The contingent use of systems development methodologies in South Africa

B Moyo

orcid.org/0000-0002-6609-8165

Thesis accepted in fulfilment of the requirements for the degree *Doctor of Philosophy in Computer Science* at the North-West University

Promoter: Prof HM Huisman

Co-promoter: Prof L Drevin

Graduation May 2021

23630590

DECLARATION

I, Benson Moyo, hereby declare that this thesis titled: “The contingent use of systems development methodologies in South Africa” is my own original work, and that the sources have been correctly reported and properly acknowledged, and that this thesis has not been submitted in any form to any university or institution for academic qualification purposes.

Signature: 

DECLARATION: LANGUAGE EDITOR AND PROOFREADER

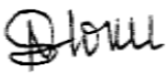
Editing and Proofreading Report

13 December 2020

This is to declare that I, Dr I. Ndlovu of the English Department, University of Venda, have proofread and edited a PhD thesis titled: “The contingent use of systems development methodologies in South Africa” by Benson Moyo.

I carefully read through the thesis, focusing on proofreading and editorial issues.

Yours Sincerely



Dr Isaac Ndlovu, PhD
Lecturer: English Department
University of Venda
Private Bag X5050
Thohoyandou 0950
South Africa
Tel.: +27 15 962 8361
Fax: +27 15 962 4749
E-mail:
isaac.ndlovu@univen.ac.za

ACKNOWLEDGEMENTS

The thesis is an artefact based on a synthesis of research work conducted under the expert supervision and guidance of Prof. Magda Huisman and Prof. Lynette Drevin. The pursuit of a higher degree takes time, patience and support from others. During this unforgettable experience, I am indebted to many people who contributed in different ways irrespective of their areas and levels of expertise. Therefore, I would like to express my appreciation to all those who supported me directly or indirectly in this PhD journey.

Firstly, I am deeply grateful to my promoter Prof. Magda Huisman for recognising my potential and providing me with this great opportunity to do research and protected me throughout this research journey. Her invaluable encouragement, insightful expert guidance, support and constructive criticism during my PhD study kept me motivated and focused. She provided timely highly consistent and professional feedback on countless iterations on each stage of the research process. This gave me the courage and the confidence during hard times. Thank you so much for making me feel I was making progress especially during the conceptual model development and the hypothesis formulation where, I displayed the depth of my weaknesses and you allowed the evolution of the conceptual model to go through even more iterations.

Secondly, I am deeply grateful to my co-promoter Prof. Lynette Drevin for her insightful detailed expert supervision and guidance. Her encouragement was an important driving force for me to navigate through my research. Your assistance is acknowledged and greatly appreciated.

This work would not have been possible without the assistance of the systems development organisations in South Africa that gave up their valuable time and provided data required for the study.

Many thanks to Pfarelo Peacenolance Moyo who was used as a conceptual test pad during the whole research journey. Thank you for your patience, support and encouragement during my PhD studies. I would not expect more than your being there for me. I love you my beautiful wife. To my two boys Nhlalwenhle Nkalipho and Nhlanhlenhle Banele, thank you for behaving well despite my failure to give you the attention you deserved during my study.

I would like to thank Dr. Gift Dafuleya for providing me with his constructive comments and brilliant suggestions. Many thanks to Prof. Barry Dwolasky for explaining the reasons behind the discrepancy between the actual systems development practice and what the systems development industry say they practice. Many thanks to Prof. Khulumani Sibanda for his advice and suggestions. To colleagues and friends throughout my study, I am extremely grateful for your support.

I would like to acknowledge the support of my parents, my late father Nhlalo Moyo, my mother Mantombi Moyo who saw me start but left before I completed this research work, and my late brother Allan Moyo. May they all rest in eternal peace. To my brothers Clenten, Danisa, Fanuel and my sisters Sifelumusa and Merlyn thank you for your encouragement, love and support.

I have tried not to omit any person who was involved or contributed towards the completion of this research, but if anyone feels left out, I beg for forgiveness.

ABSTRACT

There are two main classes of Systems Development Methodologies (SDMs) that are commonly deployed in the systems' development projects. These are the plan-driven SDM class and the agile SDM class. Previous research has focused on proving the superiority of one class over the other, without carefully analysing the rationale behind each SDM’s origins. SDM is a fundamental concept in systems development. The study argues that SDMs can be compared based on the underlying philosophical assumptions. The focus, however, posited by the study on the comparison is different from the perspective of promoting one SDM over the other. The comparison is viewed as an assessment criterion to determine the strength and the limitations of an SDM in a system’s development project-specific contextual stressors and combining them to gain synergies not possible with the deployment of any one SDM.

It is demonstrated in the literature that the evolution and transition of SDMs went through four generations: the pre-SDM, the early SDM, SDM and the post-SDM period. Each period had a specific focus on the systems' development contextual stressors. The current period is the post-SDM or the contingent use of SDMs or hybrid SDM era where, SDMs are no longer viewed as complete packages to address the systems' development projects. They are viewed as complementing each other. A conceptual definition of SDM and the role of SDMs in the systems' development field is outlined. The deployment of SDMs is often conceptualized as a once-off event, whereas it is a continuous process. The research uses the contingency theory and the theory of innovation adoption to develop a contingent use of SDMs conceptual model. The conceptual model was tested using survey data collected from the systems' development practitioners. Statistical analysis was performed and this resulted in the development of a refined contingent use of SDMs conceptual model. The study concludes that the two SDM classes are viewed as extreme opposites, but in practice, they complement each other. SDMs are combined to create hybrid SDMs or interleaved on the same systems' development project. Literature and statistical findings have been used to develop guidelines towards the contingent use of SDMs.

Keywords: agile SDM, conceptual model, guidelines, hybrid SDM, plan-driven SDM, SDM

OPSOMMING

Daar is twee hoofklasse van stelselontwikkelingsmetodologieë (SDM's) wat gewoonlik in stelsel-ontwikkelingsprojekte gebruik word. Dit is, onderskeidelik, die plan-gedrewe SDM-klas en die "*agile*" SDM-klas. Vorige navorsing het daarop gefokus om die voortreflikheid van die een klas bo die ander te bewys, sonder om die rasionaal agter elke SDM se oorsprong noukeurig te ontleed. SDM is 'n fundamentele konsep in stelselontwikkeling. Die studie voer aan dat SDM's vergelyk kan word op grond van die onderliggende filosofiese aannames. Die fokus wat die studie op die vergelyking stel, verskil egter van die uitgangspunt om die een SDM bo die ander te stel. Die vergelyking word beskou as 'n assesseringsmaatstaf om die sterkpunte en beperkings van 'n SDM in 'n stelsel-ontwikkelingsprojek-spesifieke kontekstuele stressors te bepaal, en dit te kombineer om sodoende sinergieë te verkry wat nie andersins haalbaar is deur die implementering van 'n enkele spesifieke SDM nie.

Literatuurstudies dui daarop dat SDM's deur vier generasies van evolusie en oorgang gegaan het: die pre-SDM, die vroeë SDM, SDM en die post-SDM tydperk. Elke tydperk het spesifiek gefokus op die kontekstuele stressors van stelsel-ontwikkeling. Die huidige SDM periode is bekend as die post-SDM of die gebeurlikheids gebruik van SDM's of hibriede SDM-era, waar SDM's nie meer as volledige pakkette beskou word wat die stelsel-ontwikkelingsprojekte kan aanspreek nie. SDM's word beskou as aanvullend tot mekaar. 'n Konseptuele definisie van SDM's, asook die rol van SDM's in die stelsel-ontwikkelingsveld word in hierdie studie uiteengesit. Die implementering van SDM's word dikwels as 'n eenmalige gebeurtenis beskou, terwyl dit in werklikheid 'n deurlopende proses is. Navorsing gebruik die gebeurlikheids teorie asook die teorie van innovasie aanvaarding ("*innovation adoption*") om 'n konseptuele model van gebeurlikheids gebruik van SDM's te ontwikkel. Die konseptuele model is getoets aan die hand van opname-gegewens (data) wat ingesamel is vanaf stelsel-ontwikkelingspraktisyns. Statistiese ontleding is uitgevoer en dit het gelei tot die ontwikkeling van 'n verfynde konseptuele model van gebeurlikheids gebruik van SDM's. Die studie kom tot die gevolgtrekking dat die twee SDM-klasse as uiterste teenoorgesteldes beskou word, maar dat dit in die praktyk aanvullend is tot mekaar. SDM's word gekombineer om hibriede-SDM's te skep, of 'n vervlegging te maak in dieselfde ontwikkelingsprojek. Literatuur en statistiese bevindings is gebruik om riglyne te ontwikkel vir die gebeurlikheids gebruik van SDM's.

Sleutelwoorde: "*agile*" SDM, konseptuele modelle, riglyne, hibriede-SDM's, plangedrewe-SDM's, SDM's

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AHP	Analytical Hierarchy Process
CMMI	Capability Maturity Model Integration
CIPC	Companies and Intellectual Property Commission
CVM	Competing Values Model
CUOSDM	Contingent Use of SDMs Model
DOI	Diffusion of Innovation
DTI	Department of Trade and Industry
ICT	Information Communication Technology
IT	Information Technology
IS	Information Systems
ISO	International Organization for Standardization
JCSE	Joburg Centre for Software for Engineering
KMO	Kaiser-Meyer-Olkin Measure
MICT SETA	Media, Information and Communication Technologies Sector Education and Training Authority
OMG	Object Management Group
PEOU	Perceived Ease of Use
PU	Perceived Usefulness
PMI	Project Management Institute
PCA	Principal Component Analysis
SEI	Software Engineering Institute
SIC	Standard Industrial Classification
SDLC	Systems Development Life Cycle
SDM	Systems Development Methodologies
TTF	Task Technology Fit
TAM	Technology Acceptance Model

CHAPTER ONE: INTRODUCTION AND OVERVIEW

1.1 INTRODUCTION

A brief overview of contingent use of systems development methodologies and the problem statement for the study are presented in this chapter. The research aim and objectives are formulated and the significance of the study is explained. The research methodology and research methods used in the study are briefly outlined and finally, the thesis structure and organisation is presented.

1.2 CONTEXT OF CURRENT RESEARCH

South Africa presents a wide variety of organisations that are involved in systems development. These organisations range from small to large. Some have systems development as core business and others as support service to their core products and services. The skills levels and role assignment also differ. Bigger organisations have highly specialised personnel, whereas smaller ones have limited skills base. In smaller organisations, there is a tendency of combining several roles to one individual. Each organisation irrespective of its core competences, skills base and role assignment, strives for success. Systems Development Methodologies (hereafter referred to as SDMs) gives the necessary support structure for systems development (Leau *et al*., 2012; Carroll, 2003; Fitzgerald, 1998) to achieve success.

The question is how to adopt a particular SDM and how it might be usefully tailored or combined with other systems development methodologies, to obtain the best fit with the systems development project contextual factors (Henderson-Sellers *et al*., 2014). If an organisation can manage to adopt an SDM or systems development methodologies that fit with the systems development contextual factors, failure risks would be reduced (Gill *et al*., 2018; Imani *et al*., 2017). The term adopt in this case refers to a consistent commitment of SDM use. The main idea in the contingent use of systems development methodologies, is to achieve a continuous fit of an SDM or systems development methodologies with the systems development project contextual factors.

There is no universally accepted documented guide on how to adopt an SDM from a myriad of systems development methodologies that exist (Ramsin and Paige, 2008). The systems development crisis seems to take a different form each time, for instance, initially the crisis was perceived as predominantly development process management (Avison and Fitzgerald, 2003). Later, it cascaded to user-relation dynamics (Checkland, 1981). Once the user participation was under control, it became associated with issues related to market window and change management (Chan and Thong, 2009). The polymorphic property of the systems development crisis was perhaps the reason for SDMs' explosion. With the explosion of a variety of SDMs, practitioners were confronted with an SDM adoption dilemma. Adopting an SDM from many available options was demanding and confusing, because the adoption criteria or guidelines was neither clear nor justified (Yusof *et al*., 2011; Carroll, 2003). Naumann and Palvia (1982) stated that adopting an SDM from the numerous existing SDM classes was a challenge with technological, social and financial consequences. Not only was there difficulty on the adoption among SDM classes, but also on the instances of SDM classes. Iivari *et al*. (2001) presented an SDMs classification to demystify the tenet of "methodology jungle" identified by Avison and Fitzgerald (2006). However, the suitable SDM search space is a nondeterministic polynomial hard problem. Despite the complexity of adopting a suitable SDM, it is hoped that an appropriate SDM should standardise and formalise the development process, organise work and resources and direct appropriately the perception of each member of the development team (Carroll, 2003; Fitzgerald, 1998). An SDM provides the process structures that align the development project with the organisational objectives (Young *et al.,* 2016). Therefore, the adoption of a suitable SDM is imperative.

1.2.1 SDM Adoption frameworks

There are several studies on the SDM adoption process and there is a growing literature base on the development of SDM adoption frameworks and models. However, there is little research on evaluating these in practical situations. Yaghini *et al*. (2009) proposed an SDM adoption framework based on a multi-faceted approach. The SDMs were first classified as hard or soft and then compared according to six basic features; the philosophy, systems development model, systems development scope, systems development tools, systems development background and participants. This model had limitations since it compared Soft Systems Methodology (Checkland, 1981), Structured Systems Analysis and Design Methodology (SSADM) which are SDMs instances grounded on different paradigms (Iivari *et al*., 2001). Therefore, they are not comparable due to their different philosophical underpinnings and assumptions. The criteria for determining the scope of each systems development phase was not precisely and explicitly stated in this model. Scope problems are inherent in an SDM as one of the dimensions of inconsistency (Iivari and Maansaari, 1998). Therefore, the adoption framework might suffer from objectively scoping the phases. This model introduced only a set of six SDMs and it would be challenging to include any new SDM.

Naumann and Palvia (1982) presented an SDM adoption model centred on quantitative scoring method called Delphi to collaboratively evaluate and recommend essential SDM functional characteristics. The candidate SDM was adopted based on the scores awarded to it. The drawback of this model was the subjectivity of SDM functional attributes definition and the concentration on the system development techniques and neglecting the other components of an SDM.

Cockburn (2002) put forward a decision model based on evaluating appropriateness of each member of the Crystal SDM family instances to a target systems development problem domain. The challenge of this model was that it was restricted to limited SDM instances. Rashmi and Anithashree (2009) proposed an SDM adoption framework for Rapid System Development (RSD) methodologies built on a comparative analysis of a set of essential aspects of rapid development methodology instances under consideration. However, this adoption framework was limited to Rapid System Development (RSD) methodology family.

Yusof *et al*. (2011) presented a hybrid approach to adopting an SDM based on complexity and uncertainty, quality criteria and scope of SDM phases as key factors. The researchers selected eight SDMs and stated that they are the most common ones and gave a formula for calculating the score for each SDM. The drawback of this model was that it was derived deductively from theory and validated on single systems development project in a particular organisation. As a case study, it is rich in details of a particular systems development of contextual settings, but it does not provide adequate empirical evidence to justify the benefits or to generalise the usefulness of the model.

The perspective provided by the SDM adoption frameworks gives an impression that adopting an SDM is a once-off event. Some important factors of the SDM adoption in systems development are described in the following section.

1.2.2 SDM Adoption factors

The experience, knowledge and expertise of the systems developer has significant influence in the adoption of SDMs (Yusof *et al*., 2011). These factors create a level of developer's confidence (or lack thereof) to adopt a specific SDM or SDMs. Naumann and Palvia (1982) discovered that adoption was biased towards experience and familiarity with the SDM. Experience has been consistent as an influential factor in SDM adoption (Marks *et al.,* 2017; Berente *et al*., 2015; Aitken and Ilango, 2013). However, Fitzgerald (1998) reported that Nijssen's Information Analysis Method (NIAM) was mandatory to systems development projects in the Netherlands government departments, whereas Merise is required by the French government departments. The V-Modell XT is required in German government organisations (Kuhrmann *et al*., 2011). The Structured Systems Analysis and Design (SSADM) was once mandatory not only in the United Kingdom government departments but also in Ireland, Malta, Hong Kong and Israel (Fitzgerald, 1998). The public sector, in most cases, required a high level of formalisation, control of the systems development process so as to reduce risk and uncertainty. Therefore, it is no surprise to find that government departments adopted a tried and tested SDM or SDMs at the outset. The predetermination of an SDM did not consider systems development project contextual factors as the organisation mandated the adoption of an SDM or SDMs. In most cases, organisational mandate is based on the assumption that an SDM can address all the requirements of any systems development project, irrespective of the systems development project contextual particularities (Fitzgerald *et al*., 2000). Henderson-Sellers *et al*. (2014), Ramsin and Paige (2008) and Fitzgerald (1998) are all against the assumption of applicability of an SDM across all systems development project context situations. Similarly, Rashmi and Anithashree (2009) state that SDMs should be chosen according to their suitability to a specific systems development project context. This was also in support of the view that an SDM is based on particular philosophical assumptions and its use on a systems development problem context situation should match the assumptions made or intended usage context (Mingers and Brocklesby, 1997). Avison and Fitzgerald (2006) indicate that the systems development project context factors are the rationale behind the adoption of an SDM. Therefore, the systems development contextual factors are imperative when adopting an SDM. Each SDM should be based on the assumptions about the systems development project contextual factors (Berente *et al*., 2015; Aitken and Ilango, 2013). Specific systems development contextual factors determine anew an SDM appropriate for the systems development project.

Burns and Dennis (1985) advocated a two-dimensional framework for adopting the most suitable SDM. These are the contingency approach classified systems development projects in terms of systems development project complexity and systems development uncertainty factors (Burns and Dennis, 1985). The systems development project complexity was determined through four aspects; the project size, the number of system users, the quantity of new generated information, and the complexity of generating new information (Burns and Dennis, 1985). By contrast, the systems development project uncertainty consisted of three characteristics; the level of structure, the extent of users' knowledge on their duties and system developer's experience and expertise. The SDM adoption process in this strategy involved a straightforward reading of a two-dimensional array contents based on the level of complexity and uncertainty of the systems development project. However, the drawback of this approach was that it considered only two methodology instances. A variety of systems development methodologies are available (Mirza and Datta, 2019). No single SDM can probably meet the demands and requirements of all systems development situations (Marks *et al*., 2017; Berente *et al*., 2015; Aitken and Ilango, 2013; Huisman, 2013; Clarke and O'Connor, 2012; Ramsin and Paige, 2008).

Burns and Dennis (1985) acknowledged the availability of many systems development methodologies. They also stated that each of these are best suited to specific types of systems development projects based on the level of complexity and uncertainty. However, their proposed methodology selection approach provided no explicit algorithm to expand the proposed contingency framework. The quantitative determination of systems development project complexity and uncertainty was not explicit, which made the extension of the selection framework challenging.

Systems development project contextual factors are different or can present different combinations that make even the same systems development project different over time. Some systems development project contexts have more well-understood systems development contextual settings than others. These different systems development circumstances demand different SDMs if predictable results are to be obtained. Studies reveal that each SDM addresses a conceived concern in the domain of systems development. Even in single organisational settings, the concerns may differ from one systems development project to another. Carroll (2003) found, that contingency factors affected the adoption of SDMs throughout the development process. The contingent factors approach suggests that each development situation demand an appropriately adopted SDM from a portfolio of SDMs. However, the challenge is that there is no repository for comparing all the SDMs. Thus, they cannot be classified and analysed on their normative principles, strengths, weaknesses and contextual appropriateness.

The SDM engineering went a step further and suggested the adoption of SDM fragments from a repository and constructed an appropriate framework or adapt, configure or tailor systems development methodologies to fit the specific systems development project situations (Henderson-Sellers *et al*., 2014). This is only a component of the contingent use of systems development methodologies. However, experience and a high degree of expertise are necessary to apply the SDM engineering approach.

The derivations of these adoption frameworks were more biased on theoretical deductions than empirical evidence and thus became mere pieces of advice on what was to be done at that particular time (Siau and Rossi, 2011). The adoption frameworks follow a linear view in the implementation of SDMs and this does not capture how SDMs are implemented (Fitzgerald, 1994). The existence of different SDMs reflects the dynamic nature of systems development. The linear view considers the selection of an SDM as a once-off deliberate event instead of being a continuous process of alignment of the selected SDM with the changing systems development project contextual factors. The SDMs are seldom implemented as intended by their authors (Henderson-Sellers *et al*., 2014). The implementation of an SDM is more likely to be contingent to the systems development project contextual settings than being a-priori fixed, deliberate and detailed (Henderson-Sellers *et al*., 2014). This implies that it is not only the systems development project context that is changing, but also the SDM itself should be flexible enough to be adapted, modified or combined with another to form a hybrid throughout the development process (Conboy and Fitzgerald, 2010; Carroll, 2003). In the following section, the problem statement is presented and discussed.

1.3 PROBLEM STATEMENT

Sheffield and Lemétayer (2013) and Iivari and Huisman (2007) indicate that the use of SDMs is contingent to the systems development project contextual factors and that SDMs are not necessarily fixed before the assessment of the systems development project contextual factors. Organisations should adopt, adapt, modify and tailor an SDM to fit systems development project contextual factors (Iivari and Huisman, 2007). The effort to achieve the ideal fit of the SDM within the scope of the system development project contextual factors is referred to as the contingent use of SDMs and is explained in detail in Chapter 3 of this study.

Contingent use of SDMs has recently attracted a great deal of attention, as it is a departure from the prescribing and fixing of SDMs for a system development project. Although numerous studies are available that describe the use of SDMs in systems development projects within organisations, fewer studies describe the contingent use of SDMs across organisations in South Africa. Therefore, this study investigated the state of contingent use of SDMs in systems development organizations in South Africa.

Systems development methodologies have faced challenges in addressing systems development project contextual factors since the late 1960s (Mahanti *et al*., 2012; Jiang and Eberlein, 2008). To mitigate the challenges posed by the systems development project contextual factors, numerous SDMs have been proposed (Gill *et al*., 2018; Boehm, 2006). The adoption of an SDM may be conceptualised as a dichotomous variable, that is, it can either be adopted or rejected (not adopted) (Fitzgerald, 1997). The adopted SDM may be categorised as plan-driven SDM class instance or agile SDM class instance. While there is no universal SDM (Henderson-Sellers *et al*., 2014; Huisman, 2013; Börner, 2011; Burns and Deek, 2011) that is appropriate in all system development project situations, some practitioners and researchers advocate a single SDM that is fixed for all systems development projects (Jiang and Eberlein, 2008). In addition, to a fixed SDM, it should also exclusively belong to one of the two SDM classes (Jiang and Eberlein, 2008). The advocates of one SDM class considers the extreme negative aspects of the other SDM class on specific systems development project context situation. In an experiment, however, Jørgensen (2013) shows that some systems development professionals tend to be biased and selectively find data to justify the use of an SDM that they believe to be the best regardless of the systems development project context

Claiming that one SDM class is superior from the other in all systems development project situations has been rejected as not objective and counterproductive (Jiang and Eberlein, 2008). Sheffield and Lemétayer (2013) and Iivari and Huisman (2007) indicate that the adoption of SDMs is contingent to the systems development project contextual factors. This means that an SDM can be tailored, tweaked, combined with another SDM or adapted to the requirements of the system development project contextual factors regardless of the SDM class. Empirical research is required to gain insight into the status quo of the contingent use of SDMs in organizations and to propose guidelines to assist system development practitioners.

Contingent use of SDMs research is relevant to organizations aiming at achieving the ideal fit of SDMs for specific systems development projects in specific organizations, with specific organizational cultures (Fitzgerald *et al*., 2002). The adoption of an appropriate SDM is an integral part of contingent use of SDMs intended to reduce the likelihood of systems development project failure. It also increases systems development process efficiency, improve quality of developed systems, and deliver systems on schedule and within budgetary constraints (Standish Group, 2016; Johnson and Mulder, 2016). Therefore, an appropriate SDM creates the mental models that facilitate conceptual framing that harmonise interpretation of information and clear understanding of interdependencies in a systems development problem space (Huisman, 2013).

Organisations are aware of the systems development crisis and the implications of systems development project failure on reputation, employee morale, costs and business continuity. This is one of the reasons for the adoption and continued use of one proven, tested and familiar SDM since this avoids the uncertainty associated with a new SDM. However, adopting an SDM that is proven is a necessary condition, but not sufficient as adoption does not guarantee success. An SDM might have been proven in a different set of systems development project contextual factors, but understanding when and how to adopt, adapt, and tailor an SDM appropriately or create an alternative SDM that is fit for purpose, is crucial for organisations involved in systems development.

This study shows that systems development organisations still face challenges in adopting, adapting and creating hybrid SDMs that are ideally suited for each system development project contextual factors.

1.4 RESEARCH QUESTION

This study’s research question is:
Can the contingent use of SDMs conceptual model be developed to investigate the contingent use of SDMs and guidelines proposed for the contingent use of SDMs to assist systems development organisations in South Africa?

1.5 RESEARCH AIM

The aim of the study is to develop a contingent use of SDMs conceptual model to investigate the contingent use of SDMs and propose guidelines to assist system development organisations with the contingent use of SDMs in South Africa.

1.6 RESEARCH OBJECTIVES

To achieve the research aim, the following objectives were formulated:

Research objective 1: Examine systems development methodologies in use within the systems development organisations in South Africa.
Research objective 2: Identify the contingent use of systems development methodologies in organisations in South Africa.
Research objective 3: Identify the critical success factors of the contingent use of systems development methodologies.
Research objective 4: Develop a contingent use of SDMs conceptual model and the guidelines for the contingent use of SDMs.

1.7 ETHICS CONSIDERATIONS

The research proposal for this study was approved as a **No Risk** project by the Faculty of Natural and Agricultural Sciences Ethics Committee of the North-West University of South Africa (see Appendix A). A letter seeking for the informed consent of respondents was also prepared (see Appendix B). The code of conduct for the NWU researchers was also signed (see Appendix C).

1.8 RESEARCH DESIGN AND METHODOLOGY

As research methodology, the study adopted a positivist philosophical stance. The positivist methodology strives to infer explanations from objective and scientific measures (Neuman, 2011). As a first methodological step, relevant literature was thoroughly reviewed. Knowledge gaps were identified in contingent use of SDMs in South Africa. The detailed literature review served as a theory building platform that established the context of the problem and the extent to which other researchers have contributed towards addressing it. More importantly, the thorough review of pertinent literature led to the development of a conceptual model. Constructs and their relationships were identified as well as the hypothesis formulations. A survey data generation instrument was developed to achieve the generalisability of results. The data generation instrument was in the form of a questionnaire. Each organization was asked to participate in completing the questionnaire. After agreeing to participate, a single systems development practitioner was identified for participation in the completion of the questionnaire. The researcher collected the completed questionnaire from the participants. The sample frame was purposeful so that it adequately included the subject under study and adequately represented the population of interest. This way, the sample included leading industrial players in systems development in South Africa. The organisations were identified from the Bizcommunity, ITWeb directory of companies and Rainbow nation websites.

The collected data was analysed through descriptive statistics, t-tests, X^2 tests, factor analysis, correlation and regression analysis. A detailed explanation on the research methodology is in Chapter 4.

1.9 THE RESEARCH RATIONALE AND SCOPE

Systems development projects are critical for most modern organisations. However, systems development projects have consistently shown low success rates (Gupta *et al*., 2019; Standish Group, 2016). The Chaos Report (Standish Group, 2016) found that 71% of all systems development projects are unsuccessful. Khoza and Marnewick (2020) indicate that in South Africa, 36% of the systems development projects fail. The adoption of an inappropriate SDM is usually the cause of the failure of systems development projects (Gupta *et al*., 2019; Sheffield and Lemétaye, 2013). Organisations should adopt SDMs contingently over the systems development project lifecycle, that is, systems development methodologies that match systems development circumstances at any point in time as the systems development project progresses. Unfortunately, not much research has been done to guide organisations in this regard.

The research was conducted within the South African systems development industry. The study focused on systems development organisations from all the nine provinces of South Africa and systems development practitioners were selected as respondents. Chapter 4 provides details in this regard.

1.10 THESIS ORGANIZATION

The thesis is organised into seven chapters as shown Figure 1-1. In Chapter 1, an overview of contingent use of systems development methodologies (SDMs) background is explained. The chapter presents and discusses the problem statement, significance of the research, aim and research objectives. The research methodology for the study is also briefly described. The literature survey is presented in Chapter 2. The context in which this research is undertaken is described and a detailed review of the historical perspective of systems development methodologies (SDMs) is also presented. The gaps in literature are identified. To address the gaps identified, a conceptual model is developed in Chapter 3. The theoretical basis for the contingent use of systems development conceptual model is presented. The diffusion of innovation model (DOI) is justified as the guiding theory for the development of a hybrid model for the contingent use of systems development methodologies. The development of the conceptual model of the study based on in-depth review of the pertinent literature on the contingent use of systems development methodologies is presented.

A detailed outline and justification of the research methodology followed by the research design adopted are presented in Chapter 4. The process of developing a survey data generating instrument for the study is also described in detail in Chapter 4. Furthermore, the physical implementation of the survey instruments is reported and the sample frame explained in Chapter 4. The descriptive data analysis of the collected data is presented in Chapter 5. The results of inferential statistics, discussion and interpretation of the data analysis are presented in Chapter 6. Findings in relation to the previous literature, conclusions, contributions, limitations, implications of the research and suggestions for further research in contingent use of systems development methodologies are presented in Chapter 7. The work from a generalised viewpoint and the extent to which the objectives were fulfilled are also outlined in Chapter 7.

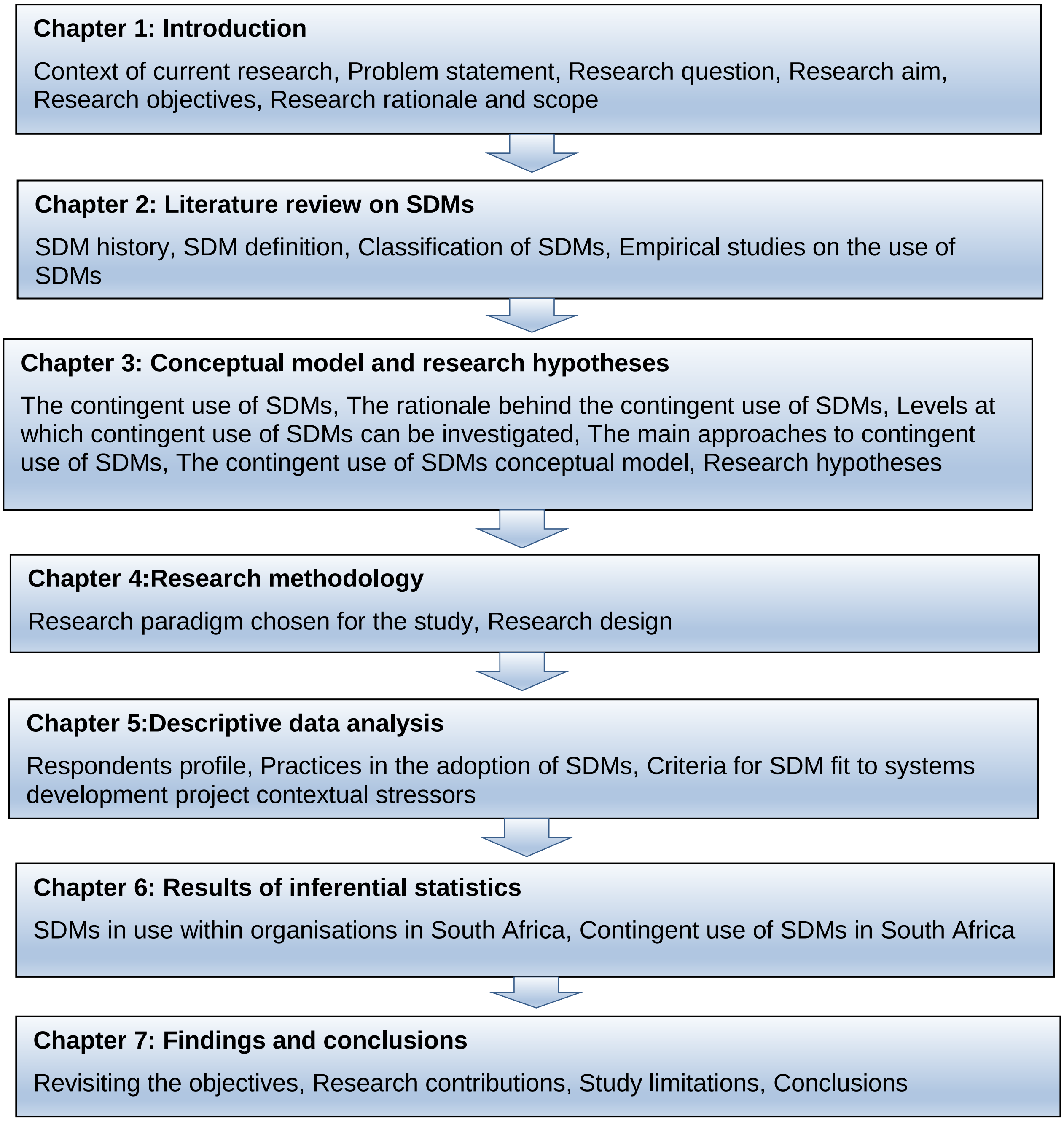


Figure 1-1: Thesis organisation

1.11 CHAPTER SUMMARY

This chapter presented an overview of the characteristics of the systems development industry in South Africa. Different categories of organisations and the skills base were described to show the context the study. The population and the sample frame was briefly described and the data generation instrument briefly outlined. A brief outline of the evolution of systems development methodologies was presented. A preliminary definition of contingent use of systems development methodologies was given to indicate the meaning adopted by the study. SDM adoption was used as surrogate for contingent use of SDMs. The information gap on the guidelines towards SDM adoption was briefly explained. The chapter identified three levels of SDMs fit which are: the systems level, the information systems development level and the developer level. The research aim and objectives were also formulated and presented. The motivation of the study was outlined, problem statement discussed and the approach and the research methodology used in the study were also explained. The layout of the chapters was presented to show the organisation of the thesis. The next chapter reviews SDMs.

CHAPTER TWO: LITERATURE REVIEW ON SDMS

2.1 INTRODUCTION

As indicated in the research rationale in Chapter 1, systems development projects are critical for most modern organisations. However, systems development projects have consistently shown low success rates (Gupta *et al*., 2019; Standish Group, 2016). The Chaos Report (Standish Group, 2016) found that 71% of all systems development projects are unsuccessful. Khoza and Marnewick (2020) report that in the context of the South African systems development industry, 36% of systems development projects fail. The adoption of an inappropriate SDM is usually the cause of the failure of systems development projects (Gupta *et al*., 2019; Sheffield and Lemétaye, 2013). There are many systems development methodologies that exist (Huisman and Iivari, 2006; Wynekoop and Russo, 1995) and researchers estimate that there are well above 1000 (Conger, 2013; Avison and Fitzgerald, 2006; Huisman and Iivari, 2006; Iivari *et al*., 2001; Iivari and Maansaari, 1998). Most research explores the development of new systems development methodologies, frameworks and models for selection, classification, understanding, applicability, usage and adoption (Henderson-Sellers and Ralyté, 2010; Avison and Fitzgerald, 2003; Iivari *et al*., 2001; Fitzgerald, 1998; Iivari and Maansaari, 1998; Wynekoop and Russo, 1995; Naumann and Palvia, 1982). The continuous explosion of systems development methodologies suggests that the software crisis is not yet over and that it is an active research area. Therefore, the following questions arise about the systems development methodology and the role it should play in systems development: Where did SDMs come from? How and why are SDMs evolving? A brief history of SDMs is outlined in the next section.

2.2 SDM HISTORY

A history of systems development methodologies provides context for this study. The history of systems development methodologies is not trivial at all. The concomitant availability of different instances and unavailability of generalised evidence on adoption and usage at any particular time compound the challenge in outlining a clear history. There is also no clear understanding of the subject under study (Wynekoop and Russo, 1995) and there is no agreed definition for SDM (Iivari and Maansaari, 1998; Huisman and Iivari, 2006) despite the Britain Computer Society's (Avison and Fitzgerald, 2006) attempt to define the concept. The history exposes the evolving understanding of the interaction between the systems development factors and the systems development methodologies. The next subsections contextualise each systems development methodology and highlight some important developments in this domain.

2.2.1 SDM ERAS

The systems development methodologies have evolved over the years. The evolution of SDMS is a continuing evolutionary process rather than a revolutionary process. This represents efforts to deal with certain limitations identified in systems development in a specific period. Avison and Fitzgerald (2006) examined the evolution of systems development methodologies and identified four eras, namely the pre-methodology, early-methodology, methodology and the post-methodology or reassessment. This put systems development methodologies into the relative time intervals of before, past, present and future. A timeline of the history of SDM is presented in Figure 2-1.

Pre-methodology	**Early-methodology**	**Methodology**	**Post-methodology**
1960 - 1970	1970 -1980	1980 - 1999	2000 and beyond
Philosophical Assumptions -Technical perspective ***Main characteristics*** -Informal -Ad hoc -Expert based ***No example***	*Philosophical Assumptions* -Technical perspective ***Main characteristics:*** -Structured -Repeatability -Phased and rigid ***Examples:*** Waterfall (Royce, 1970) Merise (Quang and Chartier-Kastler, 1991)	*Philosophical Assumptions* -Socio-technical perspective -Technical and rational perspective ***Main characteristics:*** -User participation -Structured and phased -Repeatability -Development process standardisation ***Examples:*** Structured Systems Analysis and Design Methodology (SSADM) (Eva, 1994) Rapid Application Development (RAD) (Martin, 1991) Effective Technical and Human Implementation of Computer-based Systems (ETHICS) (Mumford, 1983) Jackson System Development (JSD) (Jackson, 1983)	*Philosophical Assumptions* -Contextual stressors dynamics -SDM components and practices ***Main characteristics*** -Agility -Contingent, adaptive and iterative ***Examples:*** Rational Unified Process (RUP) (Kruchten, 2000) Scrum (Schwaber and Beedle, 2002)

Figure 2-1: Timeline of the history of systems development methodologies (Own construction)

2.2.1.1 The pre-methodology era

The pre-methodology era covers the 1960s. The systems development during this era was ad hoc, unstructured and informal in nature (Berente *et al*., 2015; Aitken and Ilango, 2013; Walters *et al*., 1994). Systems development was approached from a build and fix perspective. There were no clear factors that could be used to classify these systems development processes in a meaningful way to warrant the concept of an SDM. Systems were considered in a narrow sense as consisting of hardware and software and emphasis was on the hardware component. Developers were viewed as engineers and artists and attention was on technology. Systems development had a very strong technical orientation, and was considered a highly technical activity that depended on the experience and expertise of the developer (Avison and Fitzgerald, 2003; 2006). Moreover, systems development relied on individual systems development practitioner intuition, trial and error without much theoretical underpinning (Aitken and Ilango, 2013). It was assumed the developer understood and knew the user requirements without the user's involvement (Mahanti *et al*., 2012). However, requirements are key to user acceptance and development of an appropriate solution to any problem.

Development was based on the rule-of-thumb where the developer was expected to produce a running system and not necessarily a working system. This meant that most systems developed during this era did not meet user expectations, as there was no correspondence between the user requirements and the developed systems functionality. There were no notable mechanisms in place to manage the systems development project. The unpredictable and dynamic nature of systems development (Brooks, 1987) lead to failure of most systems development projects in terms of the Standish Group's (1995) iron triangle criteria of success. The systems development projects exceeded schedule, budget and were delivered without all the functionality. Some systems development projects were even abandoned. Progress on systems development could not be measured as there were no mechanisms to manage the systems development project. Gradually, the systems development community started considering some ways to introduce standards and discipline in the development of systems.

2.2.1.1.1 Strengths associated with SDMs in the pre-methodology era

The systems development methodologies of this era had the following benefits:

- Technically trained developers
- No bureaucratic processes involved as the development was dependant on experts.

2.2.1.1.2 Weaknesses of the SDMs in the pre-methodology era

During the pre-methodology era, systems development suffered from poor control and management of systems development projects. Developers were usually overworked and spent a large proportion of their time debugging and enhancing the systems. The system development process lacked a disciplined approach since standards had not evolved.

2.2.1.2 The early-methodology era

The period covers the 1970s, and systematic and disciplined approaches to systems development process were introduced. The systems development community started looking for solutions in other disciplines, like mechanical engineering, to introduce discipline in systems development (Fowler, 2005). It was assumed that a systems development process is rational, phased and sequential as a mechanical engineering process. Therefore, the development of systems involved an orderly sequence of transitions from one phase to the next in a sequential order as in an engineering production process (Fowler, 2005; Royce, 1970). The ground breaking work of Royce (1970) outlined the systems development life cycle (SDLC) as a conceptual and rational schema for managing and controlling systems development projects. It introduced discipline and provided guidelines to organize, plan, prioritize, budget, schedule, resource and manage and control systems development projects. The SDLC was considered a *de jure* SDM appropriate to deal with all systems development situations (Aitken and Ilango, 2013). It viewed systems development to be a logically ordered process that resembled engineering assemble plant process. The SDLC proposed by Royce (1970) became known as the waterfall model and it became the backbone of the waterfall methodology. The guiding principle of the waterfall methodology was in the division of the systems development process into separate but sequential phases. Its structure involved cascading phases whereby output from one phase acted as input to the subsequent phase as in an engineering assemble line. For instance, to proceed from systems feasibility study phase to requirements analysis and specification phase, a feasibility report had to be verified, validated and signed off first. Each phase had a clear start and a clear end. The deliverables at each phase were known, therefore, cost estimation and cost escalation avoidance were improved.

The strength of the waterfall methodology was in dividing the systems development project into manageable subprojects in the form of phases. At phase level, deliverables would be evaluated, validated and verified. This allowed division of labour, task specialisation by phase and enforcing of standards. Apart from enforcing discipline in the systems development process, the waterfall methodology enabled the development team to gather systems requirements at the outset and produce a detailed documentation. It did not rely on specific experts as it allowed skills mobility. Any SDM that invests more time and effort in the initial stages of the systems development project is viewed as front loaded.

The waterfall methodology is characterized by thorough planning and preparation at the beginning of the development process. The waterfall methodology relies on a detailed and massive up-front requirements analysis, design, documentation, and sequential implementation of predefined plans to manage the systems development process (Somerville, 2016; Meso and Jain, 2006). The waterfall methodology is called a front loaded or plan driven, because it involves thorough planning and preparation in the initial stages of systems development (Munassar and Govardhan, 2010). This addresses the real problem as the development starts after a thorough investigation and understanding of the systems development problem. Documentation is one of the important indicators of progress in this approach. Each phase should be completed and signed off as approved before starting the next phase (Munassar and Govardhan, 2010). When a phase was signed off it would have met all the required quality standards. Therefore, there was no room for feedback for the phase after sign off until the systems development project was completed, which increased the risk of errors being propagated to the next phase (Dima and Maassen, 2018). The iteration between phases are indicated using red arrows as shown in Figure 2-2. The system had to be deployed first before revisiting any of the previous phases (Somerville, 2016). Systems development decisions related to budget, schedule and functionality are made upfront and strictly adhered to throughout the process. The customer interaction takes place primarily at the beginning of the systems development project when the decisions are made, and at the end when the product is delivered.

The waterfall methodology has served and is still serving as a framework in the development of systems and creation of other SDMs. It is a de facto standard against which most systems development are compared (Mitchell and Seaman, 2009). Therefore, the waterfall methodology, will be discussed in more detail. The waterfall methodology consists of six phases; the feasibility study, requirements analysis and specification, design and specification, coding and module testing, integration and system testing, and delivery and maintenance. The typical phases of the waterfall SDM are shown in Figure 2-2 and described in Table 2-1.

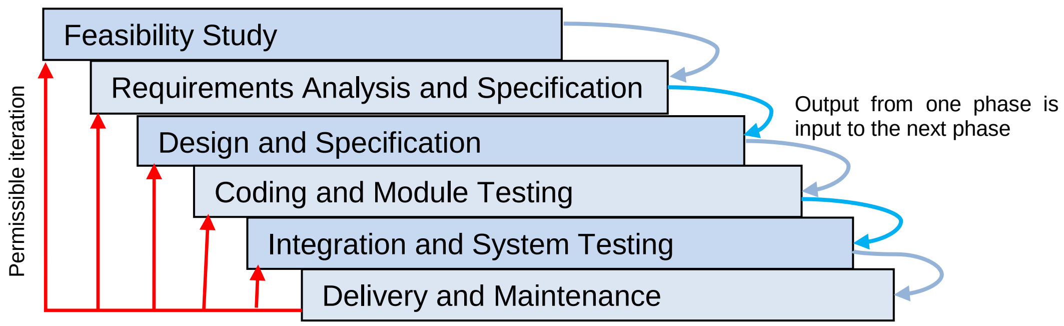


Figure 2-2: The Waterfall SDM (Royce, 1970)

The typical phases in a waterfall SDM are described in Table 2-1.

Table 2-1: Description of the waterfall SDM phases

PHASE	DESCRIPTION
Feasibility Study	This phase performs a preliminary investigation to evaluate the practicality of developing a system. The feasibility study is a critical step because the outcome will affect the entire development process. The feasibility study reviews anticipated costs and benefits, and recommends a course of action based on organisational, political, ethical, legal, operational, technical, economic and schedule factors (Avison and Fitzgerald, 2006).
Requirement analysis and specification	The requirements and analysis phase answers the questions of who will use the system, what the system will do, and where and when it will be used. Requirements are gathered to get a detailed understanding of the problem before the systems development begins. The functional and non-functional requirements of the system are identified (Somerville, 2016; Schach, 2010). The gathered requirements are selected, prioritised and specified. The specification is reviewed and signed off by the client. After signing off requirement specification, further changes to requirements are limited (Somerville, 2016; Fowler, 2005).
Design and specification	The requirement specifications are studied in this phase and the system design prepared. The phase addresses the "how" part of the system. The overall system architecture is designed. Several design activities take place in this phase: • Development of logical model of the proposed system, including process logic definition, logical data dictionary, and logical database design, testing design, conversion design, training design, evaluation design • Design of program specifications • Development of an implementation and test schedule • Designing Codification Schemes • Detailed manual procedures • Documenting the Design specification document that is signed off by the client.
Coding and module testing	The design specification is coded into system functionality features referred to as modules. These are individually tested, validated and verified (Somerville, 2016). Dijkstra (1970) states: "Program testing can be used to show the presence of bugs, but never to show their absence!". The test is in accordance with the specification and not that the module is free of bugs (Dijkstra, 1970).
Integration and system testing	All the modules developed during coding and module testing phase are integrated into a single system and tested. The system testing checks for any flaw or errors in relation to the specified functional and non-functional requirements. It is impossible to test for all possible situations or errors that could possibly occur (Dijkstra, 1970).
Delivery and maintenance	Once the system has been integrated and tested, it is deployed in the client's environment or released into the market if the development was market driven. Testing mimics what the user would do when the system is deployed. To fix those issues that come up in the client environment or market, debugging may be done or enhancement of functionality carried out or lead to the release of better versions. In the case of market driven systems, patches are released.

Each SDM tries to address some or all phases of the SDLC or waterfall methodology. Several versions of the SDLC or waterfall methodology have emerged (Somerville, 2016). The SDLC or waterfall methodology in its original version is a rigid plan-driven SDM that assumes the world as deterministic and fully observable (Leau *et al.,* 2012). The assumptions for this line of thought was that once the development starts, what was important was to keep on working on the originally prescribed plan until the end. Therefore, the systems development process was divided into sequential phases with clear activities and deliverables. It followed a "plan the work and work the plan" perspective of systems development. Once systems development started, there was only one expectation and that was a monolithic release of the developed system to the client (Boehm, 2006). The following subsection outlines the benefits of the SDLC.

2.2.1.2.1 Strengths associated with SDMs in the early-methodology era

Mirza Datta (2019) and Somerville (2016) identified several strengths associated with SDMs in this era. Some of these are:

- SDMs acted as knowledge generators and organisational memory;
- Division of the systems development project into manageable subprojects;
- Provided discipline to the systems development process;
- Easy to manage due to their rigidity as each phase has specific deliverables and a review process;
- Provided techniques for task arrangement and division of labour;
- Progress became measurable as conformance to plan was used as an indicator to avoid cost escalation.

2.2.1.2.2 Weaknesses of the SDMs in the early-methodology era

The benefits of the SDMs in this era are not short of criticisms. First, systems development is not a logical process therefore it cannot be directly inspired by engineering processes. The engineering artefacts are visible and reliable progress measurement models have been developed unlike the software systems (Fowler, 2005). Progress and success are measured by compliance to a plan which is not directly related to the development of a high-quality working system. Second, the SDMs were mostly bureaucratic and this slowed down the actual development of systems. The SDMs of this era assumed that the transformation from the problem space into the solution space was consistent with the development team's experience on similar problems. Each development situation is unique, but the early methodology SDMs did not recognize that the real world is dynamic and unpredictable (Avison and Fitzgerald, 2006). The documentation was heavy and technically oriented and was rarely updated (Avison and Fitzgerald, 2006). Flexibility rather than rigidity is required since systems development process is not a logical process but a changing process. The change may be in the form of requirements, methodology, technology or people (Williams and Cockburn, 2003). Therefore, change is an emergent inevitable phenomenon in systems development. The development of a system is a transformation or a change from the problem space to a solution space. Unfortunately, the early-methodology SDMs failed to anticipate and accommodate change. This led to the development of a solution of a non-existing problem. Some interdependencies and user requirements cannot be known from the outset and the early-methodology SDMs were unable to deal with this problem. User participation as well as interdependencies within systems development were neglected. The assumption was that the systems solution domain existed within technology irrespective of contextual factors. The systems developers were assumed to have the skills and knowledge to generate all the necessary requirements that users would need. Users were only consulted at the initial stages of development and at the end when the system was being deployed (Somerville, 2016). This led to incongruence in technological frames (Huisman and Iivari, 2006) due to lack of collaboration between the developer and the user. The problems were discovered late during testing by developers and when the system was already being deploy for use. Ignoring requirement interdependencies and users and waiting until the coding stage to test the systems led to developing a right system for an old problem. The technical view and high level of rigidity of the SDM led to failure in systems development. Failure meant that the system never worked or never met its schedule and budget, never adequately addressed user requirements, was full of bugs, did not work as per specifications, or was completely abandoned.

These challenges led to a rethinking and reconceptualization of what systems development methodologies should be. Developers realised that the early-methodology SDMs werte not realistic and did not represent systems development in practice. The logical view of development and the introduction of more discipline to systems development process to achieve measurability, predictability and controllability of the systems development process was only a necessary condition. To create sufficient conditions in systems development methodologies, user participation and flexibility was introduced. The early-methodology era achieved some form of collaboration between the developer and the users. This moved system development methodologies to the time frame of systems development referred to as the methodology era.

2.2.1.3 The methodology era

The SDMs in this era were developed through research and best practise and were expected to provide consistency, coordination, communication, documentation and other activities that guided the systems development process. period was characterised by the proliferation of systems development methodologies (Conger, 2013; Avison and Fitzgerald, 2006). The system development methodology brand names increased to above 1000 (Conger, 2013; Avison and

Fitzgerald, 2006). Iivari *et al.* (2000) referred to this explosion as an SDM jungle. Avison and Fitzgerald (2006) argue that the methodology era does not mean that every organization adopted and used an SDM for systems development, but that most organisations were using some kind of SDM. These were either commercial formal systems development methodologies (SDMs), in-house adapted from commercial SDMs, in-house developed SDMs or in-house tailored from formal SDMs (Avison and Fitzgerald, 2006). The period also was characterised by the introduction of standards such as the Capability Maturity Model Integration CMMI (SEI, 2010) and ISO 9001(ISO 9001, 2000) to improve the development process.

This explosion of systems development methodologies was triggered by the need to respond to the changing system development problem circumstances. Researchers and practitioners acknowledge that the selection of systems development methodologies is dependent on the systems development problem circumstances or contextual settings of each individual systems development project (Young *et al*., 2016; Clarke and O'Connor, 2015; Henderson-Sellers *et al*., 2014; Conboy and Fitzgerald, 2010).

