Data | |

# APPENDIX C

**Table A-3: Case 2 themes, subthemes and data**

| Themes and subthemes | Interview 3: Case 2 |
|---|---|
| **Type of work** | |
| Specific | |
| | They mostly work in the farming industry where grain and field work is present. This also includes the prices and other information of the products they work with. |
| | The developers (two of them) are new to mobile development and the apps they are working on are of the first they are working on. |
| | The two apps are a app of grain and other products information and the second app is for the workers in the field to track different aspects of their job and data. |
| Coding | |
| | Mentions MVC which is C# and web development. |
| **SDM** | |
| Agile | |
| | Develop in an agile manner, but no mention of a formal methodology. Does mention different iterative cycles or sprints they work in and how they can differ in time depending on the project. |
| | They work toward a certain publish, so they declare a specific time for different sprints or cycles and then when the work is done it is partially published. This is then discussed and work is continued afterwards. |

| **Themes and subthemes** | **Interview 3: Case 2** |
|---|---|
| Security | |
| | The development framework they use does include its own level of security, but mentions different extra levels. (discussed under security) |
| | Mentioned that security isn't always a big thing in their apps because they are data driven and the data are kept off of the app itself, it is kept in safe secure storage. |
| | Develop apps around Windows accounts to have automatic authentication without having to add extra authentication. |
| | Develop securely around different security aspects they have in all their applications. Windows accounts, HTTPS and security tokens. |
| | Make sure that development is done securely from the word go. Also use skeleton (template) development where security is already built into the IDE before start. |
| Planning | |
| Testing | |
| Mobile vs Traditional | |
| | They see app development as something new to the industry and that people see it as something that has to be developed as fast as possible to get the app published and running, before the competition steals the idea, etc. |
| **Type of client** | |

| Themes and subthemes | Interview 3: Case 2 |
|---|---|
| | Farmers and people working in the field on farms. |
| | Data aren't as sensitive as other companies and thus a big fuss isn't made of security by their clients. |
| **Storage** | |
| Physical | |
| Cloud | |
| | Everything is cloud-based storage, except onsite data are kept in a database on physical storage (might also be partly cloud-based) |
| | App data and data used via the app is transferred from and to the cloud via Wi-Fi or other wireless connections. |
| | The data aren't as sensitive so less security measures are needed, and thus cloud storage (being less secure than physical storage) is fine for what they have to do. |
| | App data aren't updated out in the field, only back at home base, even though it is cloud-based and not all on physical storage. |
| Hybrid | |
| **Practices** | |
| Security | |
| | Use Windows accounts for login purposes and this helps with security adding Windows security to their everyday work life. |
| | HTTPS calls that give extra security to any call made (call being data call and not a communication between two people) |

| Themes and subthemes | Interview 3: Case 2 |
|---|---|
| | Tokens are another thing they use to up security in certain places, to make sure specific things are more secure when needed. Tokens are encrypted with AES 64. |
| | Also adds time windows to their login and authentication to make sure that if a certain time goes by login is invalid and the person is not authorised. |
| BYOD (Personal Device) | |
| | No mention of B.Y.O.D implementation at all. |
| **Security** | |
| Specific | |
| | For their own company security they use Windows accounts as authentication and login. |
| | Sensitive data are kept off of applications and this keeps the data more secure. Because data are kept off of applications the application security doesn't have to be so high, normal security standards are applied, but nothing extra. |
| | Sensitive data can only be accessed with specific authorisation. |
| | All service calls are secured by HTTPS (moving toward HTTPS, some are still HTTP). The guideline rule is that very sensitive things use HTTPS and less sensitive things use HTTP, but in the end they want everything to be HTTPS. |
| | Tokens are encrypted and added as extra form of security, encryption is done with AES 64. |

| Themes and subthemes | Interview 3: Case 2 |
|---|---|
| | Applications have to access certain data, these apps have to go through authentication, the service calls have to be accepted and the encrypted token has to be accepted, the secret key needed to decrypt the data are embedded in code and cannot be retrieved, so the application is needed to decrypt the files. |
| | Permissions are all off by default when they start developing, and as the application needs a certain feature which requires a permission, the permission is requested. |
| | Time-sensitivity is added to certain sensitive data, and even some data which aren't as sensitive. |
| Development | |
| | The SDK they use has its own levels of security. |
| | In development itself, they do see security as important throughout the whole development lifecycle and if you don't add it when necessary, you sit with the problem later. But it should be mentioned, because they are so new to mobile app development, and because they don't work with super sensitive data, security is not high on the priority list, more of an afterthought and when needed. |
| | They know of course when they are done with their first few applications that they will have a basic template to work from and the template will contain the security needed for most common applications they are to develop. |
| | They see testing and auditing as important steps of any application, to keep it more secure. |
| Data | |

