

The use and effectiveness of system development methodologies in mobile application development

S Zacaria
25380249

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

November 2015

ACKNOWLEDGEMENTS

First and above all, I praise God, the almighty for providing me this opportunity and granting me the ability to succeed. This dissertation appears in its current form due to the assistance and guidance of several people. I would therefore like to offer my sincere thanks to all of them.

In the first place I would like to thank my supervisor, Prof. HM Huisman for her continuous guidance, support and inspiration throughout my years at the university. I would not have been able to make it this far without her. I would also like to thank all the Computer Science lecturers at the NWU for educating me and providing me with an environment in which I could learn and grow to prepare myself for the world.

I would also like to express my gratitude to Mrs. Wilma Breytenbach of the Statistical Consultation Bureau for her advice during the statistical analysis phase of the study.

Furthermore, I would like to thank my family for their encouragement and support. It is difficult to find the correct words to express my deepest thanks to my husband, and to my children. Without their unconditional love and loyal support this study would not have been possible.

ABSTRACT

System development methodologies (SDMs) have been in use since 1970 (Avison & Fitzgerald, 2006). Mobile application development is a new direction in system development. Organizations need high quality mobile applications to provide them with strategic advantages. This study aimed to determine the use and effectiveness of system development methodologies in mobile application development. In order to achieve this aim the following objectives had to be achieved:

- The challenges faced by companies that are currently developing mobile applications had to be investigated.
- Different system development methodologies that are available for mobile application development, for example Mobile-D, MASAM (Mobile Application Software based on Agile Methodology) and GMA (Generic Mobile Application) had to be identified.
- The research had to investigate if system development methodologies are being used for mobile application development in the IT industry.
- The success of mobile application development had to be determined.
- The suitability of current system development methodologies for mobile application development had to be evaluated.

This research was performed within the positivistic research paradigm by means of a survey. Data collection was done via questionnaires. Descriptive statistics, factor analysis, Reliability and Correlation coefficients were used as data analysis methods.

The most important challenges identified are usability, user interface design, a need for system development methodology, security, involvement of different teams and version control of a project. The least important challenges were unrealistic expectations, data intensive applications, separate testing for different platforms, automated testing support, utility, different language code bases, the entertainment factor, fragmentation between devices, platforms, testing and back-end issues. 60% of the respondents are using a system development methodology. It is surprising that 54/180 respondents use traditional system development methodologies. It is interesting that system development methodologies developed for mobile application development are also used and some mobile application development methodologies are not used at all. They are not following the SDM strictly. The successes of mobile application and “delivered on time” indicated a medium relationship with “strictness” of use of SDM. If we use a system development methodology strictly we can improve the success of mobile application and we can deliver the project on time.

Keywords: System development, System development methodologies, Mobile applications, Mobile applications development, Use of mobile application development methodology, Effectiveness of mobile application development methodology.

OPSOMMING

Stelselontwikkelingsmetodologieë (SOMs) is al in gebruik sedert 1970 (Avison & Fitzgerald, 2006). Mobiele toepassingsontwikkeling is 'n nuwe rigting in stelselontwikkeling. Organisasies het hoë-kwaliteit mobiele toepassings nodig om aan hulle 'n strategiese voorsprong te gee. Hierdie studie het ten doel gehad om die gebruik en effektiwiteit van stelselontwikkelingsmetodologieë vir mobiele toepassingsontwikkeling vas te stel. Ten einde hierdie doel te bereik moes verskillende kleiner doelwitte bereik word:

- Die struikelblokke wat maatskappye wat tans mobiele toepassings ontwikkel, teëkom, moes ondersoek word.
- Die verskillende beskikbare stelselontwikkelingsmetodologieë vir mobiele toepassingsontwikkeling, byvoorbeeld Mobile-D, MASAM(Mobile Application Software based on Agile Methodology) en GMA(Generic Mobile Application) moes geïdentifiseer word.
- Die navorsing moes ondersoek of stelselontwikkelingsmetodologieë gebruik word vir mobiele toepassingsontwikkeling in die IT-industrie.
- Die sukses van mobiele toepassingsontwikkeling moes bepaal word.
- Die toepaslikheid van die huidige stelselontwikkelingsmetodologieë vir mobiele toepassingsontwikkeling moes geëvalueer word.

Die navorsing is uitgevoer binne die positivistiese navorsingsparadigma deur middel van 'n opname. Data-insameling is gedoen deur middel van vraelyste. Beskrywende statistiek, faktoranalise, ens. is gebruik as data-analisemetodes.

Die belangrikste struikelblokke wat geïdentifiseer is, is bruikbaarheid, gebruikerskoppelvlakontwerp, 'n behoefte aan stelselontwikkelingsmetodologie-sekureiteit, die betrokkenheid van verskillende spanne en weergawe-beheer van 'n projek. Die minste belangrikste uitdagings was onrealistiese verwagtinge, data-intensiewe toepassings, aparte toetsing vir verskillende platforms, geoutomatiseerde toetsondersteuning, bruikbaarheid, verskillende taalkode-basisse, vermaaklikheidsfaktor, fragmentasie tussen toestelle, platforms, toetsing en rugkant-weergawes. 60% van die respondente gebruik die stelselontwikkelingsmetodologie. Dit is verrassend dat 54/180 respondente tradisionele stelselontwikkelingsmetodologieë gebruik. Dit is verder interessant dat stelselontwikkelingsmetodologieë wat ontwikkel is vir mobiele toepassingsontwikkeling ook gebruik word en dat sommige mobiele toepassingsontwikkeling glad nie gebruik word nie. Hulle volg nie die SOM streng nie. Die sukses van mobiele toepassings en “op tyd gelewer” het 'n

medium verhouding getoon met “strengheid van gebruik” van die SOM. As stelselontwikkelingsmetodologieë streng gevolg word kan die sukses van mobiele toepassings toeneem en die projek kan op tyd geleweword.

Sleutelterm: Stelselontwikkeling, stelselontwikkelingsmetodologieë, mobiele toepassings, mobiele toepassingsontwikkeling, gebruik van mobiele toepassingsontwikkelingsmetodologieë, effektiwiteit van mobiele toepassingsontwikkelingsmetodologie.

ABBREVIATIONS

Abstract Syntax Notation – ASN

Agile System Development Methodology – ASDM

Information Technology – IT

Message Sequence Charts – MSC

Mobile Application Development – MAD

Mobile Application Development Methodology – MADM

Specification and Description Language – SDL

Structured analysis, design, and implementation of information systems –STRADIS

System Development Life Cycle – SDLC

System Development Methodology – SDM

Tree and Tabular Combined Notation – TTCN

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	I
ABSTRACT	II
OPSOMMING	IV
ABBREVIATIONS.....	VI
CHAPTER 1 INTRODUCTION	1
1.1 Introduction	1
1.2 Problem statement	1
1.3 Research aims and objectives.....	2
1.4 Research method	2
1.5 Outline of the study	2
1.6 Summary	3
CHAPTER 2 LITERATURE STUDY.....	4
2.1 Introduction	4
2.2 The history of SDM	4
2.2.1 The “Pre-methodology era”	5
2.2.2 “Early methodology era”	5
2.2.3 The “Methodology Era “	6
2.2.4 The “Post-Methodology Era”	6
2.3 SDM definition	7
2.3.1 Similarities and differences between definitions.....	9
2.3.2 Definition used in this project.....	10

2.4	Classification of SDM based on philosophical paradigm	10
2.4.1	Traditional SDM.....	12
2.4.1.1	Examples of traditional SDM.....	14
2.4.1.2	Advantages of traditional SDM	14
2.4.1.3	Disadvantages of traditional SDM.....	14
2.4.2	Agile system development methodologies	15
2.4.2.1	Principles behind the agile manifesto.....	15
2.4.2.2	Examples of Agile SDM	16
2.4.2.3	Advantages of agile system development methodologies.....	16
2.4.2.4	Disadvantages of agile system development methodologies	16
2.4.3	Agent methodology.....	17
2.4.3.1	Examples for agent methodology.....	17
2.4.3.2	Advantages of agent methodology.....	18
2.4.3.3	Disadvantages of agent methodology.....	18
2.5	Empirical studies on the use and effectiveness of SDMs and MAD	18
2.6	The history of mobile technology.....	20
2.7	History of mobile applications.....	22
2.8	Definition of mobile application.....	22
2.9	Classification of mobile applications	23
2.10	Mobile application development.....	24
2.11	Mobile application distribution.....	25
2.12	Different types of platforms	27
2.12.1	Telco-centric model	28

2.12.2	Device centric model	29
2.12.3	Aggregator-centric model	31
2.12.4	Service-centric model	32
2.13	SDMs for mobile applications.....	33
2.13.1	“Mobile-D”	34
2.13.2	MASAM	35
2.13.3	“GMA”	40
2.13.4	“MODA-TEL”	42
2.13.5	Mansurov’s accelerated development methodology	43
2.13.6	Similarities and differences between Mobile-D, MASAM, GMA, MODA-TEL and Mansurov’s accelerated development methodology.....	45
2.14	Success factors / Critical characteristics / Challenges for MAD success	46
2.15	Summary	54
CHAPTER 3	RESEARCH METHOD.....	55
3.1	Introduction	55
3.2	Research paradigm	55
3.2.1	Positivistic research paradigm	55
3.2.2	Interpretive research paradigm	55
3.2.3	Critical social paradigm.....	55
3.3	Research method	56
3.3.1	Design and creation.....	56
3.3.2	Experiments	56

3.3.3	Survey	56
3.4	Data collection	60
3.4.1	Hard copies (postal survey)	60
3.4.2	Online questionnaire	60
3.5	Development of questionnaire.....	61
3.6	Response rate.....	61
3.7	Data analysis.....	61
3.7.1	Descriptive statistics	61
3.7.2	Factor analysis	63
3.7.3	Reliability	63
3.7.4	Correlation coefficients	63
3.8	Background information of the respondents	67
3.8.1	Personal background information	68
3.8.2	Company background information	73
3.8.3	Mobile application Information	74
3.9	Summary	76
CHAPTER 4 RESULTS		78
4.1	Introduction	78
4.2	Investigate The Challenges Faced By Companies which are currently developing mobile applications.....	78
4.3	Identify different system development methodologies that are available for mobile application development, for example Mobile-D, MASAM and GMA.	85

4.4	Investigate if system development methodologies are being used for mobile application development in the IT industry.	85
4.5	Determine the success of MAD	86
4.6	Evaluate the suitability of current system development methodologies for mobile application development.....	87
4.7	Summary	90
CHAPTER 5 DISCUSSION, COMMENTS AND CONCLUSION		91
5.1	Introduction	91
5.2	Investigate the challenges faced by companies which are currently developing mobile applications.....	91
5.3	Identify different system development methodologies that are available for mobile application development, for example Mobile-D, MASAM and GMA.	92
5.4	Investigate if system development methodologies are being used for MAD in the IT industry.....	92
5.5	Determine the success of MAD.	93
5.6	Evaluate the suitability of current SDMs for MAD	93
5.7	Contribution	93
5.8	Future work and limitations	93
5.9	Summary	94
BIBLIOGRAPHY		95
APPENDIX - QUESTIONNAIRE		101
PROOF OF LANGUAGE EDITION		107

LIST OF TABLES

Table 2-1:	Definitions of SDM	7
Table 2-2:	Similarities and differences between definitions	9
Table 2-3:	Principles behind the agile manifesto	15
Table 2-4:	Empirical studies on the use and effectiveness of SDMs and MAD	20
Table 2-5:	Classification of mobile application	24
Table 2-6:	Activity task	36
Table 2-7:	Activity task	37
Table 2-8:	Activity task	39
Table 2-9:	Similarities and differences between Mobile-D, MASAM, GMA, MODA-TEL and Mansurov's accelerated development methodology.....	45
Table 3-1:	Link between research objectives and research questions.....	57
Table 3-2:	Link between the research questions and the statistical technique.....	64
Table 3-3:	Summary of the country of the respondent.....	68
Table 3-4:	Home language.....	69
Table 3-5:	Gender.....	70
Table 3-6:	Age	71
Table 3-7:	Personal mobile development experience	71
Table 3-8:	IT working experience	72
Table 3-9:	Current position.....	72
Table 3-10:	Size of company	73
Table 3-11:	Business area (where n= 169)	73
Table 3-12:	Size of the project team	73

Table 3-13:	Open vacancies	74
Table 3-14:	Users of mobile application	74
Table 3-15:	Platform	74
Table 3-16:	Development time	74
Table 3-17:	Application development environment.....	75
Table 3-18:	Primary motivation	75
Table 3-19:	Success of mobile application	75
Table 3-20:	Success of mobile application	76
Table 3-21:	Use of SDM.....	76
Table 3-22:	Future change of use	76
Table 4-1:	Descriptive statistics of the characteristics of mobile application	78
Table 4-2:	Results of exploratory factor analysis on characteristics items of mobile application.....	80
Table 4-3:	Result of factor analysis	81
Table 4-4:	Results of reliability test on factors retained.	82
Table 4-5:	Descriptive statistics of the retained factors.....	84
Table 4-6:	Descriptive statistics for SDM-Type and its intensity, strictness of use, time of use, possible change	85
Table 4-7:	Descriptive statistics' for Success of mobile application, Delivered on time and Completed in budget	87
Table 4-8:	Correlation between (successes of mobile application) and (strictness, time of use)	87
Table 4-9:	Correlation between (delivered on time) and (strictness, time of use).....	88
Table 4-10:	Correlation between (completed in budget) and (strictness, time of use).....	88

Table 4-11:	Factor analysis for outcome	89
Table 4-12:	Reliability test on Success factor	89
Table 4-13:	Correlation between (Success factor) and SDM use	89
Table 5-1:	Summary of the three methodologies.....	92

LIST OF FIGURES

Figure 2-1:	Mobile application distribution processes	25
Figure 2-2:	Positive feedbacks in the two-sided mobile application market Development tools	25
Figure 2-3:	Telco-centric platform model: the Vodafone Live case	29
Figure 2-4:	Device – centric platform model: the iPhone Case	30
Figure 2-5:	Aggregator-centric platform model: the Facebook Mobile Case	32
Figure 2-6:	Service-centric platform model: the Open Social Case.....	33
Figure 2-7:	A diagram to sketch the GMA three-tier architecture	41
Figure 2-8:	The GMAApp development flow	41
Figure 2-9:	Phases in Mansurov’s accelerated development methodology	43
Figure 2-10:	Processes of accelerated development methodology.....	44

CHAPTER 1 INTRODUCTION

1.1 Introduction

This chapter discusses the problem statement, research aims and objectives and the research method and provides an outline of the study.

1.2 Problem statement

SDMs have been in use since 1970 (Avison & Fitzgerald, 2006). MAD is a new direction in system development. Organizations need high quality mobile applications to provide them with strategic advantages.

Development in mobile technologies can be classified into various generations. The first generation (1G), with low traffic capacity and poor voice quality and security, was introduced in 1950. The main difference between the 1G and the second generation (2G) is the introduction of digital technology, caller ID and the short message service. The second-and-a-half generation (2.5G) had web browsing and navigational capabilities. High data transmission requirement was the reason behind the introduction of the third generation (3G). Improvements in 3G in terms of efficiency speed and coverage paved way for the fourth generation (4G) and the fifth generation (5G) (Ayob *et al.*, 2009; Bangerter *et al.*, 2014). Dependency on mobile devices is increasing and more and more applications are introduced on a daily basis. Mobile applications can be classified into business applications, entertainment applications and personal applications.

To our knowledge there are currently only a few mobile application SDMs available. Mobile-D (Abrahamsson *et al.*, 2004), MASAM (Jeong *et al.*, 2008) and GMA (Cheng *et al.*, 2007) are some of the SDMs available for MAD. Some of the drawbacks of the above SDMs are:

- Mobile-D is suitable for a team of less than 10 people working together and not suitable to adapt to a rapid release cycle. Weak tool support is also another issue with the Mobile-D SDM.
- MASAM requires much time from planning to implementation.
- There are many security concerns when using GMA.

In order to produce high quality mobile applications, we need suitable SDMs. It is not clear whether MAD companies are using any SDMs.

When considering the above factors, one cannot ignore the necessity of investigating the use and effectiveness of system development methodologies in mobile application development.

This study is aimed at finding out the use and effectiveness of SDMs in MAD.

1.3 Research aims and objectives

The main aim is to determine the use and effectiveness of SDMs in MAD.

In order to achieve this aim the following objectives must be achieved:

- Investigate the challenges faced by companies that are currently developing mobile applications.
- Identify different SDMs that are available for MAD, for example Mobile-D, MASAM and GMA.
- Investigate if SDMs are being used for MAD in the IT industry.
- Determine the success of MAD.
- Evaluate the suitability of current SDMs for MAD.

1.4 Research method

This research was performed within the positivistic research paradigm. This included a survey. Data collection was done via questionnaires. Descriptive statistics, factor analysis, reliability and correlation coefficients were used as data analysis methods.

The research focused on all companies around the world that use system development methodologies for mobile application development. The research method included the following steps:

- Compiling a database of companies around the world doing mobile application development. This was done by using Google, market research reports, government list of companies, etc.
- Compiling a questionnaire to conduct the survey.
- Approaching project managers of selected companies through email or phone asking them for permission to conduct the survey through questionnaires.
- Using statistical methods to analyse the obtained data.

1.5 Outline of the study

The dissertation outline is as follows:

Chapter 1: Introduction – This chapter discusses the reasons why this topic is relevant for research and the aims and objective and research method are explained.

Chapter 2: Literature Study – This chapter presents a literature review of SDMs, MAD and SDMs for MAD.

Chapter 3: Research Method – The chapter offers an explanation of the research paradigm, research method, data collection and data analysis.

Chapter 4: Results – The results obtained are discussed.

Chapter 5: Discussion, Comments and Conclusion – The comments in this chapter are based on the results.

1.6 Summary

This chapter explained the problem statement, research aims and objectives, research method and outline of the study. The chapter serves as a foundation of the study. The next chapter presents the literature study. The main topics addressed include SDM, MAD and SDM for mobile application.

CHAPTER 2 LITERATURE STUDY

2.1 Introduction

SDMs have been in use since 1970 (Avison & Fitzgerald, 2006). It is important to study evolution of SDM to analyse the current trends in SDM.

This chapter discusses SDM and MADM. The following aspects are included. The blue part describes SDM, yellow part describes mobile application and the grey part describes both SDM and mobile application.

- The history of SDM
- The definition of SDM.
- A classification of SDM.
- Empirical studies on the use and effectiveness of SDM and MAD.
- The history of mobile technology.
- The history of mobile applications.
- The definition of mobile application.
- Classification of mobile applications
- Mobile application development
- Mobile application distribution
- Different types of platforms
- SDM's for mobile application.
- Success factors/Critical characteristics/Challenges of mobile application success and limitations for mobile application

2.2 The history of SDM

This section discusses the history of the emergence of the SDM. There are four identified eras (Avison & Fitzgerald, 2006:2).

- The “Pre-methodology era”(Avison & Fitzgerald, 2006:2)
- The “Early methodology era”(Avison & Fitzgerald, 2006:2)
- The ”Methodology era”(Avison & Fitzgerald, 2006:3)
- The “Post-methodology era” (Avison & Fitzgerald, 2006:7).

2.2.1 The “Pre-methodology era” (Avison & Fitzgerald, 2006:2)

Previously, the International Federation for Information Processing Technical Committee 8 (IFIP **TC8**) computer applications were developed without using SDM. This period is referred to as the pre-methodology era (Avison & Fitzgerald, 2006:2). In this era the emphasis was on programming when we consider computer applications development. Due to the restrictions on memory efforts was mainly to develop a runnable application by overcoming the limitations. In this era even though the developers were technically trained, they were not a good communicators (Avison & Fitzgerald, 2006:2). Methodology involved a “rule-of-thumb and based on experience” (Avison & Fitzgerald, 2006:2), which resulted in poor project control and management. Projects were often delivered late and completed over-budget.

Another problem included overwork, spending most of the time on correcting and improving applications (Avison & Fitzgerald, 2006:2). All these problems lead to the need of a well-organized approach for the development of information system (IS) (Avison & Fitzgerald, 2006:2). This resulted in the establishment of the first information system development methodologies.

2.2.2 “Early methodology era” (Avison & Fitzgerald, 2006:2)

The problems of the pre-methodology era lead to the following reactions:

- The importance or need for the position of a systems analyst came as a result of the increasing gratitude for the analysis and design part in system development (Avison & Fitzgerald, 2006:2).
- As the size and complexity of the organizations grew, people realized the necessity of moving away from on-off solutions for a particular problem to more integrated IS (Avison & Fitzgerald, 2006:3).
- The need for an accepted methodology came for the information system development (Avison & Fitzgerald, 2006:3).

The systems development life cycle or waterfall model, which included phases, procedures, tasks, rules, techniques, guidelines, documentations, training programmes and tools, was developed as a result of the above situations. It can be called the methodology era (Avison & Fitzgerald, 2006:3). The waterfall model consists of different stages like the feasibility study, systems investigation, analysis, design, and implementation, followed by review and maintenance (Avison & Fitzgerald, 2006:3). The use of SDLC helped to keep documentation standards. This lead to the use of a complete proposal that is also communicated to both users and computing staff .This made sure that the users were trained to use the system. The project

was divided into different phases of manageable tasks with deliverables that helped with avoiding missed cutover dates and disappointments regarding the delivered item. There is subsequently no unexpected high cost and the analysts received standard training to ensure continuity of standards (Avison & Fitzgerald, 2006:3).

But there were limitations for this approach. The following are some of the criticisms.

- It didn't fully meet the requirements of the management.
- Unambitious systems design was the another issue.
- Instability was another problem because of the frequent modification in businesses and the organization environment.
- There was inflexibility because the design processes were of an output-driven orientation.
- Because of the problems with the documentation and because there was no option for the user to see the system before it is operational, there was much user dissatisfaction.
- Because of the failure in up-to-date document update, there were issues with documents.
- In order to meet the user needs they tried to change the system, which resulted in application backlog mainly because of maintenance workload.
- "The Assumption of 'green field' development" (Avison & Fitzgerald, 2006:3). This is because new computerization of manual system.