The systems development project contextual settings consist of a unique set of systems development constraints, characteristics and concerns that should be met to achieve an optimal interaction between the SDM characteristics and the systems project contextual factors (Sheffield and Lemétayer, 2013). These systems development constraints, characteristics and concerns are hereafter referred to as the systems development project contextual stressors or simply contextual stressors. Therefore, the selection of an SDM is dependent on the contextual stressors (Marks *et al*., 2017; Young *et al.*, 2016; Clarke and O'Connor, 2015). However, contextual stressors are not static, universally applicable and are not of equal importance in all systems development projects (Marks *et al*., 2017). The pre-methodology SDMs were criticised for their underestimation of the variability of the contextual stressors (Gill *et al*., 2018; Fowler, 2004). The importance and emphasis on a specific contextual stressor may vary from one organisation to the other, from one system development project type to another and within the same systems development project over time (Berente *et al*., 2015; Aitken and Ilango, 2013). Therefore, the evolution of SDMs is dependent on the contextual stressors dynamics (Berente *et al*., 2015).

Several perspectives emerged to mitigate the weaknesses of the early-methodology era SDMs. Wynekoop and Russo (1997, pp. 48) define a system development methodology (SDM) as "*a systematic approach to conducting at least one complete phase (e.g. requirements analysis, design) of system development, consisting of a set of guidelines, activities, techniques and tools, based on a particular philosophy of system development and the target system"*. This definition is aligned with the early-methodology SDMs and is flexible in terms of following and completing each one of the phases. The condition is that at least one phase should be completed. Rigidity is also reduced by considering that the SDM consists of a set of guidelines. The inclusion of a philosophy in the definition indicates that there were as many perspectives to systems development as there were approaches to systems development.

Each perspective to systems development influenced the development of a specific SDM instances to address the concerns targeted by that particular view point (Avison and Fitzgerald, 2006). For instance, to address the soft (human) aspect as a systems development concern, Soft Systems Methodology (SSM) was developed (Checkland, 1981). The SSM perspective assumes that systems development is a social problem and therefore it is important to give attention to the people as the most affected component of a system (Yaghini *et al*., 2009). In the case of data modelling as a concern, Information Engineering (IE) (Martin, 1990) was introduced as a data centric SDM. When the concerns were on both data and process modelling, systems development methodologies such as Structured Systems Analysis and Design Methodology (SSADM) (Eva, 1994), Yourdon Systems Methodology (Yourdon, 1993) were developed. When the concern became systems complexity, Object Oriented Systems Development Methodologies were developed (Booch, 1994).

The effort to address concerns can be viewed as attempts to address the limitations of the SDLC. Systems development methodologies continued to evolve, but within the confines of plan-driven paradigms. Thompson (1967) discovered three interdependencies that may exist in a systems development project context. The interdependencies influence the characteristics of an appropriate SDM for a given systems development project context (Barlow *et al*., 2011). The three interdependencies (Barlow *et al*., 2011) are the pooled, the sequential and the reciprocal. The pooled interdependencies represent a context whereby all requirements are independent and they do not influence each other. The sequential interdependencies represent a context whereby one requirement depends on one or more requirements that are not dependent on it or vice versa. The reciprocal interdependencies represent a context whereby a requirement is dependent on another requirement if and only if it also depends on it. The plan-driven approach is supported by the first two interdependencies. The first two support logical plans and sequencing of the systems development process. However, the reciprocal interdependencies do not support linearity. The view of interdependencies as sequential and linear created a conceptual illusion in systems development process. The waterfall methodology views progress as viable through coordination and thorough planning of the systems development process. However, reciprocal interdependencies do not allow linearity in systems development. The weaknesses of viewing systems development as sequential and logical led to challenges in this era. To deal with complexity introduced by interdependencies, three approaches were clearly identified. The developments were to be data centred (Martin, 1990), functional centred (process centred)

(Jackson, 1983), or a combination of both (blended) (Eva, 1994). The functional decomposition was thought as a way of simplifying complexity in a divide and conquer strategy.

The divide and conquer strategy was called the structured approach and it attempted to improve the waterfall methodology (Griffin and Brandyberry, 2008). The structured approach took a top-down approach to system development as a system was defined first at an abstract level. Secondly, the system gradually underwent successive refinement until a detailed level of granularity was achieved. In other words, it was based on functional decomposition, that is, the system was broken down into manageable modules in a disciplined way. The approach stressed graphical description to represent, document and communicate information about the system being developed. The assumption was that graphical representation has a higher information density than structured natural language. It is also a more specific, precise, easy to understand and less ambiguous notation compared to the natural language. Structured approach used graphical notation such as data flow diagrams (DFD), unified modelling language (UML) diagrams, data structure diagrams and entity relationship diagrams (ERDs).

Most of the systems development methodologies subscribing to the structured approach were front loaded. The front loading meant that a lot of work was done in the initial phases of the systems development. Thorough system analysis and documentation was introduced, and the distinguishing feature was that most of them were structured.

The interdependency influence that led to a category of structured systems development methodologies that emphasised data were referred to as data oriented, for example, the Information Engineering Methodology (IE) (Martin, 1990). On the other hand, those that emphasised process modelling were classified as process oriented systems development methodologies. The Jackson SDM (JSD) (Jackson, 1983) was one of the popularised processes oriented SDM of this period. The data oriented systems development focused on data modelling and the process modelling focused on modelling. To take advantage of both data and process modelling blended systems development was introduced.

The Structured Systems Analysis and Design Methodology (SSADM) (Eva, 1994) was one of the most commercialised blended SDM of this era. It is a front loaded, data-driven, process-driven, rigorously documented, and orderly phased system development methodology. Each phase together with its input and output is clearly distinguishable. Thus, development is standardised, and management and control is visible. It assumes a rational and technical world view of the systems development process. The SSADM complies with the ISO 9000, and perhaps this was a strong reason for it to be adopted by both government departments and the private sector in the United Kingdom (Middleton and McCollum, 2001; Fitzgerald, 1998), Ireland, Malta, Hong Kong and Israel (Fitzgerald, 1998).

A breakthrough in systems development during this period was the development of the Object Oriented Analysis and Design Methodology (OOADM) which has a bias towards intuitive abstraction of real-world entities (Booch, 1994). The systems object-oriented SDM allows visualising the system in terms of classes and objects that exist in the real-world. The complexity is simplified as the phases of the SDLC can be followed with minimum translation from the problem space to the solution space. Up to this time, methodologies were viewed as compact knowledge bundles and what the development team could do is to select and apply.

A different perspective from improving the SDLC took place in parallel to the structured approach to systems development. The systems approach attempted to understand the nature of systems development as a complex open system where human relationships and interactions are prioritised. The systems approach was grounded on the assumption that systems development is a social problem (Lane and Oliva, 1998; Naumann and Palvia, 1982; Checkland, 1981). Checkland's (1981) soft systems methodology (SSM) is a well-known example of a systems development methodology that addressed the problem from the systems approach. This SSM included techniques such as rich pictures which helped the users understand the developmental situation and therefore point to areas that needed improvement. The Soft Systems Methodology (SSM) presented a paradigm shift that explicitly incorporated social techniques in systems development. The systems approach dealt with challenges stemming from conflicting objectives, requirements or interests. Conflict amongst concerned participants had to be amicably resolved. The SSM SDM facilitated participation of user communities at levels of development to create a sense of ownership and reduce user resistance and increase the acceptance chances. The user participation reduced the need for training during system deployment. User participation and social interaction were regarded as imperative in systems development (Lane and Oliva, 1998). The demand for change management was also minimised. However, Cavaye (1995) points out that participation was neither sufficient nor necessary to guarantee systems development success. Cavaye (1995) observes that people might have the necessary skills but refuse or sabotage a systems development project. The system might deliver the services but users might not perceive its usefulness and lack job satisfaction. No matter how well conceived a system development may be, it may face adoption challenges (Cavaye, 1995).

Technology alone is not sufficient for mapping the systems development problem space into the systems development solution space. Different users have varying requirements and different perspectives and perception (Naumann and Palvia, 1982) about systems development process and the systems being developed. Thus, the degree of contribution, the magnitude and role of the user may influence the systems development process positively or negatively. Narrowing the semantic gap between the different user groups was the dilemma of the SSM. The activities for user participation were not explicitly defined since it was not clear when and where the user was expected to start and contribute to the systems development. The kind of participation expected from the user was also not clear. The time spent negotiating and reconciling conflicting views was another interesting factor in the SSM.

Another contribution different from just improving SDLC was Mumford's (1983) proposal of the Effective Technical Human Implementation of Computer Systems (ETHICS) in 1983. ETHICS gave emphasised both user participation and technology innovation. This system development methodology emphasised the interaction of technologies and people. User participation was fundamental as it was perceived not only as a means-to-an-end but as an end itself (Hirschheim and Klein, 1994). The ETHICS took a socio-technical view of the systems development process. When no improvement to the systems development was observed, Kendall and Kendall (1993) proposed the Multiview. Multiview was a framework for combining existing systems development methodologies to address specific areas of the systems development. It attempted to combine systems development methodologies to highlight the view point that systems development methodologies should not be treated as complementing each other.

The prescription of the steps from the SDLC and improved versions of it did not address the practicalities of the systems development process. The phases and the documented version of the SDM were not followed in a linear form. The adoption of the systems development was also in question. One that stood out was the number of systems development methodologies proposed in this era in response to perceived limitations of the systems development approaches. The proliferation of systems development methodologies in this era presented a challenge of selecting the appropriate systems development methodologies (Viljoen, 2016; Henderson-Sellers *et al*., 2014).

2.2.1.3.1 Strengths associated with the SDMs in the methodology era

The prescriptive nature of the systems development methodologies of this era had the following benefits:

- Compensating staff turnover by organising work around teams and not just a few experts.
- Standardisation of work to improve knowledge transfer and productivity.
- Increase of user participation in systems development projects.
- Standards such as the ISO certification (Fitzgerald *et al*., 2002) and Capability Maturity Model Integration CMMI (SEI, 2010) and ISO 9001(ISO 9001, 2000) improved the systems development process and reduced uncertainty in achieving systems development project goals and minimised systems development project risks.

- Systems modelling techniques allowed simplifying system analysis and design.
- Support for computer-assisted systems engineering (CASE) tools to speed up the development process.

2.2.1.3.2 Weaknesses of the SDMs in the methodology era

Despite the proliferation of SDMs in the methodology era, there were not contingent to any specific systems development project contextual factors. The selection of the most appropriate SDM was a challenge because of many options and most SDMs did not have a good systems development productivity record. The rigidity of the systems development methodology inherited from the early-methodology SDM era presented challenges. The introduction of standards aggravated the situation by increasing documentation in an already document heavy systems development process.

2.2.1.4 The post-methodology era

The time interval from the late 1990s to today falls under the post-methodology era which saw the emergence of an SDM family referred to as agile SDMs (Hohl *et al.,* 2018; Barlow *et al*., 2011) that prioritises individuals and interaction, working system, customer collaboration, response to change instead of focusing on rigid plan-driven approach that value comprehensive documentation, contract negotiation, processes and tools (Hohl *et al.,* 2018). There are many individual instances of agile systems development methodologies. Some examples of agile SDM instances are Scrum (Schwaber and Beedle, 2002), Extreme Programming (XP) (Beck, 2000), Crystal (Cockburn, 2004), and Adaptive Software Development (ASD) (Highsmith, 2000). The differences between the instances is the extent to which they emphasise the Agile Manifesto values (Hohl *et al.,* 2018; Judy, 2012). The agile systems development methodologies deal with dynamic, volatile and unpredictable user requirements (Hohl *et al.,* 2018; Chan and Thong, 2009; Beck *et al,.* 2001) and embrace active participation of users throughout systems development process.

Agile SDMs typically use a spiral model (Boehm, 1988) which represents a series of iterations, or revisions, based on user feedback. As the development process continues, the systems development artefact gradually evolves through small incremental releases with short development cycles into a complete system development project artefact (Stoica *et al*., 2013). For the first time in the history of systems development methodologies, agile SDM had a manifesto (Hohl *et al.,* 2018; Beck *et al.,* 2001). The manifesto provided the values and the guiding principles. However, the agile approach has its challenges in systems development (Mirza and Datta, 2019; Kuhrmann *et al*., 2017). For example, critics have indicated that agile systems development methodologies are not suitable for developing complex systems as they are weak in providing comprehensive architecture (Young *et al*., 2016). When the team size increases, it breaks in the coordination and communication (Young *et al*., 2016). In some instances, the hybrid systems development methodologies are created, for example this is the case in the Water-Scrum-Fall (Kuhrmann *et al*., 2017; West, 2011) to address specific systems development situations. The systems development methodologies during this period considered the diverse perceptions on usage, challenge of inadequacy to respond to specific systems demands on issues like time and resources (Kuhrmann *et al*., 2017; Chan and Thong, 2009).

The era is also characterised by diverse perspectives about the relative advantage assumptions of the systems development methodologies. The period saw the first publication of a pessimistic Chaos Report by the Standish Group (1995) about the failure of systems development projects. Iivari and Maansaari (1998) presented a list of criticism about the contribution of methodologies in systems development. Critics were not advocating abandoning the use of systems development methodologies as such, but were calling for reflection on the role of systems development methodologies. A reassessment had to be done on how the systems development methodologies had helped deliver quality systems, improve productivity, provide knowledge and maintain standards. Success was no longer viewed as adherence to a plan nor conformance to it thereof, but addressing the stated and implied requirements of the system. The reappraisal considered a deviation from strict deployment of systems development methodologies to a contingent use approach. That is the adjusting, adapting, blending and creating alternative systems development methodologies according to the particular needs of each systems development project (Huisman, 2013; Fitzgerald *et al*., 2006). The contingent use of systems development methodologies also focuses on how systems development methodologies scale up or down, tailored and adapted during the development process. How the SDMs change when applied across similar context and within the same context over time (Carroll, 2003).

Previous systems development methodologies research had focused on developing new systems development methodologies, and on frameworks for selecting and understanding systems development methodologies (Siau and Rossi, 2011). The focus is now on understanding the usefulness of systems development methodologies in practice (Avison and Fitzgerald, 2006) and the possibility of combining the instances of SDMs into hybrid SDMs (Gill *et al*., 2018; Imani *et al*., 2017; Kuhrmann *et al*., 2017; Rahmanian, 2014). The hybrid SDMs address the historical flaws in systems development by not adhering strictly to neither plan-driven nor agile SDMs, but create a hybrid of SDM practices as dictated by the systems development problem situation (Gill *et al*., 2018; Imani *et al*., 2017; Isaias and Issa, 2015; Rahmanian, 2014). Therefore, a hybrid SDM is not a once-off activity, but a process that is continuously evolving through the gradual adoption of new SDM practices and combining them with old SDM practices for each systems development project context setting (Isaias and Issa, 2015; Rahmanian, 2014).

There was a gap between the theoretical conception and perception of the systems development methodologies' purported benefits and the practical usefulness of the same on deployment (Middleton and McCollum, 2001).

2.2.1.4.1 Strengths associated with SDMs in the post-methodology era

- Adaptive and flexible and designed to be tailored to the systems development project contextual factors (Gill *et al*., 2018; Imani *et al*., 2017).
- No requirement for adherence to a plan nor conformance to it thereof.
- Increase of user participation in systems development projects.

2.2.1.4.2 Weaknesses of the SDMs in the post-methodology era

Reassessment is a deviation from strict deployment of systems development methodologies to a contingent use approach which may threaten the discipline of the development process. The implementation of the contingent approach demands a high level of expertise from the developers. Therefore, there is need to manage the possibility of conflicting assumptions as the SDM may be a hybrid developed from combining components from the agile SDM class instances and plan driven SDM class instances.

The importance of the history of SDMs to the systems development research and practice is discussed in the next section.

2.3 THE RELEVANCE OF SDMS HISTORY TO SYSTEMS DEVELOPMENT PRACTICE AND RESEARCH

The history of systems development methodologies is characterised by transition in the perception of systems development. The initial philosophical stance was objectivistic in that organisations believed there existed one reality about the system to be developed, independent of any systems development practitioner and the perceived reality was the same for both the systems development practitioner and the customer. Therefore, the systems development practitioner did not have to involve customers as what he perceived was assumed to be aligned with what the customers required. The objectivistic philosophical were flawed and led to the constructivist philosophical position that believed in one reality, but also took into consideration that reality and perceived reality are different. The universal applicability assumption of systems development methodologies view proved flawed. The systems development methodologies were discussed among systems development practitioners and communication was prioritised.

Henderson-Sellers *et al.* (2014) observes that systems development methodologies were not implemented in their documented version perhaps due to discussions and view exchange between systems development practitioners. Henderson-Sellers *et al.* (2014) identifies the following levels: the SDM as documented; and the SDM as enacted and implemented by the systems development practitioners in a specific systems development project. This implied that systems development methodologies may be interpreted and experienced differently at organisational, systems development project and individual systems development practitioner level (Berente *et al*., 2015; Aitken and Ilango, 2013; Huisman, 2013). The historical perspective assists both researchers and practitioners to identify principles worthy of maintaining and those that should be discarded. Practitioners identify what to continue doing, what to start doing, or what to stop doing.

In most of the historical accounts on the evolution of systems development methodologies, contextual stressors impose different influences and shape the systems development methodologies. An SDM might be influenced by contextual stressors at a macro level component such as its philosophical approach assumptions or at a micro level component such as the systems development method. The historical perspective shows that the philosophical and the process model components are fundamental change drivers. The philosophical approach determines the structure and the form of the contextual stressors that can be considered. Therefore, the philosophical approach constrains how the systems development problem is perceived. For example, the systems development practitioners may ask the following key philosophical question: Are systems requirements gathered or constructed? (Iivari *et al.,* 2004) The answer to this question may indicate orientation towards traditional plan-driven SDM class, agile value-driven SDM class or a hybrid SDM class.

The traditional plan-driven SDM class considers systems requirements as already existing somewhere and need to be discovered or identified. Therefore, the SDM would be predictive if the assumption is that systems requirements are identifiable and can be gathered in a clear and deterministic manner. Since it is possible to gather systems requirements, then it should be feasible to draw a detailed plan for the development of a system. The seminal breakthrough of systems development methodologies introduced by Royce (1970) outlined systems development planning as a waterfall, based on the assumption that systems requirements can be gathered at the outset. The waterfall metaphor illustrates irreversibility of decisions and plans. In a waterfall, water cannot go back except through starting afresh the water cycle process. The assumptions of the waterfall model have been criticised. (Beck *et al*., 2001; Highsmith, 2000).

The agile SDM class considers systems requirements as artefacts that are gradually constructed. Instead of a detailed plan, the agile SDM class creates detailed plans incrementally as the systems requirement artefacts are gradually constructed (Fowler, 2004; Iivari *et al*., 2004). The systems development team does not attempt to draw a detailed plan of the system at the beginning of the project. The 'getting it right' is de-emphasised which makes it easier to reverse decisions (Fowler, 2004). The reversibility of decisions makes it possible to iteratively respond to changes that can happen at any stage and at any time during the development process (Fowler, 2004).

There are a variety of systems development methodologies and some of them are combined into hybrids to reduce their individual shortcomings (Špundak, 2014; Rahmany, 2012). The availability of a variety of systems development methodologies makes the selection of an SDM for a specific systems development project challenging. The contextual stressors inform the choice of an SDM for a specific project (Massey and Satao, 2012). If the contextual stressors are clear and stable, the adopting unit might adopt an instance of a traditional plan-driven SDM class, but if contextual stressors are dynamic, the adopting unit might adopt an instance of an agile value-driven SDM class.

Major variations in systems development methodologies are at the philosophical systems development approach level. The history of systems development methodologies reflects the change of emphasis based on the perceived contextual stressors. Each systems development project is unique and the choice of an appropriate SDM should be based on a detailed examination of the contextual stressors pertinent to the systems development project setting. The evolution of systems development methodologies seeks to address the past SDM shortcomings and the present challenges as well as inform the future of systems development methodologies. The historical changes in systems development methodologies and perceptions about the contextual stressors are listed below:

- Sequential and predictive systems development methodologies (it is possible to have a big system design from the outset and follow the systems development phases sequentially up to completion) (PMI, 2013).
- Iterative systems development methodologies (the need to revisit or rework and repeat systems development in cycles) (PMI, 2013).
- Incremental SDM (prioritise feature development by developing the initial version of the systems development product and letting it evolve towards a full-fledged systems development product with each iteration (Ruparelia, 2010).
- Adaptive systems development methodologies (value-driven and change-driven adoption, adaptation and adjustments to the development activities) (PMI, 2013).
- Hybrid systems development methodologies (a combination of systems development methodologies) (Gill *et al*., 2018; Isaias and Issa, 2015; Rahmanian, 2014).

Each systems development project is unique and the selection of an appropriate SDM should be based on the systems development project's contextual stressors. The systems development methodologies evolved in response to the philosophical assumptions about the variation in each systems development project's contextual stressors. An essential aspect of an SDM is its underlying set of philosophical assumptions about reality (contextual stressors). Inappropriate underlying philosophical assumptions about the reality (contextual stressors) may introduce challenges in systems development. There are many studies that are biased towards proving the universal superiority of one SDM class over the other (Versionone, 2018; Standish Group, 2016; Janes and Succi, 2012). The advocates of either the plan-driven SDM class or the agile SDM class in most cases highlight the limitations of the opponents' SDM class (Jiang and Eberlein, 2008).

Research and practice has changed this adversarial narrative towards a view in which systems development methodologies coexist and are complementary to each other and that they may be combined to tap into the capabilities of one another in addressing specific systems development project situations (Gill *et al*., 2018; Isaias and Issa, 2015; Rahmanian, 2014; Janes and Succi, 2012; Leau *et al.,* 2012; Jiang and Eberlein, 2008). Therefore, comparing SDM classes for superiority of one over the other instead of assessing how they can complement each other on specific systems development project situations is counter-productive (Rahmanian, 2014; Janes and Succi, 2012).

The next section proposes a conceptual definition of an SDM.

2.4 SDM DEFINITION

To conceptualise the notion of an SDM, there are two perspectives to consider; the consistent rigorous definition and the fundamental characteristics that constitute the concept. Table 2-2 presents some of the definitions of the SDM concept.

Table 2-2: Summary of some SDM definitions

AUTHOR	DEFINITION
Charvat (2003)	"A set of guidelines or principles that can be tailored and applied to a specific situation: it is a specific approach, templates, forms, and even checklists used over the project life cycle".
Avison and Fitzgerald (2006)	Methodology is defined by Britain Computer Society (BCS) (Avison and Fitzgerald (2006) as " a set of recommended means for information systems development or part of it which is based on logics and specific philosophy. The recommended means often contain definition of phases, procedures, activities, rules, techniques, documentations, tools and guidance. It may also include suggestions from management and organisation, approach and determining and training participations".

AUTHOR	DEFINITION
Huisman and Iivari (2006)	"An SDM is a combination of the following: i. 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. ii. 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. iii. 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. E.g. IE (Martin, 1990). iv. 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 use case diagram, sequence diagram, activity diagram". (OMG, 2013)
Viljoen (2016)	"A System Development Methodology (SDM) formally prescribes how to execute and manage the collection of process models of the methods in a specific way, together with the integrated philosophy, to develop and document an information system (IS), or any part thereof, with the help of some tools and/or custom techniques in a specific situation/environment. Each development can be broken down into smaller parts where the SDM is still enforced".

Huisman and Iivari (2006) note that it is hard to grasp "SDM" due to a lack of conventional, specific and thorough definition. Iivari and Maansaari (1998) point out that there was no authoritative source for the definition of methodology. Iivari and Huisman (2007) further point out that although SDM has been researched extensively, it is still an ambiguous concept Mingers and Brocklesby (1997) also concur SDM is subject to diverse and perhaps controversial interpretations and this may lead to different assumptions about the same problem situation.

Reed (1996) points out that distinguishing SDM from a method is not trivial. Fitzgerald (1998) and Vlasblom *et al*. (1995) use the two terms, method and methodology interchangeably and this reflects the European and North American traditions respectively (Iivari and Maansaari, 1998; Iivari *et al*., 2001). By contrast, Middleton and McCollun (2001) use the terms methodology and process model interchangeably. Wynekoop and Russo (1993) state that systems development methods incorporate both methodologies and process models. This suggests that a systems development method is a higher order construct than an SDM. These different perspectives of SDMs create conceptual inconsistency. Iivari and Maansaari (1998) identify two dimensions of inconsistency; the scope and category problems. The scope problems incorporate both the extent in which the systems development process is embraced by the methodology, and the aspects to be covered by a methodology. The category problems include differentiating SDM and systems development techniques, and the distinction between SDM and systems development approach (Iivari and Maansaari, 1998).

There is no single definition for SDM and the existing definitions have varying levels of abstraction. Some of the definitions are inconsistent whereas, some are even incompatible to each other. For example, Huisman and Iivari (2006) use both the terms, systems development method and SDM distinctly. They state that an SDM incorporated a systems development method. On the other hand, Wynekoop and Russo (1993) assert that systems development methods encompass systems development methodologies.

The definition of SDM is debatable and most researchers state their working definitions to provide a clear and precise frame of reference as shown in Table 2.2 above. However, notwithstanding the noted challenges, important aspects can be identified in the various definitions:

- That there is a philosophy or an approach behind every SDM which could be either explicitly or implicitly expressed (Maddison *et al*., 1984; Wynekoop and Russo, 1997; Iivari *et al*., 1999; Avison and Fitzgerald, 2003).
- System development methodologies include a set of methods.
- System development methodologies include a set of techniques.
- System development methodologies include a process model.
- System development methodologies serve as normative principles guiding both technical and behavioural expectations (organising, structuring and directing of each member of the development team's thinking and action) (Brinkkemper, 1996; Iivari *et al*., 2001).
- That system development methodology should be trainable, learnable, socially acceptable, justifiable and flexible (Maddison *et al*., 1984; Avison and Fitzgerald, 2003).
- That system development methodology provides an analytical framework and guidelines on what to do, how to do it, when to do it and why do it (Jayaratna, 1994).

This study does not propose a definition of SDM, but provides a frame of reference that guides all subsequent discussions. This is based on four main components identified by Huisman and Iivari (2006). The term SDM constitutes the following:

- Systems development approach(s):

This is a set of philosophical assumptions that define the nature of intervention. It embodies the goals, guiding principles, fundamental concepts and principles of the development process (Iivari, *et al*., 2001). The distinguishing factors between approaches are mainly the three philosophical dimensions:

i. Ontology- This, in its strictest meaning characterises the type of entities assumed to exist and the nature of that existence (Mingers and Brocklesby, 1997). Different categories of users, technology, and environment should be identified. Entities or components of a system and the relationships among these components should be understood. For example, in ETHICS the assumption is that there are users and technology. Both these entities are to be optimised. Therefore, the belief is that socio-technical design will close the gap between technical needs and social needs.

ii. Epistemology- This is how the perceived world is conceptualised, the possibilities of and the limitations on (Mingers and Brocklesby, 1997) the knowledge of the subject under study. It influences the interpretations and the level of abstraction and representation of concepts opening up the desire to discover more and add into the body of knowledge in the domain of systems development. It opens avenues to additional research and continuous improvement in the area as it provides a structural framework for the acquisition of knowledge (Fitzgerald, 1998) and allows sharing of similar perceptions.

iii. Praxiology- In this dimension, the focus is on how actions should be taken in an informed and reflexive way (Mingers and Brocklesby, 1997). This generates the experience and tacit knowledge as well as the development of standards and best practices in systems development. Examples of philosophical approach include structured, and socio-technical design.

- System development method(s):

A systems development method is an organized set of techniques, their interconnections and use. It consists of prescribed steps taken for accomplishing well-defined tasks, based on normative principles, a set of guidelines, activities, techniques, and tools, responding to a particular philosophy and the target system (Wynekoop and Russo, 1993). For instance, the PRojects IN Controlled Environments version 2 (PRINCE2) (OGC, 2009).

- Systems development process model:

A process model is a coherent networked sequence of activities, objects, transformations and events in the systems development (Vlasblom *et al*., 1995). It provides the logical sequence of tasks performed at various levels of detail and aggregation to achieve a particular objective in the development of a system. It is used to provide a notation or linguistic representation of a system as a perceived reality (Walden *et al*., 2015) and can be categorised using three independent features:

i. the manner in which the system is released, whether monolithic or incremental,
ii. the extent to which the functionality of the system is prescribed, and
iii. the way in which the activities are carried out, that is either iteratively or linearly (Vlasblom *et al*., 1995).

A process model articulates the assumptions of a paradigm related to the manner in which system development activities are organised in time and space (Vlasblom *et al*., 1995). Therefore, process models give different topological arrangements of activities, at different levels of abstraction.

Examples of systems development process models include Prototyping, Spiral, and Concurrent.

- Systems development technique(s):

A technique is a specific activity that has a clear and well-defined purpose within the context of a methodology. It consists of an unambiguous set of basic operations that can guarantee success if executed correctly (Iivari *et al*., 2001). The relationship between methodology and technique is viewed as that between a *what* and a *how*. The methodology specifies what type of activities should be undertaken, and the technique specifies the manner in which the underlying technical details of those types of activities are handled (Mingers and Brocklesby, 1997). Examples of techniques are Transition Diagram, and Object Diagram (OMG, 2013).

The definition is important so that systems development methodologies can be studied, evaluated, classified and compared in a systematic manner. The above definition in addition provides a comparative framework that can be used to analyse, evaluate and compare systems development methodologies. The systems development methodologies can be assessed against components of systems development methodologies expressed in the definition. One of the core components of a systems development expressed in the definition is the set of philosophical assumptions. The underlying philosophical assumptions may be analysed to determine the classification, the strengths and the limitations of an SDM. The underlying assumptions is also guided on which systems development can be combined to form hybrid systems development methodologies. Furthermore, through assessing whether the underlying philosophical assumptions hold in a particular systems development project, it is possible to select the most appropriate SDM for that systems development project. A working definition will be provided in Chapter 3. The classifications of systems development methodologies based on some of the constituents of SDM are described in the next section.

2.5 CLASSIFICATION OF SDMS

Figure 2-2 summarizes the three approaches according to which the SDMs classification is presented in this section, namely the philosophical approach, process model, manufacturing engineering inspired model and the Agile Manifesto.

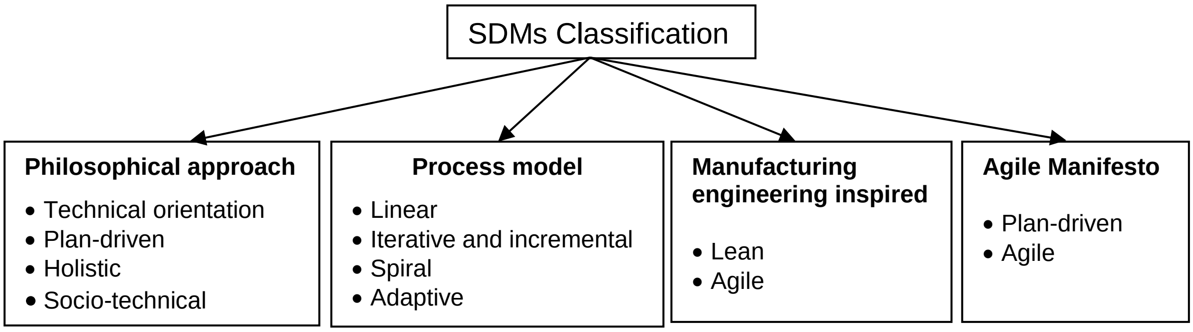


Figure 2-3: Classification of SDMs

There is no single way to classify systems development methodologies since some of the classifications may overlap. This intersection makes it challenging to carry out comparative studies. The classification of systems development methodologies may vary significantly in terms of their philosophical approaches, goals, guiding principles, fundamental concepts and principles of the development process (Iivari *et al*., 2001). The classification of systems development methodologies has a bias towards articulation of certain philosophical assumptions of the problem domain. The philosophical assumptions are based on the set of basic beliefs held by the creator of specific systems development methodologies. The philosophical assumptions include the beliefs concerning the nature of reality (ontology), the beliefs on how the knowledge about the problem domain can be acquired (epistemology), and the possible actions that can be taken (praxeology). These philosophical assumptions might be at organisational level, systems development project level, individual level or any other systems development contextual level (Iivari *et al*., 2001). Iivari and Iivari (2011) and Iivari *et al*. (2001) point out that classification at the level of methodology is based on instances of classes. Therefore, they suggested that systems development methodologies are instances of systems development approaches. Iivari *et al.* (2001) also propose a detailed four tier classification based on the underlying philosophical assumptions. The four-tier framework consists of an inheritance structure indicating system development methodology classes and their instances.

Iivari *et al*. (2001) propose classification expansion algorithm for their framework. The following components summarise the classification framework proposed by Iivari and Iivari (2011) and Iivari (2001):

- Systems development philosophical assumptions/paradigms:

The systems development philosophical assumptions/paradigms form the generic class with functionalism, social relativism, neo-humanism, and radical structuralism as instances (Iivari and Iivari, 2011; Iivari *et al.,* 2001). Each instance inherits the characteristics of the paradigm they are grounded upon.

- Systems development approaches:

The systems development approach is interpreted as a specific class of systems development methodologies with common features. Each SDM inherits features from one (single inheritance) or more (multiple inheritance) systems development approaches (Iivari and Iivari, 2011; Iivari *et al*., 2001).

- Systems development techniques:

Systems development techniques assume, is-part-of relationship, to systems development methodologies. These provide detailed concepts and notations. For example, applying the classification framework, and following a bottom up approach, an Entity Relationship Diagram (ERD) is a systems development technique. It is part of a Structured Systems Analysis and Design Methodology (SSADM), which is an instance of Structured Approach, which in turn is grounded on Functionalist paradigm (Iivari and Iivari, 2011; Iivari *et al*., 2001). Top-down or bottom-up schemes may be used to perform the classification of systems development methodologies. Classification using micro-level concepts like systems development techniques resulted in many classes which were not helpful to both research and practice. However, the framework avoids a concomitant number of systems development classes by varying the levels of abstraction. In other words, the framework organises classification at macro-level which in this case is the systems development approaches. It should be noted that macro-level or philosophical aspects are a critical issue for understanding systems development methodologies.

The introduction of new systems development methodologies and improving the existing ones continue (Griffin and Brandyberry, 2008). The classification of systems development from a philosophical view is a step towards a deeper understanding of systems development methodologies. The process model gives lower details on the philosophical underpinnings. The emphasis on certain aspects creates another avenue to further classify the systems development methodologies. Several systems development methodologies may share the same philosophical perspective, but differ at the level systems development process model, techniques or emphasis. The SDM may inherit characteristics from different philosophical perspectives. Additionally, systems development models can be employed individually or combined or interleaved to address specific situations. The classification of systems development methodologies based on a philosophical approach (Iivari and Iivari, 2011; Iivari *et al*., 2001) and engineering discipline (Fowler, 2005) are discussed in the following subsections.

2.5.1 Classification based on the philosophical approach

A philosophical approach is a set of assumptions in which an SDM or a set of methodologies are grounded. Iivari and Iivari (2011) and Iivari *et al*. (2001) indicate that the philosophical approach is an abstract point of SDM classification (Iivari and Iivari, 2011; Iivari *et al*., 2001). The classification of systems development methodologies may clarify common assumptions principles and concepts and practices shared within a group of systems development methodologies.

2.5.1.1 Technical orientation

The philosophical view point creates differences in systems development problem interpretation. The technological orientation views systems development as a technological problem that should be solved using technical skills, experience and expertise (Schach, 2010). Users do not know what they want, therefore they cannot be involved in systems development (West, 2011). The code-and-fix SDM class was prevalent during the pre-methodology era (Jayaswal and Patton, 2006). There were no formal requirements as it was not believed to be important (Schach, 2010).

According to Young (2002), approximately 85 percent of the defects in developed systems originate in the requirements. The incorrect assumptions on requirements accounts for approximately 50 percent of the defects in developed systems (Young, 2002). Schwaber and Sutherland (2012) found that typical systems development processes have more than 35 percent of their requirements changed. The ad hoc class that focused on systems development from a technological perspective failed to achieve the goal. Requirements are important and they need to be solicited, interpreted, modelled and managed accordingly (Schwaber and Sutherland, 2012).

2.5.1.2 Plan-driven (Traditional or Classical) class

The underlying assumptions are that requirements are deterministic and the problem situation is well structured, with clear objectives. The plan-driven class relies heavily on up-front detailed planning and documentation before systems development starts (Young *et al*., 2016). The plan-driven SDM class employs functional decomposition, that is, the breaking down of a systems development problem into manageable units in a disciplined manner. This plan-driven approach views systems development as a set of sequential steps from start to finish. The plan-driven SDM class is anticipatory in terms of the development approach (Leffingwell, 2011). The systems development using this class is characterised by sequential phases of planning, implementation, verification, validation and a monolithic release of a complete system at the end of the project (Boehm, 2006). System descriptions and components are represented graphically using various diagrams such as data flow diagrams, data structure diagrams, entity relationship diagrams, use case diagrams, sequence diagrams, activity diagrams and class diagrams as a way of improving communication and avoiding ambiguities of the natural language descriptions.

The plan-driven class is also referred to as a weak class of systems development methodologies. A weak class consists of a generic set of systems development methodologies purported to address problems of any nature. They use the “one size fit all philosophy”, that is, they are to guide systems development irrespective of the context of the development project (Howard *et al*., 1999). The planning and coordination of activities are rigorous. The development is hypothesised as a linear and sequential set of activities that can be managed by proper planning and the coordination of systems development process.

The plan-driven class is purported to enhance communication, reduce systems development project risks, and improve the control of the development process (Fitzgerald, 1996) through detailed documentation. It is also dependent on standard development framework that guarantees continuity, accountability and controllability of the systems development process (Avison and Fitzgerald, 2003). The management style is mostly command and control as it is also process centred (Boehm and Turner, 2005). Risk management is mitigated through thorough documentation and conformance to plans (Highsmith, 2010; Boehm and Turner, 2003). In addition, a review process is done at each phase of systems development through testing, verification and validation. Change is prevented as it is regarded as a threat to the systems development project cost, schedule and scope. That is, changes are frozen to keep the systems development project scope fixed. The change in scope may impact on systems development project resources, cost and schedule. Hence, upfront planning and designing of the systems development project prevents scope creep (PMI, 2013).

The following are some examples of systems development methodologies that fall under the plan-driven class:

- Waterfall methodology (Royce, 1970)
- Information Engineering (IE) (Martin, 1990)
- Rapid Application Development (RAD) (Martin, 1991)
- Yourdon Systems Methodology (Yourdon, 1993)
- Structured Systems Analysis and Design Methodology (SSADM) (Eva, 1994)
- Rational Unified Process (Jacobson *et al*., 1999).

The plan-driven class of systems development methodologies attempts to rigidly manage requirements. It takes a predictive stance on requirements in that once requirements are gathered they do not change a lot (Conger, 2013). However, Schwaber and Sutherland (2012) indicate that requirements are dynamic and approximately 35 percent of the requirements will change during the systems development project life cycle.

The involvement of the user only at the start and at the end when the system is being deployed is a weakness in this class. It is expensive to develop a complete development plan upfront due to unpredictability and dynamics of requirements (Conger, 2013). Change is inevitable and it should be anticipated, planed for, accommodated and managed instead of being resisted or avoided (Strode and Huff, 2015). Systems development is a process of change. It is a transformation or a change from the problem space onto the solution space. Change occurs in many ways and is always part of systems development. There may be changes in requirements due to incorrect assumptions about requirements, omitted requirements, inconsistent requirements or ambiguities within requirements (Young, 2002). Changes may occur in requirements, scope, technology, market segment and people (Highsmith and Cockburn, 2001).

There are circumstances where a plan-driven SDM class is a best fit and other circumstances where they are not appropriate. The assessment of fit is based on whether the assumptions made by the systems development hold or are valid to the systems development project situation.

2.5.1.2.1 Organisation of plan-driven SDM instances

Most plan-driven SDM class instances share the description of the roles that may be assigned to members of the development team, key artefacts that may be created and the meetings that may be convened.

2.5.1.2.1.1 Plan-driven SDM roles

Roles are task area competences required to achieve a systems development goal (Somerville, 2016). The plan-driven SDM defines a standard set of roles for guiding the specific involvement of various participants in a systems development project. These roles are not necessarily job descriptions as it is common for a single member to be assigned to more than one role in the same systems development project. Each member of the development team is assigned a role. The roles indicate the level of competence expected of the member assigned and is associated with accountability. The roles impose different levels of accountability among members of the development team which leads to task-based differences, that may constrain shared understanding between the systems development team members (Huisman and Iivari, 2006). Each member of the systems development team is strictly expected to perform the assigned tasks individually as there is recognition for individual work. Each role is associated with a level of accountability. The key roles include the Project Manager, Systems Analyst, Systems Designer, and System Quality Assurance.

- Project manager

The Project manager’s responsibility is the allocation of resources, systems development project expenditures, and the overall organisation of the systems development project.

- Systems Analyst

The systems analyst is an agent of change responsible for developing a plan for change, facilitating change and suggesting the SDM to be deployed for a particular systems development project. The systems analyst deals with the stakeholder management processes by facilitating the communication between the developers and users.

- Systems Designer

The systems designer is responsible for the user interface design, database design, program design.

- System Quality Assurance

The system quality assurance is responsible for performing various tests to the system to create and sustain the quality of the system.

2.5.1.2.1.2 Plan-driven SDM artefacts

The key artefacts of the plan-driven SDMs include the systems development project proposal, systems development project charter, systems development project plan, systems requirement specification, systems design specification, systems test plans and results, and systems deployment plan. Each of the artefacts is approved and sign-off by management.

2.5.1.2.1.3 Plan-driven SDM meetings

Documentation is used at all phases of the plan-driven SDM systems development projects. The first meeting with the customer is used to generate the systems development project proposal, systems development project charter, systems development project plan, and systems requirement specification. The meetings with the customer is minimised from the design phase after which communication is through deliverables. The main meetings are at the beginning of the systems development project and at the last stage of the systems development project.

2.5.1.3 Holistic class

The holistic class of systems development methodologies view systems development as a subset of a whole complex social problem. To understand the situation, the whole complex problem should be investigated. Social factors such as behaviour and experience of developers and users are identified as soft and technology is considered as hard. The early days of systems development did not consider users in the development of systems (Valacich *et al.*, 2012) and this had costly consequences. The social aspects made the systems development problem unpredictable and dynamic. Significantly, from the early 1980s, efforts to involve users were intensified. Checkland (1981) introduced the importance of user involvement through Soft Systems Methodology (SSM). In northern Europe, Participatory Design was introduced to underscore the importance of the social aspects in systems development process (Valacich *et al*., 2012).

The strength of the holistic class is its inclusion of the human factor in systems development. Issues related to conflict of interests are dealt with in an amicable way. The holistic class develops a sense of ownership and acceptance of the system development product.

Despite its strengths, Cavaye (1995) criticized this class of systems development methodologies stating that participation was neither sufficient nor necessary to guarantee systems development success. Cavaye (1995) further argues that the level of participation is not explicitly stated which creates role conflicts for those involved in the systems development project.

2.5.1.4 Socio-technical class

The socio-technical class emphasises the fit between the technical, operational and the social objectives of the systems development process. User participation and technology innovation are equally valued. The belief is that best development is achievable through optimisation of both users and technology. User participation is fundamental as it is perceived as central to addressing the social needs (job satisfaction) not only as a means-to-the-end but as an end itself (Hirschheim and Klein, 1994). Social and the technical factors are important aspects of systems development and fit between the technical and the social aspects is given priority in the development of systems (Mumford, 1983). The technical orientated class was the first view to be considered as important in the pre-methodology era of the systems development methodologies. From the onset, the systems were viewed from a technological perspective. This implied that requirements could be gathered completely in a systematic way and all the planning done upfront. Separating the technical and the social did not give positive results. Consequently, Mumford (1983) proposed Effective Technical Human Implementation of Computer Systems (ETHICS to achieve balance between the social and the technical aspects of systems development. In the post-methodology era, the agile manifesto highlighted the importance of the user and developer interactions as well as a working system in the systems development process (Hohl *et al*., 2018).

2.5.2 Classification based on process model

An SDM may employ one or more process models. The predominant process model will be considered for the classification. However, in some instances, it will not be explicit which process model is dominant.

2.5.2.1 Linear class

The linear models organise the systems development project as a sequence of phases throughout the whole process (Wysocki, 2011). Each phase of the systems development project consists of single-pass tasks with predefined deliverables that are signed off after completion as an indication of progress and success. Therefore, the linear class organises a systems development project as a sequence of tasks that allows the tracking, monitoring and evaluation of progress. The progress checking indicators are referred to as deliverables and these are in the form of documentation, such as requirements specification document or feasibility study document. The validation, verification and stage-by-stage deliverable signing off are in line with quality assurance and configuration management. The phases are prescribed and tasks planned strictly adhering to the set plan. Changes are neither expected nor entertained without a significant cost. Communication is used to support compliance and conformance to the set plan. Success of the systems development project is measured by conformance to the plan. However, systems development is not a linear or sequential process and therefore the linear class does not model the way systems are developed. Since requirements are not easy to determine from the outset, they are understood as development progresses. The linear class hinders precision in requirements determination as it employs a once off requirements gathering stance. Users are consulted at the beginning of the systems development project and at the end when the system is delivered. An instance of a linear SDM class is the Waterfall methodology.

2.5.2.2 The iterative and incremental class

The systems development project is divided into smaller segments and provision for change during the development process is made. The iterative and incremental class combines linear and iterative development that build a system one step at a time. Requirements are divided into multiple iterative development cycles. Requirements and design may change within or during any iteration. Each one of the iterations runs through all the development phases (i.e. analysis, design, coding and testing) and it represents an increase in details of the system features according to a predefined stage-by-stage plan for the realization of the systems development project. The iterative and incremental class allows a stage-by-stage delivery of the system to the customer (Charvat, 2003). The cycles are divided into smaller and more easily manageable tasks and this makes it more flexible and less expensive when systems development project requirements and scope change. Even though this approach allows more flexibility, it is still a linear development approach and inherits the disadvantages of the linear class.

2.5.2.3 Spiral class

The important phases in a spiral systems development class are objective setting, risk assessment and reduction, development and validation, and planning. The baseline spiral starts at objective determination phase, requirements are gathered and risk is assessed and reduced followed by the development and validation, and planning. Each subsequent spiral is built on the baseline. Each spiral involves a progression through the spiral phases and each complete spiral baseline represents one stage of a systems development project under development. The spirals consist of the same sequence of phases of objective setting, risk assessment and reduction, development and validation, and planning (Boehm, 1988). The spiral class puts the requirements at the centre of a spiral. The requirements at the centre are abstract. The details to the requirements are added with each the spiral. The spiral class is based on the incremental and iterative development approach. The spirals provide a mechanism to design and develop the system and refine the systems development project requirements incrementally until all systems development project requirements are implemented. In each subsequent spiral, incremental changes are added. The philosophy of the spiral class is that systems development project risks can be reduced by dividing large systems development projects into manageable work units. The spiral class minimizes the systems development project risk, supporting changes to the systems development project, and supporting manageable development of larger systems development projects by deconstructing a systems development project into smaller tasks and allowing for analysis of risk in each one of the systems development iterations. The systems development project cost is directly proportional to the length of the spiral. One of the shortcomings of the class is the need for intense involvement of the development team in terms of risk management, risk assessment, and risk documentation.

2.5.2.4 Adaptive class

The adaptive class is flexible and responds well to changing requirements. It is an extreme version of the iterative and incremental class. It is a time-boxed model that delivers features and accepts and expects changes during all stages of the systems development project. The adaptive SDM class responds to and accepts risks and manages them efficiently. It has three stages; assessment, evaluation and adjustment stage. The adaptive class is appropriate where there is high degree of uncertainty and complexity (Wysocki, 2011), because it is characterised by being context sensitive. Each one of the iterations seeks to capture requirements that are known and the way in which they are managed. The need to improve from the previous way is considered after evaluation. The adaptive class is flexible itself and in itself. It can adjust when there is requirement to accommodate the new changes. It anticipates and embraces change. The requirements are gathered and changed through a feedback loop as the system development process progresses (Stoica *et al*., 2013; Wysocki, 2011). An example of adaptive class is the agile family of methodologies. Despite the above advantages, the adaptive class is not suitable for mission-critical application development.

2.5.3 Manufacturing engineering inspired classification

Systems development involves multiple disciplines to handle the whole systems development process (Wohlin *et al*., 2012). To understand the origin of the lean and agile classes a description of paradigm shifts in engineering discipline is imperative. The paradigms are the mass production, lean manufacturing and agile manufacturing, because systems development methodologies have long been inspired by the engineering discipline in general and manufacturing in particular (Fowler, 2005). Below, each engineering paradigm is linked with the corresponding class of systems development methodologies.

2.5.3.1 Lean class

The mass production manufacturing paradigm with its disciplined and rigid practices inspired the structured class described in 2.5.1.2. When customers demanded customisation as opposed to standardisation, mass production became costly and a shift from mass production to lean manufacturing became necessary. The term "lean" was first coined by Krafcik (Krafcik, 1988) and became well-known after the publication of the book *The Machine that Changed the World* (Womack *et al.*, 2007). In lean manufacturing, the emphasis is on improving performance, identifying and alleviating bottlenecks and reducing waste (Dybâ and Dingsøyr, 2009). In the 1990s, the lean philosophy became popular in most industries including the systems development industry (Petersen, 2011). The lean philosophy views activities that do not add value to the customer as a waste of resources (Ikonen *et al*., 2010) and it aims to improve performance, identify and alleviate bottlenecks, minimise costs, and focus on efficiency (Poppendieck and Poppendieck, 2003).

Lean Software Development (LSD) (Poppendieck and Cusumano, 2012) is an adaptation of the lean manufacturing methodology to systems development. Kanban is another example of an SDM inspired by lean manufacturing methodology (Brechner, 2015). Kanban manages systems development emphasising continuous delivery while not overburdening the systems development team (Brechner, 2015; Ikonen *et al*., 2010). In Kanban, work is organised on a Kanban board. The board has states in which every work item passes through as they are pulled along through the in progress state, the testing state, the ready for release state, and the released state. The flow of items presents a clear view of the state of progress and the bottlenecks in the development process can be identified (Ikonen *et al*., 2010). One of the limitations of lean development was its responsiveness to change and the trade-off between cost savings and the quality of the systems development product.

2.5.3.2 Agile class

Lean approach primarily focused on cost-effectiveness while the changing environment demanded responsiveness and speed (Poppendieck and Cusumano, 2012). The lean approach implied high productivity and quality, but did not necessarily imply responsiveness. The lean systems development methodologies compromised responsiveness over cost-efficiencies (Poppendieck and Cusumano, 2012). Change has always been there in industry, but during the 1990s it was more pronounced due to the need to respond to unique customer requirements (Poppendieck and Cusumano, 2012). To maintain competitive advantage, organisations had to consider innovation, fitness for purpose, volume flexibility, variety, extreme customisation and more importantly rapid responsiveness (Poppendieck and Cusumano, 2012). Agility as a concept in engineering was introduced by a USA government sponsored research at Iacocca Institute, Lehigh University, in 1991 (Nagel and Dove,1991). The study identified important practices in various aspects of manufacturing to address the limitations of lean manufacturing. Agile class of systems development methodologies is characterised by rapid response to customer demands, delivering value to customers, being ready for change, valuing human knowledge and skills (Poppendieck and Cusumano, 2012). Agility was based not only on responsiveness and flexibility, but also the cost and quality of goods and services for which customers were prepared.