## APPENDIX D

**Table A-4: Case 3 themes, subthemes and data**

| Themes and subthemes | Interview 4: Case 3 |
|---|---|
| **Type of work** | |
| Specific | |
| | What they basically do is serve solutions to different companies that need a problem solved. (They can be seen as a small development house) |
| | They have background in different areas like: Financial, accounting, payroll, labour broking and logistic systems. And are doing new types of development every day. |
| | They can be seen as a development house, but they see themselves as a mentorship house instead of only looking at the technical part of things. |
| | The human aspect is very important to them and they believe that the more you teach the person, the better work will be done and thus they focus on the people aspect and see the development more as a tool for the person. |
| | The passion that a person has for what they do us also very important and they focus to improve the person, because they believe by improving the person, you improve the work they do. |
| | Looking into applications to enhance learning, to help people (no matter how young or old) improve themselves. |
| Coding | |
| **SDM** | |

| Themes and subthemes | Interview 4: Case 3 |
|---|---|
| | Very person driven company, development house, which sees development as the development of people and the more you develop the person, the more he/she will be able to do for you. |
| | The people in their development team are people who really want to develop and don't do it only as a job. This ensures that they do the best job possible and have a passion for what they do, and this results in great products. |
| Agile | |
| | They use agile methods, but no mention of a formal methodology. Also mentions that agile is used as a guideline in flexibility and adaptability, because why use a formal rule system when it's meant to be flexible and adaptable. |
| | Try to be as flexible and adaptable while giving good feedback on regular cycles to make sure the product is at correct quality level. |
| | Work in iterative cycles or sprints. |
| | Have specific development principles like: Separation of concerns, intent and no duplication as well as different best practices they follow. (Best practices are some of the ones from Extreme Programming) |
| Security | |
| | In development they don't really look at security except it it's really necessary and the client needs or wants it. |

| **Themes and subthemes** | **Interview 4: Case 3** |
|---|---|
| | Security isn't one of their largest things, but when they have to look at it they see it as important throughout the development lifecycle. And not only security, but no one specific element of development should be isolated as more or less important. |
| Planning | |
| Testing | |
| Mobile vs Traditional | |
| | Sees mobile and traditional software development as the same. The only difference is what the client wants and needs, not how you see them as different. The methodology shouldn't change just because the type of software changes, it depends on the software. The happier the customer, the better. So speed is a factor with mobile application development, but not only with mobile applications, with traditional software as well. But mentions that speed should not leave openings for errors. Quality is the most important. |
| | When time is a factor, the company which is developed for should prioritize their needs and development is done according to priority, this includes security. |
| **Type of client** | |
| | Have a lot of different clients being a development house and server of solutions. |
| | Clients that they have had range from financial, accounting, logistics and labour companies. There have also been some smaller companies in need of problem solving. |
| **Storage** | |

| Themes and subthemes | Interview 4: Case 3 |
|---|---|
| Physical | |
| Cloud | |
| | It all depends on the data being stored, but because of the nature of their company, cloud storage makes more sense. |
| Hybrid | |
| | They use a secure server for data storage, depending on the sensitivity of the data. But also use cloud storage for the most part, because of it being more practical. |
| **Practices** | |
| | Use a Microsoft environment and framework along with their own development practices and principles. |
| | Best Practices along with their 'separation of concerns', 'intent' and 'no duplication' policies also help them to develop efficiently. |
| | They see development (and any type of job they do or thing they work with) as a tool to help people improve themselves and thus work for them is a practice to improve on people and on themselves. |
| Security | |
| | Different security practices they do run are: SSL certificates and secure server login and authentication. |
| BYOD (Personal Device) | |
| | No mention of B.Y.O.D implementations. |
| **Security** | |

| Themes and subthemes | Interview 4: Case 3 |
|---|---|
| | In their mind-set security is a thing of "do as needed" and if it isn't much needed, leave it at basic. |
| Specific | |
| | They haven't really worked in security in the types of apps they have written other than the basic security aspects of most applications, like authentication , secure servers, SSL and policies and things to keep data more secure IF needed. |
| Development | |
| | In their opinion, security is important throughout the development lifecycle, if the applications requests or needs security. But the basics are always important. |
| Data | |
| | Data storage is a big thing when working with mobile development and they believe in data encryption on all data, to make sure it stays secure outside of applications and wherever it might reside. |

# APPENDIX E

**Table A-5: Interview 1 of Case 4 themes, subthemes and data**

| Themes and subthemes | Interview 5: Case 4 |
|---|---|
| **Type of work** | |
| Specific | |
| | This interview was done with a fraud expert in financial crime control and digital platforms. |
| | Banking applications, internal and external. |
| | Applications to help customers of bank make payment in the most secure way possible. |
| | The work they do is to make sure that apps are developed in such a way that clients feel safe to use them. |
| Coding | |
| **SDM** | |
| Agile | |
| | No specific methodology, but use different characteristics from different formal agile methodologies. Traditional way of development but with agile methods and implementations. Basically a hybrid methodology. |
| | Work in iterative intervals and the sprints are normally one to two weeks. |
| | Always work client-based and following the client requirement, end user requirements, and as flexible as possible. Changing requirements are normal so adaptability is important. |

| Themes and subthemes | Interview 5: Case 4 |
|---|---|
| Security | |
| | Developing different applications they look at authentication as one of the first things in security as they are a bank and security is important. |
| | Highest possible API and the SDK is setup to make sure the client requirements are met. |
| | Whether it is an internal application or a client facing application for people on the outside, security is always important and should never be overlooked. |
| | Development follows different standards and best practices are important. |
| | Development is always done in a secure way and data are kept out of danger of leakage and being lost. Data access levels and other policies are implemented to make sure that people using data need access to the specific data and if they don't need it they do not have access to it. Encryption is also used for data to make sure the access levels are enforced. Need to know basis. |
| | Depending on the data being transferred the encryption and security may differ. RSA and encrypted tokens could be used for extra strong encryption. |
| | The basic process is a planning session to figure out everything that needs to happen as well as a risk/threat analysis. After this security builds up as the lifecycle continues from there and extra security measures are added as needed. Each phase in development the security is assessed and addressed accordingly, thus security is always important no matter which phase of development you are currently in. |