2.2.3 The "Methodology Era "(Avison & Fitzgerald, 2006:3)

As a result of the above criticism regarding SDLC, a number of IS development approaches emerged and this became the start of the methodology era (Avison & Fitzgerald, 2006:3). According to Avison and Fitzgerald (2006:3), there are a number of movements regarding methodologies. The first movement includes all the methodologies that are an improved version of waterfall model (Avison & Fitzgerald, 2006:3). The second movement includes all new methodologies different from waterfall model (Avison & Fitzgerald, 2006:4). In this era there was an explosion or growth of SDM.

2.2.4 The "Post-Methodology Era" (Avison & Fitzgerald, 2006:7)

According to Avison and Fitzgerald (2006:7), the post-methodology era is the current situation because currently methodologies have moved beyond the pure methodology era. In this era different directions in methodology have developed. The different directions are the following

- The "Ad hoc development" (Avison & Fitzgerald, 2006:586)

- Further developments in the methodology arena (Avison & Fitzgerald, 2006:588)
- “Contingency” (Avison & Fitzgerald, 2006:587)
- “External development” (Avison & Fitzgerald, 2006:588)

Ad hoc development can be described as a return to the approach used in the pre-methodology days. They use the skills and experience of the developers (Avison & Fitzgerald, 2006:586). Another direction in the methodology is to continue developing methodologies. Some developers ascribe importance to web application, some to customer relationship management etc. Another direction is the movement towards external development with the use of packages and outsourcing. Another direction is towards contingency. In contingency a structure is presented, and the situation dictates whether to accept the tools and techniques or not.

MAD is a new direction in system development. Therefore MADM can be considered as a contingency in SDM.

2.3 SDM definition

Below all the definitions for SDM during 2000 to 2013 are listed. Eight definitions for SDM were found. Table 2-1 contains all the definitions and name of the authors and the year.

Table 2-1: Definitions of SDM

Year	Authors	Systems development methodologies
2002	Riemenschneider, Hardgrave and Davis	<i>“The methodology is comprehensive, providing process directions for the complete life cycle. It includes specifications for the use of tools at various project stages by various parties, including users. The methodology, while comprehensive, is adaptable to many different project types (e.g., small, large; new development, maintenance)”</i> (Riemenschneider et al., 2002:1143).
2003	Avison and Fitzgerald	<i>“Methodology is a recommended collection of phases, procedures, rules, techniques, tools, documentation, management, and training used to develop a system”</i> (Avison & Fitzgerald, 2003).
2006	Huisman and livari	<i>“We define a systems development methodology as a combination of the following: A systems development approach: This involves the philosophical view on which the methodology is built. It is the set of goals, guiding principles and beliefs, fundamental concepts, and principles of the systems development process that drive interpretations and actions. Examples are the structured, object-oriented and information modelling approaches.</i>

		<i>A systems development process model: A process model is a representation of the sequences of stages through which a system evolves. Some examples are the linear life-cycle model and the spiral model.</i> <i>A systems development method: A method is a systematic way of conducting at least one complete phase of systems development, consisting of a set of guidelines, activities, techniques, and tools, based on a particular philosophy and the target system. Examples include OMT, IE, etc.</i> <i>A systems development technique: Development techniques can be defined as procedures, possibly with a prescribed notation, to perform a development activity for example construction of entity relationship diagrams”(Huisman & Iivari, 2006).</i>
2009	Mnkandla	<i>“Software engineering methodologies can therefore be defined as a group of methodologies used in the development of applications. Methodologies will give details of what should be done in each phase of the software development process”(Mnkandla, 2009).</i>
2009	Jiang, Liu, and Zhao	<i>“‘Method’ or ‘Methodology’ refers to a set of related ‘methods and technologies’ that follow some ‘principals and strategies’. They can be ‘development methodology’ and ‘process methodology’. ‘Principals and strategies’ can be some development strategies, such as the well-known ‘functional decomposition’ strategy and the process strategies such as the iterative model. Development method contains the process-oriented development method and object-oriented development method, and it also includes formalization method and Clean room Software Engineering”(Jiang et al., 2009).</i>
2010	Dahiya and Jain	<i>“Software-development life-cycle methodologies provide the what (processes and deliverables), how (techniques), and who (roles) for every typical role in a software-development project, such as solution architects, business consultants, and developers”(Dahiya & Jain, 2010).</i>
2012	Hijazi, Khmour and Alarabeyyat	<i>“A software development methodology or a software development process model is an approach to the Software Development Life Cycle (SDLC) that describes the sequence of steps to be followed while developing software projects”(Hijazi et al., 2012).</i>
2012	Lam	<i>“Methodologies for application development are step-by-step procedures to carry out the development activities in different phases of a system development life cycle. A methodology has its own assumptions about the reality, its own techniques to support working principles and enforce discipline, and its own tools to generate the deliverables for activities. In other words,</i>

there are a collection of specific techniques and tools for a certain development methodology”(Lam, 2012).

2.3.1 Similarities and differences between definitions

Nine factors were identified from the definitions of the eight different authors. These are shown in Table 2-2.

Table 2-2: Similarities and differences between definitions

N o	Factors	LM 2012	HKA 2012	DJ 2010	YLC 2009	ME 2009	HI 2006	DG 2003	RHD 2002
1	System development life cycle	Yes	Yes	Yes					Yes
2	Tools	Yes					Yes	Yes	Yes
3	Techniques / Technologies	Yes		Yes	Yes		Yes	Yes	
4	Process		Yes	Yes	Yes	Yes	Yes		Yes
5	Step-by-step procedures /steps to be followed/ sequences of stages	Yes	Yes				Yes		
6	Who (roles) { solution architects, business consultants, developers, including users}			Yes					Yes
7	What { processes and deliverables}			Yes	Yes	Yes			
8	Activities	Yes					Yes		
9	System development approach		Yes				Yes		

From Table 2-2 we can see that some definitions are very complete and include many aspects, and some include only a few elements. For example, Huisman and livari's definition includes many elements like Tools, Techniques / Technologies, Process, Step-by-step procedures /steps to be followed/sequences of stages, Activities, System development approach. However, Mnkandla's definition includes only a few elements like Process, What {processes and deliverables}.

2.3.2 Definition used in this project

This definition is based on the above discussion regarding the similarities and difference between definitions of different authors. The new definition is stated below.

Systems development methodology can be defined as a philosophical view on the systematic process flow of activities that should be carried out during the different phases of development life cycle. It can be defined using the following five Ws.

- Who is responsible (solution architects, business consultants, developers, users etc.)?
- What are the processes, specifications and goals?
- Why/How will it happen (guidelines, activities, techniques, and tools)?
- When will it take place (phases)?
- Where do the above four Ws fit in a system development life cycle?

2.4 Classification of SDM based on philosophical paradigm

A SDM can be classified into seven categories based on the philosophical paradigm. The philosophical paradigm guides the process model, method, and tools and techniques used in a SDM. The following are the different categories.

- Process-oriented methodologies
- Blended methodologies
- Object-oriented methodologies
- Rapid development methodologies
- People-oriented methodologies
- Organizational-oriented methodologies
- Frameworks (Avison & Fitzgerald, 2006:389)

Process-oriented methodologies

The methodologies in this category make use of the processes-oriented techniques. Functional decomposition, data flow diagrams, decision trees and decision tables are some of the processes-oriented techniques used by the methodologies.

Blended methodologies

Blended methodology is created by blending together other methodologies. Blended methodologies use a blend of processes and data-oriented techniques. It is also called data-oriented methodologies.

Object-oriented methodologies

Object-oriented analysis (OOA) and Rational Unified Process (RUP) belongs to object-oriented methodology (Avison & Fitzgerald, 2006:451). They use objects, which is a combination of both data and processes as an approach to develop information systems. The classes of OOA ensure increased resilience to change, as old code can be reused as much as necessary. OOA has an added advantage through its use of abstraction, which allows it to only adopt aspects that are important to solving a problem.

Rapid development methodologies

The Rapid Application Development (RAD) is for the fast delivery of information systems. The RAD is also a solution for changing requirements during the development process (Avison & Fitzgerald, 2006:128). The approach involves an initial investigation, requirements definition, design, development and testing. The end product/prototype is submitted to the end users, others interested and the essential stakeholders who make the changes necessary, if at all, where after implementation and maintenance take place. RAD SDMs include the agile family Extreme Programming, Scrum, Lean Development, Dynamic Systems Development Method and Web Information Systems Development Methodology.

People-oriented methodologies

As the name indicates, people-oriented methodologies ascribe importance to the people who participate in the development of information systems and software. These methodologies make sure that they represent expertise and know-how of the people in the organization who

are affected by a new information system. Examples of people-oriented methodologies include ETHICS and KADS.

Organizational-oriented methodologies

Organizational-oriented methodologies are those that view the development of an information system as a whole, therefore adopting systems thinking, as opposed to the scientific paradigm, which breaks it down into bits and pieces. The systems thinking is used in social and management sciences where the human aspect is added. It is complicated for the human aspect to be broken down to be examined individually when each plays a part in the whole. In other words, there is no black and white, rather but grey, and they are all connected and mixed into one (Avison & Fitzgerald, 2006). The SDMs in this category include SSM, ISAC and PI.

Frameworks

According to Avison and Fitzgerald (2006), Frame is a contingent methodology that allows for different approaches depending on situations.

The section below discusses three of the above categories – traditional, agile, agent, because they are mostly used in MAD (Rahimian & Ramsin, 2008)

2.4.1 Traditional SDM

Traditional SDM is falls under the process-oriented methodology. The following is the basic structure of traditional SDM.

- Feasibility study
- System investigation
- System analysis
- System design
- Implementation
- Review and maintenance (Avison & Fitzgerald, 2006:31).

- Feasibility study

During this stage the team studies the present system, requirements of the present system, problems in meeting those requirements, and the investigation of alternative solutions to those

problems. A description must be given in terms of technical, human, organizational and economic costs and benefits of developing and operating the system. So the proposed system must be legally, organizationally, socially, technically and economically feasible. All the information, including the recommended solutions, are given to the management as a formal report and oral presentations are done by the systems analysts. The management decides whether to accept or reject the analyst's proposal (Avison & Fitzgerald, 2006:31).

- System investigation

If the management approved to proceed, the next step is the detailed investigation regarding the functional requirements of the current system, requirements of the new system, constraints, data types and the amount that has to be processed, exception conditions and problems of the current working methods. All these data are obtained by observation, interviewing, and questionnaires, searching records and documentation and sampling (Avison & Fitzgerald, 2006:32).

- System analysis

In this step the system analyst will do systems analysis by asking several questions like why do the problem exist? Why were certain methods of work adopted? Are there alternative methods? And what are the likely growth rates of data? (Avison & Fitzgerald, 2006:33).

- Systems design

The design depends on the investigation phase and questioning in the analysis phase, but the new design is similar to the previous system. The design of the computer and manual parts of the system were designed in this phase. The design documentation set includes input data and how it should be captured, outputs of the system, processes that are involved in converting the inputs to the outputs, structure of the computer and manual files that might be referenced in the system, security and backup provisions and systems testing and implementation plans (Avison & Fitzgerald, 2006:33).

- Implementation

The implementation phase includes many activities. If there is a new computer program it should be written and tested. New hardware and software should be purchased and installed if there is a need. Quality control is a major aspect of the implementation phase. Education and training of users is another major element of this phase. Documentation is another part of this

phase. Security procedures need to be tested during this phase. After all these procedures the new system can be started and the old one can be discontinued (Avison & Fitzgerald, 2006:34).

- Review and maintenance

The final stage in the system development process occurs after the system starts working. For the continued efficient running of the system, maintenance has to be done. The changes can be due to changes in organization or advancement in the technology. Maintenance can be also be due to correction of error. The evaluation of the process can help to improve the way to develop other new systems through the process of organizational learning (Avison & Fitzgerald, 2006:34).

2.4.1.1 Examples of traditional SDM

The following are some of the examples of traditional SDM

- SDLC
- STRADIS

2.4.1.2 Advantages of traditional SDM

The following are the advantages of traditional SDM

- Thorough requirements definition
- Design has been proven
- Documentation emphasis
- Planning details
- Known quantities (Rudnick, 2013)

2.4.1.3 Disadvantages of traditional SDM

The following are the disadvantages of traditional SDM

- Lack of flexibility for change
- It could be at the expense of another area
- Change is inhibited
- Test is compressed
- Less opportunity for innovation (Rudnick, 2013).

2.4.2 Agile system development methodologies

ASDM resorts under rapid development methodology. According to Moniruzzaman and Hossain (2013:2), it refers to a set of software development methods based on iterative and incremental development processes. Requirements and developments evolve through collaboration between self-organizing, cross-functional teams that allow rapid delivery of high quality software to meet customer needs and to accommodate changes in the requirements. Agile software development is emerging as an alternative to traditional plan-based software development methods.

According to Klimes and Prochazka (2006:2), there are some main common principals of agile system developments. It includes “tight, every-day cooperation between users and the development team, simplicity, early and continuous delivery of valuable software, test-driven development and changing requirements during development” (Klimes & Prochazka, 2006:2).

2.4.2.1 Principles behind the agile manifesto

Principles behind the agile manifesto are given in Table 2-3 below.

Table 2-3: Principles behind the agile manifesto (Beck et al., 2001)

No	Principles
1	<i>“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software” (Beck et al., 2001).</i>
2	<i>“Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage” (Beck et al., 2001).</i>
3	<i>“Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale” (Beck et al., 2001).</i>
4	<i>“Business people and developers must work together daily throughout the project” (Beck et al., 2001).</i>
5	<i>“Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done” (Beck et al., 2001).</i>
6	<i>“The most efficient and effective method of conveying information to and within a development team is face-to-face conversation” (Beck et al., 2001).</i>
7	<i>“Working software is the primary measure of progress” (Beck et al., 2001).</i>
8	<i>“Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely” (Beck et al., 2001).</i>
9	<i>“Continuous attention to technical excellence and good design enhances agility” (Beck et al., 2001).</i>
10	<i>“Simplicity--the art of maximizing the amount of work not done--is essential” (Beck et al., 2001).</i>

11	<i>"The best architectures, requirements, and designs emerge from self-organizing teams"</i> (Beck et al., 2001).
12	<i>"At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly"</i> (Beck et al., 2001).

2.4.2.2 Examples of Agile SDM

The following are some of the most used ASDM's.

- Extreme Programming (Beck, 2000)
- Scrum
- Dynamic Systems Development Method
- Adaptive Software Development (Highsmith, 2000)
- Crystal
- Feature-Driven Development

2.4.2.3 Advantages of agile system development methodologies

- According to Pikkarainen *et al.* (2008:332), there is a positive effect on the communication inside the development team when using agile practices.
- The use of agile methodology improves external communication. This helps the dependencies between the tasks–subtasks, feature–requirements between software development teams and stakeholders (Pikkarainen *et al.*, 2008:332).
- When using agile methodology for the development, it allows teams to welcome changes, it meets the requirements of the user, it enables teams to do quicker development and the user receives the exact system they need (Klimes & Prochazka , 2008:5).

2.4.2.4 Disadvantages of agile system development methodologies

- The main problem of agile methodology is that some of the agile methodology is only suitable for green field engineering and not for maintenance and evolution (Klimes & Prochazka2008:2).
- Another disadvantage is that the users also get the responsibility for the project success because users must communicate and cooperate with the development team every day (Klimes & Prochazka, 2008:2).

- In agile methodology developers must be able to write code that is readable. It should be according to the standards, and it should be with valuable comments. The developers also should use simplicity in coding and designing (Klimes & Prochazka, 2008:3). The skills and behaviour of the developers can therefore affect the methodology (Klimes & Prochazka, 2008:2).

2.4.3 Agent methodology

Agent methodology resorts under frameworks. According to Jennings and Wooldridge (1998), "An agent is a computer system that is situated in some environment and that is capable of some autonomous action in that environment in order to meet its design objectives".

"Agent Oriented Programming is most often motivated by the need for open architectures that continuously change and evolve to accommodate new components and meet new requirements" (Bresciani *et al.*, 2002:203).

2.4.3.1 Examples for agent methodology

- MaSE-Multiagent Systems Engineering (Dam & Winikoff, 2003:78)
- Prometheus (Dam & Winikoff, 2003:78)
- Tropos (Bresciani *et al.*, 2002)

The aim of multi-agent systems engineering is to create a complete life cycle methodology to help system developers to design a multi-agent system and to develop it. It explains all the processes from the initial stage to the implementation of the system.

Prometheus is a detailed agent-oriented software engineering methodology for non-experts like undergraduate students, which consists of three steps:

- System specification
- Architectural design
- Detailed design

Tropos is an agent-oriented software development methodology. One of the significant differences between Tropos and the other methodologies is its strong focus on early requirements analysis. Tropos consists of five different phases in its development. They are early requirements, late requirements, architectural design, detailed design and implementation.

2.4.3.2 Advantages of agent methodology

- Autonomous character of agent.
- Agent-oriented software engineering is not only for “agent systems”, it is suitable for a wide class of scenarios and applications (Zambonelli, 2010).
- The use of agent modelling to reduce the complexity of managing the emergent collective behaviour of the agents.
- Programming guidelines provided by the agent models and the ACL.
- Stratification of change into model translation, the ACL, and the inner language.
- Programming collaboration based on asynchronous text message parsing, including error messages.
- Programming tools based on reusable agents and language components (Petrie, 2001).

2.4.3.3 Disadvantages of agent methodology

The most important outstanding issues for agent-based software engineering are:

- an understanding of the situations in which agent solutions are appropriate;
- principled but informal development techniques for agent systems

2.5 Empirical studies on the use and effectiveness of SDMs and MAD

In 2003 Shine technologies (Johnson), conducted an international survey to determine the effects of using agile methods. They came up with the following results if using agile methods:

- Lower costs (Johnson , 2003)
- Better productivity (Johnson, 2003)
- Better quality (Johnson, 2003)
- Better business satisfaction (Johnson, 2003)

Another survey done in 2006 by Ambysoft (Ambler) to find the effect of using agile methods and found that 41% were using agile methodology , 65% using agile techniques ,38% reported increase business stakeholder satisfaction (Ambler, 2006).

Ambysoft did another survey in 2007 with 781 respondents and found that 69% respondents used agile techniques (Ambler, 2007). The percentage of respondents who experienced more than 90% success rate for agile projects was 44% (Ambler, 2007). The percentage of respondents with a success rate between 75% and 90% was 33% (Ambler, 2007). The

percentage of respondents who worked on adopted iterations is 98.6% (Ambler, 2007). Respondents reported iteration lengths between one and four weeks is 83% (Ambler, 2007). The survey also found that the success rate was higher in small teams than larger teams (Ambler, 2007).

In 2008 Amblysoft came with the following result from 642 respondents (Ambler, 2008).

- 69% of respondents reported that their organizations are doing one or more agile projects (Ambler, 2008).
- 61% of developers reported that their organizations do agile (Ambler, 2008). The percentage of management that thinks their organizations are doing agile are 78% (Ambler, 2008).
- The organizations doing agile beyond the pilot project phase were 82% (Ambler, 2008).
- Most of the respondents reported that when they use agile, their teams produce higher quality, higher productivity and they enjoy higher satisfaction (Ambler, 2008).
- An 82% success rate was reported for co-located teams and a 72% success rate was reported for closely located, and 60% success rate reported for significantly distributed teams (Ambler, 2008).
- The percentage of teams with a four week or less iteration length was 84% (Ambler, 2008).
- The percentage of people who experienced higher average costs when using agile was 23%, people experiencing no change in costs was 40% and very low cost was 37% (Ambler, 2008).
- When using agile the success rate was higher for co-located agile projects than non-co-located and success rate of non-co-located was higher than off-shore projects (Ambler, 2008).

In 2013 Joorabchi *et al.* conducted another survey to find out the real challenges in mobile application development. The following are their main findings.

- 76% reported the existence of multiple platforms as a challenge.
- 64% participants tested their mobile application manually (Joorabchi *et al.*, 2013).

In 2014 Dobb conducted another survey, named Dr Dobb's App Dev Priorities Survey.

The following are the main findings regarding SDM:

- 69% of the respondents were using agile development methodologies (Don, 2014).

Table 2-4: Empirical studies on the use and effectiveness of SDMs and MAD

Year	Company	No of Respondents	Result
2003	Shine technologies	131	49% stated cost reduction. 93% stated better productivity. 88% stated increase in quality. 83% reported increase in business satisfaction.
2006	Ambyssoft	4232	41% were using agile methodology. 65% using agile techniques. 38% reported increase business stakeholder satisfaction.
2007	Ambyssoft	781	69% responders used agile techniques 44% got a success rate of above 90% 33% got a success rate between 75 and 90% 98.6% of agile teams worked adopted iterations and 83% had iteration lengths between 1 and 4 weeks. The success rate is higher in smaller teams than larger teams.
2008	Ambyssoft	642	69% used agile techniques. 37% stated cost reduction. 82% stated better productivity. 77% stated increase in quality. 78 % reported increase in business satisfaction.
2013	Joorabchi	188	76% reported the existence of multiple platforms as a challenge. 64% participants are testing their mobile application manually.
2014	Dobb	528	69% of the respondents are using agile development methodologies.

From the above Table 2-4 it is clear that there are not many studies specifically examining MADM. Therefore it is necessary to conduct studies regarding MADM.

2.6 The history of mobile technology

Cell phone history can be traced back to radio telephony used in United States in 1920. In 1947 AT&T launched a car cell phone on the highway between Boston and New York. Limited frequency spectrum was the biggest problem with this network. The resourceful use of the frequency space helped to reduce the scarcity problem in modern automatic mobile

telecommunications (Dunnewijk & Hultén, 2007) Development in mobile telecommunication technologies can be classified into five generations (Dunnewijk & Hultén, 2007) as discussed below.

- The first generation (1G)

This generation can be traced back to 1950s. Analogue circuit-switched technology with FDMA (Frequency Division Multiple Access) was the basis of this generation and worked mainly in the 800-900 MHz bands. The weaknesses of this generation were low traffic capacity, unreliable handover, poor voice quality, and poor security.

- The second generation (2G)

This generation is also called GSM technology. This generation used digital technology. Even though 1G was the foundation of 2G, several new features like caller identity and messaging was introduced in 2G.