During the methodology era, many systems development methodologies, SDM hybrids, and SDM variants were developed and implemented in search of what Brooks (1987) refers to as the “silver bullet”. The initiatives attempted to address critiques levelled against systems development methodologies (Hardy *et al*., 1995). At the end of the methodology era, the concept of SDM was slowly losing credibility (Fowler and Highsmith 2001). The agile class of systems development methodologies emerged claiming to solve the problems that existed in the systems development field (Conboy and Fitzgerald, 2010). The agile class is characterised by flexibility, velocity, leanness, learning and response to change. Qumer and Henderson-Sellers (2006) define agility as the ability to accommodate changes (expected or not) in a dynamic environment being simple, economic and having quality in a short iteration strategy applying previous knowledge and experience. Conboy and Fitzgerald (2010) postulate that an agile SDM should embrace changing requirements, should be flexible and be customisable to the needs of a systems development project situation. The agile class assumes that users and the developers do not have full understanding of customer requirements at the outset of a project (Ionel, 2009). Unlike the lean class, the agile class expects, accommodates, embraces and manages change. The agile class views requirements as unpredictable and dynamic and as only being fully decodable during the systems development project (Highsmith and Cockburn, 2001; Fowler, 2005; Leau *et al*., 2012).

The agile class takes advantage of the concepts from the Participatory Design (Kuhn and Muller, 1993) which focus on user participation in systems development and the Soft Systems Methodology (Checkland, 1999) that consider user activity systems as essential and emergent in nature. The agile class of systems development methodologies are iterative and adaptive (Stoica *et al*., 2013) and when assumption fails to hold, the agile SDM adapts to the situation. In other words, the agile class goes through adaption as a learning and alignment strategy to cope with situation dynamics. It is sometimes referred to as people-centric and uses both explicit and tacit knowledge of the team members instead of diffusing information through heavy documentation (Sheffield and Lemétayer, 2013). Requirements are assumed to be emergent, rapidly changing, unknown at the beginning but discovered during the systems development project.

The assumptions of the agile class are articulated by the values and principles of the Agile Manifesto (Hohl *et al*., 2018; Beck *et al*., 2001). The Agile Manifesto was an outcome of the Agile Alliance deliberations in 2001 (Hohl *et al*., 2018; Beck *et al*., 2001). The Agile Manifesto outlined four basic tenets of agile systems development capturing the agile systems development philosophy. According to Hohl *et al*.(2018) and Beck *et al*.(2001), the agile SDM class emphasizes:

- “Individuals and interactions **over** processes and tools.”
- *“Working software* **over** comprehensive documentation.”
- *“Customer collaboration* **over** contract negotiation.”
- *“Responding to change* **over** following a plan.”

The items on the left of the word **over** are more valuable than those on the right. The items on the right represent some key characteristics of the plan-driven class of systems development methodologies, while the items on the left are a summary of the key characteristics of the agile class of systems development methodologies.

In addition, the Agile Alliance (Beck *et al.*, 2016) compiled a set of 12 principles to guide the Agile Manifesto as listed in Table 2.3 below.

Table 2-3: Agile principles

NO.	PRINCIPLE
1	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4	Business people and developers must work together daily throughout the project.
5	Build projects around motivated individuals. Give them the environment and support their needs, and trust them to get the job done.
6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7	Working software is the primary measure of progress.
8	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9	Continuous attention to technical excellence and good design enhance agility.
10	Simplicity--the art of maximizing the amount of work not done--is essential.
11	The best architectures, requirements, and designs emerge from self-organizing teams.
12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

The basic premise of the core values and principles of agile class is constant communication with the customer to allow the development process to evolve. The agile class relies on constant communication and many iterations rather than high levels of planning characterised by heavy documentation. The communication is combined with shorter cycles of development to break down a complex problem into much smaller manageable problems and prioritised tasks. The shorter cycles allow early identification of flaws in requirements assumptions and conceptions to avoid escalation of cost by focusing on a wrong problem. The strength of the agile class is the focus on customer needs (Barlow *et al*., 2011) fostering a strong social network among the entire development team, close customer collaboration (Sheffield and Lemétayer, 2013) and the inclusion of actual users in the development team (Wang *et al*., 2012a). The systems development is centred on the team and not on the phases. Responding to change is the plan, and delivery of functionality is frequently the norm.

The Agile Manifesto presents two important concepts in systems development, namely the values and the principles. The values guide the decisions, actions and behaviour of developers, whereas the principles are universal realities based on natural laws. The behaviour of a developer is governed by values whereas the consequences of that behaviour is governed by principles. The Agile Manifesto outlines two values that are user oriented and three that are not. Regarding principles, it outlines eight principles that are user oriented and four principles that are not. This is the reason why the agile SDM class is regarded as people oriented (Cockburn, 2004).

Each specific instance of the agile class focuses on specific values and principles in the Agile Manifesto. There are no standards on how an agile class instance should implement its agile values and principles. The agile SDM class emerged during the post-methodology era and consists of many SDM instances such as Scrum (Schwaber and Beedle, 2002), Extreme Programming (XP) (Beck, 2000), Crystal (Cockburn, 2004), and Adaptive Software Development (ASD) (Highsmith, 2000).

Scrum and XP are the most adopted instances of the agile class (Versionone, 2018). Scrum has been used by Microsoft, IBM, Google and other well established organizations (Versionone, 2018). Due to their popularity in industry (Versionone, 2018; Noruwana and Tanner, 2012), the two instances will be further described to show how they relate to the contingent use of systems development methodologies. The crystal systems development methodologies present a subset of agile methodology class instances that are designed with internal contingent mechanism. Crystal set of methodologies allows the organization to select one of the methodologies as the appropriate one for a systems development project (Abrahamsson *et al.,* 2017). When crystal class is used, the SDM is selected based on the systems development project size and how critical the systems development project is. Each of the methodologies on the family has a colour code to indicate how heavy they are (Abrahamsson *et al*., 2017). This study focuses on the crystal family because it relates to the contingent use of systems development methodologies.

2.5.3.2.1 Scrum SDM

The name Scrum is derived from a rugby game where a team attempts to move against its opponent as a unit passing the ball back and forth thrusting and adjusting to exploit any perceived weakness of the opponent in a self-organising manner. Scrum is based on the concept that systems development is not a defined process, but an empirical process with complex input/output transformations that may or may not be repeated under differing circumstances. Scrum explicitly recognizes that user requirements often change and that systems development cannot explicitly be planned. Scrum expects change in requirements throughout the systems development project therefore advocates a planning from iteration to iteration (Schwaber and Beedle, 2002). The iterations are presented in the form of fixed iterative cycles referred to as sprints to deliver working system (Lacey, 2012). The Scrum process is initiated when the user requirements are converted into a product backlog (Schwaber and Sutherland, 2017). The product backlog is a prioritised set of user stories (Lacey, 2012).

Scrum consists of three components, the roles, the key artefacts and the key meetings. The core roles consist of the product owner, the scrum master and the development team. The key artefacts consist of product backlog, sprint goal, sprint backlog. The key meetings include sprint planning, daily scrum and sprint review.

2.5.3.2.1.1 Scrum SDM roles

Scrum assigns responsibilities associated with specific roles to the members of the development team. Some of the main roles are:

- Product Owner

The product owner is the central point of product leadership. The product owner is an authority responsible for deciding which features and functionality to build and the corresponding priorities of each feature. It is the product owner's responsibility to maintain and communicate a clear vision of what the team is striving to achieve. The product backlog is managed by the product owner (Schwaber and Sutherland, 2017; Lacey, 2012).

- Scrum master

The Scrum master is responsible for ensuring Scrum is understood and enacted in an organization. He or she is responsible for the alignment of practices and rules in Scrum. He or she assists the product owner find techniques for effective product backlog management, helping the team understand the need for clear and concise backlog items (Schwaber and Sutherland, 2017; Lacey, 2012). The Scrum master coaches the development team in self-organizing and cross functionality and removes impediments to the development team's progress towards the delivery of sprint goals. The Scrum master practices agility and facilitates Scrum events as requested or needed. One of the key roles of the scrum master is to protect the development team and keep it focused on the important tasks at hand (Schwaber and Sutherland, 2017; Schwaber and Beedle, 2002).

- Development team

The Development Team is responsible for delivering potentially shippable product increments at the end of each Sprint. A development team may consist of 3-9 people with cross-functional skills who do the actual work (analyse, design, develop, test, technical communication, and documentation) and is self-organizing (Schwaber and Sutherland, 2017; Lacey, 2012; Schwaber and Beedle, 2002).

2.5.3.2.1.2 Scrum SDM artefacts

Scrum define several artefacts to provide the frames of reference to the expected activities for the team. Some of the main artefacts are:

- Product backlog

Product backlog is an ordered list of items that represent user requirements. The list is presented as a priority list where the highest value items are at the top. Large items are broken down into smaller manageable tasks. Features in the backlog are described as user stories. A User Story has the format (Cohn, 2004):

As a <type of user>, I want <some goal> so that <some reason>

Stories in the backlog start as an idea. At this stage, the feature is abstract, incomplete and or not actionable (Cohn, 2004). Through requirements gathering discussions, ideas are modelled into smaller user stories that are actionable. Creation of appropriate user stories are based on the following six attributes: independent, negotiable, valuable to users, estimatable, small, and testable (Cohn, 2004). The product backlog is open and editable by anyone, but the product owner is ultimately responsible for ordering the stories on the backlog for the development team. The product backlog contains rough estimates of both business value and development effort. The estimates help the product owner to estimate the timeline and may influence ordering of backlog items (Schwaber and Sutherland, 2017; Cohn, 2004).

- Sprint backlog

The sprint backlog is the list of work the development team must address during the next sprint. The list is derived by selecting stories/features from the top of the product backlog until the development team feels it has enough work to fill the sprint (Schwaber and Sutherland, 2017; Schwaber and Beedle, 2002). This is done by the development team asking: "Can we also do this?" and adding user stories/features to the sprint backlog. The team selects the most important requirements from the product backlog that can be completed in a sprint. The development team uses the experience of the previous sprint velocity (total user story points completed from each of the last sprints user stories) when selecting user stories or features for the new sprint (Schwaber and Sutherland, 2017; Cohn, 2004; Schwaber and Beedle, 2002). The stories or features are broken down into tasks by the development team. Tasks on the sprint backlog are never assigned; rather, tasks are signed up for by the team members as needed during the daily scrum, according to the set priority and the development team member skills (Schwaber and Sutherland, 2017). This promotes self-organization of the development team, and developer buy-in. In most cases, a task board is used to display the state of the tasks of the current sprint, like “to do”, “in progress” and “done” states (Schwaber and Sutherland, 2017; Cohn, 2004).

- Sprint burn down charts

The sprint burn down chart is a publicly displayed chart showing remaining work in the sprint backlog. Updated every day, it gives a quick view of the sprint progress. It also provides quick visualizations for reference and forecast progress (Schwaber and Sutherland, 2017; Lacey, 2012).

2.5.3.2.1.3 Scrum SDM meetings

Scrum has predetermined meetings that are time-boxed to minimise the time taken on meetings. Some of the important scrum meetings are:

- Sprint planning

Work to be done in the sprint is planned at sprint planning. The Scrum team reviews the product backlog and determines the highest priority items that the team can realistically and practically accomplish in the upcoming sprint. Work that is selected to be part of the sprint is called sprint backlog. It contains the stories and tasks that need to be accomplished. Sprint planning is done at the beginning of the sprint cycle (a time-box of one month or less) (Schwaber and Sutherland, 2017). The sprint planning is the time where sprint goal is set. It also prepares the sprint backlog that details the time it will take to do the work, with the entire team. The meeting allows the identification and the communication of the work that is likely to be done during the current sprint (Schwaber and Sutherland, 2017).

- Daily Scrum

Each day during the sprint, a systems development project status meeting occurs. This is called a daily scrum, or the daily stand-up. This meeting has specific guidelines: The meeting starts precisely on time. Everyone is invited but normally only the core roles speak. The meeting length is set (timeboxed) to 15 minutes. The meeting should happen at the same location and same time every day. During the meeting, each team member answers three questions (Schwaber and Sutherland, 2017):

What did I do since yesterday towards meeting the spring goal?
What will I do today towards meeting the sprint goal?
Do I see any impediments/stumbling blocks towards meeting the sprint goal?

It is the role of the Scrum master to facilitate resolution of these impediments, although the resolution occurs outside the Daily Scrum itself to keep it less than 15 minutes long (Schwaber and Sutherland, 2017).

- Sprint review

At the end of the sprint, in sprint review, the product increment is reviewed to inspect and adapt the product backlog if needed. The Scrum team presents the increment to stakeholders, sponsors and customers. The feedback is solicited and used to update the product backlog. After the review, the development team meets to conduct a sprint retrospective in order inspect itself and develop a plan for improvements to be implemented during the next sprint (Schwaber and Sutherland, 2017; Cohn, 2004).

2.5.3.2.2 eXtreme Programming (XP) SDM

eXtreme programming is a light weight software development methodology that was built around rapid iterations and places emphasis on coding development and customer interaction (Mnkandla and Dwolatzky, 2004). According to Beck (2000), XP reduces systems development project risk, improves productivity and responsiveness to changing business requirements throughout the life of a system. XP systems development project starts with a quick analysis of the entire system, and XP programmers continue to make analysis and design decisions throughout development. XP advocates putting a minimal system into production quickly and growing it in whatever directions prove most valuable (Mnkandla and Dwolatzky, 2004). A key assumption of XP is that the cost of change can be held mostly constant over time. This assumption of XP challenges the conventional tenet that the cost of changing a piece of software necessarily raises dramatically over time. Costing of the development is based on a pay as you go principle where developers estimate the value and cost of stories and the customer prioritises which stories to develop and pay only for the stories developed (Mnkandla and Dwolatzky, 2004).

XP consists of five core values and thirteen practices (Loftus and Ratcliffe, 2005). The core values are as follows: communication, simplicity, feedback, and courage. Developers communicate with customers and fellow programmers. Designs are kept simple and clean to get the job done. Early and frequent testing provides feedback, and developers can courageously respond to changing requirements. The XP practices are: the planning game, small releases, metaphor, simple design, test-driven development, refactoring, pair programming, collective ownership, continuous integration, 40-hour week, on-site customer, coding standards, and daily stand up meetings (Loftus and Ratcliffe, 2005).

2.5.3.2.2.1 eXtreme Programming (XP) SDM roles

XP allocates responsibilities associated with specific roles to the members of the development team. Some of the main roles are:

- Coach

The XP coach is responsible for maintaining relevant communication within the team. He/she advises developers when they embark on a less important task to focus on the important tasks (Beck, 2000). He/she monitors progress and makes sure the team is not spending time on complex and complicated features of the system. The coach also refreshes productive communication in case communication is deteriorating among the members of the development team (Beck, 2000).

- Customer

A real customer must sit with the team full time (Beck, 2000). The on-site customer enables an XP team to explore user stories as it needs to and gives direct access to someone who can make key decisions quickly (Beck, 2000). The customer also provides value to the company by contributing to the systems development project, thus reducing systems development project risks. The customer's decision is important as it can even determine the stories to be developed (Beck, 2000).

- Programmers

Two programmers work together taking turns to develop features of a system. The two periodically switch roles between the driver programmer and the navigator programmer. The driver programmer enters code while the navigator programmer critiques it. This is one of the XP practices that increases productivity and produces high quality systems (Loftus and Ratcliffe, 2005; Beck, 2000).

- Development team

Any team member may add to the code at any time. Everybody takes responsibility for the whole system. XP increases individual responsibility and personal power as well as appreciation of skills mix and diversity within the team. Knowledge of the system spreads around the team. The development team has continuous access to a real active customer that is someone who will be using the system or a proxy (Beck, 2000; Loftus and Ratcliffe, 2005).

2.5.3.2.2.2 eXtreme Programming (XP) SDM artefacts

XP define several artefacts as a way of guiding the development team on what is expected. Some of the common artefacts are:

- Small system feature releases

The customer chooses the smallest release that makes the most business sense. In general, every system feature release must be as small as possible. The release should contain the most valuable business requirements. The release is a result of the implementation of the highest priority user stories. The simple system is put into production quickly in iterations of customer requested features of length between one and four weeks (Beck, 2000; Loftus and Ratcliffe, 2005).

- Unit Tests cases

A unit test case is developed by the development team before and after the code is written for each feature of the system (Beck, 2000; Loftus and Ratcliffe, 2005).

2.5.3.2.2.3 eXtreme Programming (XP) SDM meetings

Meetings in XP are also predetermined and time-boxed. Some of the common meetings are:

- Planning game

The first meeting covers the exploration phase of XP methodology where the development team members meet with the customers to define the requirements of the system as “user-stories”. It is not necessary that the initial requirements fully describe the features of the final system, because the methodology is composed of series of short development cycles and the requirements are updated in each one of the iterations. The customer writes clear requirement stories any time he or she feels the need. The requirement stories consist of 2-3 sentences on a card that the user cares about, can be reasonably tested, can be estimated and prioritised. The customer may split or merge stories. Developers estimate how much effort each story will take, and how much effort the team can produce in each time interval (iteration). The development team together with the customer develop an overall plan that outlines the work to be accomplished, the schedule and the cost. When reality overtakes the plan, updates are made so that the plan continues being relevant and appropriate (Beck, 2000).

- Daily Stand-Up

Communication among the entire team is the purpose of the stand-up meeting. Every morning, a stand-up meeting is used to communicate problems, solutions, and promote the team’s shared view of the systems development process. The meeting is organised in a round table format and everyone stands up in a circle to avoid long discussions. Everyone attends, including the customer. The meeting is time boxed to a period of 15 minutes.

2.5.3.2.3 Crystal SDM

Crystal SDM comprises a set of specific instances of the agile methodology class such as Crystal Clear, Crystal Yellow, Crystal Orange, Crystal Orange web, Crystal Red, Crystal Maroon, Crystal Blue, and Crystal Violet (arranged in ascending order of weight). The characteristics of each instance are driven by several factors such as team size, system criticality, and systems development project priorities (Henderson-Sellers *et al.*, 2014; Cockburn, 2004). This Crystal family addresses the realization that each systems development project may require a tailored set of policies, practices, and processes to meet the systems development project’s unique characteristics (Abrahamsson *et al.*, 2017). The fundamental assumption is that there is no system development methodology good enough for all the cases an organization can face when developing software (Conboy and Fitzgerald, 2010). According to Cockburn (2001), each Crystal colour represents a set of methodology instances that can be tailored or adjusted according to the systems development project’s contextual factors.

Systems development projects are categorized according to the criticality of the system being produced and the size of the systems development project. Cockburn (2001) considers the systems development project size as the number of people working in a systems development project and links this number to a colour. The maximum number of people that might have to get involved in a systems development project is regarded as the measure of the systems development project's size. For instance, clear (Crystal Clear) may involve up to 6 people and yellow (Crystal Yellow) up to 20 people.

Cockburn (2001) identifies four levels of systems criticality based on what might be lost because of a failure in the produced system. The systems development project criticality levels are:

i. Comfort (C)
ii. Discretionary Money (D)
iii. Essential Money (E)
iv. Life (L)

Cockburn (2001) developed a matrix to categorise systems development projects along the systems development project size and systems development project criticality. The matrix rows represent the systems development project size (number of people) and matrix columns represent the systems development project criticality. For example, a category C20 systems development project would be a systems development project involving up to 20 people developing a system whose failure may result in loss of comfort. The Crystal Yellow configuration will be selected for the development of the system. The development team(s) selects a base methodology at the start of the systems development project (in the form of a minimal set of working conventions) and stretch-it-to-fit according to the demands of the circumstances.

The core properties of Crystal methodologies are frequent delivery, reflective improvement, and communication. The frequent delivery allows the sponsors to get critical feedback on the rate of progress of the team. The users get a chance to discover whether their original request was implemented as recommended. The reflective increment involves the identification of what both is and is not working, discuss what might work better, and make those changes in the next iteration. The teams may try, in various forms: pair programming, unit testing, test-driven-development, various levels of customer involvement, and even differing iteration lengths. Communication is key to Crystal methodology. Crystal Clear names an osmotic communication. This implies that information flows into the background where members of the team can hear it, so that they pick up relevant information as though by osmosis. In Crystal, the number of people involved changes the level of communication complexity and influences the weight of the SDM (Cockburn, 2004).

2.5.3.2.3.1 Crystal SDM roles

Crystal define several roles including:

- Sponsor

The sponsor is responsible for formulating the mission statement and trade-off priorities.

- Coordinator

The coordinator, with the help of the team, is responsible for producing the systems development project map, release plan, systems development project status, and iteration plan. He or she facilitates communication in the systems development project.

- Development team

The team is responsible for organising its structure and work conventions. It uses the reflection workshop outcomes to improve its performance. It is composed of programmers, designers, testers and other developer categories.

- Expert user

The expert user provides the user stories and prioritises the system features to be developed.

2.5.3.2.3.2 Crystal SDM artefacts

Some of the main artefacts created by Crystal are:

- Systems development map

The systems development project map is a dependency diagram identifying the major work to be done and the dependencies involved. It acts as the preliminary plan for the systems development project.

- Release plan

The Release Plan is the well organised set of deliverables and their estimated dates.

- Systems development project status

The systems development project status is a progress monitoring technique, whereby the daily progress is evaluated in relation to the release plan.

2.5.3.2.3.3 Crystal SDM meetings

Crystal has several meetings that reflect its being an SDM generator other that an SDM itself.

- Reflection Workshops

Reflection workshops are frequently held to monitor and fine tune the process. The team should pause for a short time after each delivery to reflect on its working conventions. In the reflection workshop, the team members discuss what is working well, what needs improvement, and what they are going to do differently during the next iteration.

- Daily Stand-Up

The daily stand-up meeting is a short meeting to inform each other on the systems development project status, progress and problems. The meeting is not used to discuss problems, but to identify problems.

The classification of systems development methodologies based on Agile Manifesto values are described in the following section.

2.5.4 Classification based on Agile Manifesto

The Agile Manifesto values and principles provide a comparative framework through which SDMs can be compared, evaluated, and contrasted. The agile values and principles (assumptions) divide the SDMs into two broad classes; the agile and the plan-driven classes. Comparative studies where analysis and comparison of the plan-driven class and the agile class are performed have dominated studies since the introduction of the Agile Manifesto in 2001 (Versionone, 2018; Standish Group, 2016; Leau *et al*., 2012; Highsmith, 2010; Jiang and Eberlein, 2008). The Agile Manifesto values are restated as assertions below:

- Focus should be on individuals and interactions rather than on system development processes and tools.
- Working software is more important than comprehensive documentation of the systems development project.
- Customer collaboration is more important than contract negotiation.
- The system development process should respond to change rather than rigidly following a plan.

These assertions create an overarching theoretical framework on which systems development methodologies are classified either as plan-driven or agile. This classification makes it easier to compare SDMs in terms of their similarities, differences, strengths, and weaknesses. The two classes differ with respect to their underlying assumptions on systems development project scope. The plan-driven class puts more emphasis on freezing the scope of the systems development project, while the agile class emphasises cost, schedule and quality frozen and the scope regarded as variable (Collyer and Warren, 2009). The two SDM classes have been studied using several dimensions such as systems development process model (Leffingwell, 2011), requirements dynamics (Boehm and Turner, 2004), systems development project type (Dybå and Dingsøyr, 2009), addressing change (Boehm and Turner, 2003) and organization culture (Gruver and Mouser, 2015; Iivari and Iivari, 2011; Nerur *et al*., 2005). Both plan-driven class and the agile class have their strengths and limitations (Mirza and Datta, 2019; Dhurka, 2015; Henderson-Sellers *et al*., 2014; Janes and Succi, 2012) compared to different systems development project types and situations. The selection of an instance from a systems development class should be based on both the systems development project type, project size, domain and other organisation factors (Henderson-Sellers *et al*., 2014). SDM instances from either class should be adopted and adapted to the systems development project situation and not the other way around.

The proponents of the agile class of systems development methodologies emphasise certain aspects of the systems development process and this complements the plan-driven class of systems development methodologies. This does not necessarily prove its superiority over the other class (Gill *et al*., 2018; Isaias and Issa, 2015; Rahmanian, 2014). In general, some aspects of a systems development project are addressed by instances from the agile class, while others are addressed by the instances from the plan-driven class. Instead of viewing the two classes from an adversarial perspective, Turk *et al.* (2014) and (Nerur, 2005) classify the two classes as extreme points along a spectrum depending on the degree of agility or predictive level of the SDM. JCSE (2015) and Dwolatzky (2008) demonstrate that despite being viewed as two extremes, agile SDM class and CMMI (instance of a plan-driven class) can productively coexist. This research takes a neutral stance to understand the underlying assumptions and see the strengths and the limitations of each of the two SDM classes. Debate is moving away from trying to prove that one SDM class is categorically better than the other (Gill *et al*., 2018; Isaias and Issa, 2015; Rahmanian, 2014; Jiang and Eberlein, 2008). The focus is on finding ways to select a suitable SDM for a systems development project or how to combine systems development methodologies to gain synergies not possible with the application of just one SDM (Leau *et al*., 2012).

2.6 IMPLICATIONS OF SDMS CLASSIFICATION

The classification of systems development methodologies can lead to an understanding of the important characteristics of systems development methodologies. The classification characteristics may assist practitioners and researchers understand the architecture of systems development methodologies and the reasons why some systems development methodologies are likely to be selected instead of others. It could also lead to understanding why some systems development methodologies are adopted and used as they are, or modified or viewed as obsolete.

The systems development methodologies are classified into different groups depending on the objectives of the classification. The classification in most cases provide a comparative framework through which SDM classes can be compared to each other. The SDM classification exposes the underlying philosophical assumptions of each SDM. The underlying philosophical assumptions of

SDMs can be used to compare the strengths and the weaknesses of each class or class instances of SDMs. Therefore, the understanding of the strengths and weaknesses SDM classes can be used to find ways to combine different SDM classes to gain synergies not possible with the application of just one SDM class (Leau *et al*., 2012).

There are some circumstances where each SDM class or class instance may be best fit and some systems development contexts where it might not be appropriate (Conboy and Fitzgerald, 2010; Iivari and Huisman, 2007). This classification addresses the realization that each systems development project may require a tailored SDM, a hybrid methodology or a set of practices, and processes to meet each systems development project's unique characteristics (Abrahamsson *et al*., 2017; Kalus and Kuhrmann, 2013; Conboy and Fitzgerald, 2010; Cockburn, 2004).

The next section discusses the benefits of SDMs.

2.7 BENEFITS OF SYSTEMS DEVELOPMENT METHODOLOGIES

Systems development methodologies have theoretical and practical value in the understanding of the process and product. It provides the ontology (object identification, representation, description, and the relationships and processes involved) to understand entities and entity dynamics. It allows visualisation of important aspects by abstracting the systems development process and product. Systems development methodologies help guide the selection of tools and techniques used in a systems development context. It facilitates the organisation of work and sequencing of activities in well-defined steps. They (systems development methodologies) develop technological frames to align expectations of technology and minimise incongruence in systems development (Huisman and Iivari, 2006). Success is predictable because activities are orderly planned and executed.

Systems development methodologies provide support for improved quality, productivity, and control (Huisman and Iivari, 2006). Systems development methodologies organise the development process and influence the thinking patterns of the user and the developer into a coherent well focused perspective (Huisman and Iivari, 2006). The systems development work is broken down and organised into a comprehensive and manageable structure with clear goals, principles, guidelines and techniques (Eva, 1994). It also allows standardisation of the development process and the learning of the best practices and guarantees quality by facilitating ISO certification (Fitzgerald *et al*., 2002; Middleton and McCollum, 2001). The system development methodology employment acts as quality assurance and creates trust in developers. It may facilitate achievement of ISO certification (Fitzgerald, 1998) or /and attainment of capability maturity model integration (CMMI) level status (Fitzgerald *et al*., 2002). The capability maturity model integration (CMMI) appraises an organisation's profile based on the maturity level of the SDM practice (SEI, 2010). The organisation's profile may be evaluated through the practices advocated by the systems development methodologies employed (Fitzgerald *et al*., 2002).

Systems development methodologies provide a clear division of labour, and roles and the skills required for developing systems in an organisation (Fitzgerald, 1998). Iterative steps do not only improve understanding of the system, but also facilitates development of interpersonal skills amongst actors. Systems development methodologies are knowledge repositories (Iivari and Huisman, 2007; Fitzgerald, 1998) and help in knowledge diffusion (Iivari and Huisman, 2007). The next section presents criticisms levelled against the system development methodologies.

2.8 CRITIQUE AGAINST SDMS

As already noted, there is no agreement on the definition of the concept of systems development methodologies (Wynekoop and Russo, 1995; Huisman and Iivari, 2006). The influential British Computer Society (BCS) Information System Analysis and Design Working Group (Avison and Fitzgerald, 2006) proposed what it considered to be a comprehensive definition of the SDM concept. However, researchers continue to define the concept in their own conceptualisation. For example, Mohan and Ahlemann (2011) synthesised the definition of SDM selectively from three references, Lyytinen (1987), Checkland (2000) and Avison and Fitzgerald (2003). Mohan and Ahlemann (2011, pp. 735) define SDM as a "collection of goal oriented, problem solving methods/techniques governed by a set of normative principles, beliefs, and a multi-step procedure that prescribes what to do and how to do things". This definition differentiates between methodology and method. Notably, many researchers use the concepts methodology and method interchangeably (Harb *et al*., 2015; Barlow *et al.,* 2011). These divergent views continue to present problems in research and practice.

Iivari and Maansaari (1998) discuss the scope and the category of problems associated with system development methodologies. The scope challenge shows the failure of a system development methodology to indicate coverage on the systems development process. The category problem demonstrate the failure to explicitly show the types of system development situations that can be solved by the SDM. The scope and category problems have implications on the role systems development methodologies should play in systems development process and the corresponding performance measures.

Systems development methodologies are based on philosophical assumptions. These assumptions draw certain boundaries that confine the SDM. The boundaries provide the lens for viewing systems development problems. The boundaries defined by the assumptions drive the ontological and epistemological foundations that affect the perception of practical problems. If assumptions are inappropriate, this can lead to inappropriate abstraction which fails to capture the richness of the real problem. In turn, inappropriate abstraction may result in a right solution to a wrong problem. Therefore, these boundaries limit the possible systems development activities, and solutions that are afforded by the SDM. An SDM is essential but not a sufficient condition to achieve a systems development goal. A single SDM may not provide all the necessary communication and the coordination for the systems development process activities (Brinkkemper, 1996). That is why organisations are making contingent use of systems development methodologies. When making contingent use of systems development methodologies, organisations adapt, tailor, change or even develop a new SDM to address the contextual factors of each systems development project.

An approach taken by an SDM influences the normative principles that guide behavioural expectations in activity accomplishment. For example, neglecting user participation can result in failure of a systems development project due to user resistance. Some practitioners tend to keep on using one SDM, perhaps because of challenges in knowledge and skills required to change or combine methodologies. Although some assume that systems development methodologies are applicable to all systems development problem situations (Fitzgerald, 1998), this is not feasible because no SDM can consider all philosophical approaches without conflicting assumptions (Conboy and Fitzgerald, 2010). It is unfortunate that some systems development methodologies are not backed by empirical studies and have not been proven and tested on real world scenarios (Fitzgerald, 1998). No organisation is willing to experiment on new ways until proven effective and efficient; otherwise there can be serious financial, technological and social implications. Organisations are usually biased towards familiarity with certain practices (Naumann and Palvia, 1982). Tthere is usually resistance to change (Chan and Thong, 2009). Each organization has its own unique culture deeply ingrained in in it. The SDM may not be consistent with the prevailing organization culture (Nerur *et al*., 2005).

The contribution of systems development methodologies is perceived differently by each member of the systems development team as their views are shaped by the role or position they hold within the systems (Huisman and Iivari, 2006). Therefore, different roles may lead to different views on how the system should be developed and different priorities and ratings of the same issues.

Systems development methodologies are viewed as standards. It is not easy to change a standard rapidly to match the ever-changing systems development environment. An SDM applied in similar environments may give different outcomes (Carroll, 2003), but if it is to be a standard, it should provide a baseline of best practice shared by the research and practice communities and should deliver consistent results. The next section presents a review of empirical studies on SDM use.

2.9 EMPIRICAL STUDIES ON THE USE OF SDMS

The concept “methodology use” embodies several performance related measures. In this context, it refers to the adoption and commitment to the actual exploitation of the capabilities of the systems development methodologies during systems development. The main outcomes for systems development methodologies are improvements on development process, productivity, and end products quality (Avison and Fitzgerald, 2003).

Conceptual definitions and operational measures are fundamental not only in the provision of frame of reference and conceptual congruency, but also in facilitating comparability of research results from different studies (Dybå, 2005). The operationalization of the abstract construct “methodology use” is a challenge in the comparative study of systems development methodologies. The actual use of systems development methodologies in practice is hard to establish due to conceptual definition inconsistency, and construct operationalization variances. Some researchers measure use as a dichotomous variable where responses from users can either be that they use or they do not use the system (McLean and Sedera, 2010). Other researchers measure use indirectly through a proxy like intention to use, attitude towards use (Hardgrave *et al*., 2003). Venkatesh *et al*. (2003), measured use from tracing events and system logs. The construct use is multi-dimensional and its measurement is problematic (McLean and Sedera, 2010) as researchers operationalise it in different ways. Most researchers have assessed systems development methodologies in the light of tool conceptualisation. The most appropriate concept to following an SDM rather than using it. The challenge on the proper conceptualisation and operationalisation might be the reason why researchers have not vigorously carried out empirical studies targeting systems development use, because the SDM concept is difficult to conceptualise and operationalise. Vijayasarathy and Turk (2012) point out that there is little empirical research on systems development methodology use.

Some studies assume that systems development methodologies are being used (Saeki,1998) while others argue that systems development methodologies are being followed (Huisman and Iivari, 2003). Some studies suggest that systems development methodologies have not been best practice (Carroll, 2003). Bygstad, *et al.* (2008) states that SDM use is a binary variable while Venkatesh *et al.* (2003) and Hardgrave *et al.* (2003) see it as a latent variable. This difference in opinion makes it difficult to objectively compare findings from different studies and establish the extent of systems development use.

Table 2.4 summarises findings from various studies on systems development methodologies from different countries, and different organisations. It is challenging to make general conclusions from these findings due to inconsistencies. For instance, Chatzoglou and Macauly (1996) found that 47% of the respondents never used an SDM, while Hardy *et al.* (1995) found the number to be only 18%. The reason for these disparities may be that the unit of analysis differs within the studies. Findings based on systems development projects as a unit of analysis and those based on organisations as a unit of analysis are difficult to compare. The second anomaly is perhaps the inconsistent use of the concepts methodology, method, technique, and process model. Researchers compare the same constructs, but classify them differently. For example, Eva and Guilford (1996) classify prototyping as a method, while Wynekoop and Russo (1993) consider it to be a process model. The studies are presented in ascending order based on the year the study was conducted. The survey respondents are different in each study which also limits generalizability of the findings.

Table 2-4: Summary of empirical studies of SDM use

STUDY	COUNTRY	RESEARCH METHOD	UNIT OF ANALYSIS	RESPONDENTS	TERMS USED	DEGREE OF USE	NEVER USE SDM	IN-HOUSE SDM	USED C-SDM	ADAPTED THE SDM	USE AS SPECIFIED
Jenkins *et al.* (1984)	USA	Interviews	72 proj	Single: Project leader	SDM	No	(13%)	51%	36%	81% (C-SDM)	
Erlank *et al.* (1991)	South Africa	Survey	52 org	Multiple: Project manager	SDM	No	(48%)	6%	46%		
Hardy *et al.* (1995)	UK	Survey	102 org		Systems development method	No	(18%)	38%	44%	88%	12%
Chatzoglou and Macaulay (1996)	UK	Survey	72 proj	Single: Project manager, Systems analyst, Consultant	SDM	No	(47%)	(14%)	53% *		
Eva and Guilford (1996)	UK	Survey	152 org	Single: IS manager	Systems development method	No	Included in In-house	(24%)	*		17%
Russo *et al.* (1996)	USA	Survey	92 org	Single: IS management Position	SDM	Yes	(20%)	[42%]	80% *	60%	6%
Poo and Chung. (1998)	Singapore	Survey	54 org	Single: SD manager	SDM	No	(30%)	[78%]	69%		
Fitzgerald (1998)	UK	Survey	162 org	Single: Traditional IS role	SDM	No	(60%)	26%	14% *	58%	
Iivari and Maansaari (1998)	Finland	Survey	44 org	Multiple: IS managers	Systems development method	Yes		(21%)	*		
Rahim *et al.* (1998)	Brunei Durassalam	Survey	36 org	Single: IS manager	Systems development method	Yes	(33%)	(39%)	67% *		
Fitzgerald *et al.* (2000)	Ireland	Case Study	1 organisation	Systems Engineers	Systems development method	Yes			Yes (% not specified)	Yes (% not specified)	
Khalifa and Verner (2000)	Australia and Hong Kong	Survey	Not specified	82 Senior Software Developers	Systems development method	Yes					(54.1%)

STUDY	COUNTRY	RESEARCH METHOD	UNIT OF ANALYSIS	RESPONDENTS	TERMS USED	DEGREE OF USE	NEVER USE SDM	IN-HOUSE SDM	USED C-SDM	ADAPTED THE SDM	USE AS SPECIFIED
Carroll (2003)	Australia	Case Study	1 organisation	Systems analyst, Sociologist	SDM					Single case	
Huisman and Iivari (2003)	South Africa	Survey	73organisations	73 project managers and 243 systems developers	SDM	Yes	(14%)			(60%)	
Bygstad *et al.* (2008)	Norway	Survey	78 organisations	General/IT manager	System development method	Yes	(8%)	(24%) [68%]*		(57%)	

* Some respondents indicate more than one method/methodology SDM: SDM
( ) Percentage of all respondents C-SDM: Commercial SDM
[ ] Percentage of respondents that use a method/methodology

2.10 CHAPTER SUMMARY

This chapter presented a historic perspective of systems development methodologies. The discussion showed that there is no agreement on the definition of systems development methodologies. The four main components of an SDM were identified as the systems development approach, the systems development methods, the systems development process model and the systems development techniques. The chapter presented a historical analysis and synthesis of the past, current situation and predictions for future use of the systems development methodologies. The chapter also discussed challenges affecting analysing empirical evidence from various research endeavours on the use SDM. It revealed that these challenges often emanate from conceptual definition inconsistencies, operational definitions and unit of analysis disparities. Notwithstanding challenges in conceptual frames of references of systems development methodologies, the key features that constitute a frame of reference for a definition have been identified. The discussion revealed that systems development methodologies bear both theoretical and practical value in understanding systems development and that theory and practice influence each other. Theory is refined by practice, while practice is guided by theory.

Systems development methodologies were categorised into two classes namely, the plan-driven SDM class and the agile SDM class. Each class defines a set of roles based either on the type of activity (e.g. tester), or systems development artefact (e.g. product owner) or level of specialisation (e.g. Scrum master). Whether organisations depend on experienced or novice systems development teams, there is need to organise work in a formal, coherent consistent and sustainable manner (Leau *et al*., 2012). It is assumed that systems development methodologies are adopted in practice (Versionone, 2018; Saeki, 1998). Systems development methodologies are difficult to compare since they are established upon different assumptions. Foundations that are applicable to some type of systems development projects are not applicable to others (Iivari and Huisman, 2007). These challenges make imperative to carry out an investigation into contingent use of systems development methodologies. In the next chapter a conceptual model of the contingent use of SDMs is developed and presented.

CHAPTER THREE: CONCEPTUAL MODEL AND RESEARCH HYPOTHESES

3.1 INTRODUCTION

This chapter presents the development of a conceptual model for the contingent use of systems development methodologies. The working definition of SDM is presented in this chapter. The outline of the practical appeal and selection SDM selection approaches are presented. The chapter conceptual definition of contingent use of systems development methodologies is proposed in this chapter. The explanation of the rationale behind the contingent use of systems development methodologies is presented. The explanation of the two main approaches to contingent use of systems development methodologies is presented and an outline of their theoretical support for the contingent use is given. Lastly, a conceptual model for the contingent use of systems development methodologies is developed and presented.

3.2 THE INFLUENCE OF CONTEXTUAL STRESSORS ON THE CHOICE OF SDMS

The implementation of systems development methodologies may vary from one organisation to the other, from one system development project to another and across similar systems development project contexts and within the same systems development project context over time (Berente *et al*., 2015; Aitken and Ilango, 2013) as influenced by the contextual stressors.

The search for universally context-setting independent systems development methodologies has been dismissed by researchers and practitioners (Gill *et al*., 2018; Marks *et al.,* 2017; Young *et al*., 2016; Serrador and Pinto, 2015). The relationships between contextual stressors and systems development methodologies are unique for each systems development project (Henderson-Sellers *et al*., 2014). The contextual stressors are dynamic in nature and can be interpreted in different ways perhaps that is the reason why systems development is regarded as a huge problem (Brooks, 1987), difficult and complex process (Clarke and O'Connor, 2015). The same contextual stressor might impose a different influence depending on its configuration and interdependencies with other contextual stressors over the systems development project lifecycle. Systems development practitioners must make an assessment on a wide range of contextual stressors before selecting the most appropriate SDM to adopt any specific systems development project. On the other hand, contextual stressors are not static; they undergo changes during the systems development project. The criticality of contextual stressors evolves over various stages of the systems development project life cycle and the evolution is non-deterministic.

The variation of contextual stressors exerts different strains on how systems are developed. The evolution of systems development methodologies was and is informed by the variations in the contextual stressors configurations (Marks *et al.*, 2017; Avison and Fitzgerald, 2003; Beck *et al.,* 2001; Royce, 1970). The rapid proliferation of systems development methodologies sought to address the variations in contextual stressors configurations. Considering a system functional requirement as an instance of contextual stressors, many scenarios can be identified. A system functional requirement could be stated, implied, anticipated, non-stated, emergent or dynamic. Stable system functional requirement needs a different SDM compared to a dynamic functional requirement. In most cases, a combination of system functional requirement classes may occur at the same time in a single systems development project. The fundamental requirement for an SDM is its fit to the systems development project contextual stressors (Marks *et al.*, 2017). Therefore, to select an SDM the contextual stressors should be assessed, evaluated and prioritised. Lack of knowledge of contextual stressors might limit not only the ability of the systems development practitioner to select the most appropriate SDM for the specific systems development project at hand but also adapt it accordingly (Špundak, 2014). This is exacerbated by the non-availability of a contextual stressor triage framework to assist systems development practitioners in identifying, assessing, evaluating and prioritising the most important contextual stressors for a specific systems development project and select a fit for purpose SDM.

A variety of systems development project contextual stressors thought to be important in systems development has been studied. However, no agreed set of contextual stressors has been generally agreed upon as each system development project is unique. Notwithstanding the nonexistence of two identical systems development projects, researchers consider various dimensions when organising contextual stressors. Clarke and O'Connor (2012) propose a Situational Factors reference framework that organises 44 contextual stressors into 8 categories, namely organisational, personnel, management, business, requirements, technological, application and operations. The 44 contextual factors are further divided into 170 sub-factors. Clarke and O'Connor (2012) provide an important step in identifying systems development contextual stressors. However, they do not provide any analytical framework to map systems development project contextual stressors onto the characteristics of an SDM. Therefore, the Situational Factors reference framework may serve as a checklist of contextual stressors that can be considered by system systems development practitioners. Table 3.2 shows the organisation of the contextual stressor dimensions into 8 themes from Clarke and O'Connor (2012).

Table 3-1: The classification of contextual stressors (Clarke and O'Connor, 2012)

THEME	DIMENSION
Organisational	Size, facilities, structure, stability, management commitment, maturity

THEME	DIMENSION
Personnel	Turnover, team size, culture, experience, cohesion, skill, productivity, disharmony, commitment
Management	Accomplishment, expertise, continuity
Business	External dependences, business drivers, time to market, customer satisfaction, Payment terms, opportunities, potential loss
Requirements	Feasibility, rigidity, standard, changeability
Technological	Emergent, knowledge
Application	Degree of risk, performance, predictability, size, type, complexity, connectivity, component reuse, development phase, development profile, quality
Operations	End-user, prerequisites

Viljoen (2016) categorises contextual stressors into organisational, systems development project and individual levels. Viljoen empirically measured the effects of each dimension in each category. Viljoen's (2016) classification of contextual stressors is presented in Table 3-3.

Table 3-2: The contextual stressors classification (Viljoen, 2016)

CATEGORY	DIMENSION
Organisational	Size, age, culture, uncertainty
Project	Project size, criticality, nature, system future, development time, platform, external interaction, complexity, cost, change management, development team, tools and techniques, legacy support, team size, maintenance
Individual	Gender, age, qualification, experience

In Table 3.4, a summary of contextual stressors dimensions that are commonly found in literature is presented. The contextual stressor dimensions such as organisational culture, systems development project type, systems development project size, team size, requirements dynamics, team experience and quality related issues feature frequently in literature as illustrated in Table 3-4. The design and development of a data generating instrument in Chapter 4 is informed by some of the most common contextual stressors.

Table 3-3: The contextual stressors dimensions

DIMENSION	REFERENCE
Organisational culture	Marks *et al. (*2017), Gruver and Mouser (2015), Sheffield and Lemétayer (2013), Dybå *et al*. (2012), Iivari and Iivari (2011), Vavpotič and Vasilecas (2011), Highsmith (2010), Iivari and Huisman (2007), Boehm and Turner (2003)
Organisational size	Viljoen (2016), Dybå *et al*. (2012)
Organisational structure	Clarke and O'Connor (2012)
Organisational age	Viljoen (2016), Dybå *et al*. (2012)
Individual experience	Marks *et al*. (2017), Leau *et al*. (2012), Mohan and Ahlemann (2011), Wysocki (2011), Vavpotič and Vasilecas (2011), Misra *et al*. (2009), Chow and Cao (2008), Boehm and Turner (2003), Huisman and Iivari (2002)

DIMENSION	REFERENCE
Project criticality	Rajagopalan and Mathew (2016), Vavpotič and Vasilecas (2011), Highsmith (2010), Cockburn (2007), Charvat (2003), Lindvall *et al*. (2002), Cockburn (2000)
Project complexity	Rajagopalan and Mathew (2016), Young et al. (2016), Leau *et al*. (2012), Highsmith (2010)
Project type	Marks *et al*. (2017), Clarke and O'Connor (2012), Lee and Xia (2010)
Project size	Young *et al*. (2016), Leau *et al*. (2012), Wysocki (2011), Vavpotič and Vasilecas (2011), Charvat (2003), Lindvall *et al*. (2002)
Requirements uncertainty	Young *et al*. (2016), Viljoen (2016), Rajagopalan and Mathew (2016), Wysocki (2011), Highsmith (2010), Lee and Xia (2010), Boehm and Turner (2003)
Team size	Marks *et al*. (2017), Viljoen (2016), Young *et al*. (2016), Clarke and O'Connor (2012), Leau *et al*. (2012), Vavpotič and Vasilecas (2011), Highsmith (2010), Cockburn (2007), Charvat (2003), Cockburn (2000)
Compliance	Highsmith (2010)
Team expertise	Marks *et al.* (2017), Young *et al*. (2016), Leau *et al*. (2012), Lindvall *et al*. (2002)
Team cohesion	Clarke and O'Connor (2012)
Productivity	Clarke and O'Connor (2012)
Commitment	Clarke and O'Connor (2012)
Disharmony	Clarke and O'Connor (2012)
Customer involvement	Vavpotič and Vasilecas (2011), Wysocki (2011), Misra *et al.* (2009), Chow *et al*. (2008)

In a survey of 100 organisations, Mahanti *et al.* (2012) identified contextual stressors affecting the choice of a system development methodology and determined the importance of these contextual stressors in the selection of systems development methodologies. The identified contextual stressors, arranged in order of priority, were project type, project size, project duration, project complexity, level and type of expected risk, clarity of user requirements, application domain, customer involvement, systems development practitioner experience, team size, interfaces, tools and technology, product versions and the reliability required (Mahanti *et al.,* 2012).

The identification of contextual stressors, the dependencies and the interdependencies are central in selecting the most appropriate SDM for a systems development project (Maruping *et al.,* 2009). An example of a contextual stressor could be a candidate system functional requirement. A candidate system functional requirement goes through a requirement triage process, where it is assessed, evaluated and prioritised to be developed or discarded. A candidate system functional requirement might be discarded and might be discovered later that a prioritised system functional requirement is dependent on that discarded system functional requirement. Systems development, therefore, is a complex system that may result in an emergent behaviour. The explosion of systems development methodologies attempts to incorporate new ideas and theories into the development process and management in response to ever changing systems development contextual stressors (Viljoen, 2016; Young *et al*., 2016; Rajagopalan and Mathew, 2016; Henderson-Sellers *et al*., 2014; Wysocki, 2011; Highsmith, 2010; Lee and Xia, 2010; Boehm and Turner, 2003; Avison and Fitzgerald, 2003). The design of a standardized system development methodology capable of addressing all system development project situations has been discarded (Young *et al*., 2016; Rajagopalan and Mathew, 2016; Diebold *et al*., 2015; Henderson-Sellers *et al*., 2014). The effort to create universal systems development methodologies resulted in the creation of systems development methodologies or systems development components that are pre-contextualised for domains not necessarily specific situations (Burns and Deek, 2011; Conboy and Fitzgerald, 2010). The adopting unit may select and adapt the original pre-contextualised SDM artefact on a case by case basis and at any point during the systems development project life cycle. (Diebold *et al*., 2015; Komus, 2014; Conboy and Fitzgerald, 2010).

The rapid proliferation of systems development methodologies led to an SDM selection challenge (Jeners *et al.*, 2013). The selection of an SDM from a wide range of available systems development methodologies is discussed later in this chapter.

The working definition of an SDM for this research is provided in the next subsection.

3.2.1 Working definition of systems development methodology for the study

There has been a considerable effort to arrive at an authoritative definition for SDM (Young *et al.,* 2016; Viljoen, 2016; Conger, 2013; Valacich *et al.,* 2012; Chan and Thong, 2009; Geambasu *et al*., 2011; Vavpotič and Vasilecas, 2011; Iivari and Huisman, 2007; Huisman and Iivari, 2006; Avison and Fitzgerald, 2006; Iivari *et al*., 2001; Iivari and Maansaari, 1998) as presented in Chapter 2. However, researchers continue to define the concept in different ways (Viljoen, 2016; Huisman and Iivari, 2006). However, not every researcher is proposing their own SDM definition as evidenced by Vavpotič and Hovelja (2012) and Geambasu *et al*. (2011) who both adopted the definition proposed by Avison and Fitzgerald (2006).

The approach taken by Huisman and Iivari (2006) and Viljoen (2016) covers most of the definitions. Huisman and Iivari's (2006) definition is based on the seminal breakthrough on the dynamic classification of systems development methodologies presented by Iivari *et al.* (2001). The definition outlines a perspective based on the SDM components not limited to instances of SDM classes. The definition presents a unique approach that is neither biased towards the plan-driven class of SDM nor the agile class of SDM perspectives. It provides a systematic way of understanding, comparing and evaluating systems development methodologies as presented in Chapter 2.

Based on Huisman and Iivari (2006) the following definition was formulated:

A system development methodology is a dynamic framework for developing systems consisting of the systems development approach, the systems development method(s), the systems development process model(s) and the systems development technique(s) guided and uniquely shaped by a philosophy.

This is the working definition for this study.

The theoretical appeal of SDMs is outlined in the next subsection.

3.2.2 Theoretical and practical appeal of SDMs

Different systems development methodologies produce different types of systems depending on the articulation of both the set of different philosophical assumptions and the different intensities of philosophical dimensions in which the methodology is grounded. Each SDM addresses a specific set of systems development contextual stressors (Viljoen, 2016; Clarke and O'Connor, 2015; Barlow *et al*., 2011).