| **Themes and subthemes** | **Interview 5: Case 4** |
|---|---|
| | Security is important before, during and after development and even in production. In production security is also assessed in real time and this is also addressed and acted upon. |
| Planning | |
| | Planning is an important aspect to their development and this includes threat assessment and different levels of security that is needed for the specific application. |
| | During planning testing is also planned and the testing framework is chosen and decided on. |
| | The basic process is a planning session to figure out everything that needs to happen as well as a risk/threat analysis. After this security builds up as the lifecycle continues from there and extra security measures are added as needed. Each phase in development the security is assessed and addressed accordingly, thus security is always important no matter which phase of development you are currently in. |
| Testing | |
| | During planning testing is also planned and the testing framework is chosen and decided on. |
| | Testing is done on different levels and it depends on the application and the outcome of the previous tests. A specific test framework is chosen, but as the testing proceeds, different necessary testing aspects are added. For instance if one test fails another one is set up to make sure that it gets fixed and passes the next test. All tests have to be passed at the best possible level and nothing is left out. Internal production testing as well as external testing is also done. |

| Themes and subthemes | Interview 5: Case 4 |
|---|---|
| Mobile vs Traditional | |
| **Type of client** | |
| | Being a bank they have clients all over the country and world, working with their finances. There is no limit to the type of client they may have. |
| | Mentions large companies, international companies and even individuals. |
| | Millions of transactions are done per day, thus client base is very large. |
| **Storage** | |
| Physical | |
| | Physical data storage is used for sensitive data as well as data that don't have to move around a lot, but the less sensitive data that also moves around more are kept in cloud-based storage or any type of volatile storage. Saying this, it's more of a guideline than a rule, it all depends on the specific piece of data. |
| | Encryption for data are important when in transit or at rest, no matter where the data are. |
| Cloud | |
| | Data are also stored across the whole country as well as the world, with a lot of different divisions being established in the world. For this reason data are well monitored and ensured to stay secure no matter where it travels to or even at rest in all storage areas. |
| | Encryption for data are important when in transit or at rest, no matter where the data are. |

| **Themes and subthemes** | **Interview 5: Case 4** |
|---|---|
| Hybrid | |
| | Combination of cloud and physical storage, depending on the sensitivity of the data. |
| | Data segregation is applied to different data types and classifications (the sensitivity of the specific piece of data) to make sure the correct data are kept properly secure. Enterprise data and personal data are also segregated. |
| | Encryption for data are important when in transit or at rest, no matter where the data are. |
| **Practices** | |
| Security | |
| | Some of their security practices include biometric security systems like: voice recognition (for employees and clients), fingerprint scanning, retina scanning, etc. |
| | Voice recognition is widely used for clients when making voice payments for instance. The voice is run through a system and the recognition system can tell if it is the correct person making the payment (this is of course used along with other security measures) |
| | One of the practices they implement to help them recognise different clients as well as their movements and actions is "Big Data" which is basically a huge database full of different information about clients and employees which help with decision making (done with machine learning and Artificial Intelligence) |
| | Use Fido which is extra authentication and cryptographic support to go along with their own. As well as private key policies (encryption key system) |

| **Themes and subthemes** | **Interview 5: Case 4** |
|---|---|
| | For data moving around in the company they use WorxMail for secure and configured mail services to ensure that all data moving around the company is kept safe and away from malicious intent, people or actions. Along with this enterprise and personal data are kept segregated, but at time that they can't help mixing personal and work data, the data are monitored and measures are taken to make sure the data gets transmitted safely. |
| BYOD (Personal Device) | |
| | They have B.Y.O.D implementations as well as B.Y.O.N, but to an extent. The devices or networks used for work and that use enterprise data are setup securely and frameworks are used to help keep sensitive and enterprise data safe and secure. |
| | For the off chance that data gets leaked or lost, policies are set in motion to try and remake the data as well as assess the risk of the data lost (data loss prevention). They also use remote wiping for devices that get lost and stolen. |
| **Security** | |
| Specific | |
| | Different biometric systems are used, like: voice recognition, fingerprint scanners, retina scanners, etc. |
| | Fido is an extra security measure adding another level of authentication as well as cryptographic support to employee and client systems. |
| Development | |
| | In development, no matter what application is worked on, |

| Themes and subthemes | Interview 5: Case 4 |
|---|---|
| | authentication is important. Security overall in a bank is important. |
| | Security, threat and risk levels, authentication, data access, these can all differ depending on the app or even different users. |
| | The SDK makes sure that whether internal or external apps are written, security is implemented to the fullest extent to which the app needs it. This is done after a risk/threat assessment is done to ensure that all the extra security measures are also added, the ones which the SDK does not account for. |
| | The application as well as the security is put through rigorous testing frameworks and only passes when it passes al tests to the fullest. If needed extra tests are designed for certain projects and external testers are also used at times. Testing is even done during the production phase to make sure in real-time that the application works as it should. |
| | When developing the data have to stay secure, and this is done by assigning access levels to different developers and making sure only the correct people see the sensitive data or work with the sensitive data. When the data have to move, encryption is used to keep the data secure and when the data are real sensitive, extra token encryption is done with RSA (Not mentioning which RSA it is, 128 or 256) |
| | In the planning phase risk and threat assessment is done, as the project lifecycle runs its course the security builds up, each step/phase security is assessed and dealt with accordingly. Thus it can be seen that security is important no matter what phase of development it is. |
| Data | |