- 2.5G

This generation is also known as Generation Enhanced. This generation is in between 2G and 3G, and was a stepping-stone towards 3G. This generation provides a higher data rate and additional capabilities compared to 2G, such as packet-switched connection (GPRS) in the TDMA-based GSM system, and enhanced data rates (HSCSD and EDGE), web browsing, the use of navigation and navigational maps, voice mail, fax, and the sending and receiving of large email messages.

- The third generation (3G)

Japan was the first country to introduce a 3G system. The advantages of 3G systems are higher data transmission rates, increased capacity, voice calls.

3G is based on packet-switching technology. 2G infrastructure has to change to accommodate 3G. Many features like live TV streaming, web browsing, e-mail, fax, navigation systems and video conferencing are available in 3G.

- 4G

This is the fourth generation of wireless communication. 4G is faster than 3G. There are two versions in this generation, 4G and 4G LTE. 3G or 4G network performance is evaluated on “hard” metrics, including peak data rates, coverage, and spectral efficiency.

- 5G

This is the next generation in mobile technology. It is also called the fifth generation. The transition period from 4G to 5G is called 5G Era (Bangerter et al., 2014). 5G will be 1000 times faster than the 4G mobile network. When using 5G, users will be able to quickly download films without feeling any delay.

2.7 History of mobile applications

Mobile application is not a new concept, it was available in different platforms and different versions for years (Tracy, 2012:31).

Generation one cell phones are now regarded “candy bar” phones (Tracy, 2012:31). They came with small screens like a large candy bar and they had minimal data service available for usage. Applications were developed for specific phones. Phones supporting wireless application protocols can be considered second generation phones. HTML variant wireless mark-up language (WML) was supported by these phones. It was still challenging for a developer to create applications for this generation due to differences in screen size etc.

Smart phones can be regarded as the 3rd generation phones. They were considered less application-based phones. Smart phones introduced strict control measures on its applications and it first targeted areas such as messaging and personal information management (PIM) functions like calendars and contacts.

The popularity of iPhone was behind the current generation of phones called touch-screen phones. They are application-based phones. These phones are treated as the development platform for various applications. Large display phones really helped the growth of mobile applications. Lighter Tablets are now becoming more popular and they are considered as a computing device with extra applications like messaging and PIM features. They are usually designed to support Wi-Fi 802.11 in addition to cellular 3G/4G networks.

2.8 Definition of mobile application

Mobile applications were created to satisfy the requirements of a growing population that uses cell phones to carry out various activities in day-to-day life ‘any time anywhere’ (Gasimov *et al.*, 2010).

Mobile applications can be defined as the software applications working on mobile devices, which make access to information anywhere and anytime a reality (Zhang & Adipat, 2009:294).

Mobile apps are currently evolving rapidly, but are generally targeted at a specific task, run on a mobile device, make heavy use of the data network, and have a simple delivery mechanism (i.e., Google Play Store or Apple Store) (Tracy,2012:30).

2.9 Classification of mobile applications

According to Gasimov *et al.* (2010:79) mobile applications can be classified into five categories. This includes:

- **Transactional**
The transactional category includes all the conversational, commercial and tracking and tracing applications. Examples include eBuddy, Xumii, PicTicket, Child Locate etc.
- **Content dissemination**
The content dissemination category includes all the user-requested information dissemination, advertising and mobile content sharing, for example Google map, Yahoo Weather etc.
- **Social networking**
The social networking category includes all the networking applications and entertainment applications, for example Crush or Flash, Facebook etc.
- **Personal productivity**
The personal productivity category includes all the applications that improve personal productivity, for example Microsoft Office Mobile, Remember the Milk etc.
- **Leisure**
The leisure category applications include all applications that give personal leisure time and that do not do any social exchanges with other, for example textonphone.com

Table 2.5 gives the classification of mobile application.

Table 2-5: Classification of mobile applications (Gasimov *et al.*, 2010:79)

Category	Domain	Key characteristics	Implementations
Transactional	Conversational	Short term, as-needed conversations	Namabuzz (namabuzz.com), eBuddy (ebuddy.com), Xumii (xumii.com), Truphone (truphone.com), Fring (fring.com)
	Commercial	Information and/or monetary exchanges involving one or more parties	Digby (digby.com), Mobiqa (mobiqa.com), PicTicket (matrixsolutions.com), Luupay (luupay.de)
	Tracking and tracing	User tracking and tracing via location updates from mobile, satellite, or home location register (HLR)	Trace A Mobile (traceamobile.com), Child Locate (childlocate.co.uk)
Content dissemination	User-requested Information dissemination	Delivery of information requested by users	Google Map (www.google.com/mobile/gnm), Yahoo Weather, Stock, Mail (mobile.yahoo.com), Mobile Campus (mobilecampus.com), Shazam (shazam.com), Kooaba (kooaba.com), Swiss Railways (SBB) (http://mct.sbb.ch/mct/fr/reisezeit/mobiletools/sms.htm)
	Advertising	Information delivery for marketing purposes	Cellfire (cellfire.com)
	Mobile content sharing	Data exchange using Internet, WLAN or near field communications such as peer-to-peer (P2P) file sharing	Miraveo (miraveo.com), eMule (mobile.emule-project.net)
Social networking	Networking	Facilitate the forming and maintenance of social relationships, and organizing social activities	Limbo (Limbo.com), Crush or Flash (crushorflash.com), Rabble (rabble.com), JuiceCaster (juicecaster.com), Itsmy (itsmy.com), Mocospace (mocospace.com), Mobikade (www2.mkade.com), Twitter (twitter.com), Plazes (plazes.com), GyPSii (gypsii.com), Bedo (Bedo.cn), Facebook (m.facebook.com), Xing (mobile.xing.com), Nokia Sensor (nokia-asia.com/A4416020)
	Entertainment	Collective-based leisure activities undertaken using mobile device	The Club (www.apple.com/store/iphone/appstore)
Personal productivity		Support work related activities to improve personal productivity	Microsoft Office Mobile, Remeber The Milk (www.rememberthemilk.com), OmniFocus (www.omnigroup.com/applications/omnifocus)
Leisure		For personal leisure activities that do not involve social exchanges with others	Pandora (pandora.com), Books (textonphone.com)

2.10 Mobile application development

According to (Gasimov *et al.* (2010:74), MAD is characterized by the three points below:

- Maturity of mobile network (Gasimov *et al.*, 2010:74)

The maturity of mobile network plays an important role in mobile application development. Mobile network bandwidth can be increased with the new and improved protocols (see 2.6).

- Cell phone hardware should be the newest

Because of the high bandwidth and more user-friendly internet, mobile users easily become bored. New improved technologies in mobile hardware will increase the interest of mobile developers. Developers subsequently started to develop more and more new applications (Gasimov *et al.*, 2010:74).

- Rise in demand for mobile application

The increase in demand for mobile application will lead to the development of new application.

2.11 Mobile application distribution

Mobile application distribution can be defined as the process including the development of an application, marketing, purchasing and use of this application on a mobile platform. The three main components of this process are (i) development tools, (ii) portals and (iii) devices.

Figure 2-1 below explains that the increase in users for a specific platform or a device will bring more developers into this platform and more developers means more applications for customers, which will bring even more consumers. Developers will benefit from working with more popular devices using more popular platforms as it reaches the highest number of users. Consumers have the advantage of more applications to use if they buy more popular devices (Holzer & Ondrus, 2011). This will create a positive feedback loop, as depicted in Figure. 2-2

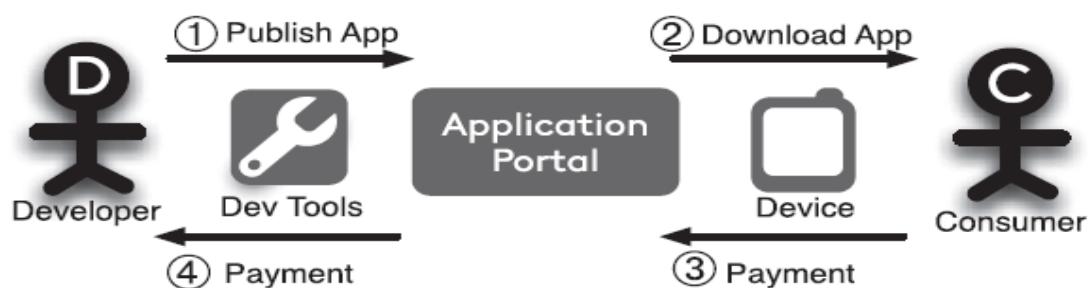


Figure 2-1: Mobile application distribution processes (Holzer & Ondrus, 2011:23)

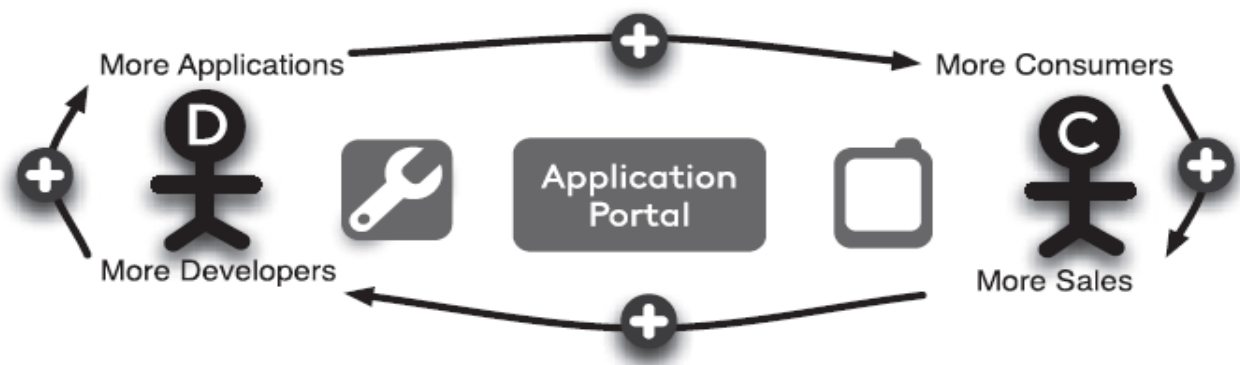


Figure 2-2: Positive feedbacks in the two-sided mobile application market Development tools (Holzer & Ondrus, 2011:24)

Development tools are nothing but software development kits (SDK) that help 3rd party developers to create applications for various platforms. They include libraries, debuggers, handset emulators etc. Platforms provide integrated development environments (IDE) to aid the development process. Some platforms have chosen to restrict access as much as they can, and other platforms have chosen to disclose the entire source code of their SDK and Operating System.

- Closed technology

In closed technology, the platform provider is in control of the strategic decisions about the platform. This is also known as cathedral model. Apple, RIM and Microsoft follow this approach and they have 50% of the world market share. The big disadvantage is that they have to employ people to develop applications for their devices.

- Open technology

In open technology platforms permit freelance developers to access SDK and OS source code. This method is also known as bazaar model. A central architect is not part of this platform. This model really motivates developers to work on this platform. This will bring down the development and maintenance cost of the operating system, in the process reducing the price of mobile applications. Linux and Google are examples of this approach.

Mobile Application Portal (MAP)

Portals work as links between developers and consumers. The MAP is a vital component in the application distribution process. An application is developed and made available to customers through an application portal. There are two different types of mobile application portals as discussed below.

1. Decentralized portal

If developers have no restriction in uploading the various applications onto the portals, those portals can be classified as decentralized portals. There is no central policy governing these portals. There is battle between various portals to bring more applications and with that more users. One of the disadvantages of this model is a lack of a full review on application at some of the portals. It will be difficult for a customer to choose the correct application. Platform providers do not need to preserve a centralized resolution as applications are uploaded mostly by freelancers. Microsoft, Nokia and Limo are using this approach.

2. Centralized portal

As the name indicates, one portal is suggested as the main portal where all the applications are uploaded. This will give the portal provider an added advantage as all the applications are loaded in one portal and customers can find the application with ease. Developers' job is also minimized as they have a one sales point. This will bring more developers and consumers.

Appstore by Apple and android by Google are examples of this approach. Apple, however, has a review process for uploading the applications while Google does not have any such restrictions on its portal. Google takes customer feedback to remove certain applications with low quality.

2.12 Different types of platforms

According to Ballon and Walravens (2008:103), defined sets of tasks combined with specific actions, activities and authorization to achieve a desired outcome can be defined as a business role.

Four roles can be identified in the mobile service platforms:

- Profile/Identity management: Management of customer data and customer preferences for multiple services and customer preferences for different user situations.
- Service provisioning/Service brokerage: The service broker defines areas where final users can retrieve, subscribe and use services and service components.
- Charging & billing: It is a necessity to have charging and billing components for the successful development of the architect.
- Service creation environment: Development and hosting tools for independent developer scan be part of this group.

A business actor facilitates or brings factors to achieve the abovementioned desired outcome.

The main actors in the field of mobile service provision are:

- Device manufacturer: They are responsible for design and manufacturing of mobile devices.
- Platform operator: They provide and operate a service platform.
- Meta Platform Operator: The role is to facilitate the information exchange between different platforms.
- Network operator: They provide network connectivity for devices.
- Portal provider: An entity that aggregates services and content, including service level authorization and single sign-on functions.
- Service provider: An actor that focuses on the production and operation of applications and services for end users.
- Service aggregator: An actor that offers a portfolio of services and bundles.

A stake holder can be an individual, institution, company or an organization with an interest in the outcome of certain actions.

The following are the different types of platforms.

- Telco-centric model (Ballon, 2009:12)
- Device-centric model (Ballon, 2009:13)
- Aggregator-centric model (Ballon, 2009:14)
- Service-centric model (Ballon, 2009:15)

2.12.1 Telco-centric model

The main characteristics of this model are:

- The client uses a portal screen to get services.
- The portal provider is part of network and platform operator. The technical tool to assist network access and portal provisioning is the responsibility of platform operator. Portal environment helps to finish service aggregation.
- Carriers will be able to make special deals to publish services on portals with service providers.
- The carrier is used as a medium between customers and service provider. Carrier gets money from user for access and service provider is paid by carrier for service delivery.

The network carrier is responsible for the four roles mentioned below, which can be classified as actor functions of platform operator, portal provider and network operator.

- Service creation: The mobile operator makes the final decision on the availability of a service through the network.
- Profile/Identity: The mobile network operator is in control of the activities of its customers. It will have records of identity of the user accessing the network.
- Service brokerage: Service developers and end users are connected by carriers. Carriers normally choose the service provider for the services for the portal.
- Charging & billing: Carriers charge the customer directly for access to the network and to the portal.

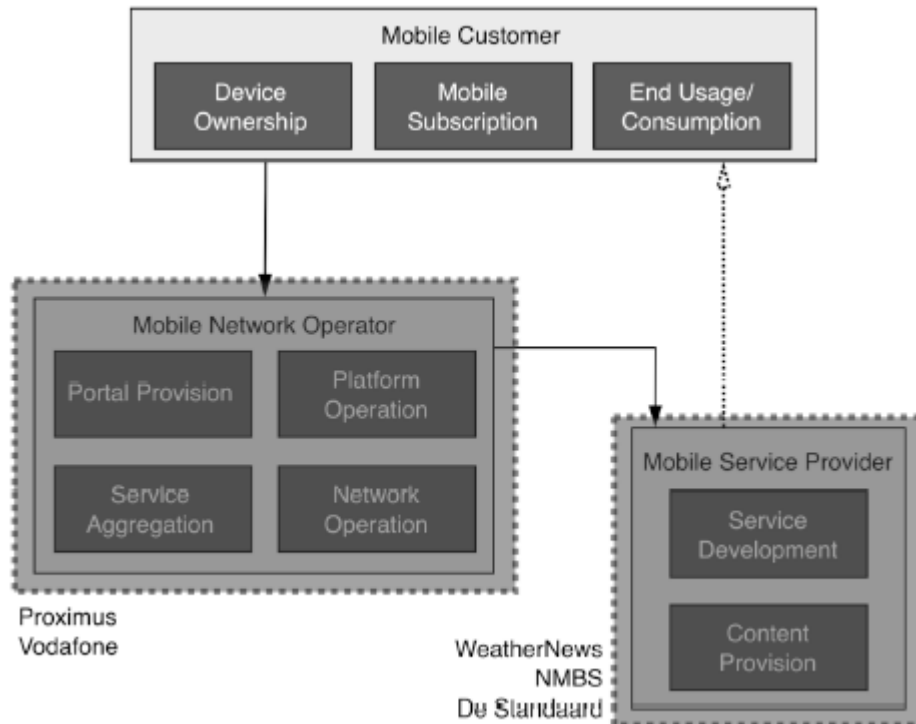


Figure 2-3: Telco-centric platform model: the Vodafone Live case (Ballon, 2009:12)

2.12.2 Device centric model

Mobile device comes with the main service platform. A good example is the Apple iPhone device. The main characteristics of this model are:

- The price paid for the device covers the integrated services also.
- Portal provisioning and platform operation activities are the responsibility of the manufacturer.
- Mobile versions of various services are provided by device manufacturer.
- Carrier can also provide services to the devices.
- PC can be used to send service to mobile devices.
- The mobile network operator works as an ISP."

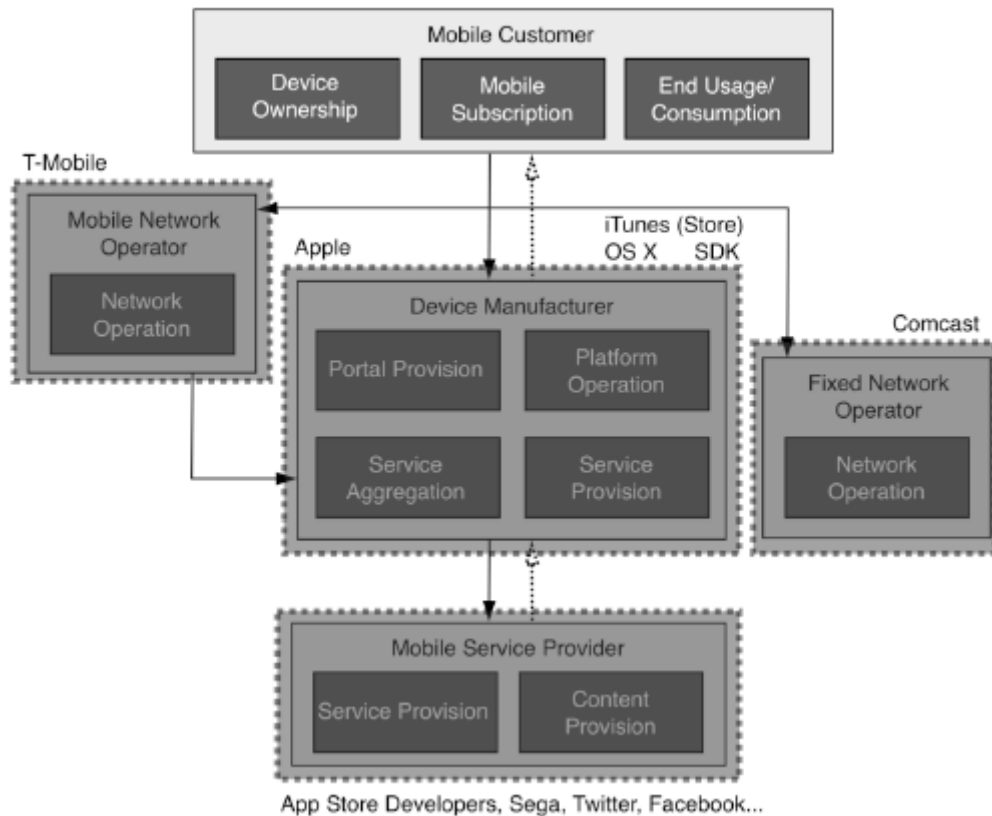


Figure 2-4: Device – centric platform model: the iPhone Case (Ballon, 2009:13)

When analysing device-centric configuration, it is clear that device manufacturer is taking control from the operator:

- Service creation: Developers are encouraged to develop applications that have to be authorized by manufacturer. Income is shared between developer and manufacturers.
- Profile/Identity: The device manufacturer gets most of the information, as he can access customer activities on the device and only limited information is with network operators.
- Service brokerage: the device manufacturer is in charge of service brokerage. The manufacturer works between developers and end users.
- Charging & billing: The network operator is in charge of access, but device manufacturer can be responsible for services.

2.12.3 Aggregator-centric model

A service aggregator independent of mobile network operator fulfils the actor role in this model. A real-life example is the Facebook Mobile platform.

The main characteristics of this model are:

- The network operator is responsible for giving access for users to use network.
- The service aggregator acts as the portal provider, which allows customers to choose services of choice.
- The service aggregator is not restricted to any specific network operator.
- The platform is operated by the service aggregator, who is defines the development language.
- The service developers create applications that can run on the portal.
- Some services can be accessed outside of portal, i.e. via the web browser(Ballon, 2009:14).

The platform owner takes the gatekeeper roles.

- Service creation: In this model services available on the portal are not strictly controlled. Any freelancer is allowed to do service creation. Aggregator won't involve in this process. Developers should adhere to the technical platform offered by the aggregator.
- Profile/Identity: This is the main function of aggregator is to keep control while spreading the reach (through data portability) over the user data.
- Service brokerage: It is the service aggregator who handles the services to the user.
- Charging & billing: User billing is done by network operator for access. Monetization in this particular model is not clear and can for example rely on advertising revenues, or various customer groups could be charged for access to the portal (Ballon, 2009:14).