The systems development methodologies result in the following benefits:

- Address the complexity, size, and other contextual stressors associated with a systems development project (Berente *et al*., 2015).
- Effective and efficient response to contextual stressor dynamics (Clarke and O'Connor, 2015; Lee and Xia, 2010).
- Break down work into a comprehensive and manageable structure with clear goals, principles, guidelines, techniques and tools (Viljoen, 2016; Conger, 2013; Hardgrave, *et al.,* 2003; Iivari, *et al*., 2000; Fitzgerald, 1998).
- Provide support for improved quality, productivity, and control (Chan and Thong, 2009; Vijayasarathy and Turk, 2008; Huisman and Iivari, 2006; Iivari, *et al*., 2000).
- Improved predictability of schedule, cost, and quality (Vijayasarathy and Turk, 2008).
- They are knowledge repositories (Iivari and Huisman, 2007) and help in knowledge management, retention and diffusion (Vijayasarathy and Turk, 2008).
- Organise the development process and influence the thinking patterns of the user and the systems development practitioner into a coherent well focused perspective (Huisman and Iivari, 2006).

- Develop technological frames to align expectations of technology and minimise incongruence, in systems development (Huisman and Iivari, 2006).
- Permit standardisation of the development process and the learning of the best practices that guarantees quality by facilitating certification (PMI, 2013; Middleton and McCollum, 2001).
- Provide a clear division of labour, roles and the skills required (Fitzgerald, 1998).
- Reduce risks in systems development process and facilitate skills transfer and mobility (Conger, 2013; Vijayasarathy and Turk, 2008; Fitzgerald, 1998; Jayaratna, 1994).
- Provide the ontology (object identification, representation, description, and the relationships and processes involved) to understand entities and entity dynamics (Conger, 2013; Huisman and Iivari, 2006; Hardgrave, *et al.,* 2003).

The systems development methodologies offer several benefits. However, these benefits are not all found in a single instance of an SDM (Rahmanian, 2014; Aitken and Ilango, 2013; West, 2011). The design of each SDM is grounded on a specific philosophical view, applicable to some specific systems development contexts. It may be assumed that users of SDMs are familiar with systems development contexts characterising the system to be developed. The knowledge of the context influences the selection of a suitable SDM for that particular systems development project.

The fact that a system development methodology should be selected indicates that the benefits may differ from one SDM to another. When there is a short systems development market window for example, agile SDM instances may be the most appropriate. However, when the system criticality is core, the most appropriate would be a plan-driven SDM. However, if the two development constraints are combined in a single systems development project, each one of the two systems development methodologies would not be appropriate. Then the selection is no longer a straightforward problem. The SDM users would have to search for the existence of the methodology that may address the requirements of the systems development project. Each systems development project is unique in terms of its development contextual stressors combination. The selection of an SDM for a specific systems development project involves trade-off decisions made by the development team after careful assessment of the contextual stressors combinations and interdependencies (Lee and Xia, 2010). An SDM is expected to fit into the contextual stressors including the various levels in which an SDM may be selected. The various levels in which the SDM can be selected, adopted and used are discussed in Section 3.7. The action taken to adapt, tailor, adjust, modify, change, discard, or create a new SDM (Henderson-Sellers *et al*., 2014; Henderson-Sellers Ralyté, 2010) to fit the systems development project circumstances is the contingent use of systems development methodologies. The concept of contingent use of systems development methodologies is described in the next section.

3.3 THE CONTINGENT USE OF SDMS

Alter (2017) defines conceptual artefacts as abstract objects that can be developed tested and improved. Alter (2017) notes that there is perhaps too much emphasis on theory building, theory testing, and theory expanding and less focus on other types of conceptual artefacts. He then makes a plea to researchers in the Information Systems discipline to consider and accept conceptual artefacts. Responding to Alter's (2017) call, this study focuses on some conceptual artefact development in the form of definitions and conceptual model development.

Conceptual definitions and operational measures are fundamental not only in the provision of frame of reference and conceptual congruency, but also in facilitating comparability of research results from different studies (Dybå, 2005). The conceptual definition of the contingent use of systems development methodologies creates a frame of reference and facilitates common interpretation at the same time averting ambiguities and incongruence that may arise from different interpretations.

3.3.1 Definition of contingent use of SDMs

Serrador and Pinto (2015) and Burns and Deek (2011) point out that systems development practitioners often change systems development methodologies to fit the specific circumstances of a systems development project. Conboy and Fitzgerald (2010) observe that most systems development projects are unique, and that the choice of an SDM or a variant thereof was contingent to contextual factors. It is not common for an SDM to be used rigidly as per its published version (Serrador and Pinto, 2015; Henderson-Sellers *et al*., 2014; Conboy and Fitzgerald, 2010). In practice, each SDM, even the one regarded as the most appropriate, was tailored (Henderson-Sellers *et al*., 2014) or adapted (Diebold *et al.*, 2015) to suit systems development project context (Viljoen, 2016; Sellers, Henderson-Sellers *et al*., 2014; Conboy and Fitzgerald, 2010; Brinkkemper, 1996; Russo *et al*., 1996). Huisman (2013) defines 'contingency' as a matching of the SDM to the systems development project and its context.

The definitions and the observations made by researchers on the set of activities that are employed to create a fit of systems development methodologies with the systems development contextual stressors (Henderson-Sellers *et al.*, 2014; Huisman, 2013; Börner, 2011; Conboy and Fitzgerald, 2010; Brinkkemper, 1996) led to the formulation of the following definition:

The contingent use of systems development methodologies is the entire set of activities that are performed to achieve an ideal fit between systems development methodologies and the systems development contextual stressors at any given point in time during the development of a system. This may involve tweaking, omitting some parts, adaptation, modification, customisation, creating new systems development methodologies from existing SDM components, combination of systems development methodologies, or creation of alternative systems development methodologies from scratch over the systems development project lifecycle.

This definition will be used throughout this study.

The contingent use of systems development methodologies focuses on the provision of the necessary support structures and processes to deal with contextual stressors. The rationale of the contingent use of SDMs is outlined in the next section.

3.4 THE RATIONALE BEHIND THE CONTINGENT USE OF SDMS

Most systems development methodologies were designed with the universal applicability assumption or at least to address most development situations. Hardly is an SDM applied as per its prescription on a systems development project to achieve an ideal fit (Clarke and O'Connor, 2015; Fitzgerald *et al*., 2003). There are always some adjustments that are done in each context to achieve an ideal fit of the SDM and the contextual stressors (Kalus and Kuhrmann, 2013; Fitzgerald *et al*., 2003). Therefore, the contingent use of systems development methodologies aims to achieving an ideal fit. The contingent use of systems development methodologies is a fundamental principle in the hybrid SDM era. Instead of continuing creating new systems development methodologies in search of a universally applicable system development methodology, the focus shifts to understanding contextual stressors in relation to the existing SDM knowledge base (Henderson-Sellers *et al*., 2014).

The contingent use of systems development methodologies take advantage of the existing knowledge of systems development methodologies. However, it does not completely abandon the creation of new systems development methodologies, but it carefully considers the specificity of systems development project contexts. The specific contextual stressors influence the type of SDM instance to be selected, the way it is tailored and used (Serrador and Pinto, 2015; Kalus and Kuhrmann, 2013). An understanding of the continuous interplay between the SDM and the contextual stressors is a basic criterion for developers to continuously tailor the SDM to achieve an ideal fit. It is envisaged that an ideal fit between an SDM and the system development contextual stressors could improve both the development process and product (Viljoen, 2016;

Huisman and Iivari, 2006; Avison and Fitzgerald, 2003; Hardgrave and Johnson, 2003). The contingent use of SDMs considers the selection of an SDM on a project-by-project basis.

The selection of SDMS as a multi-criterion problem is explained in the next subsection.

3.4.1 The selection of SDMs

The selection of an SDM is a multi-criterion problem due to the availability of many systems development methodologies and their variations and versions (Conger, 2013). Choosing the most suitable SDM for a specific systems development project persists without a straightforward answer. The selection of a suitable SDM is informed by contextual stressors (Viljoen, 2016; Clarke and O'Connor, 2015; Geambasui *et al*., 2011). What exacerbates the selection challenge is that each set of systems development context stressors is different for each system development project (Clarke and O'Connor, 2015). Avison and Taylor (1997) identify classes of systems development problem situations. These classes are: 1) well-structured problem situations with well-defined objectives and contextual stressors, 2) well-structured problem situations with clear objectives but dynamic contextual stressors, 3) unstructured problem situations with unclear objectives, 4) unknown problem situations and dynamic contextual stressors, 5) complex problem situations. Avison and Taylor (1997) give examples of the appropriate systems development methodologies to each class; however, in their conclusion, they indicate that the most likely prevalent problem situation is the fifth category. Several developments have taken place since Avison and Taylor (1997) proposed the selection of systems development methodologies given problem situations.

In 2001, the Agile Manifesto was outlined and it contributed to the handling of dynamic contextual stressors (Judy, 2012). The fluidity of the contextual stressors might be addressed by adaptability and flexibility of the hybrid systems development methodologies, that put together the strengths of individual systems development methodologies (Rahmany 2012). In the current state of systems development, an organisation has four options from which to select an SDM for a particular systems development project. These options are: selecting from the traditional SDM class, an agile SDM class, create a hybrid SDM or create a new SDM that is fit for purpose. However, the selection of an SDM is not a trivial matter, because of contextual stressors variations.

The selection of an SDM amid the contextual stressors variations can be conceptualised using the Cynefin framework. The Cynefin framework originated from the Management Sciences discipline and is applicable to Information Systems discipline (Vakoc and Buchalcevova, 2017). The Cynefin framework allows the systems development team to make decisions based on what to continue doing, what to start doing or what to stop doing (Vakoc and Buchalcevova, 2017; Snowden and Boone, 2007). Therefore, systems development practices may be continued, adopted or discarded and new systems development practices created depending on their perceived appropriateness to the systems development project context settings (Vakoc and Buchalcevova, 2017; Snowden and Boone, 2007).

The Cynefin framework classifies problem contexts into four important context settings: the simple, the complicated, the complex and the chaotic (Snowden and Boone, 2007). The simple context settings are characterised by stability and clear cause and effect relationships (Vakoc and Buchalcevova, 2017). The constraints are so well structured to be easily predictable. In such context settings, the decision making involves three stages, understanding the contextual stressors configuration, categorising the contextual stressors as deterministic and providing a response based on best practices (Snowden and Boone, 2007). In such predictable systems development project context settings, plan-driven systems development methodologies would be the most appropriate choice to deal with the contextual stressors (Snowden and Boone, 2007).

The complicated context settings entail multiple promising alternatives with clear objectives. The complicated context settings have predictable contextual stressors, but require expertise to understand. The decision making in such context settings entails analysing the contextual stressors and responding by selecting good practices that can be adapted. In the complicated context setting, a traditional plan-driven system development methodology with the view to adapt and adjust it accordingly as the context demands, would be an appropriate candidate for selection (Burns and Deek, 2011; Conboy and Fitzgerald, 2010).

The complex context settings are characterised by unpredictable contextual stressors and unclear objectives that make it difficult to establish complete requirements. Emerging patterns can be observed but not predicted. As the systems development processes unfold, systems development practitioners uncover the real relationships between and within the contextual stressors and the associated emergent properties. The decision making in such a context setting may involve probing the contextual stressors based on a typically prototyping approach and emergent practices. In that context, a hybrid of agile SDM class and traditional plan-driven methodology class may be the option (Diebold *et al*., 2015).

The chaos domain entails no clear relationships between cause and effect. The creation of a new SDM to deal with such contextual setting would be the option. The novel practices are more applicable in such settings. The contextual stressors configuration changes continuously either towards simple or chaotic contexts. Therefore, the systems development practitioners need to diagnose contextual stressors to select or create an SDM that is fit for purpose without being obstructed by perspectives developed through past experiences, training and success (Vakoc and Buchalcevova, 2017).

The Cynefin framework provides the systems development team with decision making guidelines on selecting, adopting, adapting, adjusting, modifying, discarding or creating alternative SDMs depending on the assessment of the systems development project contexts (Vakoc and Buchalcevova, 2017; Snowden and Boone, 2007). Therefore, the strength of the Cynefin framework is that it creates a shared mental model of the systems development project contextual stressors and enables the identification of the appropriate SDMs amid the variations of systems development situations.

The decision to select and adopt systems development methodologies is important, as there is consistency on empirical studies indicating that systems development methodologies are adopted in practice (Young, 2016; Hardgrave and Johnson, 2003; Saeki, 1998; Howard *et al.*, 1999). Leau *et al*. (2012) indicates that the systems development methodologies provide a relevant frame of reference to both the novice and the knowledgeable and experienced systems development practitioner in each systems development problem context. Within an organisation, Cockburn (2004) found that there was a higher propensity for inexperienced systems development practitioners to adopt an SDM more rigorously. Cockburn (2004) concludes that the experience levels of systems development practitioners show some systems development methodologies use patterns. At novice systems development practitioner stage, Cockburn (2004) stated that an SDM is generally taken axiomatically. Cockburn (2004) continues to observe that at intermediate stage, systems development practitioners gained experience and started to assess its suitability to context based on its capability as perceived by the systems development practitioner.

Finally, Cockburn (2004) found experienced systems development practitioners tended to rely mostly on their experience, knowledge, expertise and the bias caused by tacit knowledge. They used components of an SDM as they found them fit in the development and management process. An SDM component can be used independently to address an aspect of a problem situation and/or, it can be combined with other components without causing internal inconsistency (Komus, 2014).

Studies also reveal that systems development methodologies are neither used in their entirety nor rigorously as their designers proposed (Wang *et al*., 2012a; Fitzgerald, 1996). This is in line with the trends in the hybrid SDM era. Empirical studies reveal that both researchers and practitioners have passed the milestone on whether systems development methodologies are used or not (Dima and Maassen, 2018; Schlauderer *et al*., 2015; Russo *et al*., 2013; Middleton and Senapathi, 2011; Vijayasarathy and Turk, 2008).

Research also indicates that it would be difficult for a development team to self-organise without any form of guidance from an SDM (Leau *et al*., 2012; Iivari and Huisman, 2007; Carroll, 2003; Fitzgerald, 1998). Furthermore, the existence of systems development methodologies is proof that there is need to improve development control, predictability, standards, productivity and product quality imperatives (Young *et al*., 2016; Fitzgerald, 1996). Research and practice have consistently revealed that there is no one SDM that is ideal fit for all systems development contexts (Young *et al.,* 2016; Viljoen, 2016; Mnkandla and Dwolatzky, 2007; Brinkkemper, 1996; Iivari, 1989). The existing systems development methodologies are a partial solution to the systems development problem. The contingent use of systems development methodologies focuses on the creation, adaptation or selection of systems development methodologies, or components thereof, to achieve an ideal fit to the systems development project situation (Wang *et al*., 2012a).

In the next subsection, some SDM selection approaches are discussed.

3.4.1.1 SDM selection approaches

The systems development methodologies selection has been addressed from different levels of abstraction (Vijoen, 2016). Figure 3-1 illustrates the selection approaches identified by Viljoen (2016).

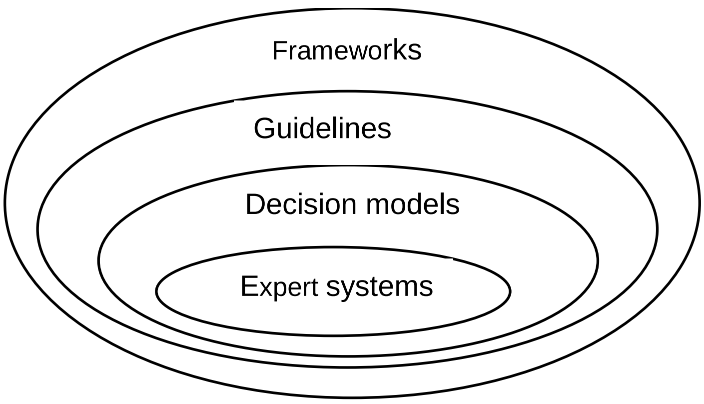


Figure 3-1: Levels of SDM selection granularity

The systems development methodologies selection approaches are frameworks, guidelines, decision models, and expert systems (Viljoen, 2016). The frameworks represent the highest level of selection abstraction. These are applicable at organisational level or by experienced systems development practitioners. The next level of SDM selection abstraction consists of selection guidelines. These standardise the selection process and facilitate consistency in SDM selection. Decision models are the next level after the selection guidelines and these are decision support systems. The finest level of granularity in the selection hierarchy is the expert systems. These are decision support systems or tools or techniques. They allow experimentation and are appropriate for novice systems development practitioners as they present expert knowledge in the selection process.

The selection approaches support the selection of the most appropriate SDM that matches specific systems development context (Viljoen, 2016; Young *et al*., 2016; Vavpotič and Vasilecas, 2012; Mnkandla and Dwolatzky, 2007; Zhu, 2002; Burns and Dennis, 1985).

3.4.1.1.1 Theoretical SDM selection framework

Viljoen (2016) proposes a theoretical SDM selection framework based on the influence of systems development project characteristics at an organisational, systems development project and individual levels. Viljoen identifies the characteristics that influence the selection of SDM selection at organisational level, systems development project level, and individual level.

3.4.1.1.2 Analytical Hierarchy Process (AHP) approach to SDM selection

Harb *et al.* (2015) proposes the application of Analytical Hierarchy Process (AHP) as a multi-criteria decision tool to systematically frame the methodology selection problem. Given a set of systems development methodologies, preference of one from or the other can be established through knowledge solicitation techniques like observations, questionnaires or interviews and subjecting the data collected to a rigorous statistical analysis. AHP converts a multi-criteria decision making process into the solution of an Eigen value problem. Eigen values have their greatest significance in that dynamic problems tend towards a steady state under some mathematical operations.

The appeal of AHP in the selection process is its ability to verify consistency of subjective measures. Ratio scales are derived from paired comparisons and both quantitative and qualitative measures can be scientifically validated. The ability to detect inconsistent judgements makes AHP useful for selecting an SDM. The framework prioritizes and evaluates SDM selection objectives and alternatives. The selected SDM should have higher weights. The framework was proposed for training novice systems development practitioners. The main drawback of this framework is that it is theoretically sound but practically tedious.

3.4.1.1.3 SDM selection criteria

Yusof *et al*. (2011) presents an SDM selection criteria based on complexity and uncertainty, quality criteria and scope of SDM phases as key factors. Yusof *et al*. (2011) propose a formula for calculating the scores for each methodology and these scores are used to rank the methodology's appropriateness. The main drawback on this framework is the determination of SDM scores.

3.4.1.1.4 SDM selection framework

Barlow *et al.* (2011) propose an SDM selection framework based on literature review and observations. The framework considers the varying team sizes, varying loyalty of the team members, and contextual factors interdependencies. The framework rates team loyalty from low to high. If the interdependencies of contextual factors are sequential irrespective of the team size and loyalty, the plan-driven SDM is the most appropriate. On the other hand, if the contextual factors are reciprocal and the team size is large, then the most appropriate SDM is the hybrid of plan-driven and agile (Gill *et al*., 2018; Isaias and Issa, 2015; Rahmanian, 2014). If the contextual factors are reciprocal and the team size is small, then the most suitable methodology would be agile. The drawback of this framework is that it is too abstract to get into the details of SDM characteristics. It just considers plan-driven and agile classes but within these SDM classes, there are variations ideal for specific situations. The framework can work at a classification level recommending either the plan-driven or agile.

3.4.1.1.5 SDM Contingent selection framework

Conboy and Fitzgerald (2010) consider characteristics of the SDM being selected for a systems development project. The first factor is the explicit statement of SDM boundaries. That is how the SDM addresses specific systems development project contextual stressors such as organisational culture, systems development project situation, and individual systems development practitioner experience and level of expertise. This assists the systems development practitioner to know under what conditions a particular SDM should be applied. When the systems development boundaries are clear, it is easier to select an SDM for a system development project. The second factor is the built-in contingency, such that an SDM itself provides the necessary guidance to be adapted, adjusted and modified to fit specific systems development project situations on demand (Conboy and Fitzgerald, 2010). The third factor is the level of independencies of SDM components. If the SDM components are dependent on each other, therefore, the whole methodology should be used to avoid internal inconsistency.

The downside of this approach is that it only considers the agile class of SDMs. Another drawback is that it considers SDM selection as a once-off event, when in practice it should be a continuous process of re-assessment of the appropriateness of the SDM. The main drawback of this framework is that it sees the predetermined conditions that are built-in as contingencies for the SDM. When the systems development project presents conditions out of the contingency variables, it is assumed that it may not achieve the optimal selection of an SDM for the specific systems development project at hand.

3.4.1.1.6 SDM selection model

Yaghini *et al*. (2009) proposes an SDM selection framework based on a multi-faceted approach. Methodologies are first classified as hard or soft and then compared according to six basic features; the philosophy, systems development model, systems development scope, systems development tools, systems development background and participants. The one that approximate the contextual factors is selected. The main drawback of this framework is that it is too theoretical and subjective.

3.4.1.1.7 Toolkit for selecting SDM

Mnkandla and Dwolatzky (2007) develop an expert system to select an SDM from an agile SDM class. The selection criteria are based on characterisation of the system, followed by the characterisation of the context. Two matrices are generated; the systems analysis matrix and the SDM selection matrix. A set of system development methodology components are identified. The identified components (referred to as agile practices) are engineered into an agile methodology. The SDM is then adapted to suit the systems development situation.

3.4.1.1.8 SDM selection matrix

Kettunen and Laanti (2005) propose an SDM selection matrix. The selection matrix is a comprehensive comparative analysis of context and how plan-driven and agile SDMs may match the contextual stressors. The matrix is detailed and is deduced from intensive review of literature. To select an appropriate SDM, the adopting unit reads through columns and rows. The first column attribute is for the contextual stressors. The subsequent column attributes are for systems development methodologies. For example, the first column and second row indicate the context where the systems development project's requirements are unclear. The second row and second column indicate the start of the comparative analysis explaining how each of the systems development methodologies addresses this particular contextual stressor.

Kettunen and Laanti (2005) select four instances of systems development methodologies from the plan-driven class, three from the agile class and consider ad hoc approach in compiling their SDM selection matrix. The drawback of this selection matrix is that it is static and consists of a sample systems development methodologies, without indicating how other instances can be added into the matrix.

3.4.1.1.9 Delphi for SDM selection

Naumann and Palvia (1982) present a selection framework, centred on a quantitative scoring method called Delphi that collaboratively evaluate and recommend the selection of an SDM. The candidate SDM is selected through scores awarded to its functionality relevancy. The drawback of this framework is the subjectivity of the methodology function definition and the concentration on the system development techniques and that it neglects the other methodology components.

3.4.1.2 Summary of the SDM selection approaches

Table 3-5 summarises studies about levels of applicability of SDM selection approaches. Some approaches cover all three; organisational, systems development project and individual levels, while others cover only one or two of these levels. The fit at organisational level appear is the most targeted by researchers. They are fewer studies that cover all three levels. This study provides empirical evidence on SDM selection at all three levels.

Table 3-4: Levels of application of SDM selection approaches

AUTHORS	PARAGRAPH	Level applicable		
		Organisational	*Project*	*Individual*
Viljoen (2016)	3.2.3.1.1	Yes	Yes	Yes
Harb *et al.* (2015)	3.2.3.1.2	Yes	Yes	Yes
Yusof *et al*. (2011)	3.2.3.1.3	Yes	Yes	No
Barlow *et al.* (2011)	3.2.3.1.4	Yes	Yes	No
Conboy and Fitzgerald (2010)	3.2.3.1.5	Yes	Yes	No
Yaghini *et al*. (2009)	3.2.3.1.6	Yes	No	No
Mnkandla and Dwolatzky (2007)	3.2.3.1.7	Yes	Yes	Yes
Kettunen and Laanti (2005)	3.2.3.1.8	Yes	Yes	Yes
Naumann and Palvia (1982)	3.2.3.1.9	Yes	No	No

The shortcomings of these frameworks are discussed in the next subsection.

3.4.1.3 The critique of SDM selection frameworks

The frameworks represent disparate ways of addressing the SDM selection problem. Most of these frameworks do not indicate the actual steps followed to arrive at certain conclusions. For instance, Yusof *et al*. (2011) selected eight systems development methodologies and formulated a formula for calculating scores for each one of the eight candidate SDM. They do not explain how the scores are generated. If the SDM is not one of the selected instances, there is no way to determine its score to compare it with other methodologies. There is weak empirical evidence on the application of these frameworks in actual systems development projects (Conboy and Fitzgerald, 2010). The SDM selection frameworks are theoretical derivations, rather than being empirically evidence based. However, the frameworks provide an insight on one aspect of contingent use of SDMS which is the selection decisions.

3.4.1.4 Rationale behind SDM selection

Despite the challenges of selecting suitable systems development methodologies, it is hoped that among other things, that an appropriate SDM should standardise the development process, organise work and resources and direct appropriately the perception of each member of the development team (Carroll, 2003; Fitzgerald, 1998). Systems development methodologies provide the knowledge bundles encapsulating the best practices serving as a language for translating the systems development problem space into the solution space.

Organisations should select an SDM contingently over the systems development project lifecycle, that is, systems development methodologies that match systems development circumstances at any point in time as the systems development project progresses. Unfortunately, not much research has been done to guide organisations in this regard. The appropriate selection of a system development methodology is purported to reduce the uncertainty in systems development. It is also expected to increase systems development process efficiency, improve quality of developed systems, and deliver systems on schedule and within budgetary constraints. Organisations are aware of the over emphasised crisis in systems development projects (and the implications of systems development project failure on reputation, employee morale, costs and business continuity (Johnson and Mulder, 2016). Some organisations would do all that is possible to achieve an ideal fit of an SDM to specific development contextual stressors. However, the fit of an SDM to the systems development project contextual stressors is not a once-off event. Regardless of the initial fit of the SDM to a set of contextual stressors, the dynamic interplay between the SDM and the systems development contextual stressors requires constant evaluation of SDM's continuous fit over the systems development project life cycle. This study refers to the effort associated with the alignment of a systems development methodology with a specific systems development situation over the systems development project life cycle as the contingent use of systems development methodologies.

The different levels in which contingent use of SDMs may be investigated are discussed in the next section.

3.5 LEVELS AT WHICH THE CONTINGENT USE OF SDMS MAY BE INVESTIGATED

An SDM is a contingent innovation (Viljoen, 2016; Huisman, 2000). Its adoption and use diffuse through different levels internal or external to an organisation. The study abstract the contingent innovation diffusion into three levels; the organisational level, the systems development project level and the individual systems development practitioner level. These levels are discussed in the next three subsections.

3.5.1 Organisational level

An organisation is a social system and can be described by its size, structure, culture and other factors. An organization culture is one of the most important organisational factors that influences communication protocols, members' behaviour, decision-making processes, practices selected and contingency strategies employed (Huisman, 2013; Iivari and Huisman, 2007; Schein, 2009). Organisation culture is an embodiment of assumptions, values, and artefacts (Schein, 2009). Therefore, policies, SDMs and organisation structure are some of the visible artefacts of organisation culture (Schein, 2009). An organization culture is the most difficult to deal with in terms of alignment with SDMs as it influences perception (Gruver and Mouser, 2015; Huisman, 2013; Iivari and Iivari, 2011). Logically, systems development practitioners need to understand the reasons and advantages of change if change is to happen; otherwise they will stick to what they know to lower the risk of failure (Bossini and Fernández, 2013).

An organisation size is also an important factor in the selection of an SDM (Viljoen, 2016; Vijayasarathy and Butler, 2015; Hoppenbrouwers *et al.*, 2011). The size of an organisation affects the way members interact and share information. The size of an organisation may allow specialisation and influence communication protocols. In most cases, the smaller the organisation in terms of the number of its members, the less formal would be the communication strategies and other support structures for the systems development activities (Barlow *et al.*, 2011). The larger the organisation, the more formal would be the support structures for the systems development activities (Barlow *et al.*, 2011). An organisation may mandate the use of an SDM and in this way, influence the diffusion of the contingent innovation in a top-down strategy. An organisation may mandate a strict and rigid application of an SDM or allow some level of deviation from it. It may set up guidelines to select the SDMs. When mandated, the systems development practitioners may start developing positive attitudes towards the SDM. They may begin to adapt, modify it or workaround it to fit the systems development project situation. In this way, some organisations may establish standards for adopting SDMs. In the case of negative attitudes, the organisation would apply its policy on such matters that are considered as resistance to change. Viljoen (2016) states that the organisation level is the first stage of SDM selection, followed by the systems development project and lastly the individual. Not only is the diffusion of SDMs top-down, but it may also be bottom-up.

3.5.2 Systems development project level

A systems development project has a clearly defined start date and an explicitly stated end date (ISO/IEC 12207, 2008). The systems development project is influenced by the contextual stressors imposed by the parent organisation such as team size (Vijayasarathy and Butler, 2015), the number of systems development projects that can run concurrently and responsibility and accountability associated with each team member (Gill *et al*., 2018; Imani *et al*., 2017; Isaias and Issa, 2015). The systems development project level marks a decision point whereby the type of intervention is considered. The type of intervention is influenced by the contextual stressors. An assessment and evaluation of how an SDM may be tailored (Diebold *et al.*, 2015) or usefully combined (Komus, 2014; West, 2011) with another to maintain an ideal fit, is considered at this level. The interdependencies are analysed iteratively scanning for an ideal fit between the SDM and the contextual stressors.

To address the contextual stressors, an SDM is selected and tailored (Vijayasarathy and Butler, 2015). It is at the systems development project level where the requirements and their interdependencies are established. The extent to which an SDM supports the type of system or systems to be developed, determines the adjustment, adaptation or tuning to be done. For instance, dealing with a systems development project involving a legacy system is different from dealing with a systems development project involving a new system. At systems development project level, there are many contextual stressors that should be considered and how those among other things, fit the systems development support structures offered by the SDM. The nature of interdependencies at systems development project level influence the level of complexity (Maruping *et al.,* 2009), which in turn influence the level of tailoring of an SDM (Kalus and Kuhrmann, 2013).

3.5.3 Individual systems development practitioner level

Regarding individual systems development practitioner level, Cockburn (2004) identifies three stages individual systems development practitioner goes through during adoption of a contingent innovation process. The stages are the novice, the intermediate and experienced systems development practitioner. At the first stage, the systems development practitioner takes systems development as it is; in the second stage, the systems development practitioner applies the SDM selectively, and in the last stage, the systems development practitioner relies on experience and expertise. Each systems development practitioner goes through these stages. However, if one takes three systems development practitioners at different stages, one would use an SDM differently. An SDM builds a mental model in each individual irrespective of the individual's experience, values, fears and beliefs (Huisman and Iivari, 2006; Carroll, 2003; Fitzgerald, 1998).

The individual systems development practitioner possesses a range of characteristics such as technical skills, communication skills, knowledge, expertise, and experience, values and beliefs that influences his or her contribution to the systems development project (Marks, *et al*., 2017; Conger, 2013; Leau *et al.,* 2012; McLeod and MacDonell, 2011; Huisman and Iivari, 2006; Cockburn, 2004; Carroll, 2003; Fitzgerald, 1998). Individual systems development practitioners should be familiar with SDM or else there would be need for training or use of other methods that increase diffusion of an innovation. It is systems development practitioners that integrate the SDM into their standard work practice. When systems development practitioners are aware of an SDM that matches the contextual stressors, they may encourage others to adopt it. The systems development practitioners in turn may keep on encouraging each other, facilitating the diffusion process.

The SDM may filter through the three different levels of abstraction, that is, the organisation level, the systems development project level and the individual level or vice versa. The levels act as filters within themselves and across levels. The lowest level has the highest level of granularity, whereas the higher levels have fewer details. The SDM can be viewed as a strategic business approach at organisation level. It can be considered together with policies and other legal and regulatory frameworks. At systems development project level, it may be viewed as a systems development project management framework. At the individual systems development practitioner level, it can be considered as a detailed guide to performing systems development tasks. The characteristics of an SDM and the contextual stressors, determine the adoption and use of an SDM. The architecture of the various levels of abstraction of SDM diffusion is illustrated in Figure 3-2.

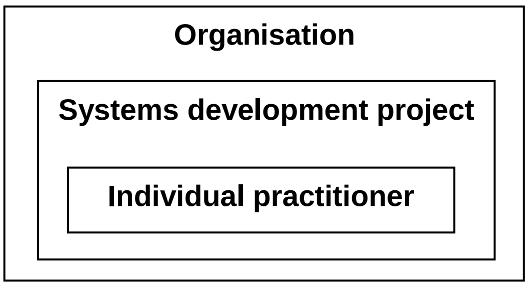


Figure 3-2: Levels of SDM diffusion (Viljoen, 2016)

The ideal fit of the SDMs provides the support framework for developing a system. The ideal fit of SDMs is the focus of contingent use of systems development methodologies. Contingent use of SDMs may be classified into two main approaches and these are presented in the next section.

3.6 THE MAIN APPROACHES TO CONTINGENT USE OF SDMS

To present a frame of reference, Henderson-Sellers *et al. (*2014) identifies three types of SDM parts, namely the fragment, the chunk and the component. The difference between these three types of SDM parts is in the level of abstraction at which each of them is described. A brief description and an example of each type of the systems methodology parts are presented in the three subsections below.

3.6.1 SDM fragment

An SDM fragment is a description of a system development methodology part at the lowest level of abstraction. It is a context independent building block of an SDM that offers the highest level of flexibility. It requires maximum effort to make it applicable in SDM construction due to its atomicity. The SDM fragment may have a producer, process or product focus (Henderson-Sellers *et al*., 2014). An SDM producer-oriented fragment can be exemplified by the concept of a tester. A tester can produce artefacts such as test plans, defect list and other test related documents independent of an SDM. Testing can be done in either traditional or agile family of systems development methodologies. The SDM process-oriented fragment instance could be source code development. Developing a source code is independent of the SDM. Lastly, the system development methodology product-oriented fragment could be exemplified by artefacts such as design templates or user stories.

3.6.2 SDM chunk

A chunk conceptualises an SDM part at a medium level of abstraction. It consists of a process and the output from such a process. The method chunk is a combination of a process-focussed fragment and a product-focussed fragment (Henderson-Sellers *et al.,* 2014). In addition, it includes knowledge about the relevant situation in which the SDM chunk can be used and its origin. An example of a system development methodology chunk is a use case model. The modelling process elicits for use cases and improves the conceptualisation of the use cases. The chunk originates from the UML (OMG, 2010) in Object Oriented Software Engineering (Henderson-Sellers *et al*., 2014).

3.6.3 SDM component

An SDM component is an independent part of an SDM that can be used as a stand alone or in combination with other compatible components (Henderson-Sellers *et al.,* 2014). It consists of a process, artefacts, a notation, and a rationale that guides its domain of applicability. It transforms one or several artefacts into a specific target artefact. The rationale consists of the principles, goals and values behind the SDM component. The rationale keeps the perspective of the systems development practitioners towards the goal. The goals are a testable state of the perceived world towards which effort is directed (Henderson *et al*., 2014). The values on the other hand are tenets viewed as important by the systems development practitioner. A systems development component can be a whole SDM, a process model, a method, or any other mature practice that preserves internal consistency of an SDM. A practical example of an SDM component is a sprint in Scrum SDM. Scrum is one of the most used SDMs in systems development industry (Versionone, 2018). A sprint involves a process system design, coding and testing. A sprint therefore can be regarded as constituting a complete system development iteration. A sprint is a fixed length of time, normally 2 to 4 weeks long. The starting point for sprint planning is the product backlog, which is the list of work to be done on the systems development project. Therefore, product backlog or user stories, coding standards, test cases and sprint backlog are the input artefacts. The output artefact is a working version of a system. Sprint retrospective is an important activity that examines the strength and the weaknesses of the current practice and tries to improve the team's performance (Lacey, 2012). The sprint retrospect addresses one of the agile principles that captures the team's reflection at regular intervals on how to become more effective, then tunes and adjusts its behaviour accordingly. This also illustrates the response to change characteristic of the change-driven SDM.

The sprint delivers a working system at the shortest possible regular intervals (Somerville, 2016; Lacey, 2012). This complies with the agile value on focusing on *working software* over comprehensive documentation (Hohl *et al*., 2018; Beck *et al*., 2001). This goal also addresses the agile principle that states that the working systems should be delivered frequently, with a preference to the shorter timescale (Beck *et al.,* 2016).

The selection phase of the sprint backlog involves the entire systems development project team that works with customers to select features and functionality from the product backlog to be developed during the sprint (Lacey, 2012). This addresses the Agile Manifesto value that prioritises *individuals and interactions* over processes and tools (Hohl *et al*., 2018; Beck *et al*., 2001). The valuing of the *customer collaboration* over contract negotiation (Hohl *et al*., 2018; Beck *et al*., 2001) is in line with the Agile Manifesto principle that business people and systems development practitioners must work together in a systems development project (Beck *et al*., 2016). The teams self-organize to determine the best way to deliver the highest priority features in each sprint (Lacey, 2012). This is based on Agile Manifesto principle that the best architectures, requirements, and designs emerge from self-organizing teams (Beck *et al*., 2016). The systems development team is organised to optimize flexibility, creativity, and productivity (Jiang and Eberlein, 2009).

This study considers the system development methodology components as the building blocks for contingent use of systems development methodologies. The two main approaches to contingent use of systems development methodologies are: 1) the contingency-based SDM selection, and 2) the situational SDM engineering. The contingency-based SDM selection assumes existence of a repository of systems development methodologies. Each SDM is matched with a specific systems development context. On the other hand, the assumption of the situational SDM engineering is that there exists a repository of independent SDM components. Appropriate systems development methodologies can be constructed using these components to match each systems development context. This implies that for the two approaches, there is need for context-specific systems development project characterization which may occur preceding systems development (ex-ante), or during systems development (on-the-fly). The two main approaches to contingent use of systems development methodologies are illustrated in Figure 3-3. The two contingent use approaches are elaborated in the following subsection.

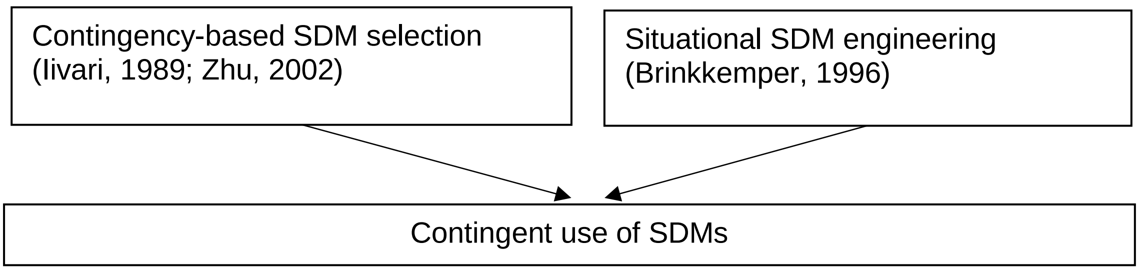


Figure 3-3: Approaches to contingent use of SDMs

3.6.4 The contingency-based SDM selection

Within the context of systems development methodologies, the underlying principle of contingency theory is based on the proposition that there is no one SDM that is necessary and sufficient to address all systems development contexts (Henderson-Sellers *et al.*, 2014; Börner, 2011; Conboy and Fitzgerald, 2010; Fitzgerald *et al.*, 2003). Each SDM works appropriately when applied to a specific situation with specific characteristics (Henderson-Sellers *et al.*, 2014). A single SDM may be necessary but not sufficient to address the requirements and demands of all systems development contexts (Henderson-Sellers *et al.*, 2014; Börner, 2011; Conboy and Fitzgerald, 2010; Burns and Deek, 2011; Rashmi, and Anithashree, 2009; Fitzgerald *et al.*, 2003;

Fitzgerald, 1998). To address this gap in systems development, Mingers and Brocklesby (1997) propose the mixing of methodologies. Rahmanian (2014), Burns and Deek (2011), West (2011), Meso and Jain (2006), and Iivari *et al*. (2001) discovered that systems development methodologies are blended into hybrids (Isaias and Issa, 2015) and adapted.

Iivari *et al*. (2001) observed that there were hundreds of systems development methodologies which differed considerably from each other. However, in practice, it is only possible for an organisation to consider a few systems development methodologies for adoption at a time. To adopt an SDM, an organisation should select it first. The selection of an SDM is a decision-making problem, and according to its nature, the overall objective of the decision requires a comparison of alternative choices. Systems development methodologies could be classified into nominal groups that are not necessarily ordered and therefore could not be ranked from the best to the worst. The choice of suitable criteria to differentiate systems development methodologies is crucial for the selection of the most appropriate SDM given a set of contextual stressors (Clarke and O'Connor, 2015). Iivari *et al.* (2001) came up with a ground-breaking classification that revealed pertinent components of an SDM. Iivari *et al.* (2001) proposed a framework to combine systems development methodologies into hybrid systems development methodologies based on abstraction at the level of SDM approaches. The perspective has a bearing on the assumptions about the systems development contextual stressors emphasised by each class of the systems development methodologies.

The framework provides a unifying structure which allows systems development methodologies to be evaluated and selected in line with the systems development contextual stressors they were designed for (Clarke and O'Connor, 2015; Henderson-Sellers *et al*., 2014; Iivari *et al.*, 2001). This study contends that systems development contextual stressors influence the contingent use of systems development methodologies. The nature of the systems development contextual stressors varies, requiring that the systems development methodologies selected in each case be aligned with the stressors of the development context at any point in time. This implies that the selection of an SDM or methodologies must be grounded in the realities of the systems development contextual stressors (Clarke and O'Connor, 2015; Henderson-Sellers *et al.*, 2014; Zhu, 2002). Zhu (2002) identifies three strands in which contextual stressors could be dealt with in practice and these are: 1) predetermined SDM or methodologies that is or are followed rigidly, 2) predetermined SDM or methodologies that may be modified at each development phase if necessary, 3) dynamic selection of an SDM or methodologies as influenced by the contextual stressors.

The three strands represent the contingency-based SDM selection. These strands are grounded on the dynamics of contextual stressors. The contextual stressors are sometimes referred to as contingency factors or simply contingencies (van Slooten and Hodes, 1996). This studies examines the contingency-based SDM adoption as identified by Zhu (2002).

The first instance is the “contingency at the outset” (Zhu, 2002, pp. 344) that assumes contextual stressors as static and thereby allows the selection of an SDM or methodologies prior the development process. The complete design specification is done up-front and the development of target systems is based on rigidly following the SDM. The selected SDM or methodologies remain invariant throughout the systems development. The SDM and the contingency variables achieve an ideal fit throughout the development process.

The second class is the “contingency with a fixed pattern” (Zhu, 2002, pp. 346) which supports deterministic selection of SDM or methodologies as the development process progresses. The possible future adjustments, variations and reconfiguration of the contextual variables are predictable and follow some known archetype. The framework assumes specific predictable expectations in different phases of a systems development life cycle. Systems development phases form decision points and allow the SDM to be changed at each stage of systems development if necessary (Khalifa and Verner, 2000). The unit of adoption continuously evaluates the relevance of the existing SDM to the systems development needs in each phase of systems development. In other words, the unit of adoption keeps searching for the most suitable SDM for each phase as the development advances in both proactive and reactive way.

The third class is the “contingency along development dynamics” (Zhu, 2002, pp. 348) which posits that the adoption of systems development methodologies, or/and methodology components should be contingent to the dynamics of the evolving systems development context. This approach suggests the existence of a high level of uncertainty in the development process. System development is not predetermined. The selection of the SDM is regarded as a response to particular configurations of contingency variables at a point in time. This approach allows multiple-decision points throughout the systems development process. The adopting unit keeps scanning for the most suitable methodology or ways to adapt or incorporate methodology components at every stage of the development process. The suitability of an SDM is interpreted as a temporal achievement of an ideal match between contingency variables. To maintain this ideal fit, the SDM has to be adapted, tweaked, adjusted, fine-tuned, reconfigured or substituted in response to the contextual factors dynamics.

The rationale behind this approach is that systems development methodologies are modular in nature. The SDM is selected as a framework referred to as base SDM. It is then possible to modify the base SDM. The modification involves the extraction of components from already existing systems development methodologies and tailoring the base SDM (Vijayasarathy and Butler, 2015; Henderson-Sellers *et al*., 2014). This is a way of addressing problem areas that may not be covered by the base SDM at hand. A component may be a process model or a method or a whole SDM (Wistrand and Karlsson, 2004). For instance, it may be the best practice such as the pair programming in eXtreme Programming (XP) or a Sprint in Scrum SDM. The development team assesses the appropriateness of an SDM component to determine whether it can be incorporated to complement the weak aspect of the base SDM (Henderson-Sellers *et al*., 2014).

3.6.5 The situational SDM engineering

Plan-driven systems development methodologies introduced rigidity in the development process. This rigidity and prescription was intended to introduce standards and improve the systems development process (Young *et al*., 2016; Barlow *et al*., 2011; Huisman and Iivari, 2006; Hardgrave, *et al.,* 2003; Middleton and McCollum, 2001; Iivari, *et al.,* 2000; Fitzgerald, 1998). However, rigidity has been criticised as a cause for lack of responsiveness to change (Conboy and Fitzgerald, 2010; Brinkkemper, 1996). Agile SDMs were proposed to introduce flexibility to systems development and address the disadvantages of the plan-driven systems development methodologies. This flexibility received criticism from research and practice for presenting high level of abstraction which led to inconsistencies in interpretation and implementation (Selic, 2009).

Situational SDM engineering tries to strike a balance between rigidity and flexibility. Flexibility on its own leads to an undisciplined way of organising systems development, while rigidity leads to failure to address the needs of a specific systems development context. The assumption of SDM engineering is that each systems development project is unique and it deserves to be treated as such. Instead of having a standard SDM prior to a systems development project, the methodology is constructed based on the systems development project contextual factors. This approach states that SDMs must be constructed using SDM components to address specific development contexts (Carroll, 2003; Fitzgerald, 1998). Therefore, each systems development project context is treated as unique and deserves a unique SDM. The starting point is not a pre-existing SDM, but the systems development contextual stressors. The process involves scanning through the systems development project problem contextual stressors. The systems development project contextual stressors determine the suitable instances of SDM components to be selected. The SDM components may be extracted from a pre-existing repository, theory, best practice or be created from scratch (Henderson-Sellers *et al*., 2014; Henderson-Sellers Ralyté, 2010). These SDM components are practices that constitute the building blocks for the SDM. The components are selected and assembled together to form a situational SDM. The constructed SDM is then applied on a specific systems development project context for which it was created for. During SDM application, learning takes place and the appropriateness of the constructed SDM is evaluated. In-house developed SDMs are developed this way.

3.6.6 The relationship between the two main approaches to contingent use of SDMs

The two approaches in their extreme cases have different starting points and rationale. The situational SDM engineering focuses on in-house construction of SDMs. The goal is to create SDMs on a systems development project-by-project basis based on specific contextual stressors (Henderson-Sellers *et al*., 2014). The SDM is targeted precisely to specific systems development project's contextual stressors. In contrast, the contingency based SDM selection focuses on acquisition of proprietary or free base SDMs. The goal is part from general base SDMs that were designed to encompass a wide spectrum of known systems development project contextual stressors. The SDM whose underlying assumption holds for the systems development project contextual stressors at hand, is selected from a repository of SDMs approved by an organisation. The one that matches the contextual stressors is based on comparing the known and available SDMs with respect to the systems development problem contextual stressors. However, because of the generic nature of the SDMs, it is the best match compared with other available alternative SDMs and not the best fit for the specific contextual stressors. To deal with specific contextual stressors, the selected SDM is then adapted, adjusted, changed, or tailored.

However, at implementation level, the two approaches tend to converge. An SDM may be constructed from scratch, but it is not common to create an SDM which becomes appropriate to a specific target context without going through some modifications (Vijayasarathy and Butler, 2015; Burns and Deek, 2011; Meso and Jain, 2006; Fitzgerald *et al.,* 2003). Therefore, in practice, each constructed SDM goes through evaluation. The evaluation may prove that the SDM is not the best fit in a specific situation due to the emergent properties of contextual stressors and the complex network of dependencies and interdependencies. The situational constructed SDM would then need to be modified to achieve ideal fit to the contextual stressors and the dependencies. This modification becomes a transition from situational SDM engineering to contingency based on SDM selection. The iterative process of creation, evaluation and improvement of SDMs result in the two approaches being complementary to each other. This study focuses on both approaches and contends that the unit of adoption creates or selects an SDM that is fit for purpose at any given point in time. The created or selected SDM is continuously reassessed for the validity of assumptions as the systems development project contextual stressors evolve.

The next section presents the theoretical base to formulate a model that describes the contingent use of system development methodologies.

3.7 THEORETICAL SUPPORT FOR CONTINGENT USE OF SDMS

The conceptual base for this study draws on insights from three theoretical models. These are: the Technology Acceptance Model (TAM), (Venkatesh and Davis, 1996); the Task Technology Fit (TTF), (Goodhue, 1995) and the Diffusion of Innovation (DOI), (Rogers, 2003). Each model focuses on different aspects of innovation adoption in a complementary manner. According to Rogers (2003), an SDM is an innovation. The TAM exploits the psychological belief attributes to explain the adoption determinants (Venkatesh and Davis, 1996). Therefore, TAM brings out the subjective perspective of SDM adoption.

The TTF focuses on the extent to which SDM characteristics fit the unit of adoption task needs (Goodhue, 1995). The matching of the task requirements and the functionality of the SDM influences adoption. The TTF approaches adoption from a rational perspective based on the interaction of the adoption unit with the SDM. The strength of TTF therefore is in the rational characterisation of the SDM and aligning its functionality with the systems development project contextual stressors. The DOI brings about the cognitive, experience and the social influence perspective of adoption (Rogers, 2003). The models address different dimensions of the adoption phenomenon. The differences in these models provide a means to complement each other in predicting and explaining the adoption phenomenon. Therefore, the resulting hybrid model is envisaged to provide a stronger explanatory basis than each model considered individually. More potent predictions and descriptions may be exposed by the lens provided by a hybrid model (Yen *et al*., 2010; Pagani, 2006; Dishaw and Strong, 1999). The TAM, TTF and the DOI models are described in the next three subsections.

3.7.1 Technology Acceptance Model (TAM)

The Technology Acceptance Model (TAM) (Venkatesh and Davis, 1996) is a causal model that was designed to predict and explain adoption. The TAM considers technology as an instance of an innovation; however, it can be applied to contexts that need acceptance and adoption (Yen, *et al.*, 2010). TAM is one of the most influential information systems models and the most commonly employed model (Lim, 2018). TAM is a well-established and a reputable model in the information systems field (Lim, 2018; Moody *et al*., 2010). It is both a parsimonious and theoretically justified model (Venkatesh and Davis, 1996; Davis *et al*., 1989) and evolved from the proposal by Davis (1989), revised by Davis *et al*. (1989) and Davis (1993), and finalised by Venkatesh and Davis (1996). Other versions of TAM are simply conceptual contextualisation to understand contextual peculiarities of contexts settings (Lim, 2018). The TAM is so influential that it has been cited more than 79,000 times on Google Scholar (Lim, 2018).

The fundamental tenets of TAM are; perceived ease of use (PEOU) and perceived usefulness (PU) (Venkatesh and Davis, 1996). This have been, still are and probably will continue to be relevant for understanding contingent innovation adoption in Computer Science and Information systems discipline (Lim, 2018). Perceived ease of use (PEOU) is the adopting unit's perception of the effort needed to adopt and use an innovation. Perceived usefulness (PU) is the adopting unit's perception of the degree to which using an innovation improves their performance. TAM suggests that when the adopting unit encounters a new contingent innovation, perceived usefulness (PU) and perceived ease-of-use (PEOU) influence the decision to adopt it (Venkatesh and Davis, 1996).

These two constructs exploit the influence of perceptions formed through past experiences, success and level of expertise in contingent use of systems development methodologies. According to Venkatesh and Davis (1996) TAM views the decision to accept, or reject a contingent innovation to be related to the adopting unit's perception of the contingent innovation, rather than the innovation itself. The perceptions are related to the adopting unit's knowledge and past experiences (Brown *et al*., 2014). The TAM is a basic powerful behavioural modelling that offers contextual flexibility and can be integrated with other models (Lim, 2018). The TAM is illustrated in Figure 3-4. Therefore, TAM represents the perceptual perspective of the contingent use of systems development methodologies.