| Themes and subthemes | Interview 5: Case 4 |
|---|---|
| | "Big Data" is another practice they use to make sure that security is up to scratch and when authenticating a client, that authentication, no matter of what type, is done correctly and that the client is really who they say they are. |
| | Private key policies as well as authentication and encryption is a must when working with data. Encryption (key policy) is used to make sure that the correct data goes to the correct person with the correct access. |
| | When looking at different data we look at user risk vs data risk, is the user is a risk, the data aren't given, if the data are a risk, the user is authenticated more harshly, etc. |
| | Any device using enterprise data are setup and configured with a security framework to make sure the enterprise data are kept safe and secure. |
| | Data loss is real and you can’t always prevent it, but risk assessments are done on lost data as well as policies and implementations to recreate lost data, to make sure that not all goes lost, but to get back as much as possible. This is also done when devices are lost, along with data and device locking and remote wiping. |
| | WorxMail application is also used to keep data in emails secure and safe from malicious intent and activities. Like mentioned before, data loss can’t be prevented completely, thus data are monitored inward and outward to make sure no malicious activity is present. |
| | Segregation of enterprise data and personal data are done, but when it can't be avoided, measures are taken to make sure data are secure and travels in a secure way via monitoring. |

| Themes and subthemes | Interview 5: Case 4 |
|---|---|
| | When developing the data have to stay secure, and this is done by assigning access levels to different developers and making sure only the correct people see the sensitive data or work with the sensitive data. When the data have to move, encryption is used to keep the data secure and when the data are real sensitive, extra token encryption is done with RSA (Not mentioning which RSA it is, 128 or 256) |
| | Data storage is secure in looking at different classifications of data are stored in different types of storage. Depending on the sensitivity. Physical and cloud storage is used, but that doesn't mean that all non-sensitive data are stored on cloud and all sensitive data are stored on physical storage. It depends on more than just the sensitivity. It also depends on how much the data have to move around. So it depends on a few things so different security measures are taken depending on the different types of data and where they are stored. |
| | Looking more at storage in an international view point, you can't really keep the data in one place and it has to move around, so other security measures are taken, for instance data encryption and data validation. But all in all different protocols and policies are in place to make sure data stays secure no matter where it is or where it is moving to. |

# APPENDIX F

**Table A-6: Interview 2 of Case 4 themes, subthemes and data**

| Themes | Interview 6: Case 4 |
|---|---|
| **Type of work** | |
| Specific | |
| | This interview was done with a lead security architect in systems and development. |
| | Background in different system development, for instance ZimSwitch and SaSwitch as well as experience with different banks (ABSA and Standard Bank) |
| Coding | |
| | They use languages like: Angular JS, Angular Node JS, Java-based languages and also at times, HTML5. |
| **SDM** | |
| Agile | |
| | Development tools are a big game shifter in present times and a lot of people use tools to help with development. This integrated with the SDK that is being used. |
| | Using agile is important in present times, and moving toward formal methodologies, whether one or characteristics from different ones, is also an important aspect of development. (waterfall approach is too slow) |
| | Responding to client requirements as quick as possible and with as short as possible cycles. Responding to business demand as fast as possible with short life cycles. |

| Themes | Interview 6: Case 4 |
|---|---|
| | Big projects might be a hybrid usage of agile and waterfall, but they will be more of the core bank banking systems, and not mobile applications. |
| Security | |
| | Data is kept as static as possible and they prefer not to move around data during development. But in cases that it need to happen, encryption is used to ensure the data stays secure. It is preferred that data gets downloaded and consumed from one specific place and not that data is stored on a device itself. |
| | They try to keep data centrally and consume it from there, they use an example like Apple and Google and even Dropbox in the ways they use central cloud storage. But only an example of central data, doesn't say that they specifically use these products (This is discussed later) |
| | When data have to be transferred a data protection Framework is followed to ensure that secure transfer is done. This steps in this framework are as follow: 1) Get ownership of data 2) Identification and classification of data 3) Integrity of data 4) Controls 5) Authentication and auditing and login |
| | Security should be present in all phases of development. From time of gathering requirements all the way through to the end. |
| Planning | |
| Testing | |
| | Testing is important as well as risk management and code reviews by peers and other employees. |
| Mobile vs Traditional | |
| | Sees all software (traditional and mobile) as the same effort value, it |