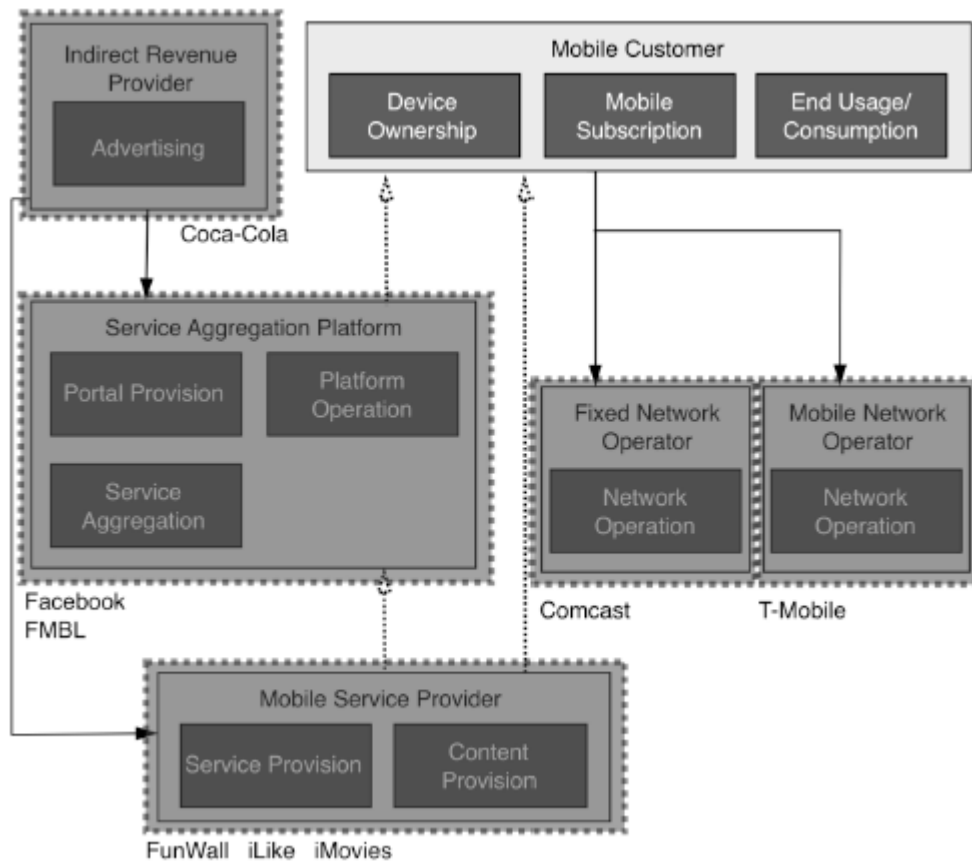


Figure 2-5: Aggregator-centric platform model: the Facebook Mobile Case (Ballon, 2009:14)

2.12.4 Service-centric model

The service-centric model can be seen as a theoretical model. It evolved from Google's Open Social initiative on open API's for social networking applications.

The main characteristics of this model are:

- The user chooses services from a portal through mobile network operator based on individual taste.
- Various platforms can be used by various service providers.
- Developers are allowed to group services and exchange information if the operator uses a meta-platform (e.g. Open Social).
- In this model, even though it is not grouped together, the services outside of the meta-platform are available.

- Individual service providers are in charge of profiles. Profiles can be shared between different service providers.
- The billing and charging methods used in this model are not clear.
- In terms of the implementation of the gatekeeper roles, these are less concentrated with one particular stakeholder than in previous models, but rather distributed over a number of individual service providers (Ballon, 2009:16)

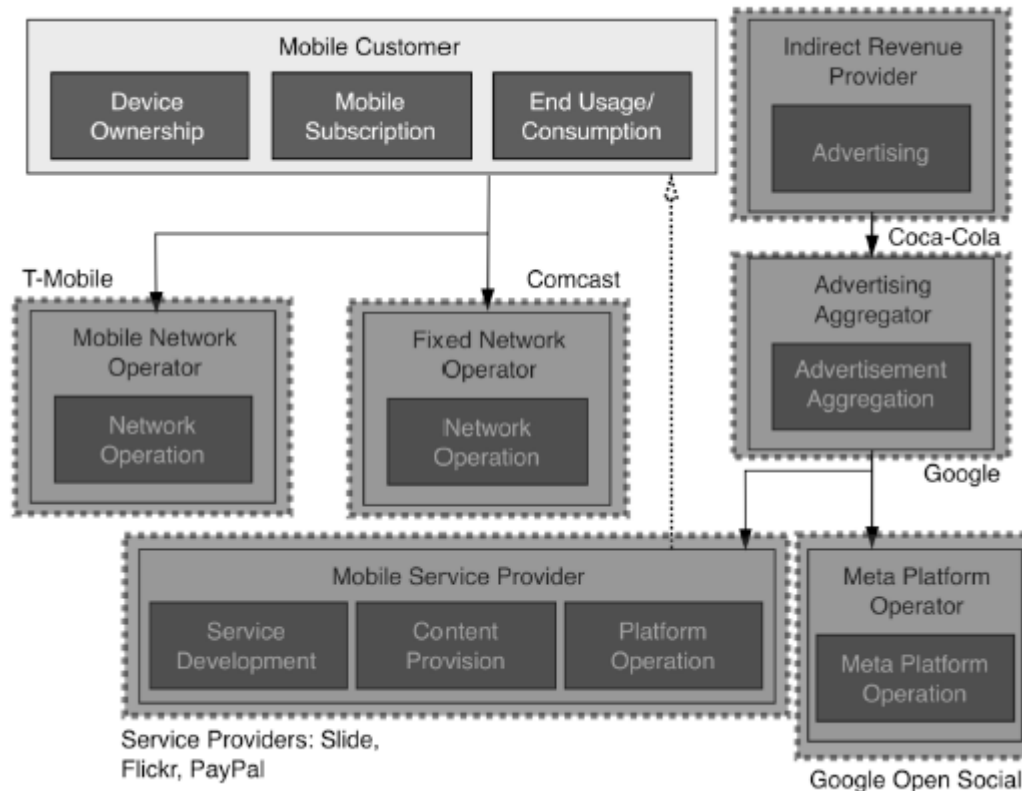


Figure 2-6: Service-centric platform model: the Open Social Case (Ballon, 2009:15)

2.13 SDMs for mobile applications

The mobile industry is widely known as a highly competitive industry, which also has many challenges to face. This section identifies different system development methodologies that are available for mobile application development in answer to research objective two.

There are many methodologies that have been specifically developed for mobile application development. Some of them are listed below:

- Mobile-D (Abrahamsson *et al.*, 2004:174)

- MASAM (Jeong *et al.*, 2008).
- GMA
- MODA-TEL (Chitforoush *et al.*, 2007:455).
- Mansurov's Accelerated Development Methodology (Mansurov, 2000)

2.13.1 “Mobile-D” (Abrahamsson *et al.*, 2004:174)

“Mobile-D” (Abrahamsson *et al.*, 2004:174) is an agile method developed by Abrahamsson *et al.* (2004:174) to overcome the challenges faced during the development of mobile application (Abrahamsson *et al.*, 2004:174). It is based on other methodologies like Extreme programming, Crystal methodologies and Rational Unified Process (Abrahamsson *et al.*, 2004:174). This method is suitable for a team containing ten or fewer developers based in the same work space working together to deliver a product within a short time span (i.e., < 10 weeks) (Abrahamsson *et al.*, 2004:175).

- When using Mobile-D for the development of a project, it is divided into five iterations called set-up phase, core phase, core-2 phase, stabilize phase and wrap-up phase. Each phase consists of three different development days:
 - “*Planning Day*” (Abrahamsson *et al.*, 2004:175),
 - “*Working Day*” (Abrahamsson *et al.*, 2004:175),
 - “*Release Day*” (Abrahamsson *et al.*, 2004:175),
 - Integration Day (Only needed if multiple teams are developing different parts of a project)

The practices of the different phases comprise nine principal elements. These elements are as follows:

- “*Phasing and pacing*” (Abrahamsson *et al.*, 2004:175),
- “*Architecture line*” (Abrahamsson *et al.*, 2004:175),
- “*Mobile test-driven development*” (Abrahamsson *et al.*, 2004:175),
- “*Continuous integration*” (Abrahamsson *et al.*, 2004:175),
- “*Pair programming*” (Abrahamsson *et al.*, 2004:175),
- “*Metrics*” (Abrahamsson *et al.*, 2004:175),
- “*Agile software process improvement*” (Abrahamsson *et al.*, 2004:175),
- “*Off-site customer*” (Abrahamsson *et al.*, 2004:175),
- “*User-Centred focus*” (Abrahamsson *et al.*, 2004:175),

According to Abrahamsson *et al.* (2004:175), when we apply the Mobile-D approach for the development of a project, we can see some positive observations:

- Progress on the project is more visible
- Technical problems are detected and resolved timely
- Responsibility is shared within the project team
- Information sharing must be effective and efficient
- High process practice coherence
- Minimal faults on the marketed products
- Phase of development is constant.

Otto (2007) identified that there was a need for a methodology for MAD that is more flexible and agile.

2.13.2 MASAM

MASAM is the short for “Mobile Application Software based on Agile Methodology” (Jeong *et al.*, 2008). It was developed by Jeong *et al.* (2008). This methodology is focused on rapid development for the fast developments in mobile application. MASAM uses various approaches like the agile approach, generating skeleton code for software platforms, providing techniques to optimize process for the rapid development of the software. Domain knowledge is stored as the pattern typed by using this domain information, we can reduce the development time and the important test item is prepared in advance. MASAM includes the following four phases.

- “*Development Preparation phase*” (Jeong *et al.*, 2008:364).
- “*Embodiment phase*” (Jeong *et al.*, 2008:365)
- “*Product Development phase*” (Jeong *et al.*, 2008:365)
- “*Commercialization phase*” (Jeong *et al.*, 2008:365)

Development preparation phase

During the development preparation phase the company prepares for product development based on market feedback. The company passes the information to the customer about the future product during this phase.

The development preparation phase activities can be described as follows:

- “*Grasping the product*” (Jeong *et al.*, 2008:364)
 - “*Product summary*” (Jeong *et al.*, 2008:364) – Here the concept of the product is created and general requirements are collected.

- “*Pre-planning*” (Jeong *et al.*, 2008:364) – The initial ideas and general requirements are organized to provide the concept of the potential product, and to identify the direction in which to gather further requirements.
- “*Product concept sharing*” (Jeong *et al.*, 2008:364)
 - “*User definition*” (Jeong *et al.*, 2008:364) – The client will get the opportunity to describe their requirements and expectations about the product in detail.
 - “*Initial product analysis*” (Jeong *et al.*, 2008:364) – Initial requirements are analysed and a formal concept for the product is developed.
- “*Project setup*” (Jeong *et al.*, 2008:364)
 - “*Development process coordination*” (Jeong *et al.*, 2008:364) – The steps needed for the development process of the product are planned, reviewed, confirmed and organized.
 - “*Project resource coordination*” (Jeong *et al.*, 2008:364) – The resources required to carry out the development project are calculated and procured.
 - “*Pre-study*” (Jeong *et al.*, 2008:364) – Initial research is done to provide context for product use, and patterns and trends in existing products of the same type.

Table 2-6: Activity task (Jeong *et al.*, 2008:364)

Activity	Task
Grasping Product	Product summary
	Pre planning
Product concept sharing	User definition
	Initial product analysis
Project Set-up	Development Process coordination
	Project resource coordination
	Pre study

Embodiment phase

The main idea behind the embodiment phase is to consolidate clients’ requirements in a formal way (Jeong *et al.*, 2008:365). This can be achieved by developing a prototype or simulated user interface and presenting it to the client. According to Jeong *et al.* (2008:365), some users don’t know what they want and developers must make use of “Application Patterns” to classify the application’s type and follow the algorithmic patterns and user interface trends found in similar applications as a guideline for design.

Activities in the embodiment phase can be described as below:

- “*Understanding user needs*” (Jeong *et al.*, 2008:365)
 - “*Story card workshop*” (Jeong *et al.*, 2008:365) – Story cards represent the main components or functionalities of a product. User requirements are used to develop story cards. Story cards are usually presented to the client for suggestions and critiques.
 - “*User interface design and non-functional requirements analysis*” (Jeong *et al.*, 2008:365) – The user interface is designed and moulded to fit their needs from the client’s requirements and suggestions.
- “*Architecture*” (Jeong *et al.*, 2008:365)
 - “*Architecture definition*” (Jeong *et al.*, 2008:365) – product architecture is planned with the overall product design from user needs and requirements.
 - “*Pattern management*” – patterns found in similar products are incorporated and utilized (Jeong *et al.*, 2008:365).

Table 2-7: Activity task (Jeong *et al.*, 2008:365)

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

“Product developing phase” (Jeong *et al.*, 2008:365)

The product is constructed iteratively and released to clients in cycles during the product development phase. The number of cycles are decided in the beginning of this phase, before development is started.

Using iterative development cycles means that requirements are systematically worked into the application, adding more functionality components after each development cycle. The client is consulted after each development cycle to ensure the application aligns with their requirements and expectations, and to allow the client to change their mind about their requirements. This leads to the possibility of new user requirements being presented after each release.

The activities of the product development can be described as below:

- “Implementation preparation” (Jeong *et al.*, 2008:365)
 - “Environment setup” (Jeong *et al.*, 2008:365) – The team allocations and development tools are determined and the development environment is also set.
 - “Development and release planning” (Jeong *et al.*, 2008:365) – the story cards are divided between teams and broken down into task cards for development, and the number of iterations and releases is determined.
- “Release cycle” (Jeong *et al.*, 2008:365)
 - “Iteration cycle” (Jeong *et al.*, 2008:365)
 - “Iteration planning” (Jeong *et al.*, 2008:365) – the story cards for the iteration will be fixed, and the required planning to develop these components is done.
 - “Implementation cycle” (Jeong *et al.*, 2008:365)
 - “Face-to-face meeting” (Jeong *et al.*, 2008:365) – the teams meet to ensure coordination between the various tasks and story cards to be developed.
 - “Incremental design” (Jeong *et al.*, 2008:365) – the overall design of the components is created or modified according to user requirements.
 - “TDD” (Jeong *et al.*, 2008:365) – performance testing is done.
 - “Refactoring” (Jeong *et al.*, 2008:365).
 - “Pair programming” (Jeong *et al.*, 2008:365) – task cards are implemented into code using a pair programming approach where two people program together.
 - “Continuous integration” (Jeong *et al.*, 2008:365) – newly developed components are integrated into the product.
 - “Feedback” (Jeong *et al.*, 2008:365) – teams meet again to give feedback on their development.
 - “Release” (Jeong *et al.*, 2008:365)
 - “Acceptance test” (Jeong *et al.*, 2008:365) – the product is released to the client for their testing and evaluation.
 - “Feedback” (Jeong *et al.*, 2008:365) – client feedback is gathered and changes in the product requirements are recorded.

Table 2-8: Activity task (Jeong *et al.*, 2008:365)

Activity	Task-1	Task-2	Task-3
Implementation Preparation	Environment Setup		
	Development Planning		
Release Cycle	Release Planning		
	Iteration Cycle	Iteration Planning	
		Implementation Cycle	Face-to Meeting
			Incremental Design
			TDD
			Refactoring
			Pair Programming
			Continuous Integration
		Feed back	
	Release	Acceptance Test	
		Feed back	

“Commercialization phase” (Jeong *et al.*, 2008:365)

The commercialization phase is where the product is made available to the market. The new product should be in accordance with the policies and laws of the country.

The activities of the commercialization phase can be described as below:

- “System test” (Jeong *et al.*, 2008:365)
 - “Acceptance tests and user tests” (Jeong *et al.*, 2008:365) – the product is tested by the client to ensure usability and acceptance of the product.
- “Product sales” (Jeong *et al.*, 2008:365)
 - “Launching test” (Jeong *et al.*, 2008:365) – the product is released to a small portion of the market to test its success.
 - “Product launching” (Jeong *et al.*, 2008:365) – the final product is released to the client.

2.13.3 “GMA” (Cheng & Yuan, 2007)

A methodology called “GMA” (Cheng & Yuan, 2007), Generic Mobile Application, was released by Cheng and Yuanin. This methodology is centred on mobile device performance, and much like the development framework, there is also a very strict physical architecture that has to be adhered to. To support different end-devices, a GMA is designed to run in any of three different modes:

- “BROWSER mode” (Cheng & Yuan, 2007:1393)
- “STANDALONE mode” (Cheng & Yuan, 2007:1393); and
- “MASTER-SALVE mode” (Cheng & Yuan, 2007:1393)

BROWSE mode is also known as “thin-client computing” (Cheng & Yuan, 2007:1393) where the device itself is only responsible for the user interface and STANDALONE mode is also known as “fat-client computing” (Cheng & Yuan, 2007:1393) where all the code, interface, execution and processing is performed on the device itself. MASTER-SALVE is also known as “distributed computing” (Cheng & Yuan, 2007:1393), where the end device is powerful enough to be able to handle the entire application, but it does not support all the functionalities required to execute the application code successfully. In this mode non-supported functionalities are executed on the server and the rest on the device.

GMA framework uses three-tier architecture to solve the problems of diverse computing power and functionalities. End-users use their own desktops or mobile devices, called GMAClient, in the front-tier to access mobile applications. In order to fit different end-devices, there are four kinds of GMAClient in this framework:

- Built-in Browser
- GMABrowser
- GMAAppSlave
- GMAAppStandalone

The middle tier includes the application server and the backend-tier includes the internet or intranet (Cheng & Yuan, 2007:1393).

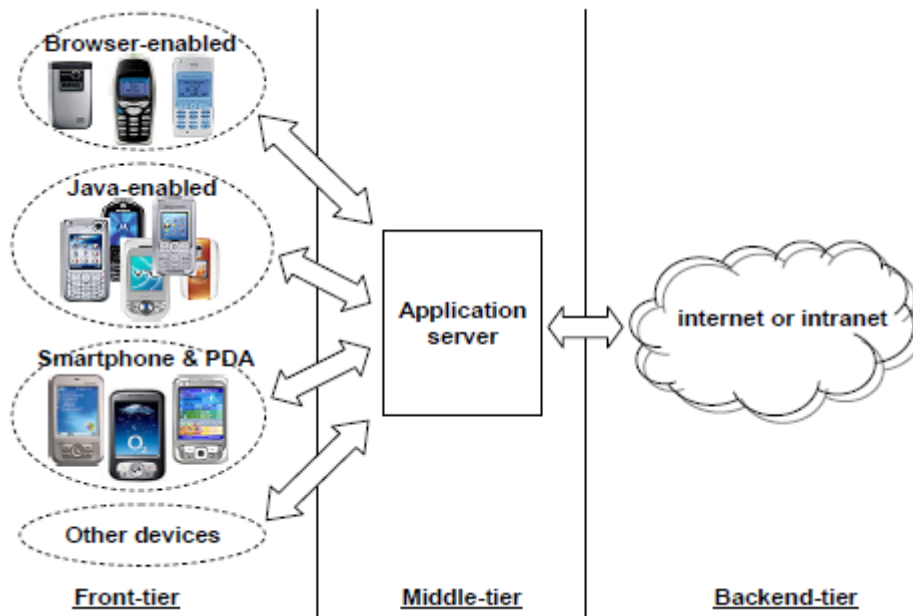


Figure 2-7: A diagram to sketch the GMA three-tier architecture (Cheng & Yuan, 2007:1393)

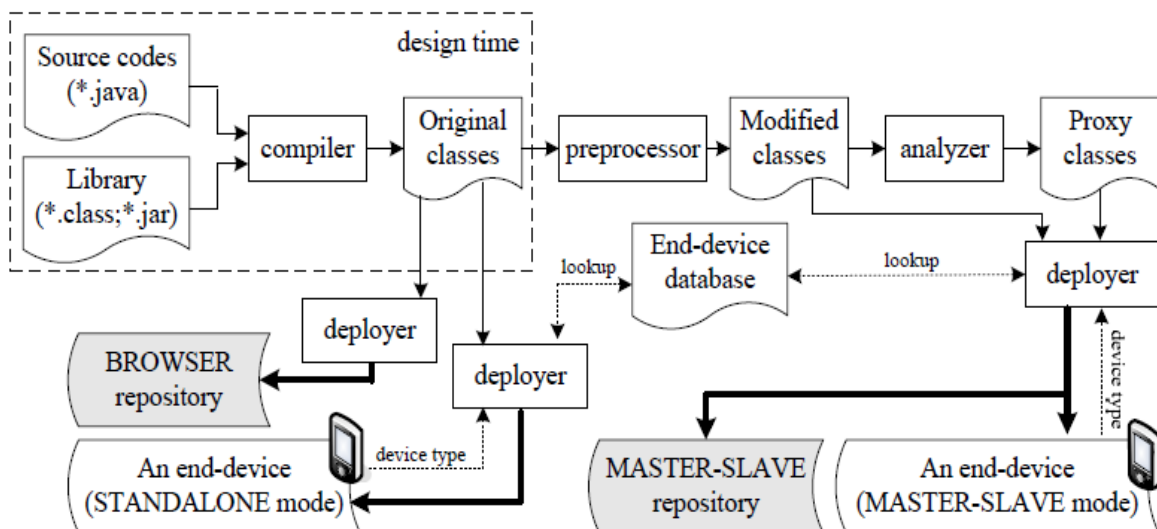


Figure 2-8: The GMA development flow (Cheng & Yuan, 2007:1402)

By using GMA, when developing a mobile application, developers do not need to worry about the computing power or the functionalities of the target end-devices and these resources will be

effectively used. Besides, a universal service interface is also proposed. Developers can use unified API to access different backend-services without background knowledge. In addition, because XML document is flexible and extensible, anyone can easily extend the end-device database to support more end-devices.

2.13.4 “MODA-TEL” (Chitforoush *et al.*, 2007:455).

“MODA-TEL” (Chitforoush *et al.*, 2007:455) is Model Driven Architecture-based software development methodology. According to Chitforoush *et al.* (2007:455), MODA-TEL is designed for distributed applications. Preparation and execution activities are separated into different phases.

This methodology contains the following phases.

1. “*Project Management*” (Chitforoush *et al.*, 2007:455): Activities that need to take place to develop the software are identified and allocated to specific teams in this phase. Quality assurance procedure are also implemented during this phase.
2. “*Preliminary Preparation*” (Chitforoush *et al.*, 2007:455): In this phase modelling and transformation needs are identified. In this phase, the final execution platform of the system and appropriate modelling language for the project are also identified. Based on the selected modelling languages, transformations between models are specified. The traceability support strategy to track requirements and changes are also implemented in this stage (Chitforoush *et al.*, 2007:455).
3. “*Detailed Preparation*” (Chitforoush *et al.*, 2007:455): Models and model transformations are determined during this phase. Depending on the needs identified in the above phase, the specifications of modelling languages and model transformations are prepared.
4. “*Infrastructure Setup*” (Chitforoush *et al.*, 2007:455): In this phase the tools are selected, for example tools for automatic or semi-automatic code generation are selected in this phase.
5. “*Project Execution*” (Chitforoush *et al.*, 2007:456): The main software development activities are done during this phase. The software development process selected in the project management phase will contribute to the selection of activities in this phase. All phases are performed in iterative-incremental mode (Chitforoush *et al.*, 2007:456). This phase may contains seven activities:
 - Requirements analysis (Chitforoush *et al.*, 2007:456),

- Modelling (Chitforoush *et al.*, 2007:456),
- Verification and validation (Chitforoush *et al.*, 2007:456),
- Transformation (Chitforoush *et al.*, 2007:456),
- Coding and testing (Chitforoush *et al.*, 2007:456),
- Integration and deployment (Chitforoush *et al.*, 2007:456); and
- Operation and maintenance (Chitforoush *et al.*, 2007:456)

2.13.5 Mansurov's accelerated development methodology (Mansurov, 2000)

Mansurov's accelerated development methodology uses formal methods and formal languages from the early development cycle to describe software. They used re-engineered versions of widely accepted telecommunication software and languages to achieve this.