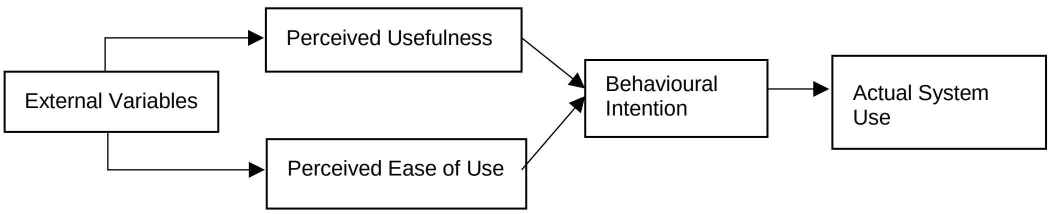


Figure 3-4 The Technology Acceptance Model (Venkatesh and Davis, 1996)

3.7.2 Task Technology Fit (TTF)

According to Goodhue (1995), a Task-technology fit (TTF) is the correspondence between the task requirements and the functionality and features of the contingent innovation. Goodhue (1995) presents the task-technology fit model as illustrated in Figure 3-5.

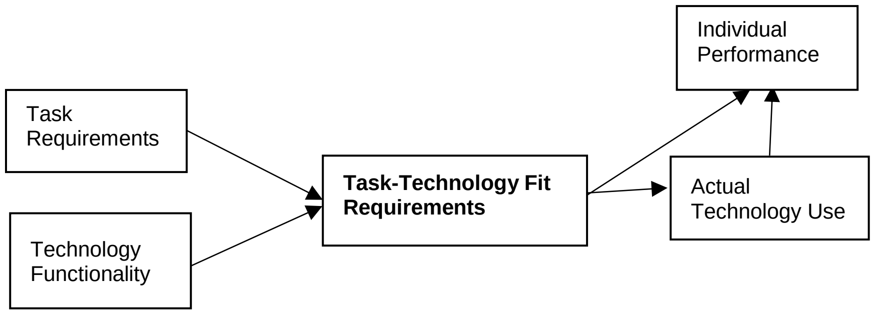


Figure 3-5: Task-technology fit (TTF) (Goodhue, 1995)

The TTF construct encapsulates the fit concept. Van de Ven and Drazin (1985) identified at least three different conceptualisations of fit, the selection approach, the interaction approach, and the systems approach. The fit as a selection approach assumes congruency between the contextual stressors and the structure. The fit as interaction approach considers interaction between context and structure. Lastly, the systems approach considers fit as interaction between contextual stressors and structural stressors. Venkatraman (1989) extended the conceptualisation of fit into six perspectives and these are: the moderation, the mediation, the matching, the covariation, the profile deviation, and the gestalts approaches. Both Van de Ven and Drazin (1985), and Venkatraman (1989) indicates that each fit conceptualisation has implications on data collection methods and the applicable statistical techniques. The fit conceptualisations are not mutual exclusive; however, the dominant fit conceptualisation should be the one explicitly indicated.

In this study, fit is conceptualised as the interaction approach explaining variations in the systems development project outcome from the interaction of the SDM and the systems development project contextual stressors. One of the data analysis techniques supported by this conceptualisation is multiple regression analysis. The fit concept measures the conditions over which a relationship is expected to hold. The SDM can be enacted, adopted to match the systems development contextual stressors. The systems development practitioner therefore, continuously observes the fit between the SDM and the systems development contextual stressors to make appropriate decisions during the systems development project. The fit might be viewed as a continuum from under-fit to over-fit. The continuous fit variable is dynamic and may fall into three regions of relevance. The two extremes regions are under-fit and over-fit regions. The under-fit is the situation whereby the interaction between the contingent innovation and the systems development project contextual stressors is below expectations. In this case, the contingent innovation will need another contingent innovation or a component thereof to complement it (Yu and Yu, 2010). The over-fit on the other extreme provides more interaction than is necessary. This may lead to high cognitive load or steeper learning curves as the need to avoid unnecessary components arises. The ideal fit region provides the highest net benefit of the adoption of a contingent innovation.

The antecedents of TTF are the interactions between the contingent innovation and the contextual stressors. First, they should be a portfolio of contextual stressors to interact with the contingent innovation. These contextual stressors are assumed to have been analysed. The fit of a contingent innovation is expected to be based on its relevance to the contextual stressors. TTF places a greater emphasis on the needs to address the contextual stressors rather than the perceptions on addressing them (Yen *et al*., 2010). Therefore, they are three possible areas in which TTF can be analysed. The gap between: 1) the contextual stressors and the contingent innovation; 2) the contingent innovation and the individual's capability; and 3) the individual's capability and the contextual stressors. The condition where no significant discrepancy exists is the ideal fit. The ideal fit is the basic assumption for adoption of contingent innovation (Rogers, 1995). A significant deviation from the ideal fit may result in a reduction in the TTF. The TTF is perceived by the adopting unit as the extent in which the contingent innovation addresses the specific contextual stressors. Table 3-6 presents the possible conditions that might lead to some discrepancies between the expected and the actual performance of a contingent innovation.

Table 3-5: Fit discrepancy conditions

FIT REGION	CONDITIONS FOR DISCREPANCIES
Under-fit	• Under-fit between the contextual stressors and the contingent innovation.
	• Under-fit between the contingent innovation and the adopting unit's capability.
	• Under-fit between the adopting unit's capability and the contextual stressors.
	• Under-fit between the contextual stressors, the adopting unit's capability, and the functionality and features of contingent innovation.
Over-fit	• Over-fit between the contextual stressors and the contingent innovation.
	• Over-fit between the contingent innovation and the adopting unit's capability.
	• Over-fit between the adopting unit's capability and the contextual stressors.
	• Over-fit between the contextual stressors, the adopting unit's capability, and the functionality and features of contingent innovation.

TTF assumes that there is an ideal fit between the contingent innovation and the contextual stressors or that a fit for purpose exists. The more fit between a contingent innovation and the contextual stressors, the higher the probability for the adoption or continued use of a contingent innovation (Pagani, 2006). The interaction between a contingent innovation and the contextual stressors should provide an ideal fit, otherwise the adopting unit might not continue using it (Yu and Yu, 2010). The adoption is based on the fit for purpose of the contingent innovation and the contextual stressors. The rejection is not categorical, but is accompanied by an explanation of why the contingent innovation is not appropriate for the contextual stressors at hand (Yu and Yu, 2010). Hence, the contextual stressors determine the appropriate contingent innovation to be adopted. In the case of a misfit, the contingent innovation can be adapted and changed.

It is envisaged that an ideal fit between an SDM and the system development contextual stressors could improve both the development process and product (Huisman and Iivari, 2006; Avison and Fitzgerald, 2003; Hardgrave and Johnson, 2003). When there is a misfit, appropriate action may be taken to improve the fit for purpose (Yu and Yu, 2010) and minimise cost escalation.

With regards to TTF, after adopting the contingent innovation, the adopting unit gains direct performance experience with it. The pre-adoption performance expectations are evaluated according to the accumulated performance evidence (Larsen *et al*., 2009). When discrepancies between the expected and actual performance are observed, the adopting unit may disconfirm the early expectations (Brown *et al.,* 2014; Sun, 2013). The experience of the adopting unit influences its future adoption behaviour (Sun, 2013). More experience is related to knowledge accumulation on how to use and why use a specific contingent innovation. In this study, experience is measured through SDMs intensity of use, vertical SDM use, and horizontal SDM use, and the total number of years the adopting unit has been using SDMs. TTF is a relevant determinant at the post-adoption phase as it is possible to evaluate the fit after interacting with the contingent innovation. The choice to adapt a contingent innovation is based on the fit discrepancy conditions (Yu and Yu, 2010; Goodhue and Thompson, 1995). The actual experience of using the contingent innovation exposes the extent of fit for purpose that in turn influences its continued use, or its discontinued use by the adopting unit (Brown *et al*., 2014; Sun, 2013; Goodhue and Thompson, 1995).

In this study, the decision to continue or to discontinue using a contingent innovation is not a once-off event, but includes various intervention strategies. The various intervention strategies are the contingent use of systems development methodologies. The contingent use of systems development methodologies increases the adopting unit's chances of confirmation of the useful practices, while minimising the chances of adopting useless practices. The adopting unit, for instance may mitigate the difference between the anticipated and the actual consequences generated by the SDM through modifying or adapting it. The modification and adaptation done to the SDM create a fit for purpose to the systems development contextual stressors (Clarke and O'Connor, 2015; Venkatesh *et al*., 2012). The TTF considers the requirements and makes a rational decision based on the configuration of the systems development contextual stressors (Brown *et al*., 2014). That is, the context settings are assessed in order to determine whether they can be addressed by best practices, good practices, emergent practices or novel practices (Vakoc and Buchalcevova, 2017). The practices consist of a complete SDM or a hybrid of systems development methodologies (Isaias and Issa, 2015) or systems development components (Rahmanian, 2014).

In Computer Science and Information Systems research, TTF has been used to investigate adoption of SDM chunks (Grossman *et al*., 2005) and post-adoption behaviours among the adopting units (Zhou *et al*., 2016).

The Diffusion of Innovation model is discussed in the next subsection.

3.7.3 Diffusion of Innovation (DOI)

The DOI is a scholarly reputable model that has shown consistent findings (Wang *et al.,* 2012b; Rogers, 2003). Rogers (1995) developed the concept of innovation diffusion as a unifying theory that cuts across disciplines (Wang *et al*., 2012b) In Computer Science and Information systems, it has been applied successfully to study adoption and use of systems development methodologies (Viljoen, 2016; Schlaunder *et al.*, 2015; Bustard *et al.*, 2013; Vijayasarathy and Turk, 2008). Russo *et al.* (2013) used relative advantage and compatibility factors from DOI to study the adoption and use of agile system development methodologies. Bustard *et al*. (2013) applied DOI to study the adoption of agile SDMs. Hardgrave *et al*. (2003) employed the DOI determinants to predict the systems development practitioner's intention to follow an SDM. Huisman and Iivari (2002) used DOI factors to predict the deployment of systems development methodologies by individual systems development practitioners in South Africa.

According to Rogers (2003), innovation is an idea, practice, or object that is perceived as new by an adopting unit, which can be an individual, group or organisation. Rogers (1995) indicates that the consideration of an innovation as new is relative to either the adopting unit and/or the context of use. Rogers (2003) specifically considers an SDM as an example of innovation.

There are two fundamental theoretical contributions made by Rogers (1995) with respect to SDM adoption. The first is the discovery of the innovation-decision process that explains how an innovation is adopted and used. The second is the development of a five factor research model that explains why an innovation may be adopted. The two theoretical contributions are discussed in the following two subsections.

3.7.3.1 Innovation-decision process

Contingent use of SDMs involves a multi-stage and dynamic decision making process that may be modelled to some extent by the innovation-decision process. The innovation-decision process describes the stages an adopting unit goes through from knowledge phase to confirmation of a contingent innovation. The innovation-decision process stages are: the knowledge, persuasion, decision, implementation and confirmation (Rogers, 2003). The stages in the innovation-decision process are illustrated in Figure 3-6. The knowledge stage involves the acquisition of information regarding the existence of a contingent innovation. The persuasion stage entails reflection on the knowledge acquired regarding the contingent innovation in terms of benefits and/or risks. The decision covers cognitive deliberation on whether to adopt or reject the contingent innovation. The implementation stage describes the perceptions post the decision to adopt or reject. The confirmation stage consists of the experiences that may be positive or negative.

3.7.3.2 Diffusion of Innovation model determinants

The model is referred to as the Diffusion of Innovation Model (DOI) and is shown in Figure 3-6.

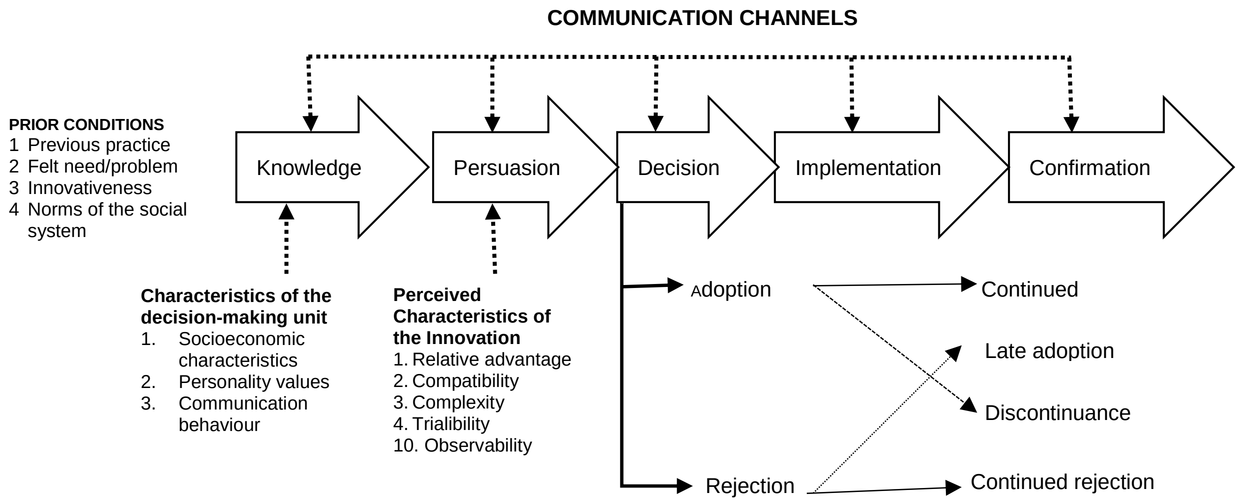


Figure 3-6: The innovation-decision process (Rogers, 1995)

The perceived attributes of a contingent innovation that influence the opinion and beliefs of the adopting unit are compatibility, relative advantage, complexity, trialibility and observability (Rogers, 2003). They are the most significant contributing factors towards the attitude formation as the individual tries to uncover the different facets of a contingent innovation and its applicability to his or her current situation or future situation.

These five most relevant contingent innovation factors explain between 49% and 87% of the variance in contingent innovation adoption (Rogers, 2003). However, with respect to systems development methodologies relative advantage, compatibility, and complexity are the most relevant determinants (Schlauderer and Overhage, 2013; Tung *et al.,* 2008; Tornatzky and Klein, 1982). Within the systems development methodologies context, relative advantage refers to the degree to which the adopted SDM or systems development component addresses the contextual stressors better than any of its precursors (Isaias and Issa, 2015). Compatibility refers to the degree to which the adopted SDM or a systems development component is perceived as being consistent with the existing values, needs, and past experiences of the adopting unit (Bhattacherjee *et al*., 2012; Roger, 2003). Complexity entails the effort and the time the adopting unit would need to use the SDM or a systems development component thereof (Isaias and Issa, 2015).

The critique of the three models is presented in the next section.

3.8 CRITIQUE OF THE MODELS

Despite being prominent conceptual lenses that are often applied in predicting adoption and use of contingent innovations, the models discussed above have been criticised.

3.8.1 TAM critique

Regarding TAM, Turner *et al*. (2010) argues that the perceived ease of use (PEU) and perceived usefulness (PU) are not consistent in the prediction of the actual use. The discrepancy is caused by the deterministic nature of TAM that assumes that the adopting unit's decision to use is determined by intention. It considers that intention results in use. This is a simplistic view to the decision process leading to adoption and use (Turner *et al*., 2010). The adoption does not include explicitly the cost implication, culture, politics, expertise, time, and resources (Brown *et al*., 2010). TAM does not reflect the variety of user task context and constraints (Yen *et al*., 2010) due to its being too parsimonious and this has led to its several extensions to accommodate different situations of adoption (Venkatesh and Bala, 2008; Disaw and Strong, 1999).

3.8.2 TTF critique

Regarding the Task-technology fit (TTF) the challenge is about who should make the assessment of fit (Goodhue and Thompson, 1995). Is it the adopting unit or an expert who should evaluate the fit? The fit is not trivial as it involves three areas of analysis and the human aspect may perceive a misfit due to knowledge level, habits, politics, or resistance to change. The TTF model takes a simplistic view of the decision to adopt a contingent innovation without considering attitude (Yen *et al*., 2010). Therefore, the TTF model does not address social constructs as it assumes the rational perspective of contingent innovation use (Yen *et al*., 2010). Adoption in TTF model is a binary variable, the contingent innovation is either adopted as it is fit or rejected as it is not fit. It does not consider that some contingent innovations may fit some tasks and not others. The contingent innovations therefore may need to be modified or tailored to fit and not to be categorically considered as misfits.

3.8.3 DOI critique

The DOI model (Bui, 2015; Rogers, 2003) takes a rational perspective where the decision taken to adopt a contingent innovation is based on an informed wilful choice of the adopting unit. Therefore, the contingent innovation adoption decision making process is based on the characteristics of the contingent innovation itself and the other relevant information acquired by the adopting unit through communication channels. The DOI does not capture explicitly the mandatory adoption of contingent innovation (Bui, 2015; Mendoza *et al*., 2013). Organisations may make the adoption of a contingent innovation compulsory (Mendoza *et al*., 2013) due to compliance issues and other reasons.

The DOI assumes that the members of a culture interact with each other, without considering the impact of interaction with members from other cultures (Bui, 2015). In systems development field, there are consultancies and other forms of interaction that may not necessarily involve members from the same population. The DOI in its original state does not capture the impact of interaction between different populations. It is logical that members from different populations may not view the positive and negative characteristics of a contingent innovation in the same way. Private sector organisations may not necessarily construe the advantages and the disadvantages of a contingent innovation the same way. In other words, applying the DOI in complex situations is difficult.

The DOI in its original form assumes a single instance of a contingent innovation at a time (Rogers, 2003) that is considered for adoption. However, multiple contingent innovations may be considered and their compatibility not only with the adopting unit, but also between the contingent innovations themselves, evaluated.

The decision to adopt is not sequential, but a complex process that involves iteration and acquisition of information, as well as discounting other information by the adopting unit (Mendoza *et al*., 2013). The DOI is presented as a simplistic linear and sequential process without any form of feedback loops. In systems development, feedback is core and affects the course of systems development as well as the SDM being used. Adoption is a binary variable where an adopting unit can be classified as either having adopted or not adopted an SDM. Instances of partial adoption, such as the adoption of an SDM component are difficult to capture in the original form of the DOI model.

The DOI captures contingent innovation as originating outside the social system, giving an impression that the adopting unit can only adopt and modify and not create a new contingent innovation. In the systems development field, contingent innovations can be created and adopted by the same creators. Therefore, the DOI has many weaknesses which should be noted before it is applied. In this study, these weaknesses are mitigated by using two models that complement each other.

A contingent use of SDMs model is presented in the next section. The model is designed as a hybrid of the three models based on contingent innovation acceptance.

3.9 THE CONTINGENT USE OF SDMS MODEL

The contingent use of SDMs model is designed from three models that are well established and reputable in the information systems field (Moody *et al*., 2010; Goodhue and Thompson, 1995). The proposed contingent use of SDMs model is presented in Figure 3-7. The model consists of three components. These are the contingent innovation, the adopting unit, and the contingent use of innovation process. The TAM provides the perceptions to evaluate the acceptance (or rejection) of the contingent innovation (SDM). The beliefs and perceptions can be influenced by knowledge or past experiences (Bhattacherjee *et al.*, 2008). Therefore, the perceived ease of use and perceived usefulness perceptions at pre-adoption might differ from post-adoption. Hence, perception is dependent on the adopting unit's experience, success (failure), and expertise or compliance issues.

The TTF emphasises the characteristics of contingent innovation with respect to the task to be accomplished. The TTF suggest that adoption of a contingent innovation should be meaningful if the adopted contingent innovation provides a fit for purpose for the task at hand (Goodhue and Thompson, 1995). The TTF complements the DOI by emphasising the alignment of specific needs or tasks with the capability of the contingent innovation. The adopting unit is always scanning the environment for better contingent innovations to improve the way it does things. In the proposed contingent use of systems development model in Figure 3-7, the DOI model is used as a decision mechanism to identify and adopt useful contingent innovations.

In the SDM model, the constructs of the three models are infused together. The perceived usefulness in TAM is related to DOI's relative advantage, while the TAM's ease of use is equivalent to DOI's complexity (Karahanna *et al*., 2006). Ease of use suggests that low cognitive effort is required to use the contingent innovation, whereas complexity implies the opposite. Ease of use is important to all stages of adoption; however, it is more relevant at the pre-adoption stage of a contingent innovation (Karahanna and Straub, 1999). When the experience of the adopting unit increases, perceived ease of use is mediated through perceived usefulness (Karahanna and Straub, 1999).

Compatibility captures the degree of disruption and magnitude of change an individual, or an organisation is likely to experience during the implementation of a contingent innovation. Compatibility from DOI is equivalent to the fit concept in TTF. Compatibility beliefs can be viewed from various fit considerations such as fit with values, fit with prior experience, and fit with existing work practices. The fit may influence the adopting unit to continue using a contingent innovation, while a poor fit might decrease the probability of the adopting unit to continue using a contingent innovation (Goodhue and Thompson, 1995). The fit may be at organisational level, systems development project level and/or at individual level. The fit directly influences usefulness or relative advantage (Tung *et al*., 2008). Perceptions of usefulness or relative advantage of a contingent innovation are a function of the fit between the contingent innovation and the adopting unit's values, prior experience and existing practices, be it at organisational, systems development project or individual levels (Karahanna *et al*., 2006). The relative advantage, ease of use, usefulness and fit are used in the model to investigate the contingent use of SDMs.

An abstract contingent use of SDMs model is illustrated in Figure 3-7. The model consists of three components, the SDMs (Contingent Innovation), the adopting unit, and the contingent use of innovation process (CUOIP). The three components are discussed in the following subsections.

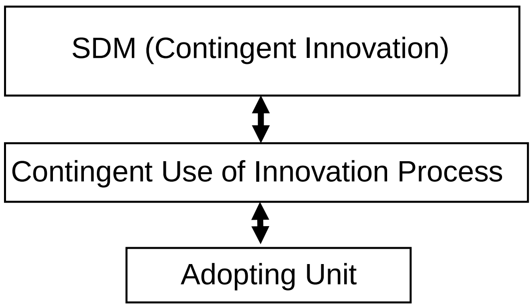


Figure 3-7: Contingent Use of SDMs Model (CUOSDM)

3.9.1 SDMs (Contingent innovation)

The systems development methodology (SDM) is an innovation (Rogers, 2003) and is regarded as a contingent innovation (Viljoen, 2016). The SDM is an object of adoption decision in the context of systems development. The adoption decision of an SDM is not made in a vacuum; rather it is driven by several characteristics of the SDM itself. The SDM characteristics influence the adopting unit to perceive problem solving from an optimistic view. Therefore, the prospective adopting unit is assumed to be a rational decision maker who makes adoption decisions based on the merits of the SDM under consideration (Huisman, 2004; Rogers, 2003). Therefore, the characteristics of the SDM are important in the adoption process (Rogers, 2003). The adoption process goes through various levels: the organisational level, the systems development project level and the individual level as presented in Section 3.7. This traversal through various levels within an organisation may be vertical or horizontal. These levels may have different views about the support structures provided by the SDMs. Huisman and Iivari (2006) found that managers had more optimistic perceptions on the adoption of SDMs than the systems development practitioners. This provided evidence that the organisational level is less detailed in terms of SDM expectations than the systems development practitioners. Organisation culture influences the adoption of SDMs (Iivari and Huisman, 2007). Therefore, in a top down adoption process, the organisation may select an SDM based on its attributes. The decision to adopt or reject an SDM is based on a rational choice of the adopting unit (Fichman, 2004; Rogers, 1995). This rational choice is directed by characteristics of the SDM itself such as perceived relative advantage/perceived usefulness, complexity/perceived ease of use, compatibility/fit (Avison and Fitzgerald, 2006; Rogers, 2003; Huisman and Iivari, 2002; Fitzgerald, 1998; Goodhue, 1995; Davis, 1989; Rogers, 1995). These characteristics are some of the high level reflection of underlying assumptions on the design of SDMs. However, these characteristics may be perceived differently at organisational, systems development project and individual levels irrespective of the underlying philosophical underpinnings considered in the design of the SDM.

Considering the top down adoption approach, an organisation may impose its alignment requirements to the SDM, to suit the strategic direction of the organisation and pass it to the systems development project level. At systems development project level, the contextual stressors influence the necessary tailoring to best fit a specific systems development project problem. Henderson-Sellers *et al*. (2014), Brinkkemper (1996) and Iivari (1989) indicate that no matter how well designed an SDM can be, it will always need to be tailored to provide the necessary fit to each specific systems development project. Therefore, amenability to tailoring is one of the important characteristics of any SDM and is dependent on the underlying assumptions considered in its design. At individual level, an SDM is evaluated and harmonised with the standard practice, expertise and experience of each individual adopter. The bottom up approach entails individuals persuading and influencing each other to adopt a specific SDM, proposing it to be adopted at systems development project level and then influencing management to adopt it at organisational level.

SDMs may be classified into two broad categories based on the underlying assumptions considered in their design: the plan-driven and the agile SDM classes as explained in Chapter 2. The plan-driven class relies on front loading. This class assumes that contextual stressors and the changes that may take place during systems development are predictable and can be predetermined (Leau *et al*., 2012). On the other hand, the agile family assumes that contextual stressors can be dynamic, varied and cannot be predetermined (Conboy and Fitzgerald, 2010). Figure 3-8 illustrates the main classification of the SDMs.

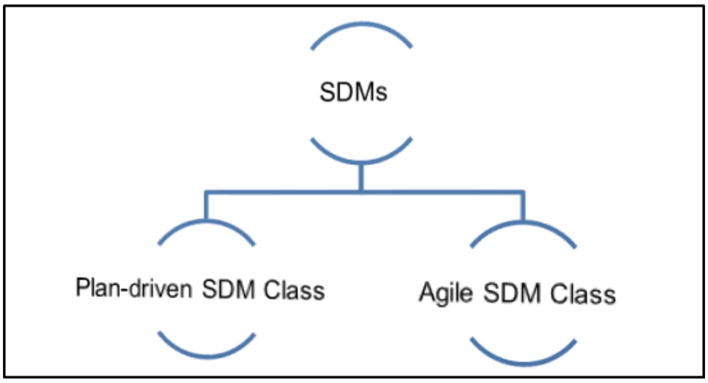


Figure 3-8: SDM classification

Each SDM class is guided by its basic assumptions about the systems development project contextual stressors. In some cases, components of different SDMs are combined to form hybrid SDMs (Rahmanian, 2014; Komus, 2014). Therefore, the characteristics of different innovations can be combined to create a fit that influences the adoption or the continued use of an SDM. The adopting unit may develop a portfolio of instances of SDMs or components of SDMs. The construction of hybrid SDMs can then be done using the portfolio as a repository of components. Adopted contingent innovations are rarely used in their normative state; they go through various contingent use processes to achieve ideal fit to the contextual stressors of specific systems development projects (Gill *et al*., 2018; Viljoen, 2016; Rahmanian, 2014; Brinkkemper, 1996).

3.9.2 The adopting unit

The next component of the CUOSDM is the adopting unit. The diffusion of SDMs occurs among the adopting units within the same contingent innovation adoption level or across different contingent innovation adoption levels. The adopting unit may be an organisation, systems development project or individual that is confronted with an SDM adoption or continued use decisions. The adopting units may perceive an SDM differently due to the adoption level, knowledge, culture, expertise and experience (Iivari and Huisman, 2007; Huisman and Iivari, 2006). The adopting unit traverses through the innovation decision process rationally evaluating the costs and benefits of their adoption choices (Bui, 2015; Huisman, 2004).

The initial choice consists of expectations driven by perceptions about the contingent innovation or knowledge from literature, consultancy, gatekeeper or any other reputable or trusted sources. The perceptions are based on a rational or utilitarian evaluation of the contingent innovation characteristics (Bui, 2015; Rogers, 2003). The innovation decision phases and the contingent innovation characteristics influence the adoption decision and the continued use of a contingent innovation (Rogers, 2003). The adopted SDM influences the organisation of the adopting units. The SDM can change the very nature of an adopting unit's work behaviour and interaction with other adopting units, be it at the same adoption level or at different adoption levels. These adopting units are expected to identify the possible misfits given an SDM.

In general, misfits are viewed from the technical, cultural, and political perspectives (Ansari *et al*., 2010). It is the responsibility of the adopting unit to identify the appropriate fit region within which a specific SDM falls. The responsibility of the adopting unit does not end on the identification of an SDM, but also evaluates it in relation to the contextual stressors. The evaluation may result in various decisions taken by the adopting unit such as adopting the SDM, adapting it to fit contextual stressors, continuing using it or rejecting it (Ansari *et al*., 2010). In the context of this study, the adopting unit is presented in a nested format. The outer most layer is the organisation, followed by the systems development project and the inner most layer being the individual. The adopting unit nested structure based on Viljoen (2017) is illustrated in Figure 3-9.

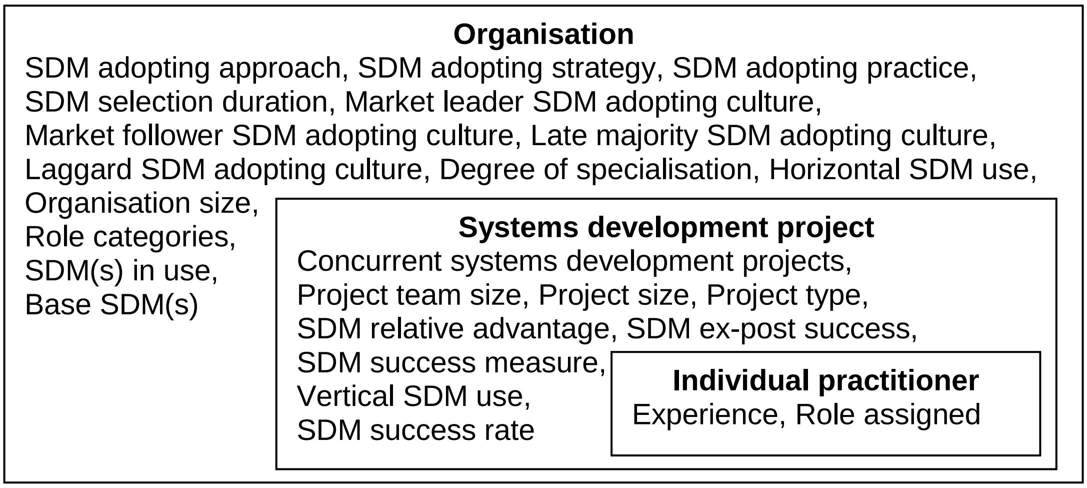


Figure 3-9: The SDM adopting unit hierarchy

The third component of the CUOSDM model is discussed in the next subsection.

3.9.3 The contingent Use of Innovation Process (CUOIP)

The Diffusion of Innovation model (Rogers, 2003) is used as an organising framework to develop the hybrid model for the contingent use of the SDMs. The Technology Acceptance Model (Venkatesh and Davis, 1996), the Task-Technology Fit model (Goodhue, 1995) are used to complement the Diffusion of Innovation model in the development of a hybrid model for the contingent use of SDMs. The three models provide the appropriate theoretical synergies for the development of the hybrid model for the contingent use of SDMs. The hybrid model comprises a combination of the determinants from all the three models, the TAM, the TTF and the DOI. These three theoretical models are complementary to each other.

TAM relies on ex-ante evaluation. That is when the adopting units encounter an SDM for the first time, perceived usefulness and perceived ease-of-use (PEOU) influence their decision to adopt it (Venkatesh and Davis, 1996). The benefits of using an SDM are compared with the effort required to use that same SDM. The TAM is concerned with the perceived psychological characteristics of an adopting unit towards adopting an SDM not necessarily the actual characteristics of the SDM. The TAM therefore, is driven by effort-orientation (perceived ease of use) and value-orientation (perceived usefulness) towards an SDM. In this regard, TAM is more appropriate at the early stages of systems development adoption decision (Turner *et al*., 2010; Yen *et al*., 2010).

Regarding TTF, the decision is biased towards post-ante matching of tasks characteristics (dealing with the contextual stressors) and the functionality offered by an SDM. The fit construct attempts to address the limitation of TAM in that it addresses the task characteristics (Dishaw and Strong, 1999). The fit has an evidence-oriented focus on how the contingent innovation supports the adopting unit's task accomplishment (Goodhue, 1995). Hence, the TTF model assumes that the positive practical experiences with the SDM and the support it provides for the tasks influences continued use (Goodhue, 1995). The DOI focuses on the learning process about the SDM from adoption to continued use. It is underpinned by the gradual reduction of ignorance (gaining experience) related to an SDM. Reduction of ignorance about a contingent innovation is fundamental in the DOI model and facilitates the adoption and use of the contingent artefact (Rogers, 2003).

Turner *et al.* (2010) and Venkatesh *et al.* (2012) encourage researchers to consider context relevance when adopting and adapting constructs. The assumptions encapsulated in the constructs of the three models TAM, TTF and DOI provide consistent results in contingent innovation research (Viljoen, 2016; Weigel *et al.,* 2014; Tornatzky and Klein, 1982). Therefore, constructs from the three models appropriately explain and interpret the object of this study. The overarching model as stated before is the DOI. The rationale behind the combination of these models is that they capture different aspects of the contingent use of SDMs and at different adoption stages. The SDM contingent use model provides a unique theoretical foundation inspired by these three models.

The contingent use of innovation process shown in Figure 3-10 is a component of the contingent use of systems development model. The innovation-decision process, hereafter referred to as the adoption decision outcome chain, is the backbone of the contingent use of innovation process. The adoption decision outcome chain is a three-phased process. The phases are pre-adoption, adoption and post-adoption. These three phases involve information-seeking and information-processing activities through which an adopting unit goes. Each phase consists of a chain of decision outcomes (DOs) on an SDM or its components. Decision outcomes (DOs) from one phase are input to the generation of other decisions in the same phase, or next phase or previous phase or phases. Therefore, progression from pre-adoption of a contingent innovation to its post-adoption is conceptualised as a nonlinear iterative process. The progression trajectories of the decision outcomes (DOs) are functions of available information, risks, and adopting unit's characteristics. The decision outcome trajectory is contingent to the feedback mechanism. The decision outcome chain allows the adopting unit to constantly monitor and evaluate the dynamics of contextual stressors during the systems development project lifecycle and respond accordingly when the need arises.

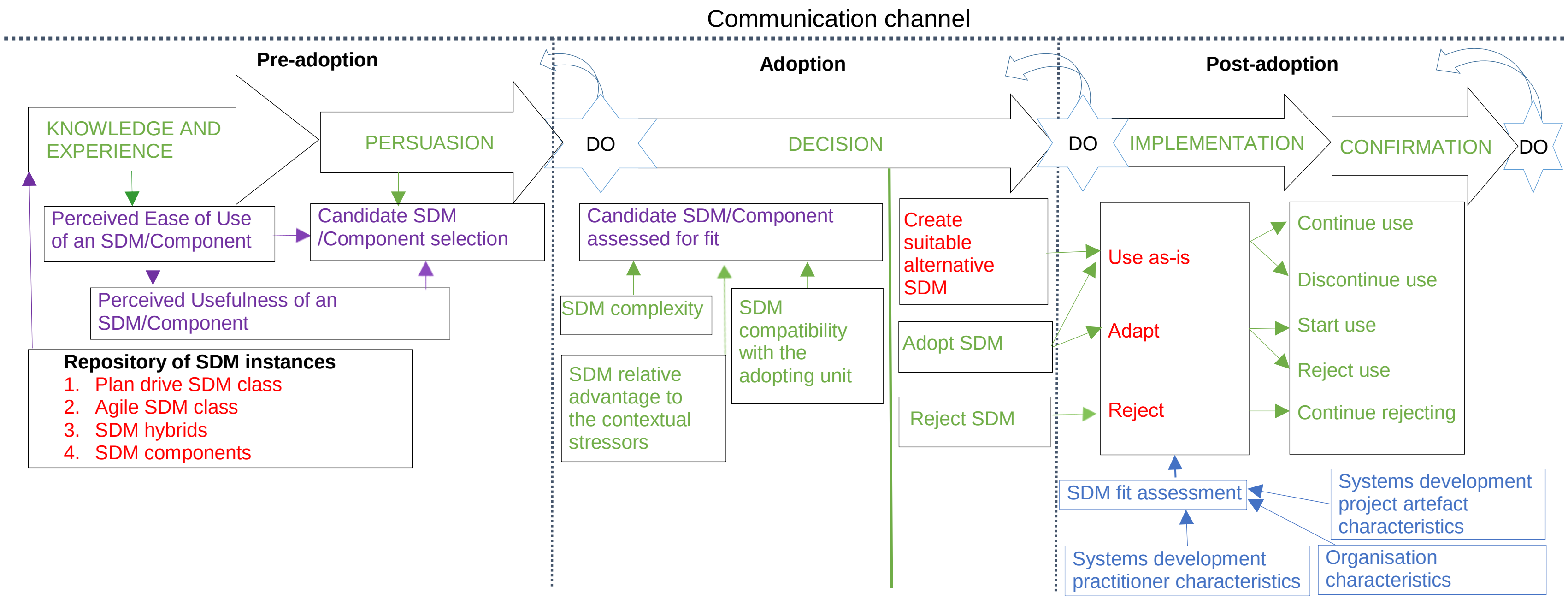


Figure 3-10: The Contingent Use of Innovation Process

<u>Key to Figure 3-.10</u>

DO stands for the decision outcome at a particular stage.

In Figure 3-10, the purple colour indicates the constructs and relationships adapted from TAM. The green colour illustrates the constructs and relationships adapted from the DOI model and the blue specifies the constructs and relationships adapted from TTF model.

3.9.3.1 Pre-adoption phase

The pre-adoption phase is the initial phase in the adoption decision outcome chain where perceptions regarding the SDM start developing. The adopting unit acquires knowledge and/or discount knowledge and reduces ignorance with regards to the costs and benefits of adopting the SDM. The TAM, TTF and the DOI determinants are important at this stage.

The two main stages at the pre-adoption phase are the knowledge and the persuasion. At the knowledge stage, the contingent innovation, in this case the SDM (SDM) or the SDM components thereof should exist and be known. The knowledge may come from literature, consultancy, gatekeeper, experience or any other reputable or trusted sources of technical and expert information. The adopting unit gains initial exposure to or/and descriptions of instances of the SDM or SDM components. The information gathering and conceptualisation leads to the development of the knowledge on how the SDM works with regards to the contextual stressors posed by the specific systems development problem situation. The adopting unit compares the known instances of SDMs to establish the appropriateness of each instance to the contextual stressors. The main decision outcome is the intention to select a candidate SDM for further detailed evaluation on the next phase. The TTF and the DOI determinants are essential at this phase of the decision outcome chain.

The persuasion stage entails the adopting unit's belief and attitude formation based on the knowledge gained from the knowledge stage regarding the candidate SDM. This stage marks the transition from the intellectual relationship with the candidate SDM to an affective relationship (Rogers, 2003). The TAM determinants are influential at this phase. Each candidate SDM is theoretically evaluated on its suitability to deal with the contextual stressors at hand. The attitude formed towards an SDM may be favourable or unfavourable. For example, after reflections about the benefits and/or risks of an SDM, an adopting unit may develop enthusiasm or curiosity, scepticism or indifference with regards to the SDM. The main objective of the persuasion stage is to trigger favourable attitude towards the adoption of a candidate SDM.

Considering Figure 3-10, at pre-adoption phase, the adopting unit may come across an SDM instance or learns about it or gets a description of an SDM that potentially fits the contextual stressors. The knowledge or experiences about the SDM becomes the main drivers to select a candidate SDM. The candidate SDM is chosen from a portfolio of SDMs instances. The portfolio consists of classes of SDMs instances and the SDM components from which candidate SDMs are selected. The selection is based on the perceived characteristics of an SDM in relation to the fit it provided to a similar context settings or simple perception of fit to the contextual stressors.

The expected favourable behaviour towards adopting a candidate SDM may be associated with the two main constructs from TAM's, namely perceived ease of use and perceived usefulness.

Perceived relative advantage is the degree to which the adopted SDM or systems development component addresses the contextual stressors better than any of its precursors (Rogers, 1995). Considering an SDM as a contingent innovation, perceived relative advantage is one of the main drivers of SDM adoption (Rogers, 2003). The higher the perceived relative advantage, the higher the probability of SDM adoption (Huisman and Iivari, 2002; Huisman, 2004). The perceived relative advantage from DOI (Rogers, 2003) and perceived usefulness from TAM (Davis *et al*, 1989) are the two value-oriented and relatively similar constructs. They are the most consistent and influential factors in relation to SDM adoption and use (Schlauderer and Overhage, 2013; Mohan and Ahlemann, 2011; Huisman, 2004; Tornatzky and Klein, 1982). The perceived relative advantage's value can be described in terms of potential benefits of an SDM such as economic gains, level of comfort, productivity, market window, conceptualisation of contextual stressors, compliance with regulatory requirements and savings on time and effort (Fitzgerald *et al*., 2006; Rogers, 2003).

SDM fit (compatibility) is the degree to which an SDM is perceived to provide a match between existing contextual stressors such as work practices, cultural values, political, technical and past experiences of the adopting units (Rogers, 2003, 1995). The fit of a system development methodology with the contextual stressors increases the probability of it being adopted and used (Rogers, 2003). SDM fit is directly proportional to the adoption of an SDM, all other factors being equal.

The nature of systems development problem does not allow experimentation on SDMs. There are no surrogate contextual factors, but a specific and unique systems development project situation each time. In this regard, Riemenschneider *et al*. (2002) observes that trialability may not be an appropriate predictor. The use of an SDM is observable but the contextual factors are different for each problem situation, therefore imitation behaviour is not applicable in systems development. Trialability and observability is not significant in SDM studies (Schlauderer *et al*., 2015; Schlauderer and Overhage, 2013; Huisman, 2004; Riemenschneider *et al*., 2002; Tornatzky and Klein 1982). Thus, observability and trialability constructs are not included in this model.

The main deliverable at the pre-adoption phase is the consideration to adopt a candidate SDM. Knowledge and persuasion stages may be complemented with experience to decide about the candidate SDM. The decision outcome of the pre-adoption phase is forwarded to the adoption phase. The adoption phase takes as input the outcome from the previous phase. The effort-oriented factor, perceived ease of use, is an indicator of the cost associated with the effort and time needed to learn using the SDM. A low perceived ease of use may act as a barrier towards the intention to adopt an SDM instance, whereas a higher perceived ease of use may positively influence the intention to select an SDM instance or instances (Davis *et al.,* 1989). Riemenschneider *et al.* (2002) argues that both perceived ease of use and perceived usefulness of an SDM has a positive influence on the intention to select a candidate SDM instance. Huisman (2004) states that relative advantage of the DOI is the most influential factor in the SDM use and acceptance.

3.9.3.2 Adoption phase

Henderson-Sellers *et al*. (2014) and Börner (2011) point out that no SDM can be used without some modification. Subscription to a single SDM is not feasible, as each systems development situation is unique. Notably, SDMs are still designed with the idea of being generic to be applied to wider problem domains, that is, different types of organisational settings, systems development projects situations, and individual user characteristics. In this way, SDM are pre-contextualised based on deterministic domain specifics. Instead of reinventing the wheel each time a systems development project is instantiated, the adopting unit selects, adapts, or tailor an SDM based on specific contextual stressors ranging from organisational, systems development project, up to individual levels (Viljoen, 2016; Huisman, 2013; Burns and Deek, 2011). In some cases, an SDM can provide a considerable fit to the contextual stressors of the development situation.

The subsequent phase from pre-adoption phase is the adoption phase. The decision outcome on the SDM instance considered from the pre-adoption phase is used as input to the adoption phase. In cases where no SDM instance exists that fits the contextual stressors, the adopting unit creates an alternative SDM from existing SDM components. The adoption phase is constituted by the decision stage. The concept of decision stage does not imply that this is the only stage in the entire contingent use of innovation process where decisions are made. The concept of decision stage is contextualised by Rogers (1995) to emphasise the decision to adopt a contingent innovation, not to imply that it is the point where decisions are made. The adoption phase is an important stage of the adoption decision outcome chain. The adopting unit provides a deliverable in the form of a set of decision choices made based on the acquired knowledge, discounted knowledge and formed perceptions about SDMs. The deliverables give a clear line of actions or activities related to adoption. The adoption related actions can be either to adopt an instance of an SDM or to reject an instance or reject all instances of SDMs under consideration and create a suitable alternative. The SDM characteristics are important at this phase, as the evaluation of appropriateness is measured by how the SDM fit the contextual stressors. The decision can also loop back to the previous phase and consider another SDM instance or a set of SDM instances or a set of SDM components. The outcomes from the adoption phase are used as input to the next phase of the decision outcome chain, the post-adoption phase.

3.9.3.3 Post-adoption phase

The third phase in the adoption decision outcome chain is the post-adoption. The post-adoption phase entails two stages, the implementation and the confirmation. The implementation stage entails the integration of the SDM into the work framework. The theoretical structure of implementation stage entails the temporal ordering of activities, steps, events and the associated experience and decisions. The interaction between the team, the SDM and the systems development project contextual stressors triggers the decision outcome chain. The adjusting, tweaking, tailoring, fine-tuning, and/or adapting are done at this stage to improve SDM use continuance and /or SDM utilitarian outcomes. The relative advantage of change would have been perceived by the adopting unit earlier on. The decision outcome from this phase determines whether the SDM was appropriate for the contextual stressors and the prospects of using the SDM or its component in future. This decision outcome chain gives insights on how the organisation, the individual member of the development team and the SDM co-evolve and inform one another over the systems development project life cycle. The post-adoption phase of the decision outcome chain is illustrated in Figure 3-11. The post-adoption phase is the focus of this study.

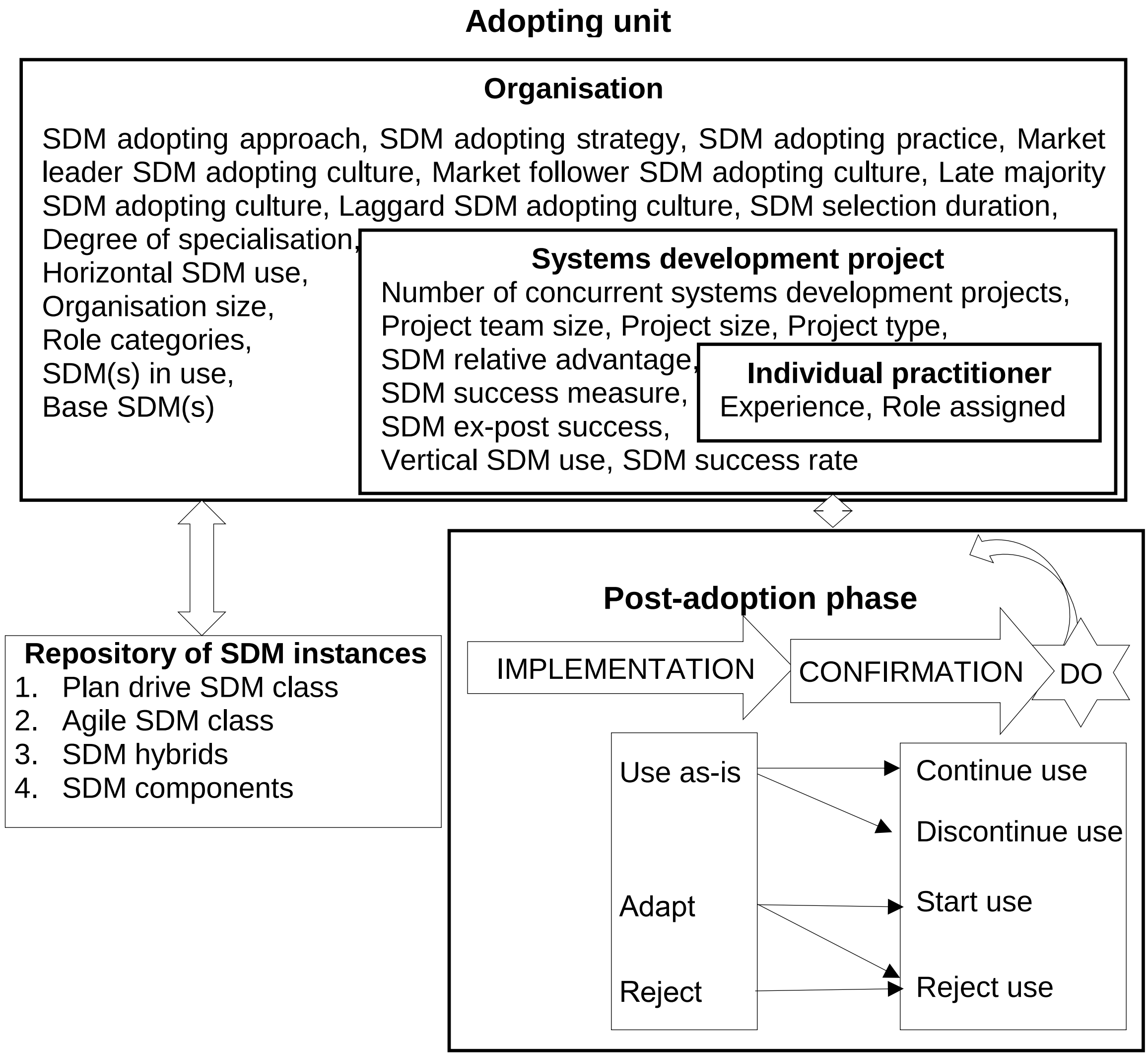


Figure 3-11: Post-adoption phase

The following subsection describes the critical success factors in relation to the contingent use of SDMs.

3.9.3.4 Contingent use of SDMs critical success factors

The systems development project contextual stressors are well documented (see Section 3.4, Table 3-2, Table 3-3, Table 3-4). The systems development project contextual stressors are commonly used in studies to compare SDMs, in which case the two compared SDMs may be instances from the same class of SDMs or from different classes of SDMs (Schlauderer and Overhage, 2013; Aitken and Ilango, 2013; Wysocki, 2011). The assumption is that, once the systems development project contextual stressors are identified and specified, the SDM is deployed and used throughout the course of the systems development project.

The contingent use of SDMs perspective does not assume the deployment of an SDM as a static process, but as a dynamic decision out chain that evolves during a systems development project (see subsection 3.11.3, Figure 3-10). Therefore, adopting an SDM for a systems development project is contingent to the evolving systems development project contextual stressors. The contextual stressors change over the systems development project lifecycle and this influences the criticality of a contextual stressor. This means that as the systems development project evolves, the importance and value of some contextual stressors change. The emergence of changes in the criticality of some contextual stressors may necessitate adaptation of an SDM in use, adjustments of an SDM in use, a workaround on an SDM in use, a discontinuance of an SDM in use or an SDM component thereof (see subsection 3.11.3, Figure 3-10). This is the core of the contingent use of SDMs. The SDM enactment and use addresses the dynamics of contextual stressors over the systems development project lifecycle.

Literature is rich in proposals related to the mapping of each systems SDM to the systems development project contextual stressors (Vieljon, 2017; Dikert *et al.,* 2016; Serrador and Pinto, 2015; Schlauderer and Overhage, 2013; Aitken and Ilango, 2013; Wysocki, 2011; Misra *et al*., 2009; Chow and Cao, 2008). Critical success factors constitute the minimum factors that can result in the satisfactory attainment of desired goals (Müller and Jugdev, 2012). In this study, the critical success factors are considered as the minimum number of systems development project contextual stressors that significantly influence the contingent use of SDMs. The focus is on the contingent use of SDMs in systems development projects.

The hypothesized relationships derived from the contingent use of SDMs conceptual model of the study are presented in the next section.