| Themes | Interview 6: Case 4 |
|---|---|
| | all depends on the specific application or software product. |
| | Mobile applications can be written to a specific baseline and features can be added, where traditional software take much longer to get to a baseline, so looking at timewise, a baseline mobile application can take longer than traditional software, but it doesn't have to and the effort isn't necessarily less. |
| **Type of client** | |
| | Being a bank they have clients all over the country and world, working with their finances. There is no limit to the type of client they may have. |
| **Storage** | |
| Physical | |
| Cloud | |
| | There's a drive toward more cloud, but a privacy office is placed in charge of permitting what data are allowed to be transferred to the cloud and what data not. It strongly depends on the sensitivity. |
| | Data are hosted internationally, but not in countries who are not friendly toward our Protection of Personal Information act. |
| Hybrid | |
| | Combination of cloud and physical storage, depending on the sensitivity of the data. |
| | Different physical storage is used as well as different cloud-based storage (volatile storage). |
| | Trying to leverage more modern types of storage, but sensitivity is still a problem and before that is addressed we will always have a hybrid system. |

| **Themes** | **Interview 6: Case 4** |
|---|---|
| **Practices** | |
| | Believe that Development tools are becoming a game changer and that it's the way of the future. |
| Security | |
| | Device and data auditing to make sure the correct person is held accountable at the correct time. Device locking and remote wiping if necessary when devices get lost or stolen. And also, people are trusted not to move data around too much, and this is monitored to make sure data stays in secure environments. |
| | But not only people and devices have to be kept secure and need rules, guidelines and policies. Applications also have to be kept secure, even after the development phase, and after security has been built into them. Applications have to run in a secure area and this when applications on devices have to access sensitive data (or any data we don't want people to see), they run in containerized environments when they are always secure and away from any malicious activities. |
| | Risk management is also done on different employees, data and applications to see what type of control has to be exercised and what type of security measures has to be taken, etc. |
| BYOD (Personal Device) | |
| | Also mention the B.Y.O.D implementation, and also give a few ways of keeping data secure when using personal devices vs using company devices. |

| Themes | Interview 6: Case 4 |
|---|---|
| | Segregation of enterprise and personal data are one of those polices. Seeing that enterprise data does not reside on a device that leaves the premise of the company. Data loss prevention. Multi factor authentication (including biometrics, login protection via username and password, email notifications and more). Containerisation. Digital rights management. MDM (is also used in general to make sure personal and company devices are managed correctly. |
| | More on personal device usage, he adds that although they have a B.Y.O.D implementation, the policy isn't completely used like it should be, but they are moving toward it because of rapid growth and different aspects of the enterprise they cannot ignore. |
| | For the reasons that they are in need of people using personal devices and it's unavoidable at times, policies are pushed down to ensure that data stays secure, some of these policies have been mentioned above, but there are more that relate to the human aspect, because the devices are not defendable anymore, they have to be configured and setup correctly, it all depends on the people, the employees using the devices. |
| | There is a tension between how much freedom employees are given in boundaries of trust versus how much control can be exercised. It depends on the type of data and the type of applications of course, as well as the people using the applications and accessing the data. |
| **Security** | |
| | Policies are set in place to make sure that all facilities are used securely. |
| | It is mentioned that policies are there, but it's difficult to enforce policies, and have control over everything that happens, so trust comes into play. |

| Themes | Interview 6: Case 4 |
|---|---|
| Specific | |
| | Triple A is important (Authentication, authorization and administration. |
| | Security controls have to be applied in regard to auditing and login, data protection and zonal protection. |
| Development | |
| | In development Testing and code reviews are an essential part of risk management. |
| | During development they also prefer for data not to move around too much and rather retrieve and consume data from a central information point. Data are not stored on devices as this can leave the data open to malicious intent and when data have to move around or be in less secure places, encryption is used to secure the data. |
| | Security is important throughout the development lifecycle from start, right to the end. |
| Data | |
| | Policies are set in place to guide employees with the moving of different devices and the data on them. One of these policies are that data should not leave the premise of the company and that as far as possible data should stay inside the company. |
| | If data are lost there are implementations for data loss and device loss. Things like remote device access help with this as well as remote wiping and data recreation when data have gone lost. |
| | Other data security policies are things like: Digital rights management, containerization, multi-factor authentication, etc. Data protection is of utmost importance to the bank, thus these policies, |

| Themes | Interview 6: Case 4 |
|---|---|
| | guidelines and principles are set into place. Also the POPI Act is also used in South Africa. |
| | There is a need-to-know basis of data, so if you don't need to know, you won't have the data, but a data protection framework is setup for this and the framework has a few different steps, and they are as follow to make sure that the correct data are viewed and edited by the correct people. This also applies when data have to be transferred. |
| | When data have to be transferred a data protection Framework is followed to ensure that secure transfer is done. This steps in this framework are as follow: 1) Get ownership of data 2) Identification and classification of data 3) Integrity of data 4) Controls 5) Authentication and auditing and login |
| | Time-sensitivity is another thing to keep in mind when talking about security. Depending on the data it can be made time-sensitive to make sure the data are more secure and only valid for a certain time. |
| | They treat different objects, systems and data with different security, which means that what is more susceptible to compromise will have higher security controls. For instance, an ATM pin will have very high security. |