This methodology is based on telecommunication formal. This includes SDL, MSC, TTCN and data description language called ASN.

A more detailed overview of the methodology is presented in Figure 2.8. This figure shows the phases of the methodology (right column), the most important models (boxes), as well as the most important information feedbacks provided by SDL tools (dashed lines).

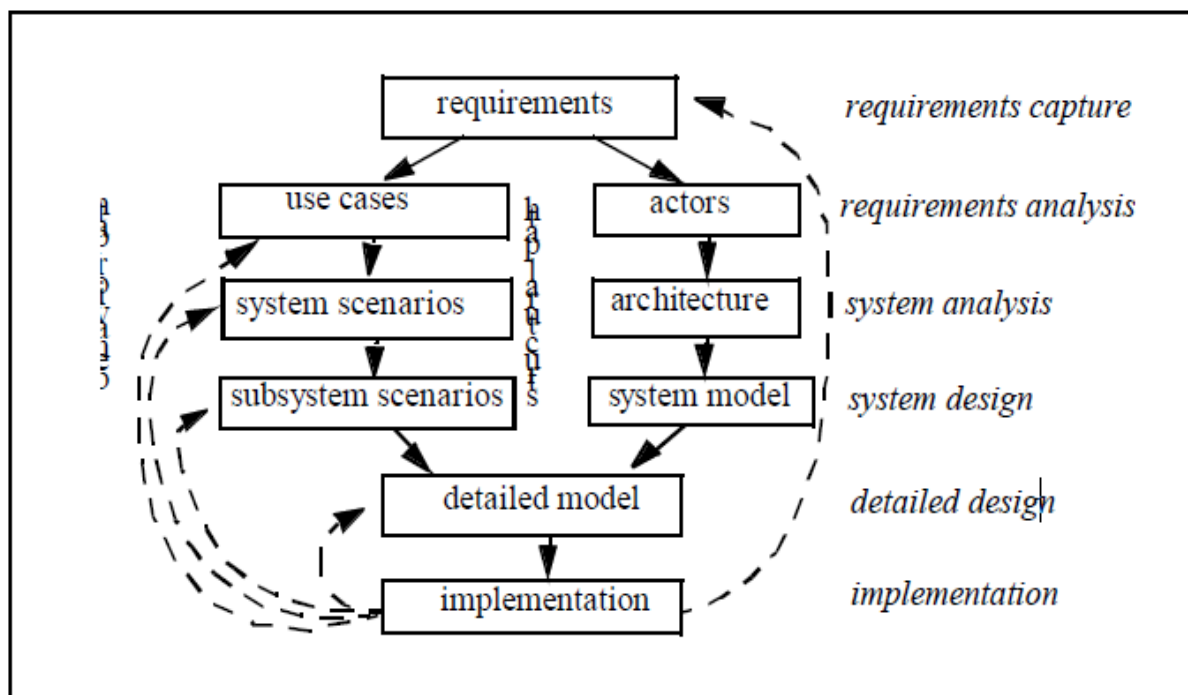


Figure 2-9: Phases in Mansurov's accelerated development methodology (Mansurov, 2000:7)

We can classify this methodology as an iterative use case based object-oriented methodology. It emphasizes the following key components:

- Capture of requirements by use-cases using the MSC language
- High-yield requirements validation using SDL requirement models
- Synthesis of SDL requirements models from approved MSC scenarios
- Seamless refinement of SDL requirement models into design models
- Adaptable code generation from the SDL models
- A high yield test design and automated test development process directed towards
 - TTCN (to allow more efficient, automated testing)
- Automatic recovery of SDL models from legacy software (Mansurov, 2000:7).

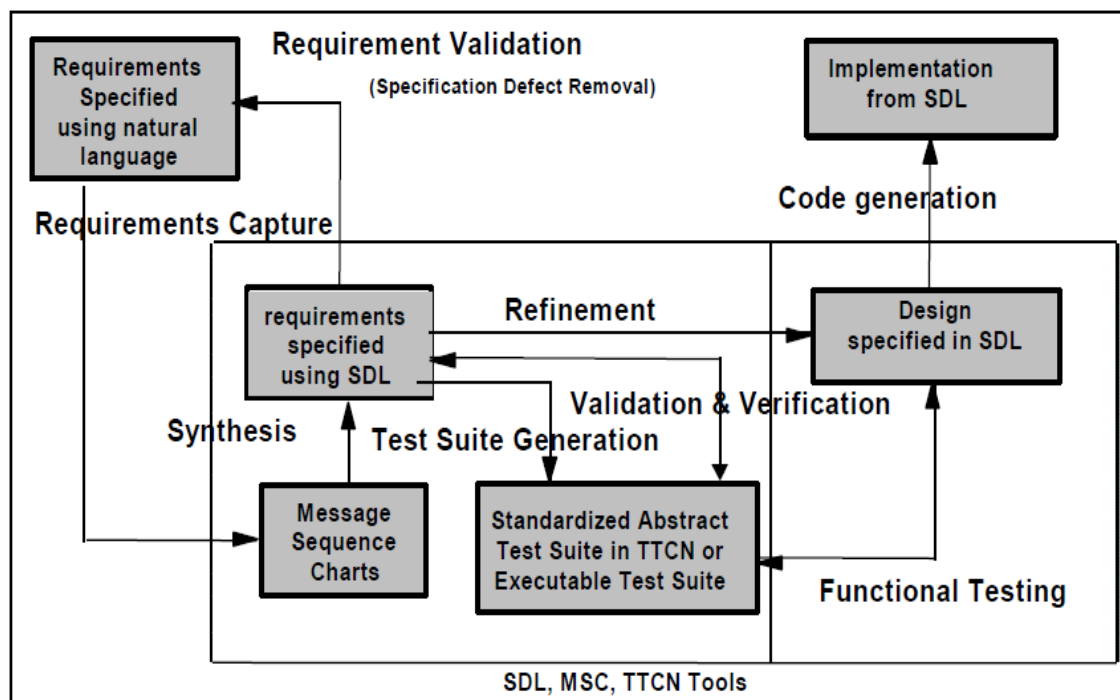


Figure 2-10: Processes of accelerated development methodology (Mansurov, 2000)

According to Mansurov (2000), TTCN tools, SDL and MSC support all processes. Because of the accelerated development methodology it provides project time and it reduces the cost of the project and time to market.

2.13.6 Similarities and differences between Mobile-D, MASAM, GMA, MODA-TEL and Mansurov's accelerated development methodology

Table 2-9: Similarities and differences between Mobile-D, MASAM, GMA, MODA-TEL and Mansurov's accelerated development methodology

	Philosophical approach	Process model	Methods(Steps)	Techniques / tools
Mobile-D	Agile development process and fast product delivery	Iterative model	Five phases of iteration: set-up phase, core phase, core-2 phase, stabilize phase and wrap-up phase and each phase consists of three different development days Planning Day, Working Day and Release Day.	Phasing and pacing, architectural line and pair programming
MASAM	Agile development process, simple and fast product delivery	Sequential and iterative	Includes four phases: development preparation phase, embodiment phase, product development phase, commercialization phase	Test driven, product summary, story card and pair programming
GMA	Adaptive Framework with client-server computing	Parallel	Three-tier architecture with front-tier, middle-tier and backend-tier	Backend services, Universal service interface, XML
MODA-TEL	Object-oriented approach	Iterative model	Project management Preliminary preparation Detailed preparation Infrastructure setup Project execution (Requirement analysis, Modelling, Verification and validation, Transformation, Coding and testing,	MDA -capable tools

			Integration and deployment, Operation and maintenance)	
Mansurov's accelerated development methodology	Based on object-oriented methodology	Iterative	Requirement capture Requirement analysis System analysis System design Detailed design Implementation	SDL models, MSC TTCN

2.14 Success factors / Critical characteristics / Challenges for MAD success

The first research objective is to investigate the challenges faced by companies currently developing mobile applications. In order to find the challenges one should identify all the challenges of a mobile application's success, success factors of mobile application and challenges for mobile application success. Below are the identified 46 items used to investigate the challenges faced by companies that are currently developing mobile application.

1. Platform Fragmentation (Shevchik, 2012).

Platform fragmentation means that there are currently numerous platforms available for example Android, iOS, etc. Selecting a platform from this wide selection is now a challenge.

2. Fragmentation within the platform (Joorabchi *et al.*, 2014)

Currently it is available in different versions of platforms. The same kind of platform got different versions. For example, Android has different versions. Selecting a version from different version is a challenge.

3. Device fragmentation (Bhatia, 2013)

Device fragmentation is another challenge faced by mobile application developers. This is occurring when some users use old version of operating system and some other users use newer versions.

4. Selection of development approach (Bhatia, 2013).

Selection of development approach is another challenge, according to Bhatia. This is because of the availability of different approaches like native, hybrid, HTML5.

5. Identifying innovative mobile use cases (Bhatia, 2013).

Another challenge in MAD is to find innovative mobile use cases.

6. Time to market (Johnson, 2014)

Another challenge is the time taken to sell the application, which means time taken for the product to reach the customer's hand. There should be a proper medium to sell the product.

7. Pressing competition (Levy, 2011)

As the competition in the field of mobile application is very high, pressing competition is a big challenge.

8. Application distribution to various markets (Levy, 2011).

Distribution of applications to various markets is another challenge faced by the MAD field.

9. Version control of the application software (Levy, 2011).

The different versions of application software should be carefully controlled (Levy, 2011).

10. Involvement of different development teams in a project.

A MAD project may contain different development teams like a front-end team, back-end team, testing team etc. Involvement of different development teams is a challenge for the success of the development.

11. Complex and lengthy integration with XML, SQL etc. (Levy, 2011)

Mobile application may include lengthy and complex integration with XML, SQL etc. This can also be a challenge.

12. Controlling the code base of different languages (Levy, 2011)

Mobile applications can be created using different languages and the controlling of code base of different languages is a challenge.

13. Ongoing effort to support new devices (Levy, 2011).

Another challenge is the effort to support new devices as they just keep popping up all the time.

14. Application security (Levy, 2011).

Application security is also a challenging factor.

15. Information security (Levy, 2011).

Another challenging factor of MAD is information security.

16. Data security (Fairchild, 2014)

Security of data is another challenge in case of mobile application.

17. Applications back-end (Tode, 2014)

Because of the limitations of the mobile devices the hosting of the application's back-end is a challenge.

18. Testing of mobile application (TechWellPresentations, 2014)

Testing of mobile application is a big challenge. Applications should be tested for different platforms.

19. The availability of the automated testing support (Joorabchi *et al.*, 2014)

The availability of automated testing support is another challenge in MAD success due to the unavailability of the automated testing facility for the testing of mobile application.

20. Developers acting as testers (Joorabchi *et al.*, 2014)

During the development of mobile application, developers may act as a tester. In this case developers acting as testers are also a challenge for the success of mobile application.

21. Testing the application for each platform separately (Joorabchi *et al.*, 2014).

For running same application in different platform, it is necessary to test application for each platform separately.

22. Selection of platform (Joorabchi *et al.*, 2014)

Open source platform and closed platforms are available in market. It is a challenge to select the platform that is whether to select open source or closed platform.

23. Need to create same application for more than one platform (Joorabchi *et al.*, 2014)

As customers are using different platform, there is a need to create same application for more than one platform.

24. Data intensive application (Joorabchi *et al.*, 2014)

Creating data intensive application is a challenge because high amount of data need to be stored on the device.

25. Budget required (Joorabchi *et al.*, 2014)

Another challenge in developing MAD is the budget required for the development.

26. Time required to building and delivering a complete mobile (Joorabchi *et al.*, 2014).

The total time taken to build and deliver a complete mobile application is a challenge when we consider the challenges in MAD.

27. Gap in MAD skills (Baker, 2014).

As everyday technology is advancing, there can be a gap in MAD skills, which is also a challenge.

28. Not enough developers have the right level of skills (Baker, 2014).

Scarcity of developers with the right level of skills is a challenge.

29. Availability of MADM (Levy, 2011).

As MAD is a new technology, there is little SDM, specifically for mobile application development. Therefore the availability of MADM is a challenge as it is difficult to select the appropriate method.

30. Need of a SDM for MAD (Levy, 2011).

MAD is a new technology. As there is a very few SDM for mobile application development, there is a need for a suitable SDM mobile application development.

31. Backlog applications (Tode, 2014)

Backlog applications can be a problem. If there are many pending applications it will slow down the development process.

32. Unrealistic expectations of the user (Accenture, 2013)

Increasing user expectation can affect the success of mobile application.

33. Entertainment (Schoknecht, 2012)

There are many criteria for rating the mobile application. One of them is the entertainment factor of the application. Therefore entertainment factor can be considered as a success factor of mobile application.

34. Utility (Schoknecht, 2012)

Another mobile rating factor is the utility factor of the application. The utility factor can therefore be considered a success factor.

35. Functionality (Schoknecht, 2012)

A mobile application should use the functionalities like camera, GPS of mobile devices. Functionality can therefore also be considered a success factor of mobile application.

36. Maintenance factor (Schoknecht, 2012)

An application should be a long-term investment. Therefore, we must update it with a regular maintains plan for the long term investment. The maintenance factor can be considered a success factor.

37. User experience (Breytenbach *et al.*, 2012:44)

As today's system development methodologies for MAD include all steps regarding the requirements, according to Breytenbach *et al.* (2012:44), it should be explained in detail to form more logical and better work method as given below.

- The needs of the user should be collected by means of surveys, interviews, and workshops
- Current applications can be studied to identify design and flow patterns for delivering similar functionalities
- Story cards are devised to illustrate the process flows within the application
- Users should be given access to story cards to get feedback before starting development
- User feedback are used to modify the story cards
- The next step is to start formal application design and development

- The application should be presented before the user for evaluation and pre-deployment testing. This phase will be the final phase of development (Breytenbach *et al.*, 2012:45).
- Applications can be changed according to user group feedback
- Final release the product for the users or for the market

38. Learnability (Breytenbach *et al.*, 2012:45)

There should be a method to ensure learnability. In order to ensure learnability in a mobile application we can use the below steps.

- We must find out the major process flows in the application after completing the design phase (Breytenbach *et al.*, 2012:45).
- Identify the skills level that is required to operate each process flow successfully (Breytenbach *et al.*, 2012:45).
- Identify the support type required to deliver the knowledge (Breytenbach *et al.*, 2012:45).
 - “Web-based help” (Breytenbach *et al.*, 2012:45)
 - “In-app help feature” (Breytenbach *et al.*, 2012:45)
 - “Support personnel” (Breytenbach *et al.*, 2012:45)

39. User Interface design (Breytenbach *et al.*, 2012:45)

In order to make sure of good user interface design we should follow the below steps

- Customer requirements are collected through surveys, interviews and workshops (Breytenbach *et al.*, 2012:45)
- Similar functionalities are designed by studying existing functions of current applications.
- Using logos, slogans or trademarks, we must create application identity.
- The software developers may not be talented enough for good graphical design, so it is better to get help from talented designers or graphic designer to set fonts, colour, form components, pictures etc. (Breytenbach *et al.*, 2012:46).

40. Efficiency (Breytenbach *et al.*, 2012:46).

In order to ensure the efficiency in a mobile application certain new approaches are proposed, including the following.

- We must set some programming standards so that developers can follow those standards to make sure uniformity and efficiency.

- We must set some guidelines for the developers to follow like how to handle the memory usage, the disposal of objects, and coding standards (Breytenbach *et al.*, 2012:46).

41. Value to user (Breytenbach *et al.*, 2012:46).

In order to ensure value to user in design and development of the system, the below steps can be followed.

- Developers must find out different important aspects of similar applications by doing research to include in the application.
- Developers must determine a marketing strategy and a market for the application, so that users will get information about the features of the application.

According to Breytenbach *et al.* (2012:46), for the continuous improvement of the application the following steps can be followed.

- Developers must include a feature in our application where users can send feedback such as suggestions or problems in the application.
- By using the feedback from the users developers can improve the application.
- Before releasing the application to the market developers must test the application by means of beta testers.

42. Compatibility (Breytenbach *et al.*, 2012:47).

In order to ensure compatibility the below method can be considered:

- Choose a multiplatform development tool;
- Do research to find out the current trend of devices because features like screen size, input methods etc. will be different for different devices (Breytenbach *et al.*, 2012:47).

43. Usability (Breytenbach *et al.*, 2012:48).

In order to ensure usability, the following steps can be practiced:

- Collect user requirements by using data collection methods like surveys, interviews and workshops;
- In order to explain the applications process flow, develop story cards;
- To help users get the idea before starting the development of the application, they should be presented with story cards;
- Change the design of the story card with the user feedback;
- Design and development of the application must begin'

- Present the new application to the user for pre-deployment evaluation and testing, and this will be the final phase of development;
- We must modify the application with the user feedback;
- Finally release the application product to the market (Breytenbach *et al.*, 2012:48).

44. “Availability and Accessibility” (Breytenbach *et al.*, 2012:48).

The following steps can be followed to ensure the availability and accessibility of the application.

- Find out the target market for the application and it should be according to the application functionalities;
- By using survey or market research, find out how this market can access mobile application (Breytenbach *et al.*, 2012:48);
- To distribute the application, create a list of most efficient vendors;
- Find out the design and cost implications to use this vendor (Breytenbach *et al.*, 2012:48).
 - Mobile platforms (Breytenbach *et al.*, 2012:49).
 - Terms and regulations (Breytenbach *et al.*, 2012:49).
 - Devices supported (Breytenbach *et al.*, 2012:49).
- Finally decide a final set of vendors for the application to distribute.

45. Adaptability (Breytenbach *et al.*, 2012:49).

The following steps can be considered to ensure adaptability:

- List a set of functionalities and features of application that might be adaptable for the user.
- Find out the situations in which the functionalities and features can be modified and identify how they can be modified.
- Find out the difficulties in design for the modification.
- Identify the most feasible modifications.
- Include the most feasible modifications in the initial design (Breytenbach *et al.*, 2012:49).

46. Security and Privacy (Breytenbach *et al.*, 2012:50).

In order to ensure security and privacy the following steps can be considered.

- Identify all the resources that the application might need to access. This should be done during the initial design phase.

- Find out all the sensitive information.
- Find out all the communication channels that are sensitive.
- In order to use throughout the development process select some coding standards.
- Identify one encryption standard to apply all the communication channels and to sensitive information.
- Apply various testing levels to make sure that all components are working optimally.
- Arrange a third party to launch a controlled attack on the system to test the vulnerabilities in the system (Breytenbach *et al.*, 2012:50).

From here onwards the above listed 46 success factors / critical characteristics / challenges of mobile application development for mobile application success will be referred to as characteristics.

2.15 Summary

This chapter discussed the available literature on this matter, the history of SDM, different definitions for the term SDM and the classification of SDM based on philosophical paradigm. Thereafter a literature review on different empirical studies on SDM and MAD SDM followed. The history of mobile technology and mobile application were examined, and definitions were offered for mobile application and the classification of mobile application was discussed. Thereafter, mobile application development, mobile application distribution, different types of platforms, SDM's for mobile application followed. Finally, we discussed all the success factors of mobile application, critical characteristics and challenges for mobile application success.

Chapter 3 describes the research methodology used to determine the answers for the research question.

CHAPTER 3 RESEARCH METHOD

3.1 Introduction

This chapter explains the research methodology used to find the answers for the research questions. The research methodology was specifically chosen to fit the research questions listed in Chapter 1

The suitable research paradigm, research method, data collection and data analysis are subsequently discussed.

3.2 Research paradigm

We can classify a research paradigm as either a positivistic, interpretive or critical social paradigm (Oates, 2006). When conducting research, a researcher can use any of the three research paradigms.

3.2.1 Positivistic research paradigm

Positivism is referred to as 'scientific method' or 'science research' (Oates, 2006). Positivist research is most commonly aligned with quantitative methods of data collection and analysis. Positivists generally assume that reality is objectively given and can be described by measurable properties that are independent of the observer (researcher) and his or her instruments. According to Labaree (2013), quantitative research deals in numbers, logic and the objective, focusing on logic, numbers, and unchanging static data and detailed, convergent reasoning rather than divergent reasoning.

3.2.2 Interpretive research paradigm

According to Oates (2006:292), interpretive research never proves or disproves any hypothesis. It will try to identify, explore and explain how all the factors in a particular social setting are related and interdependent (Oates, 2006:292). Interpretive research in information systems aims to understand the “social context”, the “social processes” through which the information system is developed. It also aims to understand how it is influenced by the social settings and how it will influence people (Oates, 2006:292).

3.2.3 Critical social paradigm

The aim of critical social research is to change the current situation in society among with regard to the social, political and cultural domination in which the research is done. Critical

research in information system and computing will identify the “power relations”, “conflicts and contradictions”. Critical research aims to empower people (Oates, 2006:296).

This study employed the positivistic research paradigm because of the use of quantitative data. In quantitative research the results are based on larger sample sizes that are representative of the population. Data was collected using questionnaires. In the positivistic research paradigm facts about the world can be discovered independently of the researcher’s personal values and beliefs.

3.3 Research method

A research method can be defined as the way to collect information. There are different research methods that can be used in positivistic research. These include design and creation, survey, experiments etc.

3.3.1 Design and creation

Design and creation focuses on developing artefacts (Oates, 2006:35). Artefacts include “constructs”, “models”, “methods”, “instantiations” (Oates, 2006:108).

3.3.2 Experiments

Experiments can be defined as “a strategy that investigates cause and effect relationships, seeking to prove or disprove a causal link between a factor and an observed outcome (Oates, 2006:127).