3.10 RESEARCH HYPOTHESES

SDM selection frameworks, guidelines, decision support systems, including expert systems are well documented (Viljoen, 2016; Young *et al*., 2016; Harb *et al*., 2015; Vavpotič and Vasilecas, 2012; Mnkandla and Dwolatzky, 2007; Zhu, 2002; Burns and Dennis, 1985) as outlined in Subsection 3.6.1.1. Furthermore, milestones in the adoption and use of SDM have been well-documented (Versionone, 2018; Imani *et al*., 2017; Young *et al*., 2016; Schlauderer *et al*., 2015; Rahmanian, 2014; Bustard *et al*., 2013; Conboy, 2009; Chow and Cao, 2008). These milestones in systems development are to this study during the contingent use of SDMs phase. Consequently, the contingent use of SDMs is associated with the post-adoption phase of the decision outcome chain. The post-adoption phase of the decision outcome chain is constituted by the implementation and the confirmation stages of the DOI’s innovation-decision process. Figure 3-12 illustrates the populated contingent use of SDMs conceptual model.

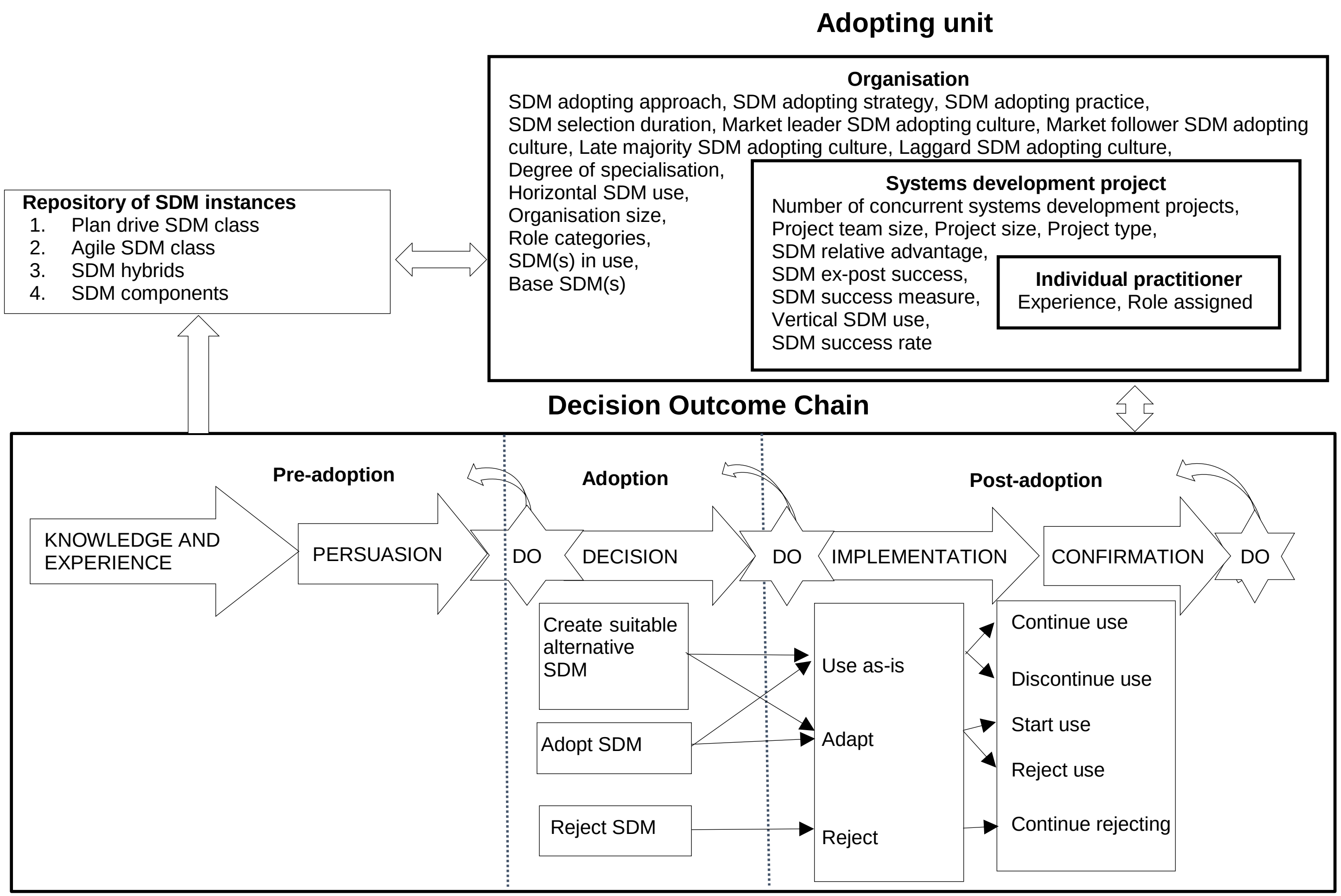


Figure 3-12: Detailed contingent use of SDMs Model (CUOSDM)

3.10.1 SDM fit assessment

At the implementation stage, a set of decisions is generated. The set of generated decisions serves as a starting point for the course of action to be taken in relation to an adopted SDM, created alternative SDM or rejected SDM. The adopting unit, after assessing the fit of the SDM to the task at hand (Goodhue, 1995), may consider the SDM use options presented in Table 3-7. The options are influenced by the SDM fit assessment (Goodhue, 1995). The SDM functionality is implicitly characterised by the SDM characteristics. The task requirements, on the other hand, are determined by the contextual stressors, whereas the adopting unit characteristics are based on experience and level of expertise among other important factors (Henderson, 2006).

Table 3-6: SDM use choice options

SDM ADOPTION CHOICE OPTIONS	SDM USE DECISION OPTIONS	SOURCE
Adopt SDM	Use SDM as-is	Viljoen (2016), Fitzgerald (1998)
	Adapt SDM to fit contextual stressors	Henderson-Sellers *et al*. (2014)
	Change SDM address the contextual stressors	Henderson-Sellers *et al*. (2014), Börner (2011), Burns and Deek (2011)
Create an alternative SDM	Use the created SDM as-is	Henderson-Sellers *et al*. (2014), Fitzgerald (1998)
Reject adoption of SDM	Reject use of an SDM	Rogers (2003)

Considering the first SDM adoption choice, the adopting unit is presented with four SDM use options. These SDM use options are: use as-is, adapt, change or reject. These options subscribe to the contingency-based SDMs selection approach of the contingent use of SDMs. The pre-packaged SDM is selected and its fit to the organisation characteristics, systems development project circumstances and individual adopter characteristics assessed. Berente *et al*. (2015) and Fitzgerald (1998) found that an SDM can be used in its original state as proposed by its creator and formalized in a publication or any documented format. The decision to use an SDM as-is exposes the SDM to practical use during which direct experience is gained, and practical, as well as evidence based evaluation, is carried out (Sun, 2013). When no discrepancies exist between the actual and the expected performance, the adopting unit confirms continued use (Sun, 2013). The confirmation stage is whereby the anticipated benefit of the SDM is compared with the actual benefits. The outcome of the comparison is used as input to a confirmatory decision to continue, or discontinue using it. The contingent use of SDMs is not a once-off outcome but rather an iterative process that is determined by the dynamics of the contextual stressors (Berente *et al*., 2015; Limayem *et al*., 2007). Therefore, the adopting unit continuously and consciously evaluate the appropriateness of the adopted SDM (Limayem *et al*., 2007).

In some cases, there exist discrepancies between the actual and the expected performance. The adopting unit may disconfirm its earlier assumptions and perceptions (Sun, 2013). The assumption for adoption is based on positive perceptions and disconfirmation indicates a decision to discontinue use (Rogers, 2003). However, this is not straightforward as the decision may iterates by providing feedback to the previous stage or stages. Feedback is the key learning characteristic of the decision outcome chain.

SDMs are designed with the intention of applying them in a wide range of systems development project contexts. The SDM creators generate assumptions based on the scope of problem domains and in anticipation of, or in response to generalised systems development project circumstances and conditions. The pre-packaged SDMs are generic in nature and too rigid. This can lead to an implementation that departs significantly from the published versions (Berente *et al*., 2015). The rigidity and heaviness of the SDM shifts focus from developing the systems development project artefact, to systems development process implementation. Burns and Deek (2011) state that adopting units adapt SDMs to fit the specific context in which they are used. Berente *et al*. (2015) concludes that each SDM is adapted to each specific context regardless of its preconceived fit to a specific context of application. Adaptation is a strategy that deals with misfit conditions (Avison and Fitzgerald, 2006). Therefore, adapting involves the activities carried out by the adopting unit to address misfits, be they technical, cultural, and/or political misfits. Ansari *et al.* (2010) argues that adaptation is a process whereby an adopting unit endeavours to achieve a fit between the contingent innovation and the contextual stressors, to increase the contingent innovation's adoption probability. The adaptation process involves changes in the implementation of an SDM, taking into consideration the technical, cultural, political, and other contextual factors. The SDM is reframed or reconfigured to increase its acceptance zone. The rationale behind adapting an SDM as contingent innovation is not necessarily to improve it, but to create a fit between the SDM and the contextual stressors.

Henderson-Sellers *et al.* (2014) points out that the need to change, modify or fine-tune the SDM arises when an adopting unit realises that the system development methodology does not provide the required and expected support structures. The modification is performed to improve the extent to which the SDM must offer the systems development support structures. The SDM modification may entail the addition of new components, omission of unnecessary components (Fitzgerald *et al.,* 2002), or combining SDMs to form hybrid SDMs (Rahmanian, 2014; Komus, 2014; West, 2011). In such context, an SDM is conceptualised as consisting of a set of components at different levels of granularity (Henderson-Sellers *et al.*, 2014). These components are the building blocks of the SDMs that can be modified or assembled to create SDMs that address the contextual stressors (Henderson-Sellers *et al*., 2014; Rahmanian, 2014).

The SDM that has been modified or improved is put to practical use (Rahmanian, 2014). The practical application of an SDM generates either positive, negative or mixed experiences. The evaluation of an SDM's merit is based on the nature of experiences. Positive experiences would likely lead to confirmed continued use. The confirmation of the anticipated benefits influences the adopting unit's intention to continue using the SDM (continuance intention) (Mohan and Ahlemann, 2011). The adopting unit continuously and iteratively carries out comparison between the anticipated benefits (perceived usefulness, relative advantage) and the actual benefits (TTF) and make choices contingently. The negative experiences would most likely influence discontinued use (Sun, 2013). The decision to discontinue is not categorical, since the adopting unit may iteratively repeat the previous stage of creating a fit of the SDM to the contextual stressors (Viljoen, 2016; Huisman, 2013; Burns and Deek, 2011). The SDM may be adapted to deploy the one that addresses the demands of the contextual stressors.

The adopting unit, after adapting the SDM may start using it. When no gaps exist between the actual and the expected performance of the SDM then, the adopting unit may confirm continued use (Sun, 2013). However, if the adapted SDM fails to meet expectations, it may be rejected and an alternative SDM created. The adopting unit will constantly and continuously engage evaluate the fit of SDM. Consequently, the following hypothesis is formulated:

H1: The SDM fit assessment positively influences the contingent use of SDMs.

3.10.2 Organisation culture

The organisation culture forms the context where SDMs are adopted and implemented. Organisation culture exerts influence in systems development (McLeod and MacDonell, 2011; Siakas and Siakas, 2007; Huisman and Iivari, 2006; Boehm and Turner, 2003).

According to Schein (2009), organisational culture can be identified by the basic assumptions, values, artefacts, and practices. According to Huisman and Iivari (2006), an SDM includes a systems development approach (goals, guiding principles, beliefs, fundamental concepts, interpretations and actions), a systems development process model, a systems development method, and a systems development technique. An SDM is part of organisational culture or a subculture thereof. It represents the cumulative learning of the organisation and creates shared mental models among team members. Furthermore, Boehm and Turner (2003) considered plan-driven and agile SDMs as two cultures in the context of systems development. An organisational culture informs an SDM and an SDM informs the organisational culture in return. Iivari and Iivari (2011) describe the relationship between the organisational culture and SDMs as evolving and dynamic. That is an organisation learns how to organise, manage and execute systems development through an SDM. It gains experience on what works in different systems development project situations. In turn the organisation improves the SDM by adapting or adjusting it to fit specific systems development project situations.

The organisational culture is one of the hardest contextual stressors to change (Schein, 2009) and every systems development practitioner is exposed to it (Russo *et al*., 2013). The past success may influence the adopting units to adhere to the values and practices that have worked successfully in the past (Boehm and Turner, 2004). Iivari and Iivari (2011) and Iivari and Huisman (2007) investigated the influence of four different culture categories on SDM deployment using the Competing Value Model and found a significant positive relationship between hierarchical culture and rational culture and the deployment of plan-driven SDMs. Sheffield and Lemétaye (2013), Iivari and Iivari (2011) and Iivari and Huisman (2007) posit that each organisational culture category favours certain classes of SDMs. Iivari and Iivari (2011) observe that some SDMs may be incompatible with certain organisational culture types. Therefore, organisational culture entails frames of references, shared values, and beliefs, collective identity, shared experiences, communication protocols, and assumptions that influence decisions in systems development (Gruver and Mouser, 2015; Sheffield and Lemétaye, 2013; McLeod and MacDonell, 2011; Iivari and Iivari, 2011; Iivari and Huisman, 2007).

The classification of organisational culture in this study was specifically targeted on the responsiveness of an organisation in adopting SDMs as contingent innovations. Rogers (2003) categorized responsiveness to the adopting of contingent innovations into five adopter classes: the innovators, early adopters, early majority, late majority and laggards. The main characteristic of innovators' culture is to embrace a contingent innovation for its own sake (Rogers, 2003). No organisation is expected to embrace an SDM for its own sake. The innovator and the early adopters are grouped together under the market leader category. Consequently, the study considered the following four categories: market leader, market follower, late majority and laggards. These would correspond to developmental, rational, hierarchical and group culture respectively. The adopter category in which an organisation falls is considered as a reflection of its SDM adopting culture. The market leader SDM adopting culture is comfortable with changing its behaviour to take advantage of the opportunities without wasting time (Pietri, 2011). The market leader SDM adopting culture is quick and flexible to adopt a new SDM or adapt an already adopted SDM contingently. Consequently, the following hypothesis is formulated:

H2a: There is a positive relationship between the market leader SDM adopting culture and the contingent use of SDMs.

The market follower SDM culture involves those organisations that take advantage of the opportunities offered by an SDM as observed from the experience of the market leader. The market follower SDM culture is goal-oriented and productivity focused and aim at what works. The adoption influence may also come from reliable communities of SDM practice to which the organisation or the team members are affiliated. There are several systems development communities present in South Africa to which organisations or team members may pay allegiance. These include the following; the Institute of Information Technology Professionals South Africa (IITPSA), Agile Alliance, Joburg Centre for Software Engineering (JCSE), ScrumAlliance, Project Management Institute (PMI), PRojects IN Controlled Environments version 2 (PRINCE2), The Institute of Electrical and Electronics Engineers (IEEE). Iivari and Huisman (2007) found a significant positive relationship between rational (market follower) organisational culture and the deployment of plan-driven SDMs. Consequently, the following hypothesis is formulated:

H2b: There is a negative relationship between the market follower SDM adopting culture and the contingent use of SDMs.

The late majority SDM adopting culture avoids the risks of breaking new ground by pragmatically weighing the costs-benefits ratio experienced by both the market leader SDM adopting culture and the market follower SDM adopting culture organisations. Control and order is important in the market follower SDM adopting culture. Iivari and Huisman (2007) found a significant positive relationship between hierarchical (late majority) organisational culture and the deployment of plan-driven SDMs. Consequently, the following hypothesis is formulated:

H2c: There is a negative relationship between the late majority SDM adopting culture and the contingent use of SDMs.

The laggard SDM adopting culture trails behind every other SDM adopting culture. They may be affiliated to some systems development communities of practice and take time to embrace change if that change is not coming from their affiliations. Generally, they are slow in adopting new approaches. Consequently, the following hypothesis is formulated:

H2d: There is a negative relationship between the laggard SDM adopting culture and the contingent use of SDMs.

3.10.3 Organisation size

Organisation size is one of the important factors in the adoption of SDMs (Viljoen, 2016; Vijayasarathy and Butler, 2015; Dybå *et al*., 2012; Turner and Zolin, 2012; Hoppenbrouwers *et al.*, 2011). The size of an organisation affects the way members interact and share information. The size influences resources, level of specialisation and applicable communication protocols (Vijayasarathy and Butler, 2015; Barlow *et al.*, 2011). The larger the organisation, the more formal would be the support structures for the systems development activities (Barlow *et al.*, 2011). Consequently, the following hypothesis is formulated:

H3: The organisation size negatively influences the contingent use of SDMs.

3.10.4 Systems development project artefact

A systems development project artefact is the focus of systems development. A systems development project is characterised by its contextual stressors. Boehm (2006) identified five systems development project contextual stressors and considered them as critical for systems development. According to Boehm (2006), personnel, requirements dynamics, organisational culture, team size, and system criticality are the core factors that influence the adoption and use of specific SDMs. There are various ways in which a systems development project artefact can be evaluated. The perceived appropriateness of an SDM is reflected by the value associated with systems development project artefact it previously produced (Shenhar and Dvir, 2007). The usefulness and appropriateness of an SDM is based on the success level of systems development artefacts it previously developed. The performance of the system development project artefact over a time scale is another measure of the SDM success (Turner and Zolin, 2012). The tried and tested SDM may be evaluated by the systems development project artefacts it successfully developed. The success history of an SDM may lead to systems development practitioners resisting changes and adaptation of the SDM in the hope of maintaining the previous success. The success of an SDM (doing it right), is evaluated by the systems development project artefact success (getting it right). Consequently, the following hypothesis is formulated:

H4: The SDM ex-post success negatively influences the contingent use of SDMs.

3.10.5 SDM success measure

Conboy and Fitzgerald (2010) dismiss the universal applicability of SDM arguing that regardless of how well an SDM may be designed, it cannot provide a perfect fit to the needs of every systems development project. Burns and Deek (2011) posit that systems development practitioners always adapt SDMs to fit specific systems development project circumstances irrespective of whether they are plan-driven or agile. Barlow *et al*. (2011) and Boehm and Turner (2003) emphasise the use of different SDMs for different systems development project context settings.

Systems development project success is regarded as a multidimensional construct (McLeod and MacDonell, 2011; DeLone and McLean, 2003) that can assume objective, subjective or both perspectives. The traditional view of systems development project success measure is based on an objective criterion referred to as the iron triangle (Versionone, 2018; Besteiro *et al*., 2015;

Kloppenborg *et al*., 2014). The iron triangle holds that if the systems development project is completed on time, within budget and meeting the specified requirements, then it is successful (Versionone, 2018; Eveleens and Verhoef, 2010). The iron triangle is important as an objective criterion, but it is inadequate in evaluating all aspects of systems development projects success (Lech, 2013). Shenhar and Dvir (2007) argue that a systems development project that meets specification does not necessarily imply a useful system development project artefact. It only implies conformance to the articulated system development product requirements, which may be complete or incomplete or even wrong. Lech (2013), Jun *et al*. (2011) and Cuellar (2010) posit that systems development project success should include both objective and subjective measures. They argue that, instead of basing the criteria on the iron triangle alone, subjective measures should also be considered, such as multiple stakeholders' perceptions of success. This gives the balance between the two dimensions, which are the objective and the subjective dimension. Huisman and Iivari (2006) found that different stakeholders had different value propositions about SDMs. This indicates subjectivity in the evaluation of SDMs and that different stakeholders are interested in different aspects of the systems development project (Huisman and Iivari, 2006). The contingent use of systems development may result in a deviation from a systems development project plan or adaptation of the SDM or creation of an alternative SDM. In such cases, the iron triangle success criterion alone may see this as a failure without considering the context leading to the deviation from the plan.

SDMs are adapted (Diebold *et al.*, 2015) to fit systems development project's contextual stressors such as the time, the cost, quality, business value, and user satisfaction. The following are some of the common measures of SDM success: delivering the systems development project artefact on time, delivering systems development project artefact with all requirements specified, delivering the systems development project artefact within cost and meeting all the objectives of the systems development project. The systems development project artefact success is interpreted as an indicator of SDM success. Diebold *et al.* (2015) argues that a systems development project artefact is a result of a successful deployment of an SDM and contingent use of an SDM. Consequently, the following hypotheses are formulated:

H5: The SDM success measure positively influences the contingent use of SDMs.

3.10.6 Individual systems development practitioner experience

One of the critical characteristics an individual systems development practitioner may bring into the systems development project, is the experience with systems development projects (Marks, *et al*., 2017; Conger, 2013; Leau *et al*., 2012; McLeod and MacDonell, 2011; Cockburn, 2004; Fitzgerald, 1998). Experience with systems development projects is how know-how and knowswhy knowledge about SDMs is consolidated and systems development project uncertainty reduced (Rogers, 2003). The experienced individual systems development practitioner would have encountered similar situations in the past and made appropriate decisions on those situations. Cockburn (2004) indicates that the individual systems development practitioner's SDM implementation patterns are influenced by their past experiences. The experienced systems development practitioner assesses the fit of the SDM to the contextual stressors and adapts it accordingly. With more experience in using various SDMs, a systems development practitioner would know which SDM works, where, when and why resulting in the systems development practitioner's flexibility in adoption, adaptation, changing and rejection of SDMs as is necessary (Jun *et al*., 2011). Consequently, the following hypothesis is formulated:

H6: The individual systems development practitioner's experience positively influences the contingent use of SDMs.

3.10.7 Systems development project team size

Individual systems development practitioners are organised into systems development project teams. The systems development project team size consists of the total number of systems development practitioners in the systems development project team. The systems development team is one of the factors that may affect the choice of an SDM (Aitken and Ilango, 2013). Boehm and Turner (2003) indicates that systems development project team size is one of the five critical factors to be considered when selecting an SDM. The systems development project team is expected to possess most of the basic skills mix needed to accomplish the development of a systems development project artefact. Some SDM classes instances have no restriction to systems development project team sizes, whereas other SDM classes instances do specify the optimal systems development project team sizes (Mahanti *et al.,* 2012). The increase in the systems development project team size may lead to an increase in the level of communication formality and development coordination challenges (Harb *et al*., 2015; Barlow *et al*., 2011). Consequently, the following hypothesis is formulated:

H7: The systems development project team size negatively influences the contingent use of SDMs.

3.10.8 Systems development project team expertise

The level of expertise, the skills set and the experience possessed by each member of the systems development team influences how each member contributes to a systems development project (McLeod and MacDonell, 2011). In most cases, the roles assigned to each member are associated with the level of knowledge the member possesses in systems development. Young *et al.* (2016) observes that more application of SDM knowledge in systems development projects subsumes the level of expertise, skills and experience. The SDM knowledge usage is measured in terms of either horizontal use, that is, across projects or vertical use, which is the intensity of SDM knowledge use. The horizontal use entails the breadth of SDM knowledge use across the development of projects. The vertical use entails the depth of SDM knowledge application on each systems development project (Russo *et al*., 2013). The high level of horizontal and vertical use of SDM knowledge may resulted in the contingent use of SDMs. Consequently, the following hypotheses are formulated:

H8a: The horizontal use of SDM positively influences the contingent use of SDMs.

H8b: The vertical use of SDM positively influences the contingent use of SDMs.

3.10.9 SDM relative advantage

The systems development methodologies offer several advantages. The advantages are not all found in a single instance of an SDM (Rahmanian, 2014; Aitken and Ilango, 2013; West, 2011). Therefore, an SDM development may be adopted to a systems development project based on its compatibility to the contextual stressors involved. Each systems development methodology is compatible and offers relative advantage to systems development project-specific contextual stressors (Berente *et al*., 2015; Clarke and O'Connor, 2015; Barlow *et al*., 2011). The fact that an SDM must be enacted, indicates that the SDM relative advantages and compatibility to systems development project-specific contextual stressors of SDMs differ from one systems development project to another (Clarke and O'Connor, 2015). The systems development practitioner's expertise in identifying the advantage of one SDM over another on a specific systems development project is important. The relative advantage of an SDM accounts for the effort needed to tailor the SDM to fit the specific systems development project (Vavpotič and Vasilecas, 2012). When the SDM is fit for purpose then there is minimum effort required to tailor it to the specific system development contextual stressors. Consequently, the following hypotheses are formulated:

H9: The SDM relative advantage to the systems development project contextual stressors positively influences the contingent use of SDMs.

A summary of hypotheses formulated for the study is presented in Table 3-8.

Table 3-7: Summary of hypotheses

HYPOTHESIS	REFERENCES
H1: The SDM fit assessment positively influences the contingent use of SDMs (see subsection 3.12.1).	Deek ,2010
H2a: There is a positive relationship between the market leader SDM adopting culture and the contingent use of SDMs (see subsection 3.12.2). *H2b: There is a negative relationship between the market follower SDM adopting culture and the contingent use of SDMs (see subsection 3.12.2).* *H2c: There is a negative relationship between the late majority SDM adopting culture and the contingent use of SDMs (see subsection 3.12.2).* *H2d: There is a negative relationship between the laggard SDM adopting culture and the contingent use of SDMs (see subsection 3.12.2).*	McLeod and MacDonell, 2011; Siakas and Siakas, 2007; Iivari and Huisman, 2007; Boehm and Turner, 2003
H3: The organisation size negatively influences the contingent use of SDMs (see subsection 3.12.3).	Viljoen, 2016; Turner and Zolin, 2012; Barlow *et al.*, 2011
H4: The SDM ex-post success negatively influences the contingent use of SDMs (see subsection 3.12.4).	Turner and Zolin, 2012
H5: The SDM success measure positively influences the contingent use of SDMs (see subsection 3.12.5).	Own construction
H6: The individual systems development practitioner's experience positively influences the contingent use of SDMs (see subsection 3.12.6).	Own construction
H7: The systems development project team size negatively influences the contingent use of SDMs (see subsection 3.12.7).	Own construction
H8a: The horizontal use of SDM positively influences the contingent use of SDMs (see subsection 3.12.8).	Own construction
H8b: The vertical use of SDM positively influences the contingent use of SDMs (see subsection 3.12.8).	Own construction
H9: The SDM relative advantage to the systems development project contextual stressors positively influences the contingent use of SDMs (see subsection 3.12.9).	Own construction

These hypotheses will be discussed in detail in Chapter 6.

3.11 CHAPTER SUMMARY

The chapter defined the concept of the contingent use of SDMs and outlined the two main approaches to contingent use of SDMs. The theoretical foundation for the contingent use of SDMs conceptual model was presented as a hybrid of three contingent innovation adoption models. Furthermore, the chapter explained the outcome decision chain for the contingent use of SDMs and developed the proposed contingent use of SDMs conceptual model. Lastly, the chapter formulated the research hypotheses. The next chapter outlines the study's research methodology and the research design.

CHAPTER FOUR: RESEARCH METHODOLOGY

4.1 INTRODUCTION

The previous chapter presented the contingent use of SDMs conceptual model and established the hypothesised relationships between the constructs. Research methodology was briefly refereed in Chapter 1, and it is described in detail in this chapter. This chapter describes the methodology used in addressing the formulated objectives stated in Chapter 1. The chapter covers the research paradigms, research design, sampling design, data collection methods, research instruments, variable operationalization and measurement, instrument pretesting and the data analysis techniques.

4.2 RESEARCH PARADIGMS

Research is guided explicitly or implicitly by some research paradigm. The research paradigms constitute the underlying philosophical assumptions about the object under study. This section outlines two important research paradigms in the field of Information Systems. Kuhn (1970) states that a paradigm represents a particular way of thinking that is shared by a community of scientists in solving problems in their field. Guba and Lincoln (1994) defines a paradigm as a basic belief system based on ontological, epistemological and methodological assumptions. Schwandt (2001) defines a paradigm as a shared world view that represents the beliefs and values in a discipline and guides how problems are solved. Morgan (2007) points out that a research paradigm is a model containing a set of assumptions about the object under study. Kuhn (1977), Oates (2006), Maree (2007) and Neuman (2011) describe a paradigm as a stance that encompasses the researcher's underlying assumptions about reality and knowledge.

In this study, a research paradigm is considered as a world view informed by philosophical assumptions based on the ontological, epistemological and the methodological dimensions shared by a research community regarding reality and knowledge. The ontological dimension focuses on the belief about the nature of reality, the epistemological dimension anchors on the ways of knowing reality, whereas the methodological dimension is concerned with how reality should be dealt with. The set of assumptions and beliefs affect every decision that is taken in a study. The ontological dimension consists of the belief about reality and considers whether there single reality or multiple realities. Therefore, ontology can be described as the perspective from which the researcher conceptualises the world. The belief that reality exists objectively regardless of the researcher's observation ascribes to a realist perspective. The realist perspective is that one truth exists, it does not change, and it is objective and generalizable. The relativist perspective asserts that multiple realities exist, shaped by context, the truth evolves and changes and only relevant to a similar context (Hudson and Ozanne, 1988). The epistemological dimension refers to the nature of knowledge, its scope, how it can be acquired, and the extent of its validity and limits. It establishes the relationship between the researcher and the researched reality (McKerchar, 2008). The methodological dimension entails how knowledge is discovered and analysed in a systematic way. The methodology is guided by the ontological and epistemological beliefs. Research paradigm influences the choice of methods in a research process (Bahari, 2010). Different research paradigms when applied to the same problem may yield different results (Bahari, 2010). Considered metaphorically, a paradigm is a lens through which a researcher chooses to view the world. Therefore, different lenses capture different aspects of reality and at different levels of detail.

The choice of a research paradigm is driven by the nature of the research problem (Bahari, 2010). These research paradigms represent a range of choices from which a research process can be structured. Each paradigm can reveal certain aspects of an object under study. The choice of a paradigm is important as it determines the features of the object under study that would be exposed or measured. It is, therefore, imperative that any research endeavour explicitly states the researcher's view of the object under study (ontology), whether from a realist perspective or a relativist perspective. The ontological dimension would then translate into epistemological foundations with regards to how the knowledge can be acquired. The ontological and the epistemological positions determine the appropriate methodology to be followed to discover the required knowledge. In turn, the methodological decisions pave for the selection of the appropriate research methods and techniques associated with data collection, analysis and processing.

In the next subsection, a brief discussion of two common paradigms in Information Systems is presented and the selected paradigm described.

4.2.1 Interpretivist paradigm

The interpretivist paradigm looks for the culturally derived and historically situated interpretation of the social reality (Bertram and Christiansen, 2014). According to Neuman (2011), the goal of the interpretivist paradigm is to understand and interpret the meaning in human behaviour, rather than to generalise and predict causes and effects. It is important to understand the motives, meaning, reasons, and other subjective experiences within a specific context and time confinement (Bertram and Christiansen, 2014). It is key to view reality as socially constructed, and subjective to understand the relativity of knowledge (Carson *et al*., 2001). The researcher starts with some prior insight of the research context, but assumes it as inadequate in exploring the phenomena. The reality is perceived as complex, multiple, and unpredictable in nature (Greener, 2008). The interpretivist paradigm allows the researcher to view the world from the informant's perspective. The ontological foundation of interpretivist paradigm is concerned about the subjective reality, not the objective reality (McKenna *et al*., 2011). Therefore, the researcher should be flexible and open to new knowledge as the study unfolds. The use of such an emergent and collaborative approach is consistent with the interpretive belief that humans can adapt and that no one can gain prior knowledge of time and context bound realities (Hudson and Ozanne, 1988). The researcher is expected to be immersed in the object under study and interfering and influencing the object under study is accepted. In summary, the interpretive paradigm investigates the object under study through interpretation of human perceptions, beliefs, experiences, shared values and meanings in point in time and at a specific social context (Geels, 2010). The interpretivist tends to use case studies, ethnographic studies, phenomenographic studies, and ethnomethodological studies as their research methods (Weber, 2004). The main dimensions of the interpretivist paradigm are presented in Table 4-1.

Table 4-1: Interpretivist paradigm dimensions

PARADIGM	ONTOLOGICAL DIMENSION	EPISTEMOLOGICAL DIMENSION	METHODOLOGICAL DIMENSION
Interpretivist	• Nature of reality is socially constructed, interpreted and subjective. • Reality evolves and changes according to context, experiences and social and time bonds. • Realities may not be generalised but may be used in similar situations.	• Researcher and the object under study are not separable • Theory develops as the research progresses and is emergent within data • The researcher and what is researched interacts. • Theory is generated by blending abstract concepts and empirical data. • Create theory as opposed to testing theory.	• More biases towards qualitative processes

4.2.2 Positivist paradigm

Reality is separate from the researcher who observes it (De Vos *et al*., 2011; Levin, 1988). The positivist paradigm views reality as an objective phenomenon (Carson *et al*., 2001). There is a single objective reality to any object under study regardless of the researcher's perspective (Hudson and Ozanne, 1988). The world is governed by laws and is ordered and structured. Therefore, reality is stable and can be observed and described from an objective viewpoint, without interfering with the object under study (Levin, 1988). Conducting research takes a controlled and structured approach (Carson *et al*., 2001). The positivist paradigm distinguishes clearly between reason and feeling (Carson *et al*., 2001). Furthermore, a clear distinction between science and personal experience and fact and value judgement is maintained. The reasoning process follows rational and logical approaches to research (Carson *et al*., 2001), thereby allowing rigorous statistical and mathematical processes to be followed to uncover single and objective reality (Oates, 2006; Carson *et al*., 2001). The positivist paradigm tends to use experiments, surveys, and field studies as the research methods (Weber, 2004). The goal of positivist paradigm is to make generalisations about reality. The main dimensions of the positivist paradigm are presented in Table 4-2.

Table 4-2: **Positivist paradigm dimensions**

PARADIGM	ONTOLOGICAL DIMENSION	EPISTEMOLOGICAL DIMENSION	METHODOLOGICAL DIMENSION
Positivist	• Reality is an objective phenomenon. • Social reality is independent of consciousness. • World is governed by physical laws and is ordered and structured.	• Researcher and the object under study are independent of each other. • Reality can be studied independently from the researcher. • Theory is first developed and then tested empirically. • Events are produced by laws therefore they are repeatable • Different researchers processing the same data set should get the same results. • Predictability and control are achievable	• More biases towards quantitative processes

4.3 RESEARCH PARADIGM CHOSEN FOR THE STUDY

The research paradigm followed in this study is predominantly positivist. The study sought to discover what is happening in each organisation in contingent use of systems development. There was objective reality that the study tried to discover independent of the researcher's knowledge of it. The positivist ontology translated into epistemology, where the questions regarding how the researcher viewed contingent use of SDMs were determined. The methodology and the methods of data generation were determined based on the positivist paradigm (Saunders *et al*., 2015; Scotland, 2012). The positivist epistemology is empirical oriented and is underpinned by observations and measurements based on statistical instruments (Saunders *et al*., 2015). This knowledge could either be acquired through experimentation, design and creation or observation. A set of constructs were defined and some inferred from observable facts and events, and were assumed to influence the object under study. The contingent use of SDMs is not based on random occurrences. There are cause-and-effect relationships in the contingent use of SDMs and these were exposed during rigorous data processing. Based on the ontological and epistemological underpinnings of the positivist paradigm, the research methodology was selected as explained in the next section.

4.4 RESEARCH METHODOLOGY

According to Schwardt (2007), research methodology refers to the framework for organising and structuring an inquiry. The selection of a research methodology is based on several factors such as previous research conducted, theory development in the field, formulated hypotheses concerning object under study, research questions and objectives formulated (Creswell, 2009). In this study, the research methodology selected is predominantly informed by the positivist ontological and epistemological views. The positivist underpinned research methodology, provides the rules to describe the world view (Tekin and Kotaman, 2013; Neuman, 2011). The positivist framework characterizes the theoretical position to reveal the relevant aspects of the object under study and addresses the purpose of the research as proposed by the researcher.

As the initial methodological step, a conceptual framework was developed through a thorough review of pertinent literature. The detailed literature review served as a theory building platform, that established the context of the problem and the extent to which other researchers have contributed towards addressing it. Knowledge gaps were identified in contingent use of SDMs in South Africa and research objectives and hypotheses formulated.

4.5 RESEARCH METHOD

The positivist ontology views reality as objective and knowable. Oates (2006) states that according to this view, research is value free and relies on precise observation and verifiable measurement. Therefore, positivist methodology focuses on objectivity in data collection, the research method and related techniques used to collect data are chosen to minimise potential bias and error at the same time allow generalisation (Dillman *et al*., 2014; Oates, 2006). The typical research methods related to the positivist paradigm include approaches, such as design and create method, experimental method, and surveys. Each is selected according to its appropriateness to the formulated research questions. The following three subsections describe some of the common methods in Information Systems and Computer Science research.

4.5.1 Design and create method

Oates (2006) explains that the design and create method focuses on the development of systems artefacts. It emphasises issues, concerns and interests that are explored and manifested through the actual creation of systems artefacts. The systems artefacts facilitate the transition between concrete object views and abstract views about the object. The system artefact function as a means of realising a concrete object that should be perceived, recognized and understood (Oates, 2006). Schön (1995) and Oates (2006) concur in that a researcher’s understanding and knowledge are related to a perspective situated within the process of praxis. This design and create method favours both the ontological and the praxiological dimensions of the philosophical view, at the same time supporting the epistemological dimension.

Within the frame of design and create method, artefacts are conceived as core in addressing research questions on a particular topic. An artefact or a product is an embodiment of theory and the body of knowledge (Oates, 2006). The design and create method can be employed to address Alter's (2017) call to consider development of artefacts as theory generation. The artefact may be in the form of models, prototypes, or complete products (Gregor, 2006). According to Oates (2006), the contribution of artefacts can be in the form of research constructs, research model, research process and instantiation as proof of applicability of theory generated. Although, it favours the Type V theory, the design and create method fits into all five types of theory identified by Gregor (2006). Research in SDMs involves the creation of the models, framework, the development process and products (Iivari and Huisman, 2007)

4.5.2 Experimental method

The relation between cause and effect, is central to the experimental method. The experimental method is more biased towards the Gregor (2006) Type IV theory. It focuses on prediction and explanation. Gregor (2006) indicates that this approach generates testable propositions and explanations on cause and effect relations. The experimental method aims at investigating the cause and effect relationships and seeks to prove or disprove a theory (Oates, 2006). Experiments are conducted and observations made with clear identification of factors that are on the causal set and the ones on the effect set. The factors should not be influenced by any other factors not regarded as either in the causal or effect set. That is, several experiments may be carried out in order to test a certain assumption. In each experiment, relevant adjustments are done to the cause variables and the perturbation on the effects variables observed. Several SDMs may be tested in a specific systems development project and the results observed. The observation may lead to identification of the most appropriate SDM for that particular project.

4.5.3 Survey method

A survey is a systematic method of collecting data from a population of interest (Dillman *et al.*, 2014; Oates, 2006). It tends to be quantitative in nature though it may be qualitative also. It focuses on collecting data from a sample of the population, such that results are representative of the population within a certain degree of error (Dillman *et al.*, 2014; Oates, 2006). Typically, a survey method is related to data collection from a large sample and seeks to discover relationships that are common across the larger population to provide generalizable statements about the object of study (Oates, 2006). A survey method is used to generate the same type of data from a representative set of the sample frame, using a standard structured data generating technique such as a questionnaire (Tekin and, Kotaman, 2013; Check and Schutt, 2011; Oates, 2006). A survey method can be deployed in two main forms, the questionnaire or the interview. The implementation of these forms includes written questionnaire, structured face-to-face interviews, structured telephone interviews, mailed questionnaires, online interviews, online questionnaires, email questionnaire (Maree, 2007; Oates, 2006). Each technique is appropriate depending on the research questions being addressed. The survey method can also be used in other research paradigms.

4.5.4 Research method selected for the study

A survey is a systematic method of collecting data from a population of interest. It focuses on collecting information from a representative sample of the population such that the results are representative of the population within a certain degree of error (Oates, 2006). The study collected data from a large group of systems development organisations in South Africa with the aim of generalising the results. The study employed a survey as the main research method. This enabled the collection of data from a large group of professionals involved in system development such as systems development project managers, systems developers, systems analysts and systems designers, and systems testers.

The organisations targeted include both those whose area of competence is software systems development and others whose area of competence is not necessarily systems development, but develop systems indoor or outsource systems development. This allowed for the investigation of the contingent use of SDMs at organisational, systems development project and individual levels. The survey method allowed the collection of large volume of data in a short space of time at a relatively low cost. The standardised and structured collection of data from a large representative sample minimises bias and allows generalised conclusions to the larger population (Dillman *et al*., 2014; Tekin and, Kotaman, 2013; Check and Schutt, 2011; Oates, 2006). The study developed a contingent use of SDMs conceptual model and the data collected was used to generalise its applicability to the systems development organisations in South Africa. The survey is replicable and allows method triangulation on data collection. Repeatability is one of the characteristics of the positivist paradigm (Dillman *et al*., 2014; Tekin and, Kotaman, 2013; Hudson and Ozanne, 1988) that clearly separates the influence of the researcher and the object of study. Additionally, the design and create method was used in the development of the contingent use of SDMs conceptual model which is an artefact (Alter, 2017). However, it is not the predominant method. The experimental method would be difficult as organisations cannot allow manipulation of SDMs as this may have financial, legal, operational and organisational implications. The survey method was used to gather data to understand the status quo in contingent use of SDMs in South Africa.

4.6 RESEARCH DESIGN

A research design is the overall strategy chosen to integrate the different components of the study in a coherent and logical way (De Vaus, 2001). It indicates how a study is conducted and is a framework for guiding a study and protecting the validity of findings by imposing controls over inhibiting factors (Burns and Grove, 2003). Therefore, the purpose of a research design is to ensure that the evidence obtained enables the researcher to address the research objectives in an unambiguous and convincing manner. That is, it allows the researcher to make valid claims on the findings on the contingent use of SDMs in South Africa.

The research design comprises the data collection instrument design, sample frame development and statistical analysis. The statistical analysis covers both the descriptive and inferential statistics. The study developed a contingent use of SDMs conceptual model in Chapter 3. This contingent use of SDMs conceptual model is used to investigate the contingent use of SDMs in South Africa. The contingent use of SDMs conceptual model was constructed as a hybrid of three adoption models: the Diffusion of innovation model (DOI) (Rogers, 2003); the Task-Technology Fit (TTF) (Goodhue, 1985) and the Technology Acceptance Model (TAM) (Venkatesh and Davis, 1996).

The research method adopted for the study was a survey, which is linked to the positivistic paradigm. The survey method, the search for causal links and statistical inference characterize the research methods used in this study, makes it mainly quantitative in nature. The next subsection discusses the survey preparation.

4.6.1 Survey preplanning

To address the formulated research objectives, a survey was considered as the most appropriate data collection method. A questionnaire was used to collect the necessary information from the sample frame.

4.6.2 Target population

To establish the study sample, a sample frame was drawn. The sample frame was defined from the accessible population, which in turn was derived from the operational definition of the target population. The target population was difficult to identify, because in practice a variety of professionals are involved in systems development. There are systems architects, systems analysts, systems engineers, systems developers, systems designers, programmers, testers and systems development project managers. Therefore, the target population consisted of a single professional in each organisation involved in systems development because, an organization was considered as the necessary condition for any systems development professional to be selected as a respondent.

To identify the target population, the Standard Industrial Classification (SIC) was reviewed and it was found that it did not classify systems development into any Industrial sector (DTI, 2008). This created some challenges in the classification of organizations into micro, small, medium or large enterprises. In the absence of an appropriate South African classification, Wendler's (2016) categorisation of organisations was adopted. The categorization led to the following classification: 1-10 employees were classified as micro enterprises; 11-50 employees classified as small enterprises, 51-100 employees classified as lower medium enterprises, 101-250 employees classified as upper medium enterprises and more than 251 as large enterprises (Wendler, 2016). The classification is inclusive of organisations that are involved in systems development as their principal area of competence and those that are involved in systems development, but not as a major area of an organisation's competence.

After completing the classification of organizations, the next step was to derive the operational definition of a systems development organization. The Intellectual Property Commission (CIPC) was reviewed again. However, according to the Companies and Intellectual Property Commission (CIPC), there is no regulatory body for Software Engineering, Software Development or Systems Development Industry (DTI, 2008; MICT SETA, 2017). In the existing company registration register, services are not explicitly stated. According to Department of Trade and Industry (DTI), the classification of companies does not indicate any organisation as involved in software development category (DTI, 2008). DTI does not allocate a separate economic sector for systems development industry in its economic classification. Instead, systems development are placed within any Industrial sector such as Finance sector, Business sector, or Other Services sector (MICT SETA, 2017).

Similarly, the South African yellow pages telephone directory did not provide explicit description of each organisation's services nor its area of competence. There was no single directory or repository for the systems development industry where all systems development organisations could be extracted. There were two challenges, the first, the operational definition of the systems development organisations, and the second was the repository where these organisations may be found.

Confronted with these two challenges, the study identified characteristics of what constitutes organisations involved in SDMs. Characteristics of systems development, that were common to all (or, at least, most of) the organisations were identified. The characteristics entailed the description of systems development, software development, software engineering or software programming and embedded systems development. The classification according to this description provided an inclusion and exclusion criterion to determine the sample frame. The description criterion covered two categories of organisations; the first set consisted of those organisations whose main competence is in systems development, software development, software engineering, software programming, or embedded systems development; the second set of organisations comprised those which are simply involved in systems development, software development, software engineering, software programming, or embedded systems development. The second set encompassed telecommunications industry, financial institutions, learning institutions and retail industry. The level of outsourcing done by financial institutions, learning institutions and retail industries made it difficult to include them in the research sample. Their inclusion created duplication of the sample units as in some cases a whole team is outsourced. Therefore, the study considered the sample frame that consists of those organisations that have competence in systems development and the telecommunication industry.

Having addressed the identification of the sample frame, the next step was to determine the boundaries for the sample frame. The Joburg Centre for Software Engineering (JCSE) is a well-established organisation formed through the collaboration between government, academia and industry to assist organisations to improve processes through the implementation of multidisciplinary best practices in South Africa and the rest of the continent (JCSE, 2016). Every year, the JCSE organises Agile Africa conference which attracts not only organisations involved in the use of agile SDM class, but also those that are involved in the traditional SDM class or both. The JCSE database of systems development organisations was identified as the possible source of respondents. However, the access to JCSE organisations involved in systems development through the JCSE database failed due to the policy on disclosure of organisational information.

In the absence of an all-encompassing repository of systems development organisations in South Africa, the next credible sources of organisations were search engines. Google was selected to search for information about systems development organisations in South Africa. However, the information provided by a search engine platform needs to be evaluated first. The boundaries for the sample frame were drawn based on the description of the inclusion and exclusion criteria to extract the relevant organisations from Google, and other websites dedicated to organisations' listing.

Most systems development organisations have web sites as a visibility strategy or an e-presence strategy. The use of a search engine in searching is important when relevance ranking is essential. The search key phrases of the query were: "systems development organisations/companies in South Africa, software development organisations/companies in South Africa, software organisations/companies in South Africa, software engineering organisations/companies in South Africa or software programming organisations/companies in South Africa, embedded systems development in South Africa”. The search generated and returned 761 000 results inclusive of the title of the webpage, the URL of the result’s webpage and the snippet. A total of 2359 organisational websites was found, however only 341 had contactable information which could be used for verifying their actual existence. The verification process was done through consultation with the South African business yellow pages directory and the Companies and Intellectual Property Commission (CIPC) Company register. Among the existing company websites, three organisations offering organisations website directories were found. The three companies were Bizcommunity, ITWeb and Rainbow nation. The bizcommunity.com website specialise in company content marketing. ITWeb.co.za website provided IT/Software companies’ profiles; and the rainbownation.com website offered a directory of South African programming organisations’ websites.

The number of organisations is presented in Table 4-3. Duplicates organisations were eliminated by listing each company once.

Table 4-3: Systems development organisations in South Africa from the four sources

WEB DIRECTORY COMPANY	NUMBER OF SYSTEMS DEVELOPMENT ORGANISATIONS IDENTIFIED
Bizcommunity (bizcommunity.com)	292
IT Web (ITWeb.co.za)	250
Rainbow nation (Rainbownation.com)	167
Google search	341
Total organisations with redundancy	1050
Total of organisations with duplicated names	477
Total organisations without redundancy	**573**

These organisations are dotted throughout the nine provinces of South Africa with a high concentration in Gauteng, KwaZulu-Natal and the Western Cape. A survey was selected as the most appropriate method for collecting relevant study data from the sample frame. The minimum selection criteria were based on the participant’s involvement in systems development. Therefore, within the organisation, the questionnaire targeted systems development project managers as priority and systems analysts, systems developers, systems designers, and other professionals involved in systems development as second option. Participation was voluntary and the respondents had the right to withdraw their participation at any time.

4.6.3 Survey planning and design

The approval of the research proposal and the registration of the title were key to deciding the type of data needed and the appropriate instrument to collect the relevant data that would respond to the formulated objectives. A survey questionnaire was used to collect data. There are many modes to administer a questionnaire such as face to face, telephonic, or self-administered. Each has its own advantages and disadvantages. The study adopted self-administered questionnaire administration. The self-administered questionnaire minimized the major source of bias in the responses as it created anonymity conditions. The respondent had time to read and analyse each question carefully before responding. This implied that the respondent could check for information to reinforce their information recall if necessary.

4.6.4 Questionnaire design

A questionnaire is a pre-formulated written set of questions to which respondents record their responses in a pre-determined order (Sekaran and Bougie, 2010). The respondents are asked to respond to questions providing a researcher with standardized data (Sekaran, and Bougie, 2010) that can be analysed and interpreted. The self-completion questionnaire instrument is easy to follow and its questions are easy to answer (Bryman and Bell, 2011). The purpose of the questionnaire design was to develop an instrument to gather reliable and relatively unbiased data from a representative sample of the target population. The questionnaire is a medium of communication between a researcher and a respondent. This conversation between the researcher and the respondent should maintain conceptual congruency between the two communicating parties. The respondent must understand and interpret the question the same way as the researcher intended. The misunderstanding, incomplete concept coverage, inconsistent interpretations, and context insensitivity may increase the non-sampling errors in data collection. To avoid challenges in questionnaire design, Sekaran and Bougie (2010) propose principles of questionnaire design. The principles of questionnaire design relate to how the questions are worded and measured, and how the entire questionnaire is organized. The principles seek to minimise respondent biases and measurement errors. The next subsection describes the three principles of questionnaire design.

4.6.4.1 Wording of the questionnaire questions

The language of the questionnaire approximated the level of understanding of the respondents (Sekaran and Bougie, 2010). The choice of words was also pitched to the assumed linguistic proficiency of the respondents, as well as their frames of references. The main objective was to word the questions in a way that could be understood by all respondents. Questionnaire question wording is a stage in the design of valid and reliable questions (Foddy, 1994). The first stage in the design of valid and reliable questions entails the researchers' being clear about the data required and designs a question, the second stage involves the respondent decoding the question in the way the researcher intended, the third stage consists of the respondent answering the question and the last stage involves the researcher decoding the answer in the way the respondent intended (Foddy, 1994). Tourangeau (1984) investigated the question-and-answer process a respondent goes through when answering a question. The process starts when the respondent reads and comprehends the question, the second phase involves the respondent's retrieval of the necessary information from long-term memory, followed by the phase whereby the respondent makes a judgement about the information needed to answer the question, and the final phase is when the respondent provides the answer to the question. The stages in the design of valid and reliable questions and the question-and-answer process were taken into consideration in the wording of the questionnaire question items.

According to Foddy (1994), Tourangeau (1984) and McKerchar (2008), the following assumptions can be made about standard and structured questionnaire questions:

- Researcher formulates clear questions that collect required data.
- All respondents understand the questions in the way intended by the researcher.
- The questions are asking for information that the respondents have and can retrieve.
- The wording of questions provides respondents with all the necessary information they require to answer them in the way intended by the researcher.

Foddy (1994) stated that the researcher should be clear of the data required. In the study the required data was generated based on the constructs in the contingent use of SDMs conceptual model, addressing the research objectives on the contingent use of SDMs in South Africa. The nature of the variables selected included both subjective feelings and objective facts. The subjective variables targeted respondent's beliefs, perceptions, and attitudes towards the aspects of the object under study. The objective variables were captured through single direct questions. The purpose of each question was carefully considered so that the variables were adequately measured.