# APPENDIX G

**Table A-7: Case 5 themes, subthemes and data**

| Themes | Interview 7: Case 5 |
|---|---|
| **Type of work** | |
| Specific | |
| | They work on web applications, mobile applications and also different line of business software, for different companies. They do development for other companies. |
| | Client relations are important to them and to make sure the applications are developed in the manner the client wants. |
| Coding | |
| **SDM** | |
| Agile | |
| | They use an agile methodology, but also no mention of a specific or formal method. |
| | Work in sprints or cycles of one to two weeks, depending on the type or amount of work (also the priority of the work being done) |
| Security | |
| | Security is important from the word go and throughout the whole lifecycle. |
| | Not all people think about security as a first thing, but when it's an important aspect of the specific project, it's always priority. |
| Planning | |
| Testing | |

| Themes | Interview 7: Case 5 |
|---|---|
| Mobile vs Traditional | |
| **Type of client** | |
| | They build applications for different large corporations and mentions banks to be one specific type. Also builds applications for smaller non-profit organisations. Also being a development house (on much larger scale) they will have all sorts of clients. |
| **Storage** | |
| Physical | |
| | They used to have only server (their own servers) for data storage, but they run hybrid (cloud and physical) now. Each employee gets a terabyte of data that they can use to store and share data and the security is what they decide, along with the normal security aspects of the cloud. |
| Cloud | |
| | Like the idea of having a one stop shop for information, information hosted centrally and using it from there. |
| Hybrid | |
| | Use both cloud and physical storage, but is trying to move as much as possible to cloud storage. |
| **Practices** | |
| | Different tools used in the work area are: Outlook, Sharepoint and other Microsoft Suite products, Trello, Jira and other tools that have mobile device counterparts. |
| Security | |

| Themes | Interview 7: Case 5 |
|---|---|
| | Don't have any corporate email application with security for data monitoring etc (like Standard Bank for instance, he also mentions that ABSA does use something similar to Standard Bank's WorxMail) |
| | They use a Microsoft framework with extra security for things like common attacks and data extraction, so that they can be prevented. They also have auditing rules so that data are audited so that they know who did what with the data, when and where. They use data version control to make sure that data can be taken to a previous version if necessary and if someone gets hold of the data files they can't do much with them because they are encrypted. |
| BYOD (Personal Device) | |
| | Different communications going out can be accessed via mobile devices. |
| | There is a type of B.Y.O.D setup, but secure to a very high extent (because of data not being extra sensitive) except that it runs through a secure VPN and other small security details being configured. No application restrictions or data restrictions on personal devices, so Mobile Device Management isn't really a thing with them, but they mention that larger companies they work for do implement policies like that. |
| **Security** | |
| | Because the work for many different companies they see a lot of different things, and security isn't always top priority, but when it is, for instance like with a bank, security is always well prioritised above the rest. |
| Specific | |

| Themes | Interview 7: Case 5 |
|---|---|
| | Common things are done to applications developed to prevent things like hackers, data extractions, SQL injections, cross site scripting, forgery and common malicious activities like them. |
| Development | |
| | Common things are done to applications developed to prevent things like hackers, data extractions, SQL injections, cross site scripting, forgery and common malicious activities like them. |
| | They believe that when developing and security is something needed in the application and one of the requirements it is important to think of and act upon security throughout the whole development life cycle from the word go! |
| Data | |
| | Authentication and authorisation are important and are used when working with systems and data. Data encryption on the other hand is more of an afterthought and it depends on the applications and the data. |
| | When transmitting data, it also depends on the data and the applications and the users that have access to the data, before thinking of encryption. |
| | If someone from the outside does get the file they can't do much with it without the decryption key because of the encrypted data. |

# APPENDIX H

**Table A-8: Case 6 themes, subthemes and data**

| Themes | Interview 8: Case 6 |
|---|---|
| **Type of work** | |
| Specific | |
| | They do software development for banks and other financial institutions and do focus on security aspects a lot. |
| | They don't specialize in mobile applications development, but they have a few people working on it. They do get exposure to that area. |
| Coding | |
| **SDM** | |
| Agile | |
| | Use agile and waterfall methods as well as different variations of them. So no formal methodology was mentioned but different agile variations are used. |
| Security | |
| | Applications are always developed in a secure way and protection against basic attacks are built into all applications. Extras are added when necessary. APIs used also have security filtering. |
| | Mentions that security should always be important in applications, but because they work with a lot of different companies it is seen at certain ones that security is neglected and then only added at a later time. This is bad practice. |
| Planning | |
| Testing | |

| Themes | Interview 8: Case 6 |
|---|---|
| Mobile vs Traditional | |
| | Mentions that a lot of people rush their mobile applications, because it looks and feels faster, but because they work on bank systems and bank applications, the features have to roll out together with the web applications and other systems. Thus the features being the same as the software equivalent's features, the mobile app will take the same time and no less effort. Quality is very important. |
| **Type of client** | |
| | They serve different development purposes for banks and other financial institutions. |
| | Very specific client base, but even if it might be less different clients, the same clients do many different projects with them. |
| **Storage** | |
| Physical | |
| Cloud | |
| Hybrid | |
| | Physical and cloud storage hybrid. File share is also enabled for data which aren’t at a too sensitive level. And if the data have to move and is sensitive, encryption is used. |
| **Practices** | |
| | Different coding practices are also put in place, but he mentions that if people need to get something out as fast as possible they tend to neglect those types of practices, and not only at their company, but all around the IT community. |
| Security | |

| Themes | Interview 8: Case 6 |
|---|---|
| | They make sure that people who work with extra sensitive information go on courses to make sure they can code securely as well as work with the data in a secure way, and the data are also audited by external companies. |
| BYOD (Personal Device) | |
| | They have a type of personal phone usage but no mention of Bring Your Own Device (This was discussed in the other interview. They do however have company laptops each that they work on. |
| **Security** | |
| Specific | |
| | Things that they think of in the sense of security are Authentication, single sign-on solutions, internal and external auditing and encryption. This along with all the basic security things built into SDKs and APIs. |
| Development | |
| | They believe that security is always important during development and not some specific area or phase should be singled out. |
| Data | |