3.3.3 Survey

In survey researchers try to obtain same kind of data from a large group of people in a systematic way. Statics analysis is used to get a pattern in in the data based on which to generalize to a big group of people (Oates, 2006:35).

This study used survey as a research method because of the following reasons: Surveys provide a wide and inclusive coverage of people and therefore the result is representative of the wider population. In addition, much information can be produced in a short time at low cost and web survey is good for people who do not have good communication skills.

Table 3-1 gives an idea of the link between research objectives and research questions. Measuring scale is indicated in brackets with each question

Table 3-1: Link between research objectives and research questions

Research Objectives	Research Questions
1. Background	Personal background information • Country (single item) • Home language (single item) • Gender(single item) • Age (single item) • Mobile development experience (single item) • IT working experience (single item) • Position (single item) Company background information • Size of company(single item) • Business area (multiple choice) Project team • Size of project team (single item) • Availability of skilled developers (any open vacancies) (yes/no) Mobile application Information • Users of mobile application (single item) • Platform (multiple choice) • Development time (single item) • Application development environment (multiple choice) • Primary motivation (multiple choice)
2. Investigate the challenges faced by companies that are currently developing mobile applications.	Characteristics (Likert Scale Score 1 to 10) • Platform Fragmentation • Fragmentation within the platform. • Device fragmentation. • Selection of development approach. • Identifying innovative mobile use cases. • Time to market.

	• Pressing competition. • Application distribution to various markets. • Version control of the application software. • Involvement of different development teams in a project. • Complex and lengthy integration with XML, SQL etc. • Controlling the code base of different languages. • Ongoing effort to support new devices as they just keep popping up all the time. • Application security. • Information security. • Data Security. • Applications back-end. • Testing of mobile application. • Availability of the automated testing support. • Developers acting as testers. • Testing the application for each platform separately. • Selection of platform (Open source or closed platform). • Need to create same application for more than one platform. • Data intensive application (High data of data to be stored on the device). • Budget required. • Time required to building and delivering a complete mobile application. • Gap in MAD skills. • Not enough developers with the right level of skills. • Availability of MADM. • Need of a SDM for MAD. • Backlog applications. • Unrealistic expectations of the user. • Entertainment. • Utility. • Functionality. • Maintenance factor(long term investment). • Learnability. • Efficiency.
--	--

	• Value to user. • Compatibility. • Usability - Ease of use. • Availability and accessibility. • Adaptability. • User interface design. • User experience on multiple devices. • Security and privacy.
3. Identify different system development methodologies that are available for mobile application development, for example Mobile-D, MASAM and GMA.	Literature study
4. Investigate if system development methodologies are being used for MAD in the IT industry.	• Type of SDM (multiple choice) + Intensity of use (Likert Scale Score 1 to 10) • Strictness of use (Likert Scale Score 1 to 10) • Time of use (single item) • Possible change (yes/no)
5. Determine the success of mobile application development.	Success of mobile application (Likert Scale Score 1 to 10) Delivered on time (Likert Scale Score 1 to 10). Completed within the budget (Likert Scale Score 1 to 10).

6. Evaluate the suitability of current system development methodologies for mobile application development.	Literature analysis plus statistical analysis of data already collected.
---	--

A survey was conducted during April- May 2015 and it was carried out all over the world, mainly in South Africa, India, Pakistan, Germany etc. This survey targeted individuals developing or managing mobile application development.

The researcher compiled a database of companies doing MAD using Google, LinkedIn etc. Thereafter the researcher compiled a list of people working in those companies using LinkedIn and Facebook. The list comprised 700 people.

Initially questionnaires were distributed electronically. Some participants requested a web questionnaire as it is more convenient for them, the researcher used Google Form to create a questionnaire. Personal emails were sent to all the participants, and weekly reminders followed.

3.4 Data collection

The most used technique to collect data in a survey is a questionnaire (Oates, 2006:93). A questionnaire is a pre-defined set of questions in a pre-determined order. There are different types of questionnaire, including hard copies (postal survey), online questionnaire etc.

3.4.1 Hard copies (postal survey)

This method entails printing the questionnaire and sending it to participants through the mail. It is also called postal survey (Wyse, 2012). They complete it and return it to the sender or the researcher goes and collects it.

3.4.2 Online questionnaire

A questionnaire can be send out by emailing the questionnaire to participants to be completed and returned by email or a researcher can create a website with questionnaire and can share the link of the website to fill the questionnaire. Both methods are called online surveys (Oates, 2006:102). In this study we used an online questionnaire. This is because as we wanted to

cover wide geographic area these methods are the most convenient and low cost method to collect data.

3.5 Development of questionnaire

We compiled a questionnaire with 26 questions. The questionnaire contained questions to capture the individual background or personal background information like the country, home language, gender, age, position, experience in mobile application and experience in IT and company background information like the size of company, business area and project teams like size of project team and the availability of skilled developers.

There were also questions regarding mobile application information like users of mobile applications, platforms, development times, success of mobile applications, on-time delivery, completed in budget, application development environment and primary motivation.

There was also a question regarding the characteristics of mobile application development. We listed 46 characteristics to scale between 1 to 10 to examine the challenges faced by the companies. There were some questions regarding SDM like the type of SDM and intensity of use, strictness of use, time of use and possible change of use of SDM. A sample of the questionnaire is included in the appendix.

We sent the questionnaire to a group of 7 co-workers for pilot test. Some of them commented that they not understand some of the terms in the question on the characteristics and they asked for descriptions of each characteristic. We subsequently added short descriptions of the terms relating to the characteristic before sending it to respondents. The pilot test responses were not included in the final data set.

3.6 Response rate

The questionnaire was sent to around 700 people and 180 responses were received. It is about a 25% response rate.

3.7 Data analysis

Statistical analysis was used to answer the research questions. The following analysis techniques were used in this study.

3.7.1 Descriptive statistics

Descriptive statistics were used in data analysis to describe the main feature of the collection of information. It is a method of organizing, displaying or describing the collected data by the use

of tables, graphs and summary measures or it can be defined as numbers that are used to summarize and describe the data. Descriptive statistics include frequency distribution, relative frequency, percentage, cumulative frequency, mean, standard deviation etc. (Mann, 2010).

- **Frequency distribution**

A frequency distribution lists all categories of data and the number of elements belonging to each listed category (Mann, 2010).

- **Relative frequency**

Relative frequency of a category of data can be calculated by dividing the frequency of that category by the sum of all frequency (Mann, 2010:30).

- **Percentage**

The percentage of a category of data can be calculated by multiplying the relative frequency of that category by one hundred (Mann, 2010:30).

- **Cumulative frequency**

Cumulative frequency can be defined as the frequency of a particular class that includes all values in a data set that lie below the upper boundary of that class (Mann, 2010:63).

- **Cumulative percentage**

Cumulative percentage of a particular class can be calculated by multiplying cumulative frequency with 100 and dividing by the total number of observations (Mann, 2010:63).

- **Mean**

Mean can be calculated by dividing the sum of all values obtained by number of values. It can be also defined as the most frequently used measure of central tendency (Mann, 2010:80).

- **Standard deviation**

The standard deviation value tells us how closely the values of a data set are clustered around the mean. If the standard deviation value is low, it indicates that the values of the data set are spread over a relatively smaller range around the mean. If the standard deviation value got a

larger value, it indicates that the data set are spread over a relatively larger range around the mean (Mann, 2010).

3.7.2 Factor analysis

Factor analysis can be defined as a statistical method that can be used to find out the inter-relationship between large numbers of variables. This can be used to explain in terms of the common factors. The objective of this technique is to get a method of condensing the information in a large number of original variables into a small set of variables without any loss of information or a minimum of loss of information. "Factor analysis is therefore motivated by the fact that measured variables can be correlated in such a way that their correlation may be reconstructed by a smaller set of parameters which could represent the underlying structure in a concise and interpretable form" (Basilevsky ,1994).

3.7.3 Reliability

The reliability test is for testing the internal consistency. The Cronbach alpha value of each retained factor after exploratory factor analysis must be checked to find out whether factors are reliable. If the Cronbach alpha ≥ 0.9 , it indicates high reliability and if the Cronbach alpha ≥ 0.8 , there is a moderate reliability. If the Cronbach alpha ≥ 0.7 , it is low reliability (Field 2009:675). However, Huisman (2000:65) states that reliability of at least 0.6 should be used with exploratory research in SDMs.

3.7.4 Correlation coefficients

Correlation coefficients tell us the strength of the relationship between two variables. It will provide no information about the amount of change in one variable as a result of any change in another variable. The Pearson correlation coefficient value lies between -1 and 1 (Mann, 2010:593). The equation is

$$r = \frac{n\sum xy - \sum x \sum y}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

where r is the sample Pearson correlation coefficient

x is the value of independent variables

y is the value of dependent variable

n is the paired data points in the sample

The link between the different items of the questionnaire and the statistical technique used is shown in Table3-2 below.

Table 3-2: Link between the research questions and the statistical technique

Item	Statistical technique used
Personnel background information • Country (single item) • Home Language (single item) • Gender (single item) • Age (single item) • Mobile development experience (single item) • IT working experience (single item) • Position (single item)	Descriptive Statistics • Frequency • Percentage • Cumulative percentage
Company Background information • Size of company(single item)	Descriptive Statistics • Frequency • Percentage • Cumulative Percentage
• Business area (multiple choice)	Descriptive Statistics • Frequency • Percentage
Project team • Size of project team (single item)	Descriptive Statistics • Mean • Standard Deviation
• Availability of skilled developers (Any open vacancies) (yes/no)	Descriptive Statistics • Frequency • Percentage • Cumulative Percentage
Mobile application information • Users of mobile application (single item) • Development time (single item)	Descriptive Statistics • Frequency • Percentage • Cumulative Percentage
• Platform (multiple choice) • Application development environment (multiple choice) • Primary motivation (multiple choice)	Descriptive Statistics • Frequency • Percentage

Characteristics (Likert Scale Score 1 to 10) • Platform fragmentation (Shevchik, 2012). • Fragmentation within the platform (Joorabchi <i>et al.</i>, 2014). • Device fragmentation (Bhatia, 2013). • Selection of development approach (Bhatia, 2013). • Identifying innovative mobile use cases (Bhatia, 2013). • Time to market (Johnson, 2014). • Pressing competition (Levy, 2011). • Application of distribution to various markets (Levy, 2011). • Version control of the application software (Levy, 2011). • Involvement of different development teams in a project. • Complex and lengthy integration with XML, SQL etc. (Levy, 2011). • Controlling the code base of different languages (Levy, 2011). • Ongoing effort to support new devices as they just keep popping up all the time (Levy, 2011). • Application security (Levy, 2011). • Information security (Levy, 2011). • Data security (Levy, 2011). • Applications back-end (Tode, 2014). • Testing of mobile application (TechWellPresentations, 2014). • Availability of the automated testing support (Joorabchi <i>et al.</i>, 2014). • Developers acting as testers (Joorabchi <i>et al.</i>, 2014). • Testing the application for each platform separately (Joorabchi <i>et al.</i>, 2014). • Selection of platform (Open source or Closed platform) (Joorabchi <i>et al.</i>, 2014).	Descriptive Statistics • Mean • Standard Deviation Factor Analysis Reliability Test
--	---

• Need to create same application for more than one platform (Joorabchi <i>et al.</i>, 2014). • Data intensive application (High data of data to be stored on the device) (Joorabchi <i>et al.</i>, 2014). • Budget required (Joorabchi <i>et al.</i>, 2014). • Time required to building and delivering a complete mobile application (Joorabchi <i>et al.</i>, 2014). • Gap in MAD skills (Baker, 2014). • Not enough developers with the right level of skills (Baker, 2014). • Availability of MADM (Levy, 2011). • Need of a SDM for MAD (Levy, 2011). • Backlog applications (Tode, 2014) • Unrealistic expectations of the user (Accenture, 2013). • Entertainment (Schoknecht, 2012). • Utility (Schoknecht, 2012). • Functionality (Schoknecht, 2012). • Maintenance factor (long term investment) (Schoknecht, 2012) • Learnability (Breytenbach <i>et al.</i>, 2012). • Efficiency (Breytenbach <i>et al.</i>, 2012). • Value to user (Breytenbach <i>et al.</i>, 2012). • Compatibility (Breytenbach <i>et al.</i>, 2012). • Usability - ease of use (Breytenbach <i>et al.</i>, 2012:48). • Availability and accessibility (Breytenbach <i>et al.</i>, 2012). • Adaptability (Breytenbach <i>et al.</i>, 2012:49). • User interface design (Breytenbach <i>et al.</i>, 2012:45). • User experience on multiple devices (Breytenbach <i>et al.</i>, 2012:44). • Security and privacy (Breytenbach <i>et al.</i>, 2012).	
Type of SDM (multiple choice) + Intensity of use (Likert Scale Score 1 to 10) Strictness of use (Likert Scale Score 1 to 10)	Descriptive Statistics • Mean • Standard Deviation

Time of use (single item) Possible change (yes/no)	
Success of mobile application (Likert Scale Score 1 to 10) Delivered on time (Likert Scale Score 1 to 10). Completed in budget (Likert Scale Score 1 to 10).	Descriptive Statistics • Mean • Standard Deviation
➤ Success of mobile application (Likert Scale Score 1 to 10) between Strictness of use (Likert Scale Score 1 to 10) Time of use (single item) ➤ Delivered on time (Likert Scale Score 1 to 10). Between Strictness of use (Likert Scale Score 1 to 10) Time of use (single item) ➤ Completed in budget (Likert Scale Score 1 to 10).). between Strictness of use (Likert Scale Score 1 to 10)Time of use (single item)	Correlation

The result of the data analysis as they corresponds to each research objective is discussed in Chapter 4.

3.8 Background information of the respondents

The following part provides details regarding the background information of the data collected. The questionnaire contains questions to capture the background information. This includes Personal background information, Company background information, background information regarding the project team; and Mobile application information.

Personal background information includes information regarding the country, home language, gender, age, mobile development experience, IT working experience and current position.

Company background information includes details of size of the company and business area of the company.

Information regarding the project team includes the size of project team, the availability of skilled developers (open vacancies)

Finally background information regarding mobile application includes Users of mobile application, Platform, Development time, Application development environment and Primary motivation.

Next comes the information received from the respondents.

3.8.1 Personal background information

Table 3-3 shows the summary of data obtained from questionnaire responses to the question on the respondents' country of residence.

Table 3-3: Summary of the country of the respondent

Country	Frequency	Percentage	Cumulative percentage
India	84	46.93	46.93
Pakistan	14	7.82	54.75
UAE	11	6.15	60.89
South Africa	37	20.67	81.56
Germany	10	5.59	87.15
USA	5	2.79	89.94
UK	1	0.56	90.50
Egypt	3	1.68	92.18
Ukraine	2	1.12	93.30
Mexico	1	0.56	93.85
Berlin	1	0.56	94.41
Tanzania	1	0.56	94.97
Japan	2	1.12	96.09
Spain	1	0.56	96.65
Turkey	1	0.56	97.21
Serbia	1	0.56	97.77
Australia	1	0.56	98.32
Indonesia	1	0.56	98.88
Canada	1	0.56	99.44
France	1	0.56	100

The results from Table 3-3 show that 46.93% respondents are from India, and 20.67% from South Africa.

The respondents speak different languages. Table 3-4 shows the summary of data obtained from questionnaire responses about the home language.

Table 3-4: Home language

Home language	Frequency	Percentage	Cumulative percentage
Malayalam	39	21.67	21.67
Hindi	20	11.11	32.78
Tamil	6	3.33	36.11
Urdu	16	8.89	45.00
Bengali	3	1.67	46.67
Telugu	6	3.33	50.00
Gujarati	2	1.11	51.11
Arabic	11	6.11	57.22
Afrikaans	7	3.89	61.11
IsiZulu	3	1.67	62.78
isiXhosa	1	0.56	63.33
English	30	16.67	80.00
Shona	1	0.56	80.56
French	2	1.11	81.67
Sesotho	2	1.11	82.78
Swati	3	1.67	84.44
Spanish	3	1.67	86.11
Serbian	1	0.56	86.67

Marathi	3	1.67	88.33
Turkish	3	1.67	90.00
Sepedi	2	1.11	91.11
Indonesia	1	0.56	91.67
N.Sotho	1	0.65	92.22
Ukrainian	2	1.11	93.33
Punjabi	2	1.11	94.44
Greek	1	0.56	95.00
Chinese	1	0.56	95.56
German	2	1.11	96.67
Polish	1	0.56	97.22
Dutch	1	0.56	97.78
Portuguese	1	0.56	98.33
Italian	2	1.11	99.44
Kiswahili	1	0.56	100

Table 3-4 shows that 21.67% respondents are Malayalam-speaking, 16.67% are English speaking people and 11.11% are Hindi-speaking people.

Table 3-5 shows the details of the gender of the participants of the survey

Table 3-5: Gender

Gender	Frequency	Percentage	Cumulative percentage
Male	175	97.77	97.77
Female	4	2.23	100

From the Table 3-5 it is clear that 97.77% participants were male.

Table 3-6: Age

Age	Frequency	Percentage	Cumulative percentage
20 – 25	73	40.78	40.78
26 – 30	79	44.13	84.92
31 – 35	20	11.17	96.09
36 – 40	4	2.23	98.32
46 – 50	2	1.12	99.44
Above 50	1	0.56	100

Table 3-6 makes it clear that 44.13 % participants are aged between 26 and 30 and 40.78% are between 20-25.

Table 3-7: Personal mobile development experience

Personal mobile development experience(no of application)	Frequency	Percentage	Cumulative percentage
0	3	1.71	1.71
1	3	1.71	3.43
2	3	1.71	5.14
3	16	9.14	14.29
4	10	5.71	20.00
5	11	6.29	26.29
6	12	6.86	33.14
7	4	2.29	35.43
8	7	4.00	39.43
9	5	2.86	42.29
10	27	15.43	57.71
11	6	3.43	61.14
12	11	6.29	67.43
14	1	0.57	68.00
15	17	9.71	77.71
16	2	1.14	78.86
18	1	0.57	79.43
20	14	8.00	87.43
21	1	0.57	88.00

25	5	2.86	90.86
30	6	3.43	94.29
40	7	4.00	98.29
50	2	1.14	99.43
55	1	0.57	100

Table 3-7 shows that 15.43% respondents have worked in 10 application development projects, four percent of respondents had experience of about 40 projects and few, 0.57%, people had a chance to work in 55 projects, which is the great number.

Table 3-8: IT working experience

Working experience in years	Frequency	Percentage	Cumulative percentage
0 to 1	5	2.81	2.81
1 to 2	26	14.61	17.42
2 to 4	68	38.20	55.62
4 to 8	59	33.15	88.76
More than 8 years	20	11.24	100

From Table 3-8 it is clear that 38.20% people got 2 to 4 years of experience in the IT field and only 11.24% participants got more than 8 years of IT experience.

Table 3-9: Current position

Group	Frequency	Percentage	Cumulative percentage
Developers Group	165	93.22	93.22
Management Group	12	6.78	100

Table 3-9 shows that 93.22% participants are from the development group, which included Android developers, Mobile application developers, programmer analysts etc. and 6.78% respondents are from management group, which included Director, Product manager, CEO etc.

3.8.2 Company background information

Table 3-10: Size of company

Size of company	Frequency	Percentage	Cumulative percentage
1 – 10	25	14.12	14.12
10 – 50	48	27.12	41.24
50 – 250	39	22.03	63.28
More than 250	65	36.72	100.00

Table 3-10 shows that 36.72 % companies have more than 250 employees.

Table 3-11: Business area (where n=169)

Business area	Frequency	Percentage
Insurance	2	1.18
Entertainment, social media, game development	14	8.28
Government	18	10.65
Education	22	13.02
Manufacturing	22	13.10
Banking	26	15.38
Others	46	27.38
IT, Software development, web development	80	47.34

From Table 3-11 it is clear that 47.34% companies are doing software development or web development.

Table 3-12: Size of the project team

	N	Mean	Standard deviation
Size of the project team	173	16.64	36.35

It is clear that the average size of a development team is ± 17 people from Table: 3-12

Table 3-13: Open vacancies

Open vacancies	Frequency	Percentage	Cumulative percentage
Yes	76	42.94	42.94
No	101	57.06	100

From Table 3-13 it is clear that 42.94% respondents reported that their company has open vacancy.

3.8.3 Mobile application Information

Table 3-14: Users of mobile application

Users of mobile application	Frequency	Percentage	Cumulative percentage
Internal Uses	4	2.26	2.26
External Use	56	31.64	33.90
Both	117	66.10	100

From Table 3-14 it is clear that 66.10% people are using the developed mobile application for their internal and external uses.

Table 3-15: Platform

Platform	Frequency	Percentage
Android	110	62.50
iOS	121	69.14
HTML5	27	15.25
Mac OS	22	12.43
Others	13	7.34

Table 3-15 shows that 69.14% respondents are using iOS and 62.50% are using Android.

Table 3-16: Development time

Development time (in weeks)	N	Mean	Standard deviation
	147	14.99	14.34

From the table we can see the average time taking to complete a project is 15 weeks, which is almost four months.

Table 3-17: Application development environment

Application development environment	Frequency	Percentage
Native	154	89.02
Hybrid	37	21.39
Mobile web or HTML5	19	10.98
Others	3	1.74

Table 3-17 shows that 89.02% are using the Native application development environment and 21.39% are using Hybrid.

Table 3-18: Primary motivation

Primary Motivation	Frequency	Percentage
Profit	101	59.06
Extension of company portfolio to attract more external clients	95	55.56
Brand recognition	63	36.84
Others	33	19.30

Table 3-18 shows that 59.06% respondents primary motivation for MAD is profit.

Table 3-19: Success of mobile application

Success of last mobile application developed (Score 1 to 10)	Frequency	Percentage	Cumulative percentage
1	1	0.58	0.58
2	1	0.58	1.17
4	2	1.17	2.34
5	11	6.43	8.77
6	14	8.19	16.96
7	30	17.54	34.50

8	65	38.01	72.51
9	24	14.04	86.55
10	23	13.45	100

Table 3-20: Success of mobile application

Success of mobile application(score 1 to 10) 1= not successful and 10 = extremely successful	N	Mean	Standard deviation
	171	7.75	1.54

From Table 3-20 and Table 3-21 it is clear that the success of mobile applications is good.