If some questions are either not understood or are interpreted differently by the respondent, then they would not meet the criteria for validity and reliability (McKerchar, 2008; Foody, 1994). The wording also determines whether the question is open-ended or closed-ended. Open-ended questions were used to capture categories that could have been left out by the closed ended questions. The closed questions required the respondents to make choices among a set of alternatives provided. Most of the questionnaire questions were closed ended and used the six point Likert-like rating scales.

Closed-ended questions help the respondents to go through the question-and-answer process (Tourangeau, 1984) faster as the several alternatives helped the information retrieval phase of the process. Closed ended questions also help the researcher to code the information easily for subsequent analysis. Ambiguity in questions which includes questions that have more than one answer was avoided. The questions were also checked to eliminate Phrasing that lead the respondents to give responses that the researcher wants were also eliminated. The last aspect in wording was the length of each question.

4.6.4.2 Measurement in questionnaire design

The principles of measurement ensure that the data collected are appropriate to test the hypotheses or address the research questions. These refer to the scales and scaling techniques used in measuring variables. There are at least two types of variables, the objective variables that can be measured objectively using physical measurements and the subjective variables that are hard to accurately measure (Zwanenburg, 2015). The subjective variables were operationalised by identifying their dimensions and converting these dimensions into observable and measurable elements (Zwanenburg, 2015). The four scales that can be applied to measure variables are the nominal, ordinal, interval, and ratio scales. There is progressive increase in precision in quantifying data from the nominal to the ratio scales (Sekaran and Bougie, 2010). All four scales were considered in the design of the questionnaire, with the ordinal scale being the most predominant. The rating scales selected as the most appropriate for the design of the questionnaire were the unbalanced Likert-like scale and the unbalanced numerical scale due to the nature of the data required for the study.

4.6.4.3 The dependent variable

The focal dependent variable of the study was the contingent use of SDMs. This dependent variable was measured through the developers' perception on how to achieve an ideal fit between an SDM and the systems development contextual stressors. The contingent use of SDMs variable is subjective, because it is dependent on the developers' perception of SDM level of variation in SDM implementation during the systems development project. To address the challenge of subjectivity and bias responses, internal reliability and validity of each question was assessed using the Cronbach's α. A set of five Likert type items was developed. Boone and Boone (2012) indicate that multiple Likert scale items can be combined to generate a composite variable appropriate for several statistical techniques including regression analysis. The composite variable can be generated by adding multiple item values (Boone and Boone, 2012), or calculating the average of the multiple item values (Keith, 2015; Boone and Boone, 2012). To apply regression analysis on the Likert type items, a composite independent variable (e.g. SDM relative advantage) was generated by calculating the mean from its Likert type items.

4.6.4.4 Consideration of data gathering ethics

A consent form was prepared which also served as an introduction letter by stating the identity of the researcher and of the institutions he is associated with. The letter also outlined the purpose of the research and the questionnaire. The questionnaire only sought to gather information on systems development practices, rather than respondents' confidential details. The respondents were assured that the data they provided would be coded to avoid it being linked to any specific name, be it that of an organization or individual. Respondents were assured that anonymity would be maintained when the results of the research are reported in scientific journals or presented at local or international conferences. The rights of the respondents were explained. To provide frames of reference, some concepts were defined in the letter.

The questionnaire was organised into nine parts and simple and clear instructions were provided on how to complete the items in each part. The questionnaire form informed the respondents that they could request for a copy of the research's findings. The questionnaire form concluded with an expression of appreciation for the participation of the respondents.

4.6.5 Questionnaire development and pretesting

Based on the principles of questionnaire design, the first step in the development of the questionnaire, entailed the generation of all potential items that would respond to the constructs in the contingent use of SDMs conceptual model. The items were also generated through in-depth and intensive literature review and consultation with experts and the promoters. Consultation with experts on questionnaire design in the field of Software Engineering, Information Systems Development was done. The experts examined acceptability of the questionnaire items and related it with the research objectives of the study and aligned the questionnaire items to the contingent use of SDMs conceptual model constructs. The questionnaire items were then formulated into questionnaire questions.

The formulation of the questionnaire questions evolved through an iterative incremental refinement process between the researcher and the promoters. The initial draft of the set of questionnaire questions was sent to two questionnaire design experts in Software Engineering and two colleagues in the Department of Business Information Systems to check whether it was measuring what it was intended to measure and to evaluate the extent to which items on the questionnaire were representative of the domain under study. This was a step towards establishing validity and reliability as advocated by Foddy (1994). One double barrel question was identified and refined. Also, three questions which were irrelevant were removed. When it passed the face validity, content validity (Markus and Borsboom, 2013) and cognitive load, it was then sent to two language experts to eliminate ambiguity, check phraseology accuracy and to edit grammar and do language proofreading. When the questionnaire met the linguistic requirements, the next step was to carry out a pilot test.

4.6.6 Questionnaire piloting

Questionnaire piloting ensured the clarity and acceptability of questionnaire items. The selected pilot respondents were allowed to make open ended comments and to propose additional items or issues not included in the draft questionnaire. The variability of answers was checked and analysed to establish the level of consistency in questions comprehension and interpretation by respondents. The length of the questionnaire was examined in terms of flow of questions, relevance of items to respondents and acceptability to respondents of the information requested.

The five organisations that were selected for the pilot study generated a data set which was not incorporated into the main data set of the study. The questionnaire expert analysed the response patterns in terms of the question-and-answer process (Tourangeau, 1984), the characteristics of valid and reliable questions (Foddy, 1994) and questionnaire design principles (Sekaran and Bougie, 2010). The questionnaire expert certified all questionnaire items as comprehensible, as allowing for a fair recall or average cognitive load, as not threatening the professional status of respondents and as not swaying respondents to specific answers. The average time to complete the questionnaire was approximated at twenty-seven minutes.

4.6.7 Questionnaire structure

The questionnaire consisted of nine parts. The first and the second parts of the questionnaire solicited for factual information. There was no need to search for information or calculate the values since information was easily available to any individual involved in systems development. Some of the questions were reverse scored to avoid response pattern bias. The remaining parts were measured on an unbalanced six-point Likert-like scale. The respondent was expected to have a clear choice and a solid decision as there was no provision for a neutral position. However, to cater for possible missing information, the “Other specify” option was included in some questionnaire items. The model of the designed questionnaire for the study is presented in Table 4-4. The study is anchored on four objectives presented in Chapter 1 as follows:

Research objective 1 (RO1): Examine systems development methodologies in use within the systems development organisations in South Africa.
Research objective 2 (RO2): Identify the contingent use of systems development methodologies in organisations in South Africa.
Research objective 3 (RO3): Identify the critical success factors of the contingent use of systems development methodologies.

Research objective 4 (RO4): Develop a contingent use of SDMs conceptual model and the guidelines for the contingent use of SDMs.

Table 4-4: Design of study questionnaire

Objective	Questionnaire Part	Question	Question item	Reason for question	Question type	Research variable	Source
RO1	Part 1: Personal information	1. What is the primary role you hold in systems development projects?		Determine if the role of an employee has an influence on the SDM in use.	The work position is determined by selecting from a list of systems development role categories: -systems development manager/leader, -systems developer, -systems analyst, -systems designer, - other specified category if not listed.	Role category	Own construction
		2. What other role do you assume in systems development in your organisation?		Determine whether employees assume multiple roles.	Multiple roles are determined by selecting from a list of systems development roles: -systems development manager/leader, -systems developer, -systems analyst, -systems designer, - other specified position if not listed.	Degree of work specialisation	Own construction
		3. How many years of experience do you have in systems development projects?		To determine if experience has an impact on the SDM in use.	The years of experience in systems development projects is indicated by selecting from an interval: 0-5years, 6-10years, 11-15years, 16-20years, 21 or more years.	Experience	Own construction based on Young *et al*. (2016)
		4. What percentage of your the system development project time is spent on selecting an SDM?		To determine the weight of SDM selection process in terms of time allocation.	The respondent indicates the estimated percentage of systems development project time spent on selecting the SDM for a system development project.	SDM selection duration	Own construction
	Part 2: Organisational information	1. What is the size of your organisation in terms of the number of employees?		To determine if the size of an organisation influences the contingent use of SDMs.	The number of employees is determined through multiple choice: 1-10 employees (micro enterprise), 11-50 employees (small enterprise), 51-100 employees (lower medium enterprise) 101-250 employees (upper medium enterprise), 251 or more (large enterprise).	Organisational size	Own construction
		2. What is the size of your systems development team in terms of the number of employees?		To determine if the size of the system development team has influence on the contingent use of SDMs.	The size of the systems development team is determined through multiple choice: 1-5 employees 6-15 employees 16-30 employees 31-50 employees 51 or more	Team size	Own construction based on Young *et al*. (2016)
		3. How many systems development projects may run concurrently in your organisation?		To determine if the number of systems development projects running concurrently has an impact on the contingent use of SDMs.	The respondent indicates the possible number of systems development projects that may run concurrently.	Concurrent systems development	Own construction
	Part 5: Selection, adoption and use of SDMs	3. What is the proportion of systems development projects that are developed using SDM knowledge in your organisation?		To determine the extent of SDM knowledge application in systems development projects.	The vertical use of SDMs is measured by selecting a percentage interval corresponding to the proportion of systems development projects using SDM knowledge: 0%, 1-20%, 21-40%, 41-60%, 61-80%, Over 80%.	Vertical use	Huisman and Iivari (2007)

Objective	Questionnaire Part	Question	Question item	Reason for question	Question type	Research variable	Source
		4. What is the proportion of people who apply SDM knowledge in projects in your organisation?		To determine the extent of SDM use across projects in an organisation.	The horizontal use of SDMs is measured by selecting a percentage interval corresponding to the proportion of people using SDM knowledge: 0%, 1-20%, 21-40%, 41-60%, 61-80%, Over 80%.	Horizontal use	Huisman and Iivari (2007)
	Part 9: SDMs in use and the extent of their usage in organisations.	1. To what extent is your organisation using the following SDM(s)?	Items [a-o]	Identifying SDMs in use in organisations.	The intensity of SDMs in use is assessed using a six-point Numerical scale (1=not used, 6=intensive use). A list of SDMs is provided and an option to specify an SDM is given in case the SDM in use is not included in the list.	SDM in use	Own construction
		2. How long has the current SDM been in use in your organisation?		Need to determine if the organisation uses a base SDM.	The base SDM in an organisation is determined by selecting a time interval from: -less than a year, 1-2 years, 3-5 years, 6-10 years, over 10 years, or do not know. This question is combined with Part 3 question 1, Part 5 question 1 and 2 to determine levels of SDM tailoring and adaptation.	Base SDM	Own construction
RO2	Part 3: Experience with the selecting SDMs.	1. When do you select an SDM or a combination of SDMs for systems development projects?		Determine if an organisation uses SDMs contingently.	The SDM contingency-based selection is determined through multiple choice: -selection at outset, -selection at phases of the SDLC, -selection at any point of the SDLC, -not selected but created.	SDM adopting strategies	Own construction
		2. How do you select an SDM or a combination of SDMs for a systems development project in your organisation?		Determine the type of SDM selection practices.	The SDM adoption practice is determined by selecting from multiple choice options: -SDM adoption guidelines or/and policies, -SDM adoption frameworks, -SDM adoption best practice and experience, -any other, specified practice if not included in the list.	SDM adopting approaches	Own construction
	Part 4: System development selection criteria.	1. What is the rating of the following statements considering the criteria used to select an SDM for a project in your organisation?	[a, e-i, p] [b-d, j-o, q]	Need to determine the influence of the systems development project contextual stressors on the contingent use of SDMs. Need to determine the influence of the SDM capability on the contingent use of SDMs.	The relationship between contextual stressors and the contingent use SDMs is assessed using a six-point Likert-like scale (1=totally disagree, 6=totally agree). The contextual factors considered are the: project type, team size, team structure, problem domain, culture, politics, and market window. The relationship between SDM capability and the selection of an SDM is assessed using a six-point Likert-like scale (1=totally disagree, 6=totally agree). The SDM capability factors considered are: the system criticality, expertise, experience, requirements management, early discovery of defects, costs, familiarity, and compliance to standards.	SDM fit assessment	Own construction
		3. How do you rate the following statements based on the practices used by your organisation to justify the selection of an SDM from a repository of available SDMs?	[a-k]	Need to determine relative advantage determinants for the contingent use of SDMs.	The relative advantage of an SDM is assessed using a six-point Likert-like scale (1=totally disagree, 6=totally agree). The statements cover the following performance areas of SDMs: requirements management, failure history, success history, communication and coordination, budget and schedule, defect detection and removal,	SDM relative advantage	Own construction based on DOI (Rogers, 2003)

Objective	Questionnaire Part	Question	Question item	Reason for question	Question type	Research variable	Source
					contextual appropriateness, and complying with standards.		
	Part 5: Selection adoption and use of SDMs.	1. How responsive is your organisation in selection and adopting a new SDM?		Need to determine if the organisation's adopter category has an influence on contingent use of SDMs.	Organisation's innovativeness is determined through multiple choice: -market leader (innovators) -market follower (early adopters) -conservative (majority) -static (laggards)	SDM adopting culture	Khalifa and Verner (2000)
		2. Which option (s) describe (s) the standard practice on SDM selection in your organisation?		Need to determine the extent of contingent use of SDMs.	The contingency-based SDMs selection practice is determined through multiple choice: -one SDM selected, -set of SDMs selected, -in-house developed SDM, -selected and adapted SDM, -do not use any SDM.	SDM adopting practices	Own construction
		3. To what extent do you deviate from the prescription of the SDM in use in your organisation?	[a-e]	To determine the level of contingent use of SDMs in organisations.	The extent of contingent use of SDMs is assessed using a composite score derived from a six-point Likert-like scale (1=totally disagree, 6=totally agree).	Contingent use of SDMs	Own construction
	Part 7: Systems development project product quality measurement after the use of selected SDM/methodologies.	1. With reference to practices in your organisation, how do you rate the following statements relating to the quality of developed systems?	[a-m]	To determine the influence of the systems development project quality measures in evaluating SDMs.	The perceptions of SDM capability on quality of a system developed is assessed using a six-point Likert-like scale (1=totally disagree, 6=totally agree). The statements cover the following system quality areas: meeting requirements, meeting implied requirements, response time that meet user expectations, accurate output, ease of use, fault tolerance, reliability, failure recovery, performance consistency and scalability.	SDM ex-post success	Own construction
	Part 8: Strategies matching the SDMs to systems development projects in an organisation.	1. How do you rate the following statements when matching the current SDM with the problem domain characteristics of the system development project in your organisation?	Items [a-h]	To determine the fit perception between the SDM and the systems development project domain characteristics.	The relative SDM fit with systems development project characteristics is assessed using a six-point Likert-like scale (1=totally disagree, 6=totally agree). The statements include fitness for purpose, fitness of purpose, contingency approach, SDM engineering.	SDM compatibility	Own construction
RO3	Part 6: Success measure for adopted SDM (s).	1. How do you rate the following statements when evaluating the success of an adopted SDM?	[a-h]	To determine if SDM success measure influences the contingent use of SDMs.	The influence of SDM success factors is assessed using a six-point Likert-like scale (1=totally disagree, 6=totally agree) based on systems development project budget, time, schedule, user satisfaction and business value.	SDM success measure	Own construction based on Khalifa and Verner (2000)
		2. How do you rate your success on adopted SDM/ (s)?		To determine if the SDM success rate influence the contingent use of SDMs.	The SDM success rate is determined by selecting a success rate percentage. 0%, 1-20%, 21-40%, 41-60%, 61-80%, Over 80%.	SDM Success rate	Own construction
RO4	Part 1 to Part 9	All					

4.6.8 Questionnaire administration

A one-page request letter was sent to organisations explaining the purpose of the survey and requesting the prospective respondents to participate. The clause to protect the identities of respondents and their organisations was included in the request letter. The letter also provided assurance of confidentiality of the information and stated the rights of the respondents. In each organisation, a contact person was identified. The contact person was referred to in this study as a gatekeeper and where possible correspondence went through the gatekeeper.

The initial plan was for the researcher to administer questionnaires in person. This was envisaged to improve response rate from the study population through personal contact and face to face request for cooperation. The initial communication was established through a telephone conversation. The researcher called each company from the study sample and made introductions. The first contact was to request for permission and the protocols necessary to administer a questionnaire in each target organisation. The name of the gatekeeper in each company and contact details were recorded. The researcher explained to the gatekeeper the purpose of the questionnaire and the rights of the participants. The gatekeeper then indicated whether the organisation may participate or not. Each organisation unwilling to participate was requested to state the reasons for declining the invitation.

Mixed modes of contact were used to request a total of 573 organisations eligible to participate in the survey. A total of 558 (97.4%) organisations were contacted first telephonically to request their participation in the survey. The remaining 2.6% were contacted via email. The refusal rate was 35.6% of the contacted eligible organisations. A questionnaire package consisting of a consent letter and a self-administered questionnaire was sent to each one of the 369 eligible organisations that agreed to participate in the survey. The first preference was the systems development project manager. However, in the case of the manager not being available, other systems development professional were considered to complete the questionnaire. Email and telephone follow-ups were made to the organisations that agreed to participate to increase the response rate. Finally, a total of 162 questionnaires were completed and returned giving a response rate of 28.3%, which is acceptable (Sekaran and Bougie, 2010). However, only 27.1% were usable and 1.2% were discarded due to missing data. The questionnaire administration process is shown in Figure 4-1.

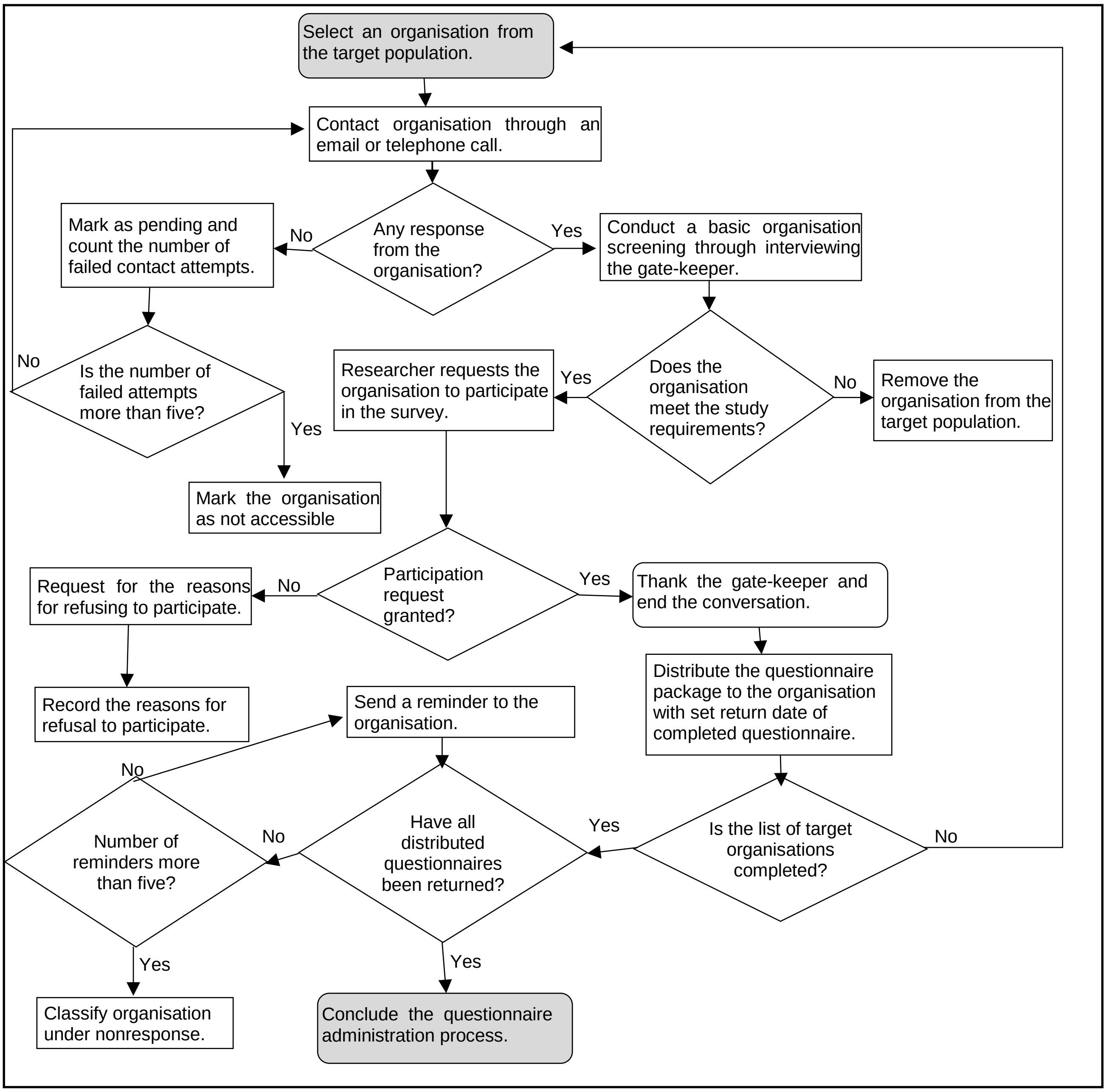


Figure 4-1: Questionnaire administration process

The data collection covered all organisations identified as the target population, in this case 573. The telephonic communication established with the target organisations requesting them to participate was successful, however, it revealed some information related to the reluctance to participate in data collection. The information included decisions at organisational level that may be legal or/and managerial related. The legal included organisational policies on nondisclosure of organisational information. There would be a need to consult with legal counsel for some organisations to participate, which could have financial implications. At managerial level, participation was seen by some as distracting the systems development team from focusing on the critical tasks. The task switching from systems development tasks to questionnaire completion was viewed by some as a disturbance that may significantly compromise the product quality and increase the development time.

Some of the reasons stated by the prospective respondents at individual level were the tight deadlines and demanding requirements. Of course, some organisations and individual developers did not state any reasons; they just declined resulting in a refusal rate of 35.6%.

4.7 DATA ANALYSIS AND INTERPRETATION

Marshall and Rossman (2014) describe data analysis as the process of organising, structuring and deriving meaning to gathered data. The purpose of data analysis irrespective of whether the data is qualitative or quantitative, entails the application of techniques to expose patterns and relationships hidden in the data and drawing valid conclusions (Albers, 2017; Oates, 2006). The data analysis, in this study was performed mainly to investigate the contextual relevance of the following relationship issues: existence of relationships (statistical significance); the degree of the relationships (correlations), and the type of relationship (direction and strength). The research objectives in Chapter 1 and the hypotheses formulated in Chapter 3 informed the data analysis methods to be performed and the planning of the desired data analysis. Results led to the development of the data collection instrument presented in the previous section. The collected data was predominantly quantitative. In the next subsection, quantitative data analysis methods used are discussed.

4.7.1 Quantitative data analysis

The data collected and the form in which it was collected, was determined by the desired results based on the research objectives and hypotheses (Albers, 2017). The research data was predominantly numerical and the analysis was performed in order connect the results of the statistical analysis with the study's objectives and draw meaningful conclusions.

4.7.1.1 Descriptive statistics

The first level of quantitative data analysis was based on descriptive statistics. The descriptive statistics were performed by analysing one variable at a time (univariate analysis). The descriptive statistics constituted the first level of data analysis to summarise the collected data and extract the underlying patterns and understand the distribution of the collected data (Albers, 2017). The statistical measures included the frequency distributions, the mean, median, and the standard deviation.

The frequency distributions were used for the demographic variables and other variables to indicate the characteristics of responses. For example, the respondents were divided into systems development managers/leaders, systems developers, systems analysts, systems designers and other systems development role categories. The grouped frequency distribution was used to summarise data collected from an individual respondent or from a group of respondents classified according to their role categories. The standard deviation statistic showed how much variation or dispersion from the mean existed within the collected data. Most of the questionnaire item responses ranged from 1 representing totally disagree to 6 representing totally agree. A small value of a standard deviation indicated the clustering of points relatively closer to the mean. The large standard deviation indicated the clustering of responses at the opposite extremes of the scale.

4.7.1.1.1 Screening and cleaning the data for analysis

It was essential that the data set was examined and checked for errors before conducting any statistical data analysis. There are basically two common errors on data sets (Pallant, 2016): the data entry errors and the missing data. It was critical that the data was screened and cleaned before any analysis could be performed on it as suggested by Zhang and Yuan (2016).

The data was checked for errors committed during data entry, outliers and missing data. The values that fell outside the range of possible values for the variable were checked and the source of error investigated. The values were either corrected or deleted depending on the source of error. The data capture errors were corrected by referring to the questionnaire that provided the data and editing the data entry. This was done through running the frequencies using IBM SPSS version 25 and checking for the minimum and maximum values for the variables and checking for missing and valid values (Pallant, 2016).

The missing data was considered as the absence of one or more values within the generated study dataset due to a respondent’s failure to provide a response to one or more items in the questionnaire. The missing data may distort study findings if not treated appropriately (Zhang and Yuan, 2016; Enders, 2010). The assessment and treatment of missing data are outlined in the next subsection.

4.7.1.1.2 Assessment and treatment of the missing data

The assessment of the amount of missing data within the data set was viewed in two ways. First, the percentage of missing data values per study respondent and second the percentage of study respondents that omitted certain data values. The study focused on the SDM contingent use related variables. Any item not answered might introduce a bias on the findings. The aim of assessing the pattern of missing data was to identify if data values were missing in a particular manner.

There are three patterns that missing data could be compared (Enders, 2010). The first is the Missing Completely At Random (MCAR), where analysis indicates no patterns in the missing data. The second pattern is Missing At Random (MAR) where missing data values are significantly related to some other variable within the dataset. Lastly, the Not Missing At Random (NMAR), where missing data pattern is related to the variable that has the missing values (Enders, 2010). Each one of these patterns can be treated by using a technique that is selected depending on the research focus. Several techniques for handling the missing data has been developed (Enders, 2010) and they include; techniques such as List-wise deletion, Pairwise deletion, Mean substitution, Regression imputation and many others. Each technique is suitable under certain assumptions about the pattern of missing data (Enders, 2010).

A total of 7 (4.3%) completed questionnaires had key data values missing. In this case the key data values are those data elements that are linked to the SDM contingent use related variables. Further assessment of the missing data patterns revealed that this was a result of omission by respondents. No relationship could be established with other variables which led to the missing data classified as Missing Completely At Random (MCAR). Therefore, the missing data satisfied the assumption of MCAR and the Listwise deletion or Case deletion was the most appropriate technique to produce unbiased estimates during data analysis (Bannon, 2015). The Listwise deletion or Case deletion discards the whole data record, in this case, the whole data from questionnaire having missing key data elements were removed from the data set. However, after removing the cases with missing key data, there were still missing data, but not too serious to lead to a biased analysis. The Exclude case pairwise option was selected to deal with the remaining cases with missing data. The Exclude pairwise option excludes the case only if they are missing data required for specific analysis, but still provides the analysis for which they have the data elements (Pallant, 2016). There were 7 (4.3%) cases with missing key data values for the study and these were removed from the data set. This number is within the acceptable data loss range (Bannon, 2015).

The data screening also established the distribution of data set for parametric assumptions (Gravetter and Wallnau, 2014; Hair *et al*., 2018). The descriptive statistics such as Skewness and Kurtosis values were calculated. The Skewness value provides an indication of the normality of the distribution of the data (Pallant, 2016; Gravetter and Wallnau, 2014; Hair *et al*., 2018). The Skewness range of -1 to 1 was used to evaluate normality of a distribution. In the data set used for the study the calculated Skewness fell within the range -1 and 1. The Kurtosis values indicate the extent to which the data cluster is in the tails of a distribution (Pallant, 2016; Hair *et al*., 2018;

Gravetter and Wallnau, 2014). The Kurtosis values range was set at -2 to +2 as recommended by Gravetter and Wallnau (2014). The calculated Kurtosis values for the data set fell between the acceptable range for the normality of the distribution. The assumption of homoscedasticity was tested using the residuals and the scatterplot showed that the data was randomly distributed.

Pallant (2016) provided guidelines for the Cronbach's alpha level reliability interpretation summarised in Table 4-5. The Cronbach's alpha was used to indicate good internal consistency of the items in the scale, whereas the dimensionality of the scale was determined by Factor Analysis. The Factor Analysis description and application in this study is discussed in the next subsection that deals with the exploration of relationships among variables.

Table 4-5: Cronbach's alpha level reliability interpretation guidelines

CRONBACH'S ALPHA LEVEL	INTERPRETATION
$\alpha \geq .9$	Excellent reliability
$\alpha \geq .8$	Good and preferable reliability
$\alpha \geq .7$	Acceptable reliability
$\alpha \geq .6$	Questionable reliability
$\alpha \geq .5$	Poor reliability
$\alpha \leq .5$	Unacceptable reliability

4.7.1.2 Inferential statistics

A range of statistical techniques that may be used to make inferences about the sample data are covered by inferential statistics. Depending on the intended purpose, these statistical techniques may be classified into three categories: evaluating differences, examining relationships, and making predictions. The study focused on the last two categories of statistical techniques: examining relationships, and making predictions. Inferential statistics were used in this study as a second level of quantitative data analysis. To explore relationships among variables, several statistical techniques were used. The correlation technique was used to describe the strength and direction of the linear relationship between two variables in the study. A Pearson correlation matrix was used to explore the nature, direction, and significance of the bivariate relationships of the variables. The correlation matrix gave an indication of how closely related the variables under investigation were.

Precursor to regression analysis the Principal Component Analysis (PCA) was applied as a data reduction technique whereby many variables were reduced to a more parsimonious meaningful, interpretable set of factors (Albers, 2017; Tabachnick and Fidell, 2013). A PCA was conducted before regression analysis, because of the many variables in the study. These variables could lead to collinearity that would upset the results of regression analysis leading to none of the predictor variables being statistical significant. To verify the suitability of the collected data for PCA, the Kaiser-Meyer-Olkin Measure of Sampling Adequacy (KMO) was generated. It measured the probability value (*p*-value) that the observed outcome would happen. The *p*-value should be 0.6 or above, with values above 0.7 classified as good, values from 0.8 and above classified as great and values above 0.9 are classified as superb and values below 0.5 as unacceptable (Pallant, 2016; Field, 2013). This implies that the smaller the *p*-value the stronger the evidence against the null hypothesis.

The Bartlett's Test of Sphericity was used to test the difference between the correlation matrix for the variables and the structure of the identity matrix which determines the factorability of the correlation matrix. The Bartlett's Test of Sphericity is significant at $p \leq 0.05$. The decision strategy to determine the number of components to be extracted was based on the Kaiser criterion, Parallel analysis and the Screeplot.

The multiple linear regression analysis was performed on the data set to explore the relationships between one continuous dependent variable and many independent variables. The objective of multiple linear regression analysis was to predict the changes in the dependent variable in response to changes in the independent variables (Hair *et al*., 2018; Keith 2015). This was done to test the hypotheses formulated in Chapter 3 and to test the predictive capability of the proposed contingent use of SDMs model developed in the same chapter. According to Keith (2015), the generic form of the linear regression model is:

$$\boldsymbol{Y} = \boldsymbol{\beta_0} + \sum_{\boldsymbol{i=1}}^{\boldsymbol{k}} (\boldsymbol{\beta_i X_i}) + \boldsymbol{\varepsilon} \qquad 4.1$$

where Y is the dependent or explained variable, β_0 is the y-intercept, β_i is the regression coefficient, $X_1,..,X_k$ are the independent or explanatory variables and ε is the error term or disturbance. One advantage of multiple linear regression is that its statistics also capture the magnitude of effects (Keith, 2015). The standardized regression coefficient is one measure of the relative importance of an independent variable in predicting the dependent variable. In this study, the relative importance of each independent variable was accounted for, and the variable that got high values on standardized regression coefficient was a critical success factor (CSF). However, before applying the standard multiple regression analysis, a set of tests were done to test whether the assumptions of the linear regression analysis hold for the study data set. The sample size was adequate based on Tabachnick and Fidell (2013) who suggested a formula to calculate the sample size as $N > 50 + 8m$ (where m is the number of explanatory variables). Specific linear regression assumptions were tested for included normality, multicollinearity and singularity and outliers. The normality of the distribution was tested through checking the Skewness and Kurtosis values of the data, plotting histograms of the standardised regression residuals and plotting the normal P-P plot of regression standardised residuals. The outliers were checked through performing residuals scatterplots that were generated during a regression procedure and observing the standardised residuals more than 3.3 or less than -3.3. The multicollinearity and singularity was tested by checking the correlation matrix for correlation values above 0.9 and the variance inflation factor (VIF) below 10.

The statistical tests provided the basis for interpreting the study data set and drawing valid conclusions from it (Albers, 2017; Tabachnick and Fidell, 2013). The statistical significance does not show the practical significance of results (Coe, 2002). The effect size statistic indicates the practical significance of the results (Albers, 2017). A large effect size reinforces the argument that the finding of statistical significance is meaningful (Albers, 2017; Tabachnick and Fidell, 2013). The effect size therefore allows ascertaining of the practical significance of statistical significance. Based on the guidelines of Cohen (1988), $f^2 \geq 0.02$, $f^2 \geq 0.15$ and $f^2 \geq 0.35$ represent small, medium and large effect sizes respectively. The study uses the Cohen's (1988) f^2 for the effect size calculations and interpretation. The statistical analyses of the data collected are presented in the next two chapters.

4.8 CHAPTER SUMMARY

The literature survey indicated gaps and how some variables were conceptualised, operationalized and measured in the work that preceded this research. This chapter described the research paradigm, research methodology, research design, sampling design and data collection methods. The chapter also discussed the research instrument and operationalization and measurement of variables and tabulated the measurement of parameters for each research objective. The survey research method was selected and a questionnaire developed as a data generation instrument and the data analysis procedures and techniques were also explained. Statistical data analyses are presented in the next two chapters. Chapter 5 presents the descriptive statistics and Chapter 6 inferential statistics.

CHAPTER FIVE: DESCRIPTIVE DATA ANALYSIS

5.1 INTRODUCTION

The research methodology and data generation method were described in Chapter 4. This chapter presents the descriptive analysis of the data collected through the data generation instrument. The descriptive data analysis was used to summarise, interpret and extract the pertinent information from the gathered data. The descriptive data analysis was also performed to uncover patterns in the collected study data including checking for errors, missing data and the normality of the distribution. Chapter 6 presents the characteristics of the empirical study data that were used to determine the appropriate statistical techniques for inferential statistical analysis. The descriptive data analysis in this chapter is organised according to the study's questionnaire structure.

5.2 RESPONDENT PROFILE

A set of four questions were asked to gather information about the individual's position, experience and the average time they spent in the selection of SDMs for systems development projects. The frequency distributions on roles and experience of respondents were tabulated.

5.2.1 Roles assigned to respondents

The level of expertise, the skills set and the experience possessed by each member of the systems development project team is assumed to influence the role each member may be assigned in a systems development project (McLeod and MacDonell, 2011). The role assigned to each member of the development team are task-based. Some of the common roles include systems development project manager/leader, scrum master, product owner, systems analyst, systems developer, systems designer, systems architect, systems tester and systems programmer. In some cases, the naming of roles depends on the type of SDM class adopted. In Structured Systems Analysis and Design Methodology (SSADM), which is a plan-driven class instance, there is a systems development project manager, whereas in Scrum an agile SDM instance at the same level, the role is referred to as scrum master. In some cases, as demonstrated during the study, some systems development project team members assumed multiple roles.

The contribution of each systems development project team member is influenced by the roles they are assigned within a team, systems development project or organisation (McLeod and MacDonell, 2011). Generally, members in an organisation are organised by roles assignments

(McLeod and MacDonell, 2011). These roles are each associated with accountability as a way of handling relationships, be it in a team, systems development project or organisation. The main roles assigned to respondents within organisations are summarised in Figure 5-1. Respondents were presented with a list of four common roles found in systems development. The respondents were asked to choose from the list. However, in cases where all the options did not match the description of the respondent's role, the "Other" field was provided to collect the specific role not included. The dominant respondents were systems development project managers (38.1%), followed by systems analysts who constituted 26.5% of the respondents. The systems designers and those who selected the "Other" option constituted 5.2% and 9.0% of the respondents respectively. In most organisations, senior positions such as systems development project manager and systems analyst fall within the decision-making role category. The distribution of the role categories of the respondents though randomly selected, are consistent with the state of the ICT skills in South Africa as reported in the 2019 JCSE-IITPSA ICT Skills Survey (Schofield and Dwolatzky, 2019). It is logically assumed that the occupation of senior/leadership positions of the respondents influenced the selection, as well as how SDMs were used in systems development projects, hence valid conclusions could be drawn from the responses related to the contingent use of SDMs in South African organisations.

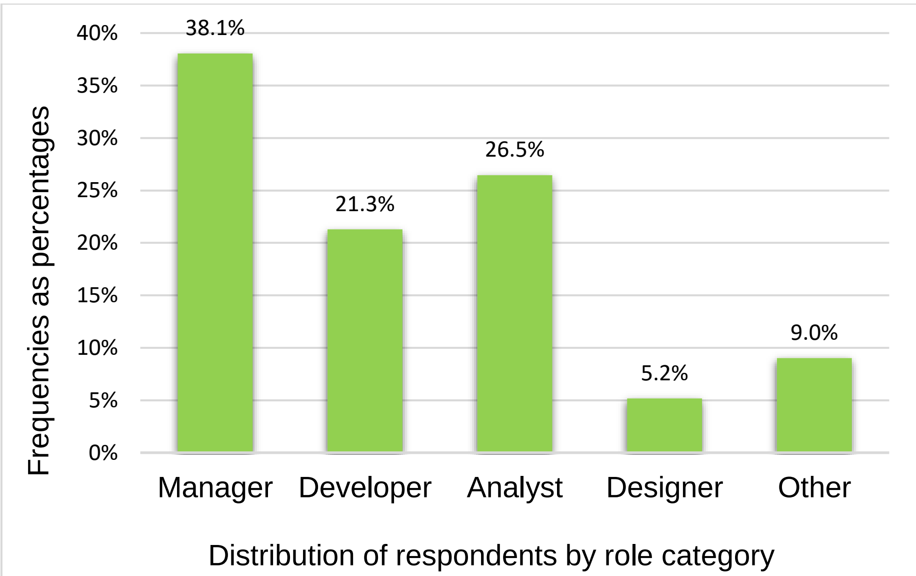


Figure 5-1: The role categories of respondents

When the respondents selected the "Other" option on the role categories, they were further asked to specify their role category titles. The role categories specified in the "Other" option for the primary role categories of the respondents are presented in Table 5-1. Again, Table 5-1 shows that more participants consisted of senior positions in systems development within their organisations. This suggested that the generated data relate to sufficient experiences and influence in systems development projects. Thus, the data can be used to make valid generalizations on systems development projects.

Table 5-1: Respondents that selected the "Other" option on primary role category

Role category	N	Frequency as a percentage
Agile Coach	3	1.9%
Agile Consultant	1	0.6%
Chief Architect	3	1.9%
Product Owner	1	0.6%
Senior Programmer	2	1.3%
Quality Assurance Engineer	2	1.3%
Scrum Master	1	0.6%
Senior Tester	1	0.6%
Total frequency	**14**	**9.0%**

5.2.2 Respondents that held more than one role category

The respondents were requested to indicate if they held more than one role category within their organisation. More than a fifth (24.5%) of the respondents indicated that they held at least more than one role category in their organisation at the time of the study as presented in Table 5-2. The systems developer had the highest percentage of 12.9%, followed by the systems analyst with a 5.2%. The demand for systems developers is also indicated in the 2019 JCSE-IITPSA ICT Skills Survey (Schofield and Dwolatzky, 2019). For instance, a systems development practitioner may serve as a systems analyst, developer and a tester in a systems development project, or assume different roles in different systems development projects in the same organisation. The respondents with multi-skills may provide relevant data required for the study as they may have had exposure to the whole range of systems development project roles.

Table 5-2: Respondents that held more than one role category

Role category	N	Frequency as a percentage
Systems Development Project Manager	3	1.9%
Systems Developer	20	12.9%
Systems Analyst	8	5.2%
Systems Designer	1	0.6%
Other	6	3.9%
Total frequency	**38**	**24.5%**

5.2.3 Individual experience of the respondents

The experience of each systems development practitioner with systems development projects may influence the extent to which an SDM is adopted, understood and used (Marks, *et al*., 2017; Barlow *et al*., 2011). Experience with systems development projects of the respondents is summarised in Table 5-3. The respondents with less than six years of experience in systems development projects constituted 18.7% and those with experience ranging between six and ten years constituted 21.9%. A total of 27.7% of the respondents had between eleven and fifteen years of experience in systems development projects. More than a fifth (20.6%) of the respondents had experience ranging between sixteen and 20 years in systems development projects. The rest (11.0%) of the respondents had over 20 years of experience in systems development projects. The meaningful experience on systems development practice possessed by the respondents strengthened the relevance of the data collected in terms of its validity towards generalisation.

Table 5-3: Individual experience of respondents with systems development

Experience in years	N	Frequency as a percentage	Cumulative Percentage
0-5 years	29	18.7.6%	18.7%
6-10 years	34	21.9%	40.6%
11-15 years	43	27.7%	64.4%
16-20 years	32	20.6%	89.0%
21 or more years	17	11.0%	100%

5.2.4 Time dedicated to the SDM selection

The distribution of systems development project time dedicated to the selection of SDMs is shown in Table 5-4. Some respondents (12.9%) indicated that they did not allocate any time of the systems development project (that is zero percent of the systems development project time) deliberating on the selection of an SDM. Others (43.9%) indicated dedicating from 1% to 10% of the total systems development project time to the selection of an SDM. Those who indicated between 10% and 31% of the systems development project time being dedicated to the SDM selection constituted 41.3% of the respondents.

Table 5-4: Percentage of systems development project time dedicated to SDM selection

Percentage of systems development project time dedicated to SDM selection	Frequency	Percentage of respondents
0%	20	12.9%
1-10%	68	43.9%
11-20%	45	29.0%
21-30%	19	12.3%
31-40%	2	1.3%
41-50%	1	0.6%
Average percentage of systems development project time dedicated to SDM selection:12.0%		

5.3 ORGANISATION PROFILE

Statistics on some of the contextual stressors imposed by an organisation on a systems development project are provided in the next subsection.

5.3.1 Organisation size

Organisations were categorised into four enterprise classes according to Wendler (2016) namely, the micro (1 - 10 employees), small (11 - 50 employees), lower medium (51 - 100 employees), upper medium (101 - 200 employees) and large (201 or more employees) as presented in Table 5-5. The size of an organisation determines the type and amount of resources available and the communication protocols among other important characteristics (Baker, 2012). The respondents came from organisations of varying sizes. Most respondents came from large enterprises which constituted 42.6% of organizations, followed by the medium enterprises which constituted 33.6% of the organisations. The small and micro enterprises constituted 15.5% and 8.4% of the organizations respectively. This data is representative of large, medium, small and micro enterprises and is assumed to generate valid generalizations across systems development projects in these organisational categories.

Table 5-5: Organisation size

Organisational size	Organisation category	N	Frequency as a percentage	Cumulative Percentage
1-10 employees	Micro Enterprises	13	8.4%	8.4%
11-50 employees	Small Enterprises	24	15.5%	23.9%
51-100 employees	Lower Medium Enterprises	30	19.4%	43.2%
101-250 employees	Upper Medium Enterprises	22	14.2%	57.4%
251 or more employees	Large Enterprises	66	42.6%	100%

5.3.2 Systems development project team size

The systems development team is responsible for executing the core tasks required to complete a given systems development project. Boehm and Turner (2003) argue that the systems development project team size is one of the five important factors in the selection of an SDM. Barlow *et al.* (2011) indicate that the systems development project team size is an important factor in determining the most appropriate SDM for a system development project and suggest that small teams should use the agile SDM class and the large teams should use the plan-driven SDM class. Young *et al*. (2016) state that the effectiveness of communication to coordinate systems development project activities deteriorate with the increase on the systems development project team size. Barlow *et al*. (2011) indicate that systems development projects team sizes that are less than 5 members are small and the systems development project team sizes of more than 5 members are large and complex. Most systems development project teams (78.7%) had more than 5 members at the time of study as shown in Table 5-6. The largest systems development project teams had more than 50 members representing 8.4% of the respondents. The small systems development project teams (between one and 5 team members) had 21.3%

respondents. The collected data was representative of both small and large systems development project teams.

Table 5-6: Systems development project team size

Systems development project team size	N	Frequency as a percentage	Cumulative percentage
1 - 5 employees	33	21.3%	21.3%
6 - 15 employees	67	43.2%	64.5%
16 - 30 employees	32	20.6%	85.2%
31 - 50 employees	10	6.5%	91.6%
51 or more employees	13	8.4%	100%

5.3.3 Number of systems development projects that can run concurrently

The data set on the number of SDMs that can be implemented concurrently by each organisation showed a skewness value of 1.438 as shown in Table 5-7. This indicates data values that are clustered on the left. The Skewness value is greater than 1. A further step was taken to assess the normality of the distribution of this data set. In Table 5-8, the Kolmogorov-Smirnov test of normality that is shown with Sig. value of 0.000, indicates the violation of normality assumption (Tabachnick and Fidell, 2013). In this case, the nonparametric statistic or the median was used to summarise data as opposed to the mean which is a parametric statistic (Tabachnick and Fidell, 2013). The interquartile range statistic of value 6 was used to summarise data set variability for the group.

Table 5-7: Data distribution for the number of systems development projects that can run concurrently

N		Mean		Median	Std. Deviation	Skewness		Kurtosis		Interquartile Range
Valid	Missing	Statistic	Std. Error	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error	Statistic
155	0	5.589	0.382	4	4.76	1.438	0.195	1.59	0.387	6

Table 5-8: Test of normality of the distribution of the number of systems development projects that can run concurrently

Kolmogorov-Smirnov[a]			Shapiro-Wilk		
Statistic	df	Sig.	Statistic	df	Sig.
0.220	155	.000	0.827	155	.000

a. Lilliefors Significance Correction

5.3.4 Systems development project team members involved in the selection of an SDM

The collected data revealed that the selection decision of the SDM was determined by systems development project team members in different role categories. The selection of an SDM or SDMs for a systems development project involved many individuals within an organisation. The number of stakeholders responsible for making decisions on the selecting of an SDM for each systems development project is presented in Table 5-9. Without specifically identifying the stakeholders, 43.9% of the respondents indicated that the responsibility of selecting an SDM or SDMs for systems development projects, rested in less than six members of an organisation as shown in Table 5-9. The study data set also revealed that 83.9% of the respondents associated SDM selection decisions with less than 16 members of the organisation.

Table 5-9: Systems development team members involved in the selection of an SDM

Number of systems development project team size involved in SDM selection	N	Frequency as a percentage	Cumulative percentage
1 - 5 members	68	43.9%	43.9%
6 - 15 members	62	40%	83.9%
16 - 30 members	9	5.8%	89.7%
31 - 50 members	7	4.5%	94.2%
51 or more members	9	5.8%	100%

5.3.5 Perception on accountability per role in a systems development project

McLeod and MacDonell (2011) consider systems development success as a multifaceted construct consisting of technical, economic, social and political dimensions. Systems development success is objective, subjective and temporary and the triple constraints cannot adequately measure it (Zwikael *et al*., 2014). To measure a systems development project's success both process and product success should be taken into consideration (Zwikael *et al*., 2014). The respondents were asked to indicate, based on their perception of systems development project failure, the level of accountability per role. The question sought to expose responsibility and accountability associated with the systems development roles (Misra *et al*., 2009). The contribution on the selection of an SDM is associated with the level of accountability and responsibility for the actions and decisions embedded within a role (Misra *et al*., 2009). More than a third (35.5%) of the respondents indicated that the accountability on the systems development project manager is high, that is, 80% or more as shown in Figure 5-2. An equal number (14.8%) of the respondents indicated for both the intervals 40-59% and 60-79% blame on the systems development project manager when a systems development project failed.

Perception about accountability associated with the systems developers is generally low. The results are shown in Figure 5-2 where 32.3% of the respondents indicated that the level of accountability of a systems developer was below 20%. The accountability percentage interval 21 - 39% was indicated by 20.6% of the respondents. The accountability percentage interval 40 - 59% was shown by 18.1% of the respondents; the accountability percentage interval 60 - 79% was indicated by 14.2% of the respondents and the accountability interval from 80% and above was indicated by 14.8% of the respondents as presented in Figure 5-2.

The majority (38.1%) of the respondents indicated that the level of accountability of the systems analyst is 80% or more. The extent of blame perceived to be associated with the systems analyst when the systems development project fails is shown in Figure 5-2. A high perception (80% or more) of accountability associated with a systems designer was indicated by 7.1% of the respondents. An accountability interval of 60 - 79% was indicated by a few respondents (12.9%) when the systems development project failed. The accountability percentage intervals associated with the systems designer is shown in Figure 5-2.

Those who selected the "Other" option constituted 3.9% of the respondents. Most (83.3%) respondents who selected the "Other" option indicated the accountability when the systems development project fail to be 80% or more. These included systems quality assurance (QA) professionals, systems testers and the implemented SDM. The implemented SDM was the only non-human component in a systems development project that was associated with the blame on systems development project failure.

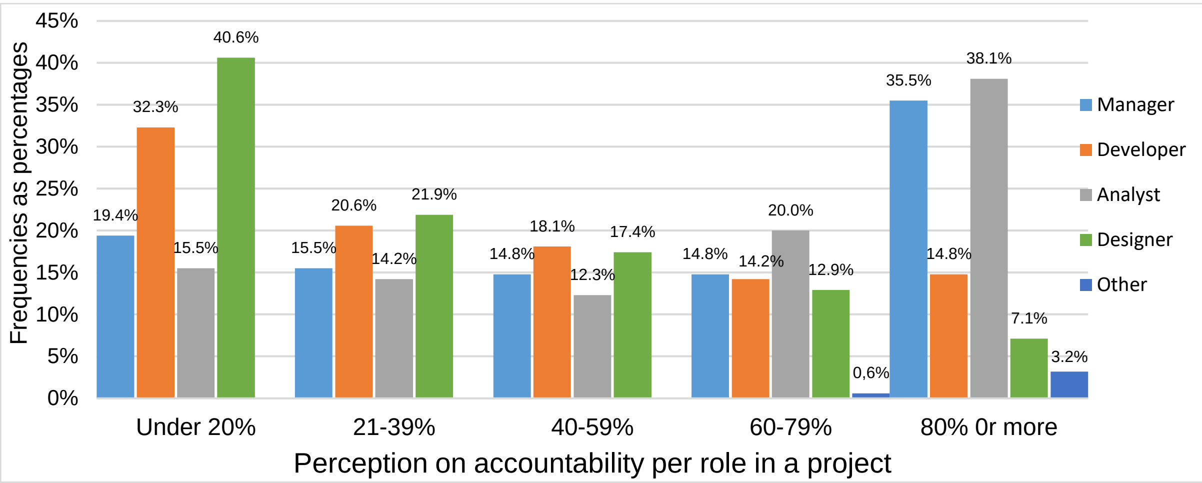


Figure 5-2: Perception on accountability per role in a systems development project

5.4 PRACTICES IN THE ADOPTION OF SDMS

The adoption of an SDM is contingent to the contextual stressors. The adoption stages involve the pre-adoption, adoption and post adoption phases. Adoption may be guided by organisational polices, SDM adoption frameworks or best practice. In terms of adoption strategies, organisations may implement a pre-selected SDM, or select an SDM based on fixed-points such as iterations or phases or at any point as dictated by the contextual stressors (Isaias and Issa, 2015; Rahmanian, 2014; Zhu, 2002).

Massey and Satao (2012) found that contextual stressors are the main basis for determining the choice of a specific SDM for a particular systems development project. Zhu (2002) points out that organisations may deal with the adoption of SDMs in different ways based on the contextual stressors. The respondents were presented with two questions. The first question requested each respondent to state the SDM selection strategy they use. A list of selection strategies was provided so that the respondent may select. The respondent could select more than one option if it met their organisational practice, as it is possible to have each of the strategies implemented based on the systems development project's contextual stressors. The second question asked the respondents to indicate their SDM adoption approach which could be through guidelines, adoption frameworks, best practice and experience (Huisman, 2000).