# APPENDIX I

**Table A-9: Case 7 themes, subthemes and data**

| Themes | Interview 9: Case 7 |
|---|---|
| **Type of work** | |
| Specific | |
| | The company does contract work for different companies. This interview was an overall view of different things they do and how they do them as well as a little more focus on the specific contract they have with a bank, Rand Merchant Bank. |
| | He spoke mostly of banks and investment banks. |
| Coding | |
| | Json, Javascript, C# and HTML5 are all examples of what languages they use. All depending on the type of app being developed. |
| **SDM** | |
| Agile | |
| | Use different agile methods. Basically they use an iterative approach to development (which can be seen as cycles or sprints in development) |
| | They normally used the traditional waterfall approach, but it takes too long in present times of development. Also much less flexible. |
| | The sprints they do are normally two weeks in length and they have daily stand-ups, which are meetings each morning during the sprint to give an executive summary of what's going on in the project at the moment. Thus planning is important and that they stay on track. |

| Themes | Interview 9: Case 7 |
|---|---|
|  | They also weekly demos, or a demo every two weeks, depending on the project. This helps the client know what is going on in the project.  Keeping the client happy is important. |
|  | The development is very much client-based and what the client inputs is very important. The develop from an ownership point of view which is basically from the clients point of view to make sure the client is happy and stays involved. |
| Security |  |
|  | The make a big point of segregating internal and external facing applications and to exclaim that they are different types of applications and that they will differ. |
|  | Mentions that there is no one specific part of development where security should be more or less important, but it all depends on the different application types. |
|  | Internal applications have lower security because the people using them have specific authorisation and that is enough for the data running via those applications, but when it comes to external facing applications, security is always very important from the word go, because in banks the data belongs to the customer and that makes the risk factor high. |
| Planning |  |
|  | Development is done to a base point to have a working application with all the features needed and then extra features are added where and when needed. |
| Testing |  |
| Mobile vs Traditional |  |
|  | Development is done to a base point to have a working application |

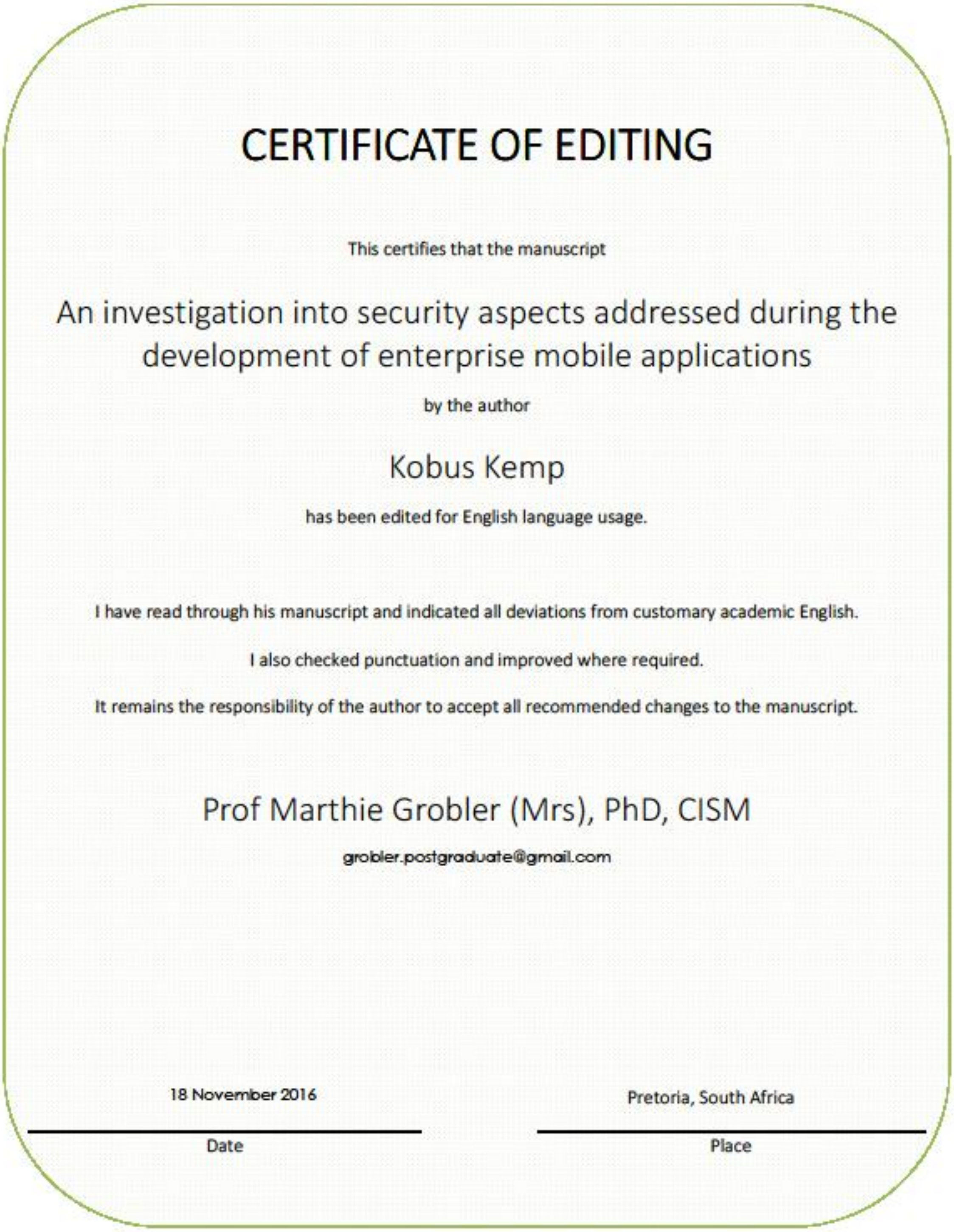
| Themes | Interview 9: Case 7 |
|---|---|
| | with all the features needed and then extra features are added where and when needed. |
| **Type of client** | |
| | They do work for many different companies being a contracting company and have many different contractors at various different companies. But one specific one mentioned was an investment bank and other banks. Also mentions smaller companies which are much less into large scale things. |
| **Storage** | |
| Physical | |
| | Physical storage mostly, because of cloud being a big risk for investment banks or any banks. |
| Cloud | |
| | Cloud systems are used, but they have private cloud initiatives where the cloud isn't completely hosted on the Internet, but rather on the intranet. So it's cloud-based, but hosted locally. The private cloud can be seen as physical storage but it's different. |
| Hybrid | |
| **Practices** | |
| | In development they look at two different types of development and for each they have different practices. Internal and external. Internal they don't look at too much looking as the applications will be internal facing and they are on the network and they are secured by authentication and if you are authenticated by the company with that specific application, they are happy. External facing applications or client facing applications are a different story. When it comes to external facing applications security teams gather to assess risks |