Table 3-21: Use of SDM

Are you using any SDM	Frequency	Percentage	Cumulative percentage
Yes	97	60.25	60.25
No	64	39.75	100

From Table 3-21 it is clear that 60.25% respondents are using SDM and 39.75% are not.

Table 3-22: Future change of use

Looking for a new SDM for mobile application development	Frequency	Percentage	Cumulative percentage
Yes	28	26.17	26.17
No	79	73.83	100

Table 3-22 shows that 26.17% respondents reported that they want a new SDM for MAD and 79% reported they don't want any new SDM.

3.9 Summary

This chapter presented the research methodology used to answer the research questions. The chapter started off by discussing the research paradigm used in this research. After that the chapter discussed different research methods and the research method used in this study. The chapter concludes with a description of data collection, development of questionnaire, response rate and data analysis techniques. In this process the section also discusses the background information provided by the respondents.

Chapter 4 describes the results. This result is presented as they relate to the research objectives.

CHAPTER 4 RESULTS

4.1 Introduction

In this section, the results of the research are discussed and analysed. The analysis and discussion of the results aim to answer the research objectives stated in Chapter 1.

4.2 Investigate The Challenges Faced By Companies which are currently developing mobile applications

In order to identify the challenges faced by the companies that are developing mobile applications, the questionnaire included 46 characteristics of MAD for respondents to scale between 1 and 10 where 1 = not important at all and 10 = extremely important. Table 4-1 is sorted from the highest mean to the lowest mean.

Descriptive statistics of the 46 characteristics of mobile application is shown below Table 4.1. The table is sorted according to the mean value.

Table 4-1: Descriptive statistics of the characteristics of mobile application

Characteristics	N	Mean	Standard deviation
Data security	168	8.95	1.66
Testing of mobile application	165	8.93	1.44
User interface design.	162	8.81	1.54
Usability - ease of use.	161	8.77	1.65
Information security	167	8.75	1.83
Value to user.	164	8.69	1.74
Security and privacy.	160	8.6	1.70
Application security	169	8.56	1.98
Involvement of different development teams in a project	168	8.54	1.78
Version control of the application software	169	8.4	1.97
Efficiency.	161	8.37	1.66
Time required to building and delivering a complete mobile application.	164	8.37	1.78
User experience on multiple devices.	161	8.34	2.03
Selection of development Approach	172	8.30	2.09

Functionality.	165	8.18	1.88
Availability and accessibility.	161	8.1	1.87
Compatibility.	160	8.04	2.05
Applications back-end	167	8.04	1.96
Testing the application for each platform separately	164	8.03	2.09
Identifying innovative mobile use cases	170	7.94	1.98
Budget required.	165	7.92	2.01
Time to market	172	7.90	2.06
Learnability.	160	7.84	1.88
Platform fragmentation	173	7.83	2.36
Adaptability.	160	7.8	2.00
Maintenance factor (long-term investment)	162	7.70	1.95
Ongoing effort to support new devices as they just keep popping up all the time	169	7.53	2.19
Fragmentation within the platform	168	7.52	2.09
Gap in MAD skills.	164	7.45	1.93
Not enough developers with the right level of skills	163	7.28	2.39
Utility	164	7.27	2.01
Need to create same application for more than one platform	165	7.27	2.31
Application distribution to various markets	167	7.20	2.09
Device fragmentation	169	7.19	2.27
Availability of MADM.	163	7.16	2.30
Selection of platform (Open source or closed platform).	167	6.99	2.43
Availability of the automated testing support	166	6.97	2.38
Controlling the code base of different languages	168	6.92	2.27
Pressing competition	167	6.90	2.44
Need of a SDM for mobile application development	162	6.84	2.39

Data intensive application (High data of data to be stored on the device).	166	6.8	2.13
Developers acting as testers	168	6.78	2.4
Complex and lengthy integration with XML,SQL etc.	168	6.74	2.22
Entertainment.	163	6.72	2.39
Backlog applications	161	6.34	2.47
Unrealistic expectations of the user	166	6.31	2.44

As we collected data on a large number (46 characteristics) of variables, some of these characteristics may correlate with each other. This means some of them may be measuring the same construct. Therefore, it is better to reduce the 46 characteristics to a smaller number of factors. A principal component analysis (PCA) was conducted on the 46 characteristics in Table 4.1 with oblique rotation (oblimin).

The Kaiser's Measure of Sampling Adequacy (MSA) = 0.85, which is a good value, since accepted values for MSA are 0.5 barely acceptable, 0.5 - 0.7 mediocre, 0.7 - 0.8 good, 0.8 - 0.9 great and more than 0.9 is superb (Field, 2009:647). The obtained MSA value is shown in Table 4-2.

Eigenvalues of the correlation matrix were created with a total = 46, and from 46 characteristics 12 factors were retained according to Eigenvalue-one criterion with MINEIGEN = 1. This means we can retain and interpret any characteristics (factors) with an Eigenvalue greater than 1 and the number of factors extracted should be equal to the number of the Eigenvalue or variance greater than 1 (Field, 2009:640).

The communalities vary between .90 and .55, the total number of factors retained is 12, % variance explained is 72.72% and communalities vary between .90 and .55. All this information is shown in Table 4-2.

Table 4-2: Results of exploratory factor analysis on characteristics items of mobile application

MSA	N	Number of factors retained	% Variance explained	Communalities varies between
0.85	120	12	72.72	.90 and .55

We retained 12 factors from the 46 items (characteristics). The retained 12 factors with its items are shown in Table 4-3.

Table 4-3: Result of factor analysis

Construct (12 Factors)	Items in each retained factor (Characteristics)
Security	Information security. Data Security. Application security. Security and privacy.
Platform/device compatibility and adaptability	Selection of platform. (Open source or closed platform) Compatibility. User experience on multiple devices. Need to create same application for more than one platform Adaptability. Availability and accessibility.
Quality of application	Learnability. Efficiency. Maintenance factor (long-term investment). Functionality. Value to user.
Fragmentation between devices and platforms	Fragmentation within the platform (Different versions of platform available, for example Android has many versions.) Device fragmentation (Occurs when some users use old version of operating system and others use newer versions.)
Time and budget pressures	Time to market. Pressing competition. Budget required. Time required to building and delivering a complete mobile application. Availability of MADM.
Lack of development skills	Not enough developers with the right level of skills. Gap in MAD skills.
Testing and back-end issues	Developers acting as testers. Identifying innovative mobile use cases. Applications back-end. (The hosting of the application's back-end because of the limitation of the device) Complex and lengthy integration with XML,SQL etc.
Involvement of different teams and version control of project	Involvement of different development teams in a project. (Front-end team, Back-end team,

	Testing team etc.) Version control of the application software. Testing of mobile application.
Unrealistic expectations, Data intensive application and separate testing for different platforms	Unrealistic expectations of the user. Backlog applications.
Usability ,user interface design and need of SDM	User interface design. Usability - Ease of use.
Automated testing support, Utility, different language code base and entertainment factor	Availability of the automated testing support. Utility. Controlling the code base of different languages. Entertainment.
Platform fragmentation and selection of development approach	Selection of development Approach. (Native, Hybrid, HTML5 etc.) Platform Fragmentation (Wide selection of platforms for example Android, iOS, etc.).

Reliability

Reliability tests were used to test the internal consistency. The Cronbach alpha value of each retained factor is given in Table 4-4. If the Cronbach alpha ≥ 0.9 it means the instrument has high reliability and if Cronbach alpha ≥ 0.8 it has a moderate reliability. If the Cronbach alpha ≥ 0.7 it has low reliability (Field 2009:675). However, Huisman (2000:65) states that reliability of at least 0.6 should be used with exploratory research in SDMs. From Table 4-4 it is clear that reliability was acceptable at level 0.6 Huisman (2000:65).

Result of reliability test is shown in Table 4-4

Table 4-4: Results of reliability test on factors retained.

Construct	N	Items(Characteristics)	Cronbach's alpha value
Security	158	Information security. Data security. Application security. Security and privacy.	.88
Platform/device compatibility and adaptability	153	Selection of platform (Open source or closed platform). Compatibility. User experience on multiple devices. Need to create same application for more	.86

		than one platform Adaptability. Availability and accessibility.	
Quality of application	158	Learnability. Efficiency. Maintenance factor (long-term investment). Functionality. Value to user.	.85
Fragmentation between devices and platforms	167	Fragmentation within the platform. (Different versions of platform available, for example Android has many versions.) Device fragmentation (Occurs when some users use old version of operating system and others use newer versions).	.72
Time and budget pressures	157	Time to market. Pressing competition. Budget required. Time required to building and delivering a complete mobile application. Availability of MADM.	.78
Lack of development skills	162	Not enough developers with the right level of skills. Gap in MAD skills.	.68
Testing and back-end issues	163	Developers acting as testers. Identifying innovative mobile use cases. Applications back-end.(The hosting of the application's back-end because of the limitation of the device) Complex and lengthy integration with XML,SQL etc.	.71
Involvement of different teams and version control of project	161	Involvement of different development teams in a project. (Front-end team, Back-end team, Testing team etc.) Version control of the application software. Testing of mobile application.	.7
Unrealistic expectations, Data intensive application and separate testing for different platforms	161	Unrealistic expectations of the user. Backlog applications.	.65
Usability, user interface design and need of SDM	160	User interface design. Usability - Ease of use.	.74
Automated testing support, utility, different language code base	159	Availability of the automated testing support.	.68

and entertainment factor		Utility. Controlling the code base of different languages. Entertainment.	
Platform fragmentation and selection of development approach	172	Selection of development Approach. (Native, Hybrid, HTML5 etc.) Platform fragmentation. (Wide selection of platforms for example Android, iOS, etc.)	.6

Table 4-5 shows the descriptive statistics of the above retained factors.

Table 4-5: Descriptive statistics of the retained factors

Factors	N	Mean	Standard deviation
Usability, user interface design and need of SDM	163	8.79	1.42
Security	169	8.71	1.56
Involvement of different teams and version control of project	171	8.59	1.43
Quality of application	165	8.16	1.45
Platform fragmentation and selection of development approach	173	8.05	1.94
Platform/device compatibility and adaptability	167	7.74	1.64
Time and budget pressures	172	7.64	1.57
Lack of development skills	165	7.37	1.89
Fragmentation between devices and platforms	170	7.36	1.92
Testing and back-end issues	172	7.36	1.58
Automated testing support, utility, different language code base and entertainment factor	170	6.96	1.66
Unrealistic expectations, data intensive application and separate testing for different platforms	166	6.30	2.11

From Table 4.5 it is clear that the most important challenges were (Usability, user interface design and need of SDM), (Security) and (Involvement of different teams and version control of project) and the least important challenges were (Unrealistic expectations, Data intensive

application and separate testing for different platforms), (Automated testing support, Utility, different language code base and entertainment factor), (Fragmentation between devices and platforms) and (Testing and back-end issues).

4.3 Identify different system development methodologies that are available for mobile application development, for example Mobile-D, MASAM and GMA.

This objective was addressed in Chapter 2.

4.4 Investigate if system development methodologies are being used for mobile application development in the IT industry.

To answer this question, we focused on various aspects of SDM use, namely the type of SDM that is used, the intensity of use, the strictness of use, the time of use; and if the respondents foresee a possible change in SDM use in future .

The descriptive statistics for the above mentioned aspects are shown in Table 4.6

Table 4-6: Descriptive statistics for SDM-Type and its intensity, strictness of use, time of use, possible change

Type of SDM	SDM	N	Mean (Intensity)	Standard deviation
Traditional method	Traditional system development life cycle	54	5.44	2.54
Agile method	eXtreme Programming	30	6.07	2.66
Traditional method	Rational Unified Process	25	5.96	2.11
Mobile development methodology	Mobil-D	16	5.25	2.67
Mobile development methodology	Mansurov's Accelerated Development Methodology	12	5.75	2.6
Mobile development methodology	MASAM	11	6.09	0.94
Agile	Agile	11	5.27	3.46
Agile	Scrum	11	6	3.26
Mobile development methodology	GMA	0	.	.
Telecommunication	MODA-TEL	0	.	.

	Strictness of use	96	6.94	2.1
	Time of use	93	26.58	22.57
	Possible change	107	1.74	0.44

60% of the respondents said that they are using SDM and 40% respondents said that they are not using any SDM. From Table 4.6 it is surprising that 54 of the 180 respondents (30%) use the traditional system development life cycle (SDLC). In total 44% of the respondents use a SDM that could be classified as a traditional type of SDM. Furthermore, 29% of the respondents use an Agile type of SDM, and the 22% use system development methodologies that were developed for MAD. It is clear that some mobile application development methodologies are not used at all, for example GMA and MODA-TEL. This is given in red colour.

When we consider the intensity of use, the highest mean value reported was 6.09 out of 10. This indicates that the SDM are not used very intensively.

The strictness of use received a mean of 6.94/10 which means that the respondents are not following the SDM 100%. This is in line with the current trend towards the contingent use of SDM. SDM are adapted to fit the project and the context in which it should be applied.

The respondents were asked to indicate how long they have been using the SDM in their company. The respondents were captured in number of months. The mean value received is 26.58 which indicate that they using it for more than 2 years.

The respondents were also asked to indicate if they plan any changes towards the use of SDM in their organizations. The responses were captured as follows: Yes =1 and No =2. The mean value for this measurement was 1.74, which indicates that the majority of respondents don't plan any change. More precisely, 74% of the respondents indicated that they don't plan any change regarding SDM use, while 26% indicated that they do plan possible changes.

4.5 Determine the success of MAD

In order to examine the success of current system development methodologies for MAD we focussed on Success of mobile application, Delivered on time; and Completed in budget. Descriptive statics for Success of mobile application, Delivered on time and Completed in budget is shown in the Table 4.7.

Table 4-7: Descriptive statistics' for Success of mobile application, Delivered on time and Completed in budget

	N	Mean	Standard deviation
Success of mobile application	171	7.75	1.55
Delivered on time	167	7.40	2.02
Completed in budget	166	7.25	2.27

From Table 4-7 Success of mobile application received a mean of 7.75/10 and Delivered on time received a mean of 7.40/10 and Completed in budget received a mean of 7.25/10. From this result it is clear that people are happy with the success of mobile application and delivered on time. However, the budget is problematic.

4.6 Evaluate the suitability of current system development methodologies for mobile application development.

In order to evaluate the suitability of current system development methodologies for MAD, SDM use and outcome was correlated.

For SDM use the items Strictness of use and Time of use were taken, and for outcome the items Success of mobile application, Delivered on time, and Completed in budget.

Correlation between (successes of mobile application) and (strictness, time of use) is shown in Table 4.8

Table 4-8: Correlation between (successes of mobile application) and (strictness, time of use)

	Pearson Correlation Coefficients	Prob > r under H0:Rho=0	Number of observations
Strictness	0.4	0.0004	95
Time of use	0.2	0.1033	92

From table 4.8 it is clear that successes of mobile application has a medium relationship with strictness of use. This is because there is a medium relationship if the Pearson Correlation Coefficient = .3 (Field 2009:170)

Correlation between (delivered on time) and (strictness, time of use) is shown in Table 4-9.

Table 4-9: Correlation between (delivered on time) and (strictness, time of use)

	Pearson Correlation Coefficients	Prob > r under H0:Rho=0	Number of observations
Strictness	0.4	0.0012	95
Time of use	0.2	0.0502	92

From table 4.9 it is clear that delivered on time has a medium relationship with strictness of use of SDM. This is because there is medium relationship if Pearson Correlation Coefficient = .3 (Field 2009:170)

The correlation between (completed in budget) and (strictness, time of use) is shown in table 4.10

Table 4-10: Correlation between (completed in budget) and (strictness, time of use)

	Pearson Correlation Coefficients	Prob > r under H0:Rho=0	Number of observations
Strictness	0.1	0.4231	94
Time of use	0.1	0.3447	91

Table 4.10 shows there is small relation with (completed in budget) for (strictness) and (Time of use). This is because there is small relationship if Pearson Correlation Coefficient = .1 (Field 2009:170)

Factor analysis for outcome is shown in Table 4.11

Table 4-11: Factor analysis for outcome

	Standardized scoring coefficients
Success of mobile application	0.34939
Delivered on time	0.45583
Completed in budget	0.44493

Success of mobile application, Delivered on time and Completed in budget resulted in one factor called “success factor”

The reliability test on Success factor is given in below Table 4.12

Table 4-12: Reliability test on Success factor

Construct	N	Items(Characteristics)	Cronbach's alpha value
Success factor	162	Success of mobile application Delivered on time Completed in budget	0.71

As the Cronbach alpha value is 0.71, “Success factor” is reliable and is accepted (Field, 2009:675).

Correlations between (Success factor) and system methodology use (strictness and time of use) is then calculated Correlation between (Success factor) and SDM use is given below in Table 4.13

Table 4-13: Correlation between (Success factor) and SDM use

	Pearson Correlation Coefficients	Prob > r under H0:Rho=0	Number of observations
Strictness(question 23)	0.3	0.0034	96
Time of use(question 24)	0.2	0.0537	93

It is clear that there is a medium effect between success of mobile application (Success factor) and the strictness of use. This is because there is medium relationship if Pearson Correlation Coefficient = .3 (Field 2009:170).

4.7 Summary

The aim of the study is to investigate the suitability of systems development methodologies for mobile application development. For this we did the following objectives.

- Investigate the challenges faced by companies which are currently developing mobile applications.
- Identify different system development methodologies that are available for mobile application development, for example Mobile-D, MASAM and GMA.
- Investigate if system development methodologies are being used for MAD in the IT industry.
- Determine the success of current system development methodologies for mobile application development.
- Evaluate the suitability of current system development methodologies for mobile application development.

The most important challenges were (Usability, user interface design and need of SDM), (Security) and (Involvement of different teams and version control of project) and the least important challenges were (Unrealistic expectations, Data intensive application and separate testing for different platforms), (Automated testing support, Utility, different language code base and entertainment factor), (Fragmentation between devices and platforms) and (Testing and back-end issues). 60% of the respondents said that they are using SDM and 40% respondents said that they are not using any SDM. It is surprising that 54/180 of the respondents use the traditional system development life cycle. It is also clear that agile system development methodologies are also used. It is interesting that system development methodologies developed for MAD are also used. It is clear that some MAD methodologies are not used at all. Most of the people are happy with the current development methodologies. They are following only the SDM, but they are not following it 100% strictly. The success of mobile application has medium relationship with “strictness” of use of SDM. It is clear that “delivered on time” has a medium relationship with “strictness” of use of SDM. The data also shows that there is a medium effect between success of mobile application and the strictness of use. If we use SDM strictly we can improve success of mobile application and we can deliver the project on time.

Chapter 5 describes the discussion, comments and conclusion.

CHAPTER 5 DISCUSSION, COMMENTS AND CONCLUSION

5.1 Introduction

The purpose of this study was to determine the use and effectiveness of system development methodologies in mobile application development. This chapter gives the summary of the research objectives, summary of the research method and the findings. The research objectives were as follows.

- Investigate the challenges faced by companies which are currently developing mobile applications.
- Identify different system development methodologies that are available for mobile application development, for example Mobile-D, MASAM and GMA.
- Investigate if system development methodologies are being used for MAD in the IT industry.
- Determine the success of MAD.
- Evaluate the suitability of current SDMs for MAD.

The study used a survey as a research method. The survey was conducted during April- May 2015 and it was carried out in 20 countries, mainly in South Africa, India, Pakistan, Germany etc. The researcher compiled a database of companies doing MAD using Google, LinkedIn etc. Thereafter the researcher compiled a list of people working in those companies using LinkedIn and Facebook. Initially the questionnaires were electronically distributed. Some participants requested a web questionnaire as it is more convenient for them. The researcher used Google form to create questionnaire. Emails were sent to all the participants, followed by weekly reminders to increase the responses. We targeted individuals developing or managing mobile application developments. In this study we used electronic questionnaires and web questionnaire for data collection.

5.2 Investigate the challenges faced by companies which are currently developing mobile applications

The study revealed that the most important challenges were Usability, user interface design, Need of SDM, Security, Involvement of different teams and version control of project. The least important challenges were Unrealistic expectations, Data intensive application, separate testing for different platforms, Automated testing support, Utility, different language code base and entertainment factor, Fragmentation between devices and platforms, and Testing and back-end issues.

5.3 Identify different system development methodologies that are available for mobile application development, for example Mobile-D, MASAM and GMA.

This objective was addressed in Chapter 2. Table 5.1 below gives a summary of the three methodologies.

Table 5-1: Summary of the three methodologies

	Mobile-D	MASAM	GMA
Philosophical paradigm	Agile development process and fast product delivery	Agile development process, simple and fast product delivery	Adaptive framework with client-server computing
Process model	Iterative model	Sequential and iterative	Parallel
Method	Five phases of iteration set-up: phase, core phase, core-2 phase, stabilize phase and wrap-up phase and each phase consists of 3 different development days Planning Day, Working Day and Release Day.	Includes four phases in development: Preparation Phase, Embodiment Phase, Product Development Phase, Commercialization Phase	Three-tier architecture with front-tier, middle-tier and backend-tier
Tools	Phasing and pacing, architectural line and pair programming	Test driven, product summary, story card and pair programming	Backend services, universal service interface, XML

5.4 Investigate if system development methodologies are being used for MAD in the IT industry.

In our study 60% of the respondents said that they are using SDM and 40% respondents said that they not using any SDM. We found that 30% of the respondents use the traditional system development life cycle. In total 44% of the respondents use a SDM that could be classified as traditional type. It is also clear that agile system development methodologies are also used. Furthermore, 29% of the respondents use an agile type of SDM. It is interesting that 22% of the respondents use SDMs developed for MAD. But some MAD methodologies like GMA, MODA-TEL are not used at all. The respondents are using SDMs for more than 2 years. But they are not using SDMs very intensively. The results also show that the respondents are not following the SDM 100%. This shows that the current trend is towards the contingent use of SDM. SDM are adapted to fit the project and the context in which it should be applied. Most of the

respondents are happy with the current SDMs. 74% of the respondents indicated that they don't plan any change regarding SDM use, while 26% indicated that they do plan possible changes in the use of SDM.

5.5 Determine the success of MAD.

In order to find out the success of MAD we focussed on different aspects like Success of mobile application, Delivered on time and Completed in budget. From the study it is clear that people are happy with the success of the mobile application as we received a mean value of 7.75 / 10 for the success of mobile application and delivered on time received a mean value of 7.25 / 10 which is also closer to 10. However, the budget got a mean value of 7.25 / 10 which is problematic, so there is a room for improvement.

5.6 Evaluate the suitability of current SDMs for MAD

The study revealed that successes of mobile application have a medium relationship with strictness of use. In addition, delivered on time has a medium relationship with strictness of use of SDM. There is no relation with completed in budget for Strictness and Time of use. It is clear that there is a medium effect between success of mobile application and the strictness of use of SDM.

5.7 Contribution

The study reveals that to increase the success of mobile application and to deliver the project on time, developers must follow the SDM strictly.

5.8 Future work and limitations

In order to find out the use and effectiveness of SDM for MAD data collection was conducted mainly in South Africa, India, Pakistan, Germany etc. and only 180 responses were received. In future work one can increase the number of responses to above 500 by doing survey in more countries. In order to solve the challenges in MAD, one can try to develop a new SDM for MAD that will address all the factors of MAD success.

From Table 3-22 it is clear that 26.17% respondents reported that they want a new SDM for MAD because of the following reasons.

- One of the respondents found that certain improvements are required in the current SDM. According to him the communication between the groups should improve. Also,

there should be a central section where every member can view the progress in graphical/flowchart manner. This will help them considerably.

- Another respondent suggested that there is a missing phase, "Design Integration Phase" and he suggested that it should be in between design and the development phase.
- Another respondent reported that the current methodology is not flexible enough for the various mobile applications they are building.
- Another respondent reported that they carry SDM for mobile application, which is applicable for all projects such like java and .Net, but some of the processes are not required for mobile applications and can be removed.
- Another participant reported that the existing methodology focuses only on developing systems that are more formal and larger. Using such SDM for a mobile developer won't work, especially if the developer comes up with the idea does all the development himself as a start-up. The respondent suggested a new SDM for start-up developers.
- One of the respondents needed more agile SDM with quick sprint cycles and quick release and proper documentation.
- SDM should support work on multiple projects and rapid changing requirements.
- One respondent was looking for a SDM that helped them to deliver the project on time, within budget, and the expected quality.

This data can also take into account for future work in developing a new SDM for MAD.

5.9 Summary

This chapter briefly provided summary of the research objectives, summary of research method findings and pointed out options for future work.

BIBLIOGRAPHY

- Abrahamsson, P., Hanhineva, A., Hulkko, H., Tuomas Ihme, T., Jaalinoja, J., Korkala, M., Koskela, J., Kyllonen, P. & Salo, O. 2004. Mobile-D: an agile approach for mobile application development. Conference on object-oriented programming systems, languages, and applications. Companion to the 19th annual ACM SIGPLAN, Vancouver, 24-28 October.
http://www.researchgate.net/publication/221322054_Mobile-D_an_agile_approach_for_mobile_application_development Date of access: Apr. 10 2014.
- Ambler,. 2008. Agile adoption rate survey results: February 2008
<http://www.ambysoft.com/surveys/agileFebruary2008.html> Date of access: 15 Nov. 2013.
- Ambler, 2007. Agile adoption survey: March 2007.
<http://www.ambysoft.com/downloads/surveys/AgileAdoption2007.ppt> Date of access: 15 Nov. 2013.
- Ambler, 2006. Agile adoption rate survey: March 2006.
<http://www.ambysoft.com/downloads/surveys/Agile> Date of access: 11 Nov. 2013.
- Avison, D. & Fitzgerald, G. 2006. Information systems development: Methodologies, techniques and tools. 4th ed. Berkshire, England: McGraw Hill.
- Avison, D. & Fitzgerald, G. 2006. Methodologies for developing information systems: A historical perspective. *The past and future of information systems: 1976–2006 and beyond*, IFIP international federation for information processing, 214:27-38.
- Avison, D.E. & Fitzgerald, G. 2003. Where now for development methodologies? *Communications of the ACM*, 46(1):78-82.
- Baker, I. 2014. Skills gap still the biggest challenge in mobile app delivery.
<http://betanews.com/2014/10/28/skills-gap-still-the-biggest-challenge-in-mobile-app-delivery/> Date of access: 20 Sep. 2015.
- Ballon, P. 2009. Platform types and gatekeeper roles: The case of the mobile communications industry. Summer conference, Denmark, 17-19 June 2009.
<http://www2.druid.dk/conferences/viewpaper.php?id=5952&cf=32> Date of access: May 20 2014.
- Ballon, P. & Walravens, N. 2008. Competing platform models for mobile service delivery: The importance of gatekeeper roles. 7th international conference on mobile business, Barcelona, 7-8 July 2008.

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4570169&queryText%3DCompeting+Platform+Models+for+Mobile+Service+Delivery%3A+the> Date of access: 20 May 2014.

Bangerter, B., Talwar, S., Arefi, R. & Stewart, K. 2014. Networks and devices for the 5G era. *Communications magazine, IEEE*, 52(2):90-96.

Beck, K., et al. 2000. Extreme programming explained: Embrace change. 2nd ed. Addison-Wesley.

Beck., K., et al. 2001. Manifesto for agile software development. <http://agilemanifesto.org/> Date of access: 8 May 2014.

Bhatia, N. 2013. Avoid these 6 mobile app development challenges. <http://www.propelics.com/avoid-6-mobile-app-development-challenges/> Date of access: 21 Sep. 2015.

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F. & Mylopoulos, J. 2004. Tropos: An agent-oriented software development methodology *Kluwer academic publishers. manufactured in the Netherlands.*, 8(3):203-236.

Breytenbach, M.C., Toit, H.D. & Gouws, M.L. 2012. The development and evaluation of a mobile application development methodology. Potchefstroom: NWU. (BSc Hons in Information Technology). (Unpublished).

Cheng, M.C. & Yuan, S.M. 2007. An adaptive and unified mobile application development framework for java. *Journal of information science and engineering*, 23(5):1391-1405.

Chitforoush, F., Yazdandoost, M. & Ramsin, R. 2007. Methodology support for the model driven architecture. 14th Asia-pacific software engineering conference, 2007. APSEC 2007. Aichi, 4-7 Dec. 2007.

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4425887&queryText%3DMethodology+support+for+the+model+driven+architecture> Date of access: 14 May 2014.

Dahiya, D. & Jain, P. 2010. Enterprise systems development: Impact of various software development methodologies. *International journal of advancements in computing technology*, 2(4):77-87.

Dam, H.K. & Winikoff, M. 2003. Comparing agent-oriented methodologies. Aois 2003, Melbourne, Australia, 14 Jul. 2003. http://link.springer.com/chapter/10.1007/978-3-540-25943-5_6#page-1 Date of access: 10 June 2014.

- Don, M. 2014. 2015 app dev priorities survey change will do you good.
<http://reports.informationweek.com/abstract/22/12540/SOA-App-Architecture/2015-App-Dev-Priorities-Survey.html> Date of access: 24 January. 2015.
- Dunnewijk, T. & Hultén, S. 2007. A brief history of mobile communication in Europe.
Telematics and informatics, 24(3):164-179.
- Fairchild, C. 2014. Mobile app development: 5 worst security dangers.
<http://www.informationweek.com/mobile/mobile-applications/mobile-app-development-5-worst-security-dangers/d/d-id/1204488> Date of access: 20 November 2014.
- Field, A. 2009. Discovering statistics using SPSS. 3rd ed. London: Sage publications.
- Gasimov, A., Tan, C.H., Phang, C.W. & Sutanto, J. 2010. Visiting mobile application development: What, how and where. Mobile business and 2010 ninth global mobility roundtable (ICMB-GMR), 2010 ninth international conference, Athens, Greece, 13-15 June 2010.
<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5494784&newsearch=true&queryText=Visiting%20Mobile%20Application%20Development:%20What,%20How%20and%20Where> Date of access: 25 June 2014.
- Highsmith, J.A.,. 2000. Adaptive software development: A collaborative approach to managing complex systems. Dorset House pUB.
- Hijazi, H., Khmour, T. & Alarabeyyat, A. 2012. A review of risk management in different software development methodologies. *International journal of computer applications*, 45(7):8-12.
- Holzer, A. & Ondrus, J. 2011. Mobile application market: A developer's perspective.
Telematics and informatics, 28(1):22-31.
- Huisman, H.M. 2000. The deployment of systems development methodologies: A South African experience. Potchefstroom: PU for CHE. (PhD.).
- Huisman, M. & Livari, J. 2006. Deployment of systems development methodologies: Perceptual congruence between IS managers and systems developers. *Information & management*, 43(1):29-49.
- Livari, J., Hirschheim, R. & Klein, H.K. 2001. A dynamic framework for classifying information systems development methodologies and approaches *Journal of management information systems / Winter 2000 - 2001*, 17(3):179-218.

Jennings, N. & Wooldridge, M.J. 1998. Agent technology: Foundations, applications, and markets. Springer.

Jeong, Y.J., Lee, J.H. & Shin, G.S. 2008. Development process of mobile application SW based on agile methodology. 10th international conference on advanced communication technology, 2008. ICACT 2008. Gangwon-Do, 17-20 February.

http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4493779&tag=1 Date of access: 14 June 2014.

Jiang, Y., Liu, L., Lv, W. & Zhao, C. 2009. Metadata oriented software development method. 2nd international conference on biomedical engineering and informatics, 2009. BMEI '09. Tianjin, 17-19 October 2009.

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5305826&queryText%3DMetadata+Oriented+Software+Development+Method> Date of access: 14 March 2014.

Johnson. 2003. Agile methodologies survey result

http://www.shinetech.com/agile_survey_results.jsp Date of access: 10 Nov. 2013.

Johnson, G. 2014.

The time-to-market advantage in mobile application development.

<http://magicapplicationplatform.blogspot.co.za/2014/04/the-time-to-market-advantage-in-mobile.html>

Date of access: 23 Jun. 2015.

Joorabchi, M.E., Mesbah, A. & Kruchten, P. 2013. Real challenges in mobile app development. Empirical software engineering and measurement, 2013 ACM / IEEE international symposium , Baltimore, MD, 10-11 October 2013.

http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6681334&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6681334 Date of access: 21Sep. 2015.

Klimes, C. & Prochazka, J. 2006. New approaches in software development. *Acta electrotechnica et informatica*, 6(2):1-6.

Labaree, R. 2013. Quantitative methods -organizing your social sciences research paper.

<http://libguides.usc.edu/content.php?pid=83009&sid=615867> Date of access: 5 Apr. 2014.

Lam, M. 2012. Methodologies,tools, and techniques in practice for web application development. *Journal of technology research*. Jul 2012, 3(1):1-20.

Levy, N. 2011. Mobile development challenges. <http://www.slideshare.net/levynir/mobile-development-challenges-10288414> Date of access: 20 Sept. 2015.

- Mansurov, N.N. 2000. Formal methods for accelerated development of telecommunications software. *In ISP collected papers 1999-2000*, 11-36. (In press.).
- Mnkandla, E. 2009. About software engineering frameworks and methodologies. Africon '09. , Nairobi, 23-25 Sept. 2009. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5308117 Date of access: 12 Mar. 2014.
- Moniruzzaman, A.B.M. & .Hossain,D.S.A. 2013. Comparative study on agile software development methodologies. *Global journal of computer science and technology* , 13(7):1-25.
- Oates, B.J. 2006. Researching information systems and computing. 1st ed. 1 Oliver's Yard 55 City Road London EC1 Y 1SP: SAGE Publications Ltd.
- Otto, C.H. 2007. The use of systems development methodologies in the mobile telecommunication software in S.A. Potchefstroom: NWU. (MSc.).
- Pikkarainen, M., Haikara, J. ,Salo, O ,Abrahamsson, P.,& Still ,J. 2008. The impact of agile practices on communication in software development. *Empirical software engineering*, 13(3)303-337.
- Rahimian, V. & Ramsin, R. 2008. Designing an agile methodology for mobile software development: A hybrid method engineering approach. Research challenges in information science, 2008. RCIS 2008. second international conference, Marrakech, 3-6 June 2008. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4632123&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4632123 Date of access: 1 Sept. 2015.
- Riemenschneider, C.K., Hardgrave, B.C. & Davis, F.D. 2002. Explaining software developer acceptance of methodologies: A comparison of five theoretical models. *IEEE transactions on software engineering*, 28(12):1135-1145.
- Rudnick, B. 2013. Agile versus traditional – A tale of two methodologies. *Ground System Architectures workshop*, 1(1):1-22.
- Schoknecht, P. 2012. 5 factors of mobile application success. <http://www.clickz.com/clickz/column/2144961/factors-mobile-application-success> Date of access: 20 Sept. 2015.
- Shevchik, L. 2012. Top 3 challenges facing mobile development. <https://blog.newrelic.com/2012/11/15/top-3-challenges-facing-mobile-development/> Date of access: 20 Sept. 2015.

TechWellPresentations. 2014. Mobile application testing: Challenges and best practices .
http://www.slideshare.net/TechWellPresentations/bw4-30357244?qid=012c8441-96be-4d15-94ed-e8a3a22bc3a8&v=default&b=&from_search=12 Date of access: 21 Sept. 2015.

Tode, C. 2014. Significant app development backlog costs companies revenue, customers: Report. <http://www.mobilemarketer.com/cms/news/research/19002.html> Date of access: 20 Sept. 2015.

Tracy, K.W. 2012. Mobile application development experiences on Apple's iOS and android OS. *Potentials, IEEE*, 31(4):30-34.

Wyse, S.E. 2012. Which is more effective: Paper-based surveys or online surveys?.
<http://www.snapsurveys.com/blog/which-is-more-effective-paper-based-surveys-or-online-surveys/> Date of access: 12 Sept. 2015.

Zambonelli, F. 2010. Agent-oriented software engineering.
<http://www.agentgroup.unimore.it/wiki/images/6/6b/AOSE.pdf> Date of access: 5 Mar. 2015.

Zhang, D. & Adipat, B. 2009. Challenges, methodologies, and issues in the usability testing of mobile applications. *International journal of human-computer studies*, 18(3):293-308.

APPENDIX - QUESTIONNAIRE



Private Bag X05, Noordbrug
South Africa 2522
Tel: 018 299-2000
Fax: 018 299-2999
Web: <http://www.nwu.ac.za>

School of Computer, Statistical and Mathematical Sciences

Dear Sir/Madam

My name is Smitha Zacaria and I am a Master student at North West University South Africa. For my research project, I am examining mobile application development. The goal of the project is to identify the use and effectiveness of system development methodologies in mobile application development. Because you are an expert in the mobile application field I am inviting you to participate in this survey by completing the attached questionnaire. It will only take 15 minutes of your time.

If you choose to participate in this project, please answer all questions as honestly as possible and return the completed questionnaire to mail2smithav@yahoo.co.in.

Thank you for taking the time to assist me in my educational endeavors. The data collected will be treated confidentially, and provide useful information regarding the development of new system development methodology for mobile application development.

We thank you for your participation and look forward to hear from you.

With Kind Regards

Student: Smitha Zacaria

E-Mail: mail2smithav@yahoo.co.in

Prof. Magda Huisman

Research Director

E-Mail: Magda.Huisman@nwu.ac.za

North West University

South Africa

Questionnaire

[Please select the answers and save the file and send it back]

1. Indicate the country in which you are working.

[Click here to enter text.](#)

2. Please enter your home language.

[Click here to enter text.](#)

3. Please select your gender.

☐ Male ☐ Female

4. Please select your age group.

☐ Below 20 ☐ 20 - 25 ☐ 26 - 30 ☐ 31 - 35 ☐ 36 - 40 ☐ 41 - 45 ☐ 46 - 50 ☐ Above 50

5. Total number of employees in the company.

☐ 1 to 10 ☐ 10 to 50 ☐ 50 to 250 ☐ More than 250

6. What is your business area?

☐ Banks ☐ Insurance ☐ Government ☐ Education ☐ Manufacturing ☐ Others

[Click here to specify if 'Others'.](#)

7. Is the mobile application that you develop for internal or external uses?

☐ Internal uses ☐ External uses ☐ Both

8. How many are working on the mobile application project team?

[Click here to enter answer.](#)

9. Is there any open vacancies due to the shortage of skilled developers in the company?

☐ Yes ☐ No

If "Yes" how many?

[Click here to enter answer.](#)

10. How many years of IT working experience do you have?

☐ 0 to 1 ☐ 1 to 2 ☐ 2 to 4 ☐ 4 to 8 ☐ More than 8

11. What is your current position?

[Click here to enter text.](#)

12. How many mobile applications have you personally developed or take part in the development process?

[Click here to enter answer.](#)

13. Select the platform you using for mobile application development?

☐ Android ☐ iOS ☐ HTML5 ☐ Mac ☐ Others

[Click here to specify if 'Others'.](#)

14. How long does it normally take to build a mobile application?

[Click here to enter answer.](#)

15. How successful was the last mobile application development project that you were involved with? Please score between points 1 to 10 where 1 = not successful and 10 = extremely successful

[Please select the score.](#)

16. How often is your mobile application projects delivered on time that is within the original planned time frame on a scale from 1 – 10, where 1 = never, 5 = sometimes and 10 = always.

[Please select the score.](#)

17. How often is your mobile application projects completed within the original planned budget on a scale from 1 – 10 , where 1 = never ,5 = sometimes and 10 = always.

[Please select the score.](#)

18. Which application development environment or model are you leaning towards when developing mobile applications?

☐ Native ☐ Hybrid ☐ Mobile web or HTML5 ☐ Others

[Click here to specify if 'Others'.](#)

19. What is your primary motivation for mobile application development?

☐ Profit ☐ Extension of company portfolio to attract more external clients ☐ Brand

☐ Others

[Click here to specify if 'Others'.](#)

20. Please score the importance of the following characteristics regarding mobile application development on a scale 1 to 10; where 1 = Not important at all; 10 = extremely important.

Sample Characteristics	4
Platform Fragmentation. (Wide selection of platforms for example Android, iOS, etc.)	Please select the score.
Fragmentation within the platform. (Different versions of platform available, for example Android has many versions.)	Please select the score.
Device Fragmentation. (Occurs when some users use old version of operating system and others use newer versions.)	Please select the score.
Selection of development Approach. (Native, Hybrid, HTML5 etc.)	Please select the score.
Identifying innovative mobile use cases.	Please select the score.
Time to market.	Please select the score.
Pressing Competitions.	Please select the score.
Application Distribution to various markets.	Please select the score.
Version control of the application software.	Please select the score.
Involvement of different development teams in a project.(Front end team, Back end team, Testing team etc.)	Please select the score.
Complex and lengthy integration with XML,SQL etc.	Please select the score.
Controlling the code base of different languages.	Please select the score.
Ongoing effort to support new devices as they just keep popping up all the time.	Please select the score.
Application security.	Please select the score.
Information security.	Please select the score.
Data Security.	Please select the score.
Applications back end.(The hosting of the application's back end because of the limitation of the device)	Please select the score.
Testing of mobile application.	Please select the score.
Availability of the automated testing support.	Please select the score.
Developers acting as testers.	Please select the score.
Testing the application for each platform separately.	Please select the score.
Selection of platform.(Open source or Closed platform)	Please select the score.
Need to create same application for more than one platform.	Please select the score.
Data intensive application.(High data of data to be stored on the device)	Please select the score.
Budget required.	Please select the score.
Time required to building and delivering a complete mobile application.	Please select the score.
Gap in mobile application development skills.	Please select the score.
Not enough developers with the right level of skills.	Please select the score.

3

Availability of mobile application development methodology.	Please select the score.
Need of a system development methodology for mobile application development.	Please select the score.
Backlog applications	Please select the score.
Unrealistic expectations of the user.	Please select the score.
Entertainment.	Please select the score.
Utility.	Please select the score.
Functionality.	Please select the score.
Maintenance factor. (long term investment)	Please select the score.
Learnability.	Please select the score.
Efficiency.	Please select the score.
Value to user.	Please select the score.
Compatibility.	Please select the score.
Usability - Ease of use.	Please select the score.
Availability and accessibility.	Please select the score.
Adaptability.	Please select the score.
User interface design.	Please select the score.
User experience on multiple devices.	Please select the score.
Security and privacy.	Please select the score.

21. Are you using any system development methodology for mobile application development?

☐ Yes ☐ No

(If 'No' you only need to answer question no 26)

22. If you using a system development methodology(SDM) for mobile application development please select the one you are using and indicate how intensively you are using it on a scale from 1 to 10 where 1 = very few components of the SDM and 10 = all components of the SDM.

- | | |
|--|--------------------------|
| ☐ Traditional system development life cycle | Please select the score. |
| ☐ Rational Unified Process | Please select the score. |
| ☐ eXtreme Programming | Please select the score. |
| ☐ MASAM | Please select the score. |
| ☐ Mobile-D | Please select the score. |
| ☐ GMA | Please select the score. |
| ☐ MODA-TEL | Please select the score. |
| ☐ Mansurov's Accelerated Development | Please select the score. |

☐ Others

[Click here to specify if 'Others'. Please select the score.](#)

23. How strictly are you using the system development methodology? You can score points from 1 to 10 where 1= Very loosely and 10 = Very strictly for all the projects.

[Click here to select the score.](#)

24. How long have you been using the system development methodology in the company?

[Click here to enter text.](#)

25. Are you looking for a new system development methodology for mobile application development?

☐ Yes

☐ No

[Click here to enter reasons if 'Yes'.](#)

26. Why are you not using any system development methodology? Explain the process of development you are following.

[Click here to enter reasons and process](#) .

Thank you for participating in this survey.It is highly appreciated.

PROOF OF LANGUAGE EDITION



Director: CME Terblanche - BA (Pol Sc), BA Hons (Eng), MA (Eng), TEFL

22 Strydom Street
Baillie Park, 2531

Tel 082 821 3083
cumlaudelanguage@gmail.com

DECLARATION OF LANGUAGE EDITING

I, Christina Maria Etrechia Terblanche, hereby declare that I edited the
research study titled:

**The use and effectiveness of system development methodologies in
mobile application development**

for S Zacaria for the purpose of submission as a postgraduate dissertation.
Changes were suggested and implementation was left to the discretion of
the author.

Regards,

CME Terblanche

Cum Laude Language Practitioners (CC)

SATI accr nr: 1001066

PEG registered