| Themes | Interview 9: Case 7 |
|---|---|
| | and the audience the app will be facing. Depending on the information they gather, different security protocols and practices are put into place for the apps. |
| | Development tools are used to accelerate development and make it easier on developers, these tools are also used to help make development more secure, meaning that development is done in a more secure way and that the application is secure in the end. |
| Security | |
| | They also have a company device policy where they all get company iPads (as well as laptops) which is on the true network but they were set up in a secure way as well as management of access, of who can access the device and thereby the network. |
| | Development tools are used to accelerate development and make it easier on developers, these tools are also used to help make development more secure, meaning that development is done in a more secure way and that the application is secure in the end. |
| BYOD (Personal Device) | |
| | Don’t have a B.Y.O.D implementation, but they do have a guest network where you can connect with your mobile device as well as devices like laptops, etc. This access is enforced by pin-code, which isn’t very secure, but it's basically impossible to access files from the network, because sensitive files are segregated from the open network. The network is used to access Internet, emails and specific applications. |

| Themes | Interview 9: Case 7 |
|---|---|
| | They also have a company device policy where they all get company iPads (as well as laptops) which is on the true network but they were set up in a secure way as well as management of access, of who can access the device and thereby the network. |
| **Security** | |
| | Looking at banks they always have top-notch security being very paranoid about data loss and their client happiness. |
| Specific | |
| | Laptops (as well as iPads) issued to the employees are setup with only what is necessary thus making them more secure. |
| | Another level of security is using a DMZ (Demilitarized Zone, which basically puts the application in a secure area where it can work outside of the public domain. It puts the application in a specific part of the network so it  only has access there and nowhere else, and applications can only work in that area. |
| | If you want the ability to really break something in the bank or get access to data you don't have permission to look at the only real way to do so is by social engineering, the people aspect. The technical aspects are all covered, so all that remains are people letting you down and making a gap in security. |
| | But even people have it difficult of connecting or getting access to anything. Just for an employee to get access to the VPN, just to connect to his workstation from somewhere else, is a big thing. Legal documents have to be signed and good justification has to be given for why you want the access. And even if you pass those, it all has to go through the correct channels to make sure access is given. |

| Themes | Interview 9: Case 7 |
|---|---|
| Development | |
| | During development data are kept safe by assigning access permissions to different developers and they give you an account which is basically an extra level of authentication and extra level of security. This account also has different levels of security access and depending on who you are your account gets more or less access to data, etc. |
| | Security is important to them throughout the whole development lifecycle and not any specific phase. |
| Data | |
| | The guest network mentioned before increases security for anyone from the outside connecting to a part of the company's network. |
| | Authentication of some sort to each of the networks, and different applications and different people get access to different data depending on their access rights. |
| | Cloud is a very big security risk to them, thus most of their storage is physical, but they are looking into private cloud storage, which is internal (local) cloud storage and would be more secure. |

# APPENDIX J



## CERTIFICATE OF EDITING

This certifies that the manuscript

An investigation into security aspects addressed during the development of enterprise mobile applications

by the author

Kobus Kemp

has been edited for English language usage.

I have read through his manuscript and indicated all deviations from customary academic English.

I also checked punctuation and improved where required.

It remains the responsibility of the author to accept all recommended changes to the manuscript.

Prof Marthie Grobler (Mrs), PhD, CISM

grobler.postgraduate@gmail.com

18 November 2016

Date

Pretoria, South Africa

Place

Figure A-1: Certificate of editing