5.4.1 SDM adoption approaches

Results of respondents' responses are summarised in Figure 5-3. There are many SDM adoption approaches (Young *et al*., 2016; Vavpotič and Vasilecas, 2012). These may include SDM adoption based on guidelines, SDM adoption based on policies, SDM adoption based on frameworks, SDM adoption based on best practices and experience (Harb *et al.,* 2015). Results for SDM adoption approaches are displayed in Figure 5-3. The dataset used in the study indicated that the majority (78.7%) of organisations indicated that their SDM selection was informed by best practices and experience.

The SDM adoption based on adoption frameworks and the "Other" option were indicated by 11% and 0.6% of respondents respectively. The adoption frameworks assist systems development project teams to adopt SDMs based on the assumptions made by the adoption frameworks on systems development project contextual stressors. The framework assumptions may not hold in some situations. The study data set also revealed that 24.5% of the organisations that participated employed SDM adoption based on guidelines or polices for adopting SDMs.

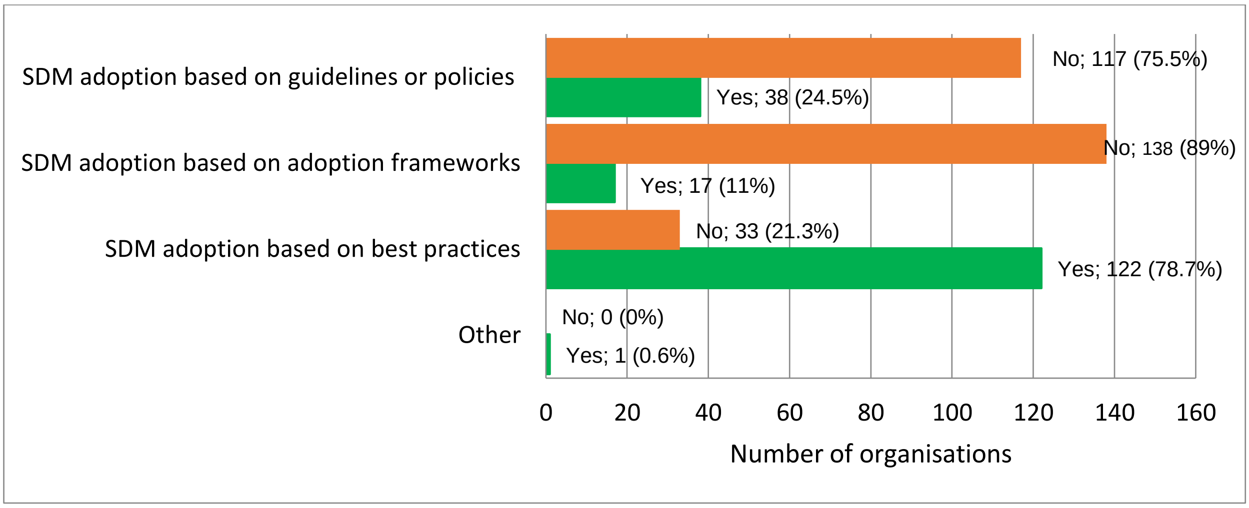


Figure 5-3: SDM adopting approaches

5.4.2 SDM adopting strategies

Results for SDM adopting strategies revealed by the study dataset are shown in Figure 5-4. The strategy to create alternative SDMs was indicated by 21.3% of the respondents. The SDM contingency along system development dynamics was indicated by 48.4% of the respondents. The SDM contingency with fixed pattern was indicated by 51%, SDM contingency at outset was indicated by 24.5% of the respondents, and the "Other" option was indicated by 1.9% of the respondents. The most dominant SDM adopting strategies at the time of study were SDM contingency with fixed pattern (51%) and the SDM contingency along development dynamics was 48.4% of the organisations that participated.

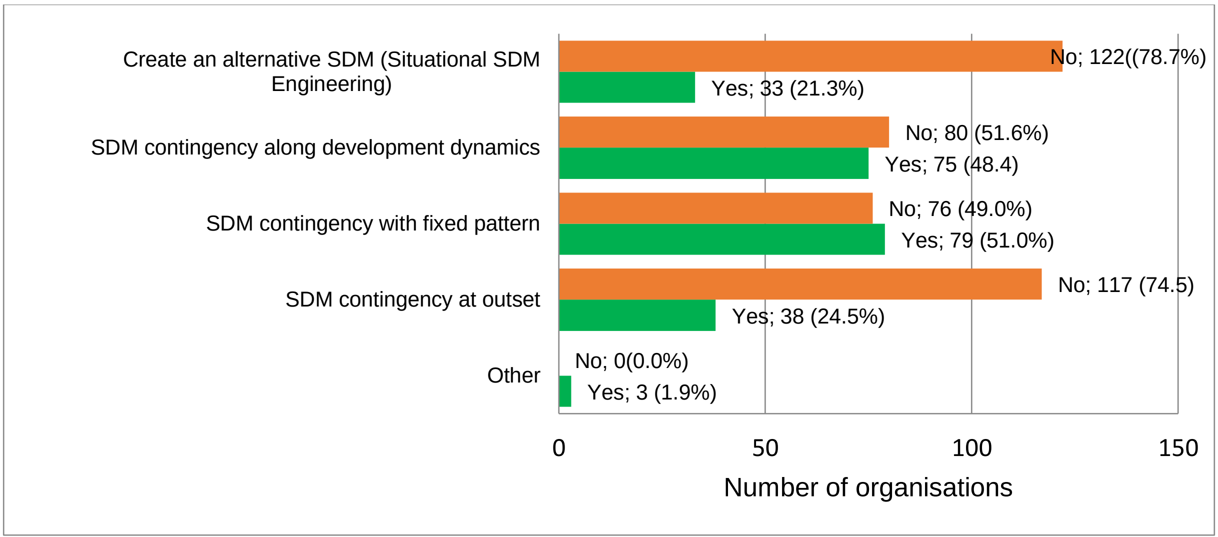


Figure 5-4: SDM adopting strategies

5.5 ADOPTING SDMS FOR SYSTEMS DEVELOPMENT PROJECTS

Boehm (2006) states that there are five critical factors that are core to adopting an SDM for a systems development project, namely the personnel, requirements dynamics, organisational culture, team size, and system criticality. Mahanti *et al.* (2012) indicate that there are sixteen important factors that are core to adopting an SDM, namely the project type, project size, project duration, project complexity, level and type of expected risk, clarity of user requirements, application domain, customer involvement, developer experience, team size, interfaces, tools and technology, product versions and the reliability required. Clarke and O'Connor (2015) identify 44 relevant factors when adopting an SDM for a systems development project.

Marks *et al.* (2017) and Young *et al*. (2016) also provide a list of factors that are key to adopting an SDM or combination of SDMs for a systems development project. The different researchers' identification of contextual stressors varies at the level of granularity. The contextual stressors can be categorised into three main themes; the organisational, systems development project and individual. A set of seventeen factors were established and were measured on a 6-point scale from 1 representing totally disagree to 6 totally agree. The identified contextual stressors were not biased towards a particular class of SDMs. The results about the influence of contextual stressors in adopting SDMs for systems development project are presented in the next subsection. A set of three questions were asked that solicited the respondents to provide information about the importance of adopting SDMs for their systems development projects, contextual factors they considered important in adopting SDMs for their systems development projects and the justification they made for the adoption SDMs for their systems development projects within their organisations.

5.5.1 Importance of adopting an SDM or a combination of SDMs

Respondents were asked to indicate how important adopting SDMs was for their systems development projects. Responses ranged from 1, representing extremely unimportant, to 6, representing extremely important. There was no room for neutrality and responses of 3 represented unimportant and 4 represented important. The mean response score for this item was high (*M=5.34,* on a 6-point numerical scale) with little dispersion (*SD=0.84*), indicating that the respondents considered the adopting of an SDM or a combination of methodologies for the success of systems development projects as very important at the time of the study. In fact, 51.6% of the respondents said adopting an SDM or a combination of SDMs for the success of a systems development projects was extremely important, 33.1% said it was quite important, and 13.5% regarded it as important as shown in Table 5-10.

Table 5-10: Importance of adopting an SDM or a combination of SDMs

Importance of adopting an SDM or a combination of SDMs for a systems development project	N	Frequency as a percentage
Extremely unimportant	1	0.6%
Quite unimportant	1	0.6%
Important	21	13.5%
Quite important	52	33.5%
Extremely important	80	51.6%

5.5.2 SDM characteristics that inform adopting an SDM or a combination of SDMs

A total of 11 factors was established and presented to the respondents in the form of statements to rate their experience on each factor within their organisations on a 6-point scale. The rating indicate how they agreed or disagreed with those statements (that is from 1 representing totally disagree to 6 representing totally agree). The responses provided by respondents are summarised in Table 5-11. The mean responses for the questionnaire items were in the range of

3.96 - 5.05 (on a 6-point scale), indicating agreement by respondents on the SDM characteristics they use to justify adopting an SDM or a combination of SDMs for a systems development project. More than three quarters (76.7%) of the respondents indicated 5 points and above, which implied absolute agreement that the way each SDM breaks down systems development project activities was regarded as an attractive attribute that increased the probability of adopting it for a systems development project. How an SDM handles bug detection, how SDM deals with systems development project costs, how the SDM complies with systems development standards, and how it matches the systems development problem domain were indicated by a mean above 4.50 (on a 6-point scale).

Table 5-11: Justification of adopting one SDM over another for a systems development project

N=155	Mean	Std. Dev	Frequencies as percentages					
			1	2	3	4	5	6
Different SDMs break down work differently.	5.05	0.91	0.0	0.0	7.7	15.5	40.6	36.1
Different SDMs detect defects differently.	4.79	1.08	0.6	3.2	6.5	25.2	35.5	29.0
Different SDMs have different cost implications in a systems development project.	4.76	1.17	0.0	8.4	6.5	13.5	43.9	27.7
Different SDMs comply with standards differently.	4.57	1.16	2.6	3.2	8.4	27.7	36.8	21.3
Different SDMs match with systems development project problem space with varying degrees.	4.52	1.11	0.0	5.8	11.6	27.7	34.8	20.0
Different SDMs have different failure histories.	4.41	1.05	0.6	7.1	8.4	28.4	45.8	9.7
Different SDMs have different success histories.	4.25	1.15	3.9	4.5	12.3	29.0	42.6	7.7
Different SDMs facilitate delivery of systems development projects within same budgetary constraints differently.	4.11	1.15	3.2	7.1	12.3	38.1	31.6	7.7
Different SDMs facilitate delivery of systems development projects within same schedule constraints differently.	4.10	1.28	7.1	3.9	12.3	35.5	31.0	10.3
Different SDMs identify critical systems development project stakeholders differently.	4.08	1.13	1.3	9.0	15.5	37.4	27.7	9.0
Different SDMs disseminate common systems development project vocabulary to identified stakeholders differently.	3.96	1.23	3.9	9.0	18.1	34.8	24.5	9.7

A Cronbach's analysis was conducted on the eleven items and the Cronbach's alpha level was 0.781, an acceptable level of reliability (Pallant, 2016; Field, 2013). Further analysis indicated that the item "Different SDMs have different failure histories" had a corrected inter-total correlation of less than .3, and was therefore removed from the scale. The remaining 10 items had a Cronbach's alpha level of 0.784 which is classified as acceptable.

Data assessment factorability was performed prior to subjecting the 10 items to a principal component analysis (PCA). The Kaiser-Meyer-Olkin measure of sampling adequacy (KMO) value was 0.783 which is classified as good, indicating that the sample size was large enough to reliably extract factors (Field, 2013). To test whether the variables did not correlate too highly or too lowly with other variables (Field, 2013) the Bartlett's Test of Sphericity was conducted. A Bartlett's Test of Sphericity, $\chi^2(45) = 349.015$, $p < 0.0001$ was found indicating that the correlations between the items were sufficiently high (Pallant, 2016; Field, 2013). The Kaiser criterion was considered as a factor extraction strategy. The Kaiser criterion revealed three components with eigenvalues greater than 1 (Pallant, 2016). The eigenvalues above 1 for the three components were: 3.473, 1.274, and 1.021 respectively. The three components explained 57.7% of the variance. The screeplot was analysed and it indicated a clear break between the first and the second component. This suggested extraction of a single component. Furthermore, a Parallel Analysis based on Monte Carlo PCA was performed and the first three criterion values for a randomly generated data matrix of the same size (10 variables x 155 respondents) (Pallant, 2016) generated through 50 replications as shown in Table 5-12. The Parallel Analysis decision strategy involved comparing the actual eigenvalue generated from PCA, with the corresponding criterion value randomly generated by Monte Carlo PCA. When the actual eigenvalue generated from PCA was found to be greater than the corresponding criterion value randomly generated by Monte Carlo PCA for Parallel Analysis, the component was retained; otherwise it was removed as illustrated in Table 5-12. For example, the first component has the actual eigenvalue from PCA of 3.473 and it was compared with 1.4318 which was the criterion value from Parallel Analysis. In the last column of Table 5-12, the decision was to retain component one, because the actual eigenvalue from PCA was greater than the criterion value from Parallel Analysis. The other two components failed to meet the criteria of retention. Therefore, only one component was generated.

Table 5-12: Component retention strategy using parallel analysis

Component number	Actual eigenvalue from PCA	Criterion value from Parallel Analysis (randomly generated from Monte Carlo PCA)	Decision
1	3.473	1.432	Retained
2	1.274	1.285	removed
3	1.021	1.182	removed

The retained component is shown in Table 5-13.

Table 5-13: Component structure for justifying adopting an SDM

Factors indicative of the justification of adopting an SDM (F0)	SDM adoption justification (F1)
All items	Different SDMs break down work differently.
	Different SDMs detect defects differently.
	Different SDMs have different cost implications in a systems development project.
	Different SDMs comply with standards differently.
	Different SDMs match with systems development project problem space with varying degrees.
	Different SDMs have different failure histories.
	Different SDMs have different success histories.
	Different SDMs facilitate delivery of systems development projects within same budgetary constraints differently.
	Different SDMs facilitate delivery of systems development projects within same schedule constraints differently.
	Different SDMs identify critical systems development project stakeholders differently.
	Different SDMs disseminate common systems development project vocabulary to all identified stakeholders differently.
Cronbach α	**0.784**

5.5.3 SDM fit assessment criteria for a systems development project

A total of 17 factors was established and presented to the respondents as items on a 6-point scale each. The respondents were requested to evaluate each item according to the existing practice in their organisations. The question item response options ranged from 1 representing totally disagree to 6 representing totally agree. The responses provided by respondents are summarised in Table 5-14. A mean response of 4.5 would indicate a strong agreement and a mean of 5 and above would indicate absolute agreement and confirmation of the practical existence of the study variable (Pallant, 2016). The mean responses for the eight items ranged from 4.50 - 5.30 (on a 6-point scale) as shown in Table 5-14. The standard deviations were small, indicating that the data points were clustered close to the mean. The small standard deviations indicated that the calculated means were most likely to represent the observed data (Byrne, 2001) and more precise portrayal of the variables of interest (Hair *et al*., 2018). Therefore, response means for these aforementioned questionnaire items were more likely to be representative of the respondents' experience on the contextual stressors that influence the selection of an SDM or a combination of SDMs for a systems development project.

A set of three items out of the seventeen items had calculated means greater than 5 points. The first highest calculated mean (5.35) corresponded to the item on systems development project size, with 92.9% of respondents indicating a 5 or a 6 (on a 6-point scale) reflecting a strong agreement on the influence of the systems development project size in the selection of an SDM or a combination of SDMs. The second mean (5.23) corresponded to the item on the degree of meeting the user requirements as a determinant for the adopting an SDM for a systems development project, with 85.8% of the respondents confirming the absolute influence by rating it at 5 or more points (on a 6-point scale). The third mean (5.11) corresponded to the item on system development project criticality, which was rated from 5 points and above 6 (on a 6-point scale) by 78.7% of the respondents.

Table 5-14: SDM fit assessment criteria for a systems development project

N=155	Mean	Std. Dev	Frequencies as percentages					
			1	2	3	4	5	6
Systems development project size influenced the adoption of a specific SDM.	5.35	0.65	0.0	00	1.3	5.8	49.0	43.9
The degree of meeting the user requirements influenced the adoption of a specific SDM.	5.23	0.83	06	2.6	0.0	11.0	44.5	41.3
Systems development project criticality influenced the adoption of a specific SDM.	5.11	0.85	0.6	0.0	1.9	18.7	42.6	36.1
Systems development team size influenced the adoption of a specific SDM.	4.97	0.99	1.3	0.6	5.2	17.4	43.2	32.3
Systems developer experience influenced the adoption of a specific SDM.	4.97	0.99	0.6	1.3	4.5	21.9	36.8	34.8
Systems development team structure influenced the adoption of a specific SDM.	4.59	0.97	0.6	1.3	8.4	36.1	35.5	18.1
Systems development project problem domain influenced the adoption of a specific SDM.	4.48	1.23	3.2	3.9	12.9	21.9	37.4	20.6
The systems development project artefact's market window influenced the adoption of a specific SDM.	4.48	1.21	2.6	5.2	9.7	25.8	37.4	19.4
Emphasized systems development project artefact quality factors influenced the adoption of a specific SDM.	4.43	1.05	0.6	4.5	11.0	32.9	36.8	14.2
The systems development project costs influenced the adoption of a specific SDM.	4.42	1.20	0.6	5.8	14.8	31.0	25.2	22.6
Systems developer expertise influenced the adoption of a specific SDM.	4.42	1.25	1.9	6.5	14.8	20.6	36.8	19.4
SDM's capability to early discover problems in a specific systems development project influenced its adoption.	4.20	1.16	1.3	6.5	18.1	32.9	27.7	13.5
Systems development project artefact post release defect history influenced the adoption of a specific system development methodology.	3.84	1.24	1.9	16.8	15.5	35.5	31.9	8.4
System development methodology familiarity influenced its adoption for a specific systems development project.	3.66	1.58	7.7	16.8	29.0	14.8	11.0	20.6
Stakeholder politics influenced the adoption of a specific SDM.	3.66	1.42	7.1	17.4	18.7	25.8	21.3	9.7
Compliance to systems development standards influenced the adoption of a specific SDM.	3.52	1.25	6.5	12.9	31.0	26.5	18.7	4.5
Stakeholder culture influenced the adoption of an SDM for a specific systems development project.	3.50	1.01	5.2	18.1	26.5	26.5	20.0	3.9

In addition, a Cronbach's analysis was conducted on the 17 items. The ten items shown in Table 5-15 indicated corrected inter-total correlations less than 0.3 and were then removed from the scale due to low reliability value (Pallant, 2016).

Table 5-15: Corrected inter-total correlations of items with low reliability values

Item	Corrected Item-Total Correlation
The degree of meeting the user requirements influenced the adoption of a specific SDM.	.205
Systems development team size influenced the adoption of a specific SDM.	.211
Systems developer expertise influenced the adoption of a specific SDM.	.280
System development methodology familiarity influenced its adoption for a specific systems development project.	.287
SDM's capability to early discover problems in a specific systems development project influenced its adoption.	.217
Emphasized systems development project artefact quality factors influenced the adoption of a specific SDM.	.137
Systems development project artefact post release defect history influenced the adoption of a specific system development methodology.	.266
Stakeholder culture influenced the adoption of an SDM for a specific systems development project.	.139
The systems development project artefact's market window influenced the adoption of a specific SDM.	.255
Systems development team structure influenced the adoption of a specific SDM.	.167

The Cronbach's alpha coefficient value for the remaining seven items was 0.61 and this indicated that the seven items had adequate inter-item reliability.

The seven items were subjected to a principal component analysis (PCA). Precursor to performing a PCA, the data was assessed for factorability. The Kaiser-Meyer-Olkin measure of sampling adequacy (KMO) value was 0.693 which is classified as adequate, indicating that the sample size was large enough to reliably extract factors (Field, 2013). To test whether the variables did not correlate too highly or too lowly with other variables (Field, 2013) the Bartlett's Test of Sphericity was conducted. The Bartlett's Test of Sphericity of $\chi^2(21) = 105.818$, $p < 0.0001$ indicated that the correlations between the items were sufficiently high (Pallant, 2016; Field, 2013). The decision strategy to determine the number of components to be extracted was first based on the Kaiser criterion, which considers components with eigenvalues greater than 1 (Pallant, 2016). The PCA showed the presence of three components with eigenvalues above 1 explaining: 31.1%, 15.6%, and 14.4% of the variance respectively. However, the screeplot showed a clear break between the first and the second components, suggesting retention of one component for further analysis.

The resulting single component was the SDM fit and is presented in Table 5-16. The generated Cronbach alpha for the component is also shown. The component had Cronbach alpha level of 0.609 which is above the minimum of 0.5 (Field, 2013).

Table 5-16: Component structure for SDM fit assessment criteria

Consideration during the adoption of an SDM (F0)	SDM fit assessment criteria (F1)
All items	Systems development project criticality influenced the adoption of a specific SDM.
	Systems development project problem domain influenced the adoption of a specific SDM.
	Systems development project size influenced the adoption of a specific SDM.
	Compliance to systems development standards influenced the adoption of a specific SDM.
	Stakeholder politics influenced the adoption of a specific SDM.
	Systems developer experience influenced the adoption of a specific SDM.
	The systems development project costs influenced the adoption of a specific SDM.
Cronbach α	**0.609**

5.6 ORGANISATIONAL RESPONSIVENESS TO ADOPTING SDMS

Part 5 of the questionnaire consisted of five questions. The first question solicited information on the responsiveness of the organisation with respect to adopting an SDM. The second question focused on the standard practice in adopting SDMs by an organisation. The third and the fourth questions sought to collect data on the horizontal and vertical use of SDM knowledge in an organisation respectively. The fifth question concentrated on the data related to the level of adaptation needed to adopt SDMs and the extent of creating alternative SDMs for systems development projects.

5.6.1 Organisational SDM adopting culture categories

During the study, 24.5% of the organisations were market leader SDM adopting culture as shown in Table 5-17. This was the second highest percentage after the market follower SDM adopting culture which constituted 29%. The late majority SDM adopting culture organisations constituted 20.0% and the laggards SDM adopting culture had 15.5% of the organisations that participated. A low proportion (5.8%) did not indicate their choices and were treated as unidentified as shown in Table 5-17.

Table 5-17: The organisational SDM adopting culture categories

Organisational SDM adopting culture	N	Frequency as a percentage	Cumulative percentage
Market leader	38	24.5%	24.5%
Market follower	45	29.0%	53.5%
Late majority	31	20.0%	73.5%
Laggard	32	20.6%	94.1%
Unidentified	9	5.8%	100%

5.6.2 Standard practice on adopting SDMs

The second question item in Part 5 of the questionnaire wanted to gather data related to the way different organisations adopt and adapt SDMs or create alternative SDMs for their systems development projects. The results for the question are shown in Figure 5-5. The respondents could select more than one option regarding practices they used in adopting SDMs for systems development projects. The dataset used for the study indicated that 77.4% of the respondents adopt an SDM and adapt it for each systems development project situation. A high proportion (70.3%) of the respondents indicated their standard practice in adopting SDMs was based on selecting and adopting SDMs from in-house developed SDMs for each systems development project situation. The dataset also revealed that 42.6% of the respondents do not only adopt a single SDM, but a set of SDMs and combine them for each systems development project situation, while, 39.4% adopt an SDM from a standard set of SDMs for each systems development project situation, and 17.4% create an alternative SDM for each systems development project situation respectively. There were no respondents that indicated non-use of SDMs in their systems development projects.

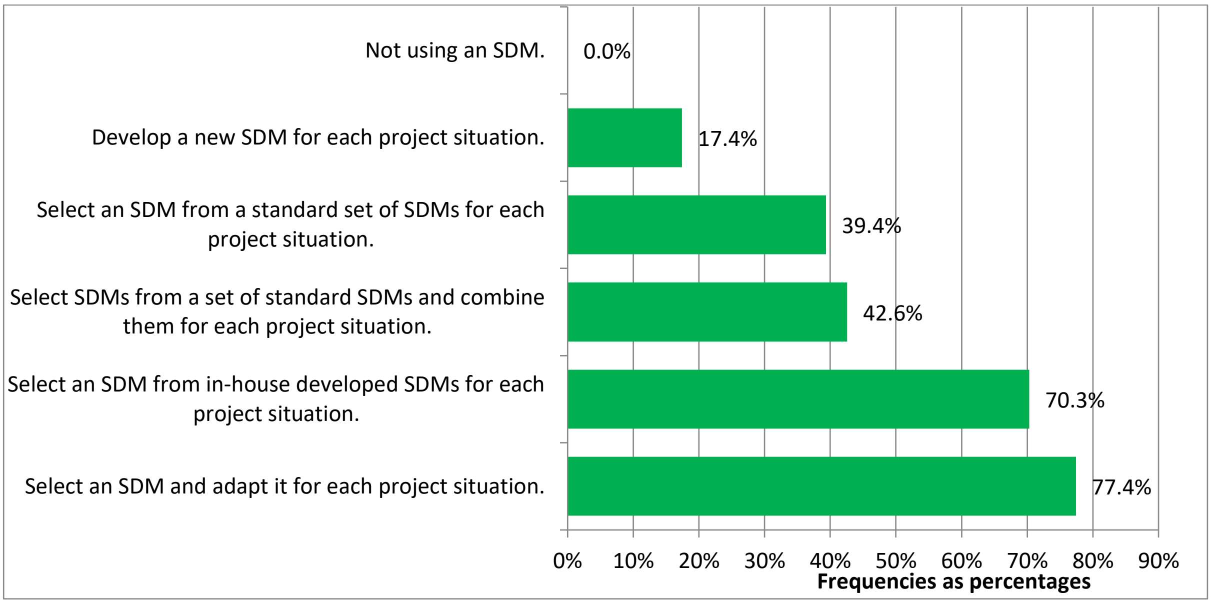


Figure 5-5: Standard practice in adopting SDMs

5.6.3 Horizontal use of SDMs

The results in Table 5-18 show that not even one (0.0%) of the respondents indicated non-use of SDM knowledge in systems development projects in their organisation. The question gathered information about the breadth of SDM knowledge use in systems development projects across systems development projects in an organisation. Of the total respondents, those that indicated the use of SDM knowledge across projects to be between 1% and 20% were 1.3%. The respondents who indicated the use of SDM knowledge across systems development projects to be between 21% and 40% were 7.1%. Most (67.1%) of the respondents indicated the application of SDM knowledge in systems development projects across their organisations to be above 60%. This indicated a strong penetration of SDM knowledge usage breadth in organisations and this increases the validity of the data collected from the respondents.

Table 5-18: Percentage interval of horizontal use of SDMS in organisations

Percentage interval of horizontal use of SDMs	N	Frequency as a percentage	Cumulative percentage
0.0%	0	0.0%	0.0%
1 - 20%	2	1.3%	1.3%
21 - 40%	11	7.1%	8.4%
41 - 60%	38	24.5%	32.9%
61 - 80%	53	34.2%	67.1%
Over 80%	51	32.9%	100%

5.6.4 Vertical use of SDMs

This question gathered information about the proportion of systems development practitioners that applied SDM knowledge in systems development projects in their organisations. The vertical SDM use relates to the intensity of SDM knowledge used by the members of the systems development project team. The results for vertical use of SDMs in organisations are shown in Table 5-19. The results show that none (0.0%) of the respondents indicated any existence of a proportion of systems development projects that did not rely on the SDM knowledge application in their organisations. Of the total respondents, those that indicated the proportion of systems development projects that applied SDM knowledge to be between 1% and 20% were 2.6%. The respondents who indicated the proportion of systems development projects that applied SDM knowledge to be between 21% and 40% were 5.2% and for the range between 41% and 60% they were 27.1% of the respondents. Most (65.2%) of the respondents indicated the application of SDM knowledge in systems development projects in their organisations to be above 60% at the time of study.

Table 5-19: Vertical use of SDMS in organisations

Vertical use of SDMs	N	Frequency as a percentage	Cumulative percentage
0.0%	0	0.0%	0.0%
1 - 20%	4	2.6%	2.6%
21 - 40%	8	5.2%	7.7%
41 - 60%	42	27.1%	34.8%
61 - 80%	62	40.0%	74.8%
Over 80%	39	25.2%	100%

5.6.5 Variability in the implementation of SDMs

The implementation of SDMs may vary from one organisation to the other, from one system development project to another and across similar systems development project contexts and within the same systems development project context over time (Berente *et al*., 2015; Aitken and Ilango, 2013). A question intended to measure variability of SDM implementation between organisations, between system development projects within the same systems development project, was posed to respondents. The responses are summarised in Table 5-20. A total of 5 possible variations on the implementation of SDMs was established from studies and presented to the respondents to rate their experience on each variation instance. The question item responses ranged from 1 representing totally disagree to 6 representing totally agree. The obtained data is summarised using, means, standard deviations and frequencies for each response as shown in Table 5-20. Mean responses for the five items are in the range of 3.0 - 4.2 with standard deviations in the range of 1.31 - 1.71. A lowest mean is generated for the case of no deviation from the SDM prescription.

Table 5-20: Variability in the implementation of SDMs

N=155	Mean	Std. Dev	Frequencies as percentages					
			1	2	3	4	5	6
Deviation from the SDM prescription was caused by the need to reconfigure some components.	4.2	1.34	5.2	6.5	18.7	23.2	30.3	16.1
Deviation from the SDM prescription was caused by the need to remove some irrelevant components.	4.0	1.31	3.9	8.4	23.9	25.8	24.5	13.5
Deviation from SDM prescription was caused by the need to address some missing components.	3.8	1.39	7.7	9.0	22.6	29.0	19.4	13.3
We created an alternative SDMs based on components from existing SDMs.	3.8	1.71	12.3	14.8	18.1	14.2	18.1	22.6
No deviation at all from SDM prescription.	3.0	1.70	19.4	31.0	16.8	7.1	12.3	13.5

A Cronbach's analysis was conducted on the 5 items and Cronbach's alpha value for the 5 items was 0.71. This indicated that the 5 items had adequate inter-item reliability.

Before performing PCA, the suitability of data for factor analysis was assessed. The Kaiser-Meyer-Olkin measure of sampling adequacy (KMO) value was 0.741, which is classified as good (Pallant, 2016; Field, 2013). To test whether the variables did not correlate too highly or too lowly with other variables (Field, 2013) the Bartlett's Test of Sphericity was conducted. The Bartlett's Test of Sphericity of $\chi^2(10) = 171.016$, $p < 0.0001$ indicated that the correlations between the items were sufficiently high (Pallant, 2016; Field, 2013). The Kaiser criterion revealed that there was only one component with eigenvalues greater than 1 (Pallant, 2016) that explained 48,8% of the variance. The screeplot also showed a clear break between the first and the second component indicating that there was only one component. This single component was then extracted and retained.

As there was a single component there was no need to simplify the interpretation of the component by performing a Promax with Kaiser Normalization rotation, as the component is the solution. The simple structure with only one component is shown in Table 5-21.

Table 5-21: Component structure of SDM implementation variability

SDM implementation variability (F0)	Contingent use of SDMs (F1)
All items	No deviation at all from SDM prescription.
	Deviation from SDM prescription was caused by the need to address some missing components.
	Deviation from the SDM prescription was caused by the need to remove some irrelevant components.
	Deviation from the SDM prescription was caused by the need to reconfigure some components.
	We created an alternative SDMs based on components from existing SDMs.
Cronbach α	**0.708**

5.7 SUCCESS MEASURES OF SDMs ON A SYSTEMS DEVELOPMENT PROJECT

Karlsen *et al*. (2005) suggested that systems development project success criteria should be defined from the outset by the various stakeholders and be flexible to reflect changes that might have occurred during a systems development project. Two questions were asked. The first question with eight items was presented to the respondents to gather data related to the success criteria used by organisations when evaluating SDMs. The second question measured the recorded success on their selected SDM.

5.7.1 SDM Success measures

Table 5-22 shows that the traditional success criteria of strict budgetary compliance, meeting time schedule and user satisfaction, were highly rated by respondents with calculated means of 4.35 and above on a 6-point scale. More than 70% of the respondents indicated their level of agreement to be 5 or above with regards to the application of the triple constraints (cost, schedule, scope) as SDM success measure in their organisations. This indicated that the triple constraints are still highly rated as a measure of success in systems development. Any deviation from the iron triangle components led to a rating lower than 4.30 (on a 6-point scale).

Table 5-22: Success measures of an SDM on a systems development project

N=155	Mean	Std. Dev	Frequencies as percentages					
			1	2	3	4	5	6
Success was achieved when the adopted SDM led to the completion of the systems development project within budget.	4.68	1.17	1.9	3.9	7.7	23.2	36.8	26.5
Success was achieved when the adopted SDM led to the completion of the systems development project on time.	4.61	1.08	0.6	3.2	10.3	28.4	34.8	22.6
Success was achieved when the adopted SDM led to the development of a systems development project artefact that satisfied the user.	4.39	1.23	2.6	5.8	11.0	29.7	32.3	18.7
Success was achieved when the selected SDM led to the completion of the systems development project artefact with all the features and functionality as initially specified.	4.38	1.29	3.2	5.2	15.5	23.2	32.3	20.6
Success was achieved when the adopted SDM led to the completion of a system development project artefact with high quality.	4.35	1.38	0.6	3.9	5.8	20.0	35.5	34.2
Success was achieved when the adopted SDM led to the completion of the systems development project within minor deviations from the schedule.	4.28	1.16	2.6	6.5	9.0	37.4	31.6	12.9
Success was achieved when the adopted SDM led to the completion of the systems development project with minor deviations from the budget.	3.99	1.28	3.2	9.7	20.0	33.5	19.4	14.2
Success was achieved when the adopted SDM led to the completion of the systems development project artefact with some of the features and functionality initially specified.	3.00	1.43	0.0	4.5	9.0	36.8	32.3	17.4

A Cronbach's analysis was conducted on the 8 items and the Cronbach's alpha level for the 8 items was 0.751. This is the generally acceptable Cronbach's alpha level (Hair *et al*., 2018).

Prior to performing PCA, the suitability of data for factor analysis was assessed. The Kaiser-Meyer-Olkin measure of sampling adequacy (KMO) value was 0.779, which is classified as good (Pallant, 2016; Field, 2013). The Bartlett's Test of Sphericity was conducted and there was a Bartlett's Test of Sphericity of χ^2 (21) = 218.518, p<0.0001. This indicated that the correlations between the items were adequately high (Pallant, 2016; Field, 2013). The Kaiser criterion revealed that there were only one component with eigenvalue greater than 1 (Pallant, 2016) that explained a total of 40.2% of the variance. The screeplot showed a clear break between the first and the second component supporting the extraction of one component. The component is referred to as the SDM success measure and is presented in Table 5-23.

Table 5-23: Component structure of SDM success measures

SDM success measures (F0)	SDM success measure (F1)
All items	Success was achieved when the adopted SDM led to the completion of the systems development project within budget
	Success was achieved when the adopted SDM led to the completion of the systems development project on time.
	Success was achieved when the adopted SDM led to the development of a systems development project artefact that satisfied the user.
	Success was achieved when the selected SDM led to the completion of the systems development project artefact with all the features and functionality as initially specified.
	Success was achieved when the adopted SDM led to the completion of a system development project artefact with high quality.
	Success was achieved when the adopted SDM led to the completion of the systems development project within minor deviations from the schedule.
	Success was achieved when the adopted SDM led to the completion of the systems development project with minor deviations from the budget.
	Success was achieved when the adopted SDM led to the completion of the systems development project artefact with some of the features and functionality initially specified.
Cronbach α	**0.751**

5.7.2 Rating success of adopted SDMs

The data collected was based on self-reported success, which could be affected by the perception of each respondent, as well as the definition of SDM success. However, the criteria were expected to be biased towards the triple constraints as the main criteria shared by most respondents during the study. The results for rating the success of adopted SDMs are shown in Figure 5-6. The results show that none (0.0%) of the respondents indicated the total failure of adopted SDMs. Of the total respondents, 5.8% indicated the success of the adopted SDMs in use to be between 1% and 20%. The number of respondents who indicated the success of the adopted SDMs to be between 21% and 40% was 8.4%. Respondents who indicated the success of the adopted SDMs to be between 41% and 60% were 19.4%. A total of 40% of the respondents indicated that the success of the adopted SDMs was above 80% at the time of study. A high proportion (66.5%) of the respondents indicated the success of adopted SDMs to be above 60%.

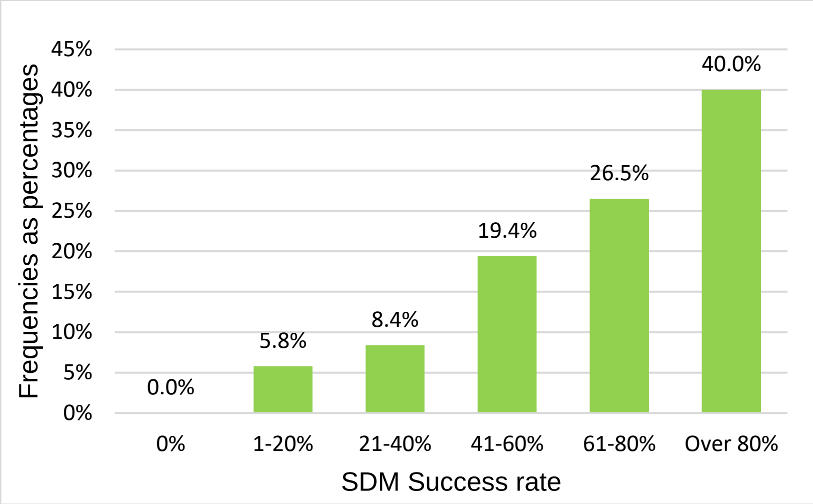


Figure 5-6: Success rate for SDM in use

5.8 SUCCESS MEASURES OF SDM THROUGH THE QUALITY OF THE SYSTEMS DEVELOPMENT PROJECT ARTEFACT

Notwithstanding the fact that systems development project success definition depends on the individual, systems development project or organisational perspective (Kerzner, 2017; Müller and Jugdev, 2012), this study measured the systems development project success from the perceived success of the adopted SDM. The success of the adopted SDM was measured from both the systems development process and the developed product perspectives. The question presented to respondents, gathered data related to the criteria used to measure the success of an adopted SDM through the systems development product developed.

5.8.1 Success measures of adopted SDM through the quality of the developed systems development project artefact quality characteristics

The perceived systems development project success measure was based on the broad success dimensions. A total of 13 items was established and presented to the respondents to rate systems development project artefact quality post the SDM use. The SDM is rated as important, based on the characteristics of the artefact they produced in the past or the purported artefact quality. The question item responses ranged from 1 representing totally disagree, to 6 representing totally agree. Respondents' responses are summarised in Table 5-24. The mean responses for the thirteen items were in the range of 4.70 - 5.14 (on a 6-point scale), representing a summary of observed data indicating agreement by respondents on the evaluation criteria of an adopted SDM based on the systems development project artefact quality. The results also indicate low standard deviations with respect to the calculated means, which indicated that the observed data points were closer to the means. The smaller the standard deviations compared to mean values, the higher the chances that the means were representing the variables under investigation (Byrne, 2001). The responses rated at 5 or more on SDM success measure, based on the systems development artefact characteristics were: provided features that met stated requirements (79.4%), provided features that met implied requirements (65.8%), response time that met user expectations (71.6%), provided accurate output for users (75.5%), user friendly (69.6%), maintained level of performance under stated conditions for a stated period of time (fault tolerance) (62.6%), stable and unlikely to fail (69.6%), recovered data and restored optimal functioning during failures (fail safe) (60.0%), gave consistent results when used (reliable) (76.1%), accommodated modification to remove defects (amenability) (67.1%), easily adapted to new specifications or changed environment (69%) and had a long life expectancy (63.9%).

Table 5-24: Success measures of selected SDM through the developed systems development project artefact quality characteristics

N=155	Mean	Std. Dev	Frequencies as percentages					
			1	2	3	4	5	6
Quality was achieved when the systems development project artefact developed using the SDM met stated requirements.	5.14	.798	0.0	0.0	2.6	18.1	42.6	36.8
Quality was achieved when the systems development project artefact developed using the SDM gave consistent results when used.	5.00	.947	0.0	1.3	7.1	15.5	42.6	33.5
Quality was achieved when the systems development project artefact developed using the SDM provided accurate output for users.	4.98	.922	0.6	0.0	6.5	17.4	44.5	31.0
Quality was achieved when the systems development project artefact developed using the SDM had response time that met user expectations.	4.95	.889	0	0.6	5.2	22.6	41.9	29.7
Quality was achieved when the systems development project artefact developed using the SDM was stable and unlikely to fail.	4.93	1.001	0.6	0.6	7.1	21.9	36.1	33.5
Quality was achieved when the systems development project artefact using the SDM accommodated changes suggested by the user.	4.90	.981	0.0	2.6	7.1	16.8	45.2	28.4
Quality was achieved when the systems development project artefact developed using the SDM was easily adapted to new specifications or changed environment.	4.89	.997	0.6	1.9	4.5	23.9	38.7	30.3
Quality was achieved when the systems development project artefact developed using the SDM was easy to use.	4.81	1.043	0	2.6	11.0	16.8	41.9	27.7
Quality was achieved when the systems development project artefact developed using the SDM accommodated modification to remove defects.	4.81	.952	0.0	2.6	5.2	25.2	42.6	24.5
Quality was achieved when the systems development project artefact developed using the SDM met implied requirements.	4.76	1.007	0.0	3.9	5.8	24.5	41.9	23.9
Quality was achieved when the systems development project artefact developed using the SDM recovered data and restored optimal functioning during failures.	4.76	1.051	0.6	1.9	7.1	30.3	31.0	29.0
Quality was achieved when the systems development project artefact developed using the SDM maintained its level of performance under stated conditions for a stated period.	4.72	.984	0	2.6	7.7	27.1	40.0	22.6
Quality was achieved when the systems development project artefact developed using the SDM had a long-life expectancy.	4.70	1.185	1.3	5.2	7.7	21.9	35.5	28.4

A Cronbach's analysis was conducted on the 13 items and the Cronbach's alpha level for the 13 items was 0.841. This indicated a good internal consistency reliability within the 13 items. This is regarded as a preferable Cronbach's alpha value (Pallant, 2016).

The suitability of data for factor analysis was evaluated prior to performing PCA. The Kaiser-Meyer-Olkin measure of sampling adequacy (KMO) generated a value 0.851 which is regarded as great (Pallant, 2016; Field, 2013). The Bartlett's Test of Sphericity was conducted and the results revealed the Bartlett's Test of Sphericity of $\chi^2(78) = 554.277$, $p < 0.0001$. This indicated that the correlations between the subscale items were adequate (Pallant, 2016; Field, 2013). The Kaiser criterion extracted three components with eigenvalues greater than 1 (Pallant, 2016), resulting in a total of 54.5% of the variance. The screeplot showed a clear break between the first and the second components, supporting the retention of a single component.

One component was retained as shown in Table 5-25.

Table 5-25: Component structure of SDM success measured through the quality of the developed systems development project artefact characteristics

SDM success through developed artefact quality characteristics (F0)	SDM ex-post success (F1)
All items	Quality was achieved when the systems development project artefact developed using the SDM met stated requirements.
	Quality was achieved when the systems development project artefact developed using the SDM gave consistent results when used.
	Quality was achieved when the systems development project artefact developed using the SDM provided accurate output for users.
	Quality was achieved when the systems development project artefact developed using the SDM had response time that met user expectations.
	Quality was achieved when the systems development project artefact developed using the SDM was stable and unlikely to fail.
	Quality was achieved when the systems development project artefact using the SDM accommodated changes suggested by the user.
	Quality was achieved when the systems development project artefact developed using the SDM was easily adapted to new specifications or changed environment.
	Quality was achieved when the systems development project artefact developed using the SDM was easy to use.
	Quality was achieved when the systems development project artefact developed using the SDM accommodated modification to remove defects.
	Quality was achieved when the systems development project artefact developed using the SDM met implied requirements.
	Quality was achieved when the systems development project artefact developed using the SDM recovered data and restored optimal functioning during failures.
	Quality was achieved when the systems development project artefact developed using the SDM maintained its level of performance under stated conditions for a stated period.
	Quality was achieved when the systems development project artefact developed using the SDM had a long-life expectancy.
Cronbach α	**0.841**

5.9 CRITERIA FOR SDM FIT TO SYSTEMS DEVELOPMENT PROJECT CONTEXTUAL STRESSORS

The assessment of SDMs, to ascertain their appropriateness to the systems development project contextual stressors, is presented in the next subsection. Respondents indicated the actual experiences in assessing the appropriateness of adopted SDMs to systems development contextual stressors in their respective organisations.

5.9.1 Assessing SDM's relative advantage to project contextual stressors

The question gathered data related to the respondents' experiences in assessing the relative advantage of SDMs in dealing with systems development contextual stressors. The question consisted of eight items with response weights ranging from 1 (representing totally disagree) to 6 (representing totally agree). The responses provided by respondents are summarised in Table 5-26. The mean responses for the eight items were in the range of 3.71 - 4.88 (on a 6-point scale). The adaptability of an SDM according to the demands of the contextual stressors indicated a mean score of 4.88, which was the highest mean score for the eight items. Table 5-26 presents the statistics for the items on assessing the SDM relative advantage to systems development project contextual stressors.

Table 5-26: Criteria for assessing relative advantage of SDMs to systems development project contextual stressors

N=155	Mean	Std. Dev	Frequencies as percentages					
			1	2	3	4	5	6
The SDM is reconfigured as the systems development project situation dictates.	4.88	0.918	0.0	0.6	7.1	23.2	41.3	27.7
SDMs complement each other in a systems development project in my organization.	4.68	1.093	1.3	1.9	10.3	25.2	36.8	24.5
SDM in use complies with fitness for purpose (doing things right) in each systems development project in my organization.	4.63	1.299	3.2	3.9	11.0	20.0	31.0	30.3
Use of an SDM is grounded in the realities of each systems development project situation in my organization.	4.56	1.495	5.8	8.4	5.8	16.8	30.3	32.9
The adopted SDMs or a set of SDMs components suit contextual characteristics of each systems development project situation in my organization.	4.55	1.228	3.2	2.6	12.3	23.2	35.5	23.2
The SDM in use is configured in each phase of the systems development life cycle during systems development project.	4.54	.982	0.0	0.6	11.0	42.6	25.2	20.6
The SDM in use complies with fitness of purpose (doing the right things) in each systems development project in my organization.	3.83	1.491	10.3	10.3	16.1	24.5	27.1	11.6
The SDM in use matches with all systems development situations.	3.71	1.512	7.1	18.1	20.0	21.9	17.4	15.5

A Cronbach's analysis was conducted on the eight items and the Cronbach's alpha level for the eight items was 0.716. This suggested an acceptable internal consistency reliability within the eight items (Pallant, 2016).

An assessment of the suitability of data for PCA was performed. The Kaiser-Meyer-Olkin measure of sampling adequacy (KMO) generated a value 0.776, which is classified as good (Pallant, 2016; Field, 2013). The Bartlett's Test of Sphericity was conducted, and a Bartlett's Test of Sphericity of $\chi^2(28) = 182.132$, $p < 0.0001$ was obtained. This indicates that the correlations between the items were adequate (Pallant, 2016; Field, 2013). The Kaiser criterion extracted two components with eigenvalues greater than 1 (Pallant, 2016), indicating a total of 47.0% of the variance. However, the screeplot showed a clear break between the first and the second component supporting the retention of only one component for further analysis. Furthermore, a Parallel Analysis based on Monte Carlo PCA was performed and the criterion values for a randomly generated data matrix of the same size (8 variables x 155 respondents) (Pallant, 2016) through

50 replications gave results shown on Table 5-27. The Parallel Analysis decision strategy compared the actual eigenvalue generated from PCA with the corresponding criterion value randomly generated by Monte Carlo PCA. When the actual eigenvalue generated from PCA was found to be greater than the corresponding criterion value randomly generated by Monte Carlo PCA, the component was retained. Otherwise it was removed as shown in Table 5-27. The Parallel Analysis supported the results from the screeplot to retain only one component as shown in Table 5-28.

Table 5-27: Component retention strategy using parallel analysis

Component number	Actual eigenvalue from PCA	Criterion value from Parallel Analysis (randomly generated from Monte Carlo PCA)	Decision
1	2.708	1.3501	Retained
2	1.053	1.2136	Removed

No rotation was necessary as there was only one component. The Cronbach alpha's level was the same as all the items constituted a single component. The resultant component matrix with a single factor is shown in Table 5-28.

Table 5-28: Component matrix for assessment of relative advantage of SDMs to systems development project contextual stressors

SDM relative advantage in organisations (F0)	SDM-relative advantage (F1)
All items	SDMs complement each other in a systems development project in my organization.
	Specific use of an SDM is grounded in the realities of each situation in my organization.
	SDM in use complies with fitness of purpose (doing the right things) in each systems development project in my organization.
	The adopted SDMs or a set of SDMs components suit contextual characteristics of each systems development project situation in my organization.
	SDM in use complies with fitness for purpose (doing things right) in each systems development project in my organization.
	The SDM is reconfigured as the systems development project situation dictates.
	The SDM in use matches with all systems development project situations.
	The SDM in use is configured in each phase of the development life cycle during systems development project.
Cronbach α	**0.716**

5.10 THE INSTANCES OF SDMS AND THE INTENSITY OF SDM USAGE IN ORGANISATIONS

The focus on the instances of SDMs and the intensity of their use within organisations were presented on this part of the questionnaire. The questions addressed the first objective of the study, which examined SDMs in use within the systems development organisations in South Africa.

5.10.1 Intensity of use of SDM instances in organisations

The question gathered data related to the specific SDMs in use within the systems development organisations and the intensity of their use. The question consisted of fourteen items with responses ranging from 1 (not at all) to 6 (to a higher intensity) on the intensity of use of a particular SDM. Respondents' responses are summarised in Table 5-29. The calculated means of the responses for the fourteen items were in the range of 1.19 - 5.50 (on a 6-point scale). The higher score indicated the greater intensity of use of the observed SDM. Only one respondent selected the other option indicating the intensive use of SSM and the item was removed from further statistical analysis as it was not representative. The intensity of use of Scrum SDM indicated the highest calculated mean score of 4.58 and had a low standard deviation of 0.973. The low standard deviation with respect to the calculated mean scores indicated a small variability on observed data, which meant that the mean scores were closer to the observed data (Hair *et al.*, 2018). Furthermore, the results revealed that 60.6% of the respondents indicated the use of Scrum SDM to a high intensity. Table 5-29 shows that only 1.3% of the respondents indicated that they were not using Scrum SDM at all. Although the intensity of use for SDMS varied, there was no SDM that was not used at all.

The Yourdon Systems Methodology (YSM) had the least calculated mean of 1.19 and a standard deviation of 0.507. A total of 86.5% of the respondents indicated that they did not use the Yourdon Systems Methodology (YSM) at all.

According to the calculated mean scores in Table 5-29, Scrum SDM, Rapid Application Development, Rational Unified Process (RUP), Home grown or in-house developed SDMs, PRINCE2 (Projects In Controlled Environments version 2), Water-Scrum-Fall, Microsoft Solutions Framework (MSF), Kanban, Structured Systems Analysis and Design Methodology (SSADM), XP (Extreme Programming), Crystal family, Jackson Systems Development (JSD) are at least used to a low intensity in systems development projects in organisations. The calculated mean scores in Table 5-29 indicate that Jackson Systems Development (JSD), Merise, Yourdon Systems Method (YSM) had very low intensity of use. The respondents that indicated not using Jackson Systems Development (JSD) at all constituted 36.1%. 39.4% respondents indicated that they did not use Merise at all, and 86.5% indicated no usage of Yourdon Systems Method (YSM). The Yourdon Systems Method (YSM) was removed from any further analysis due to a high percentage (86.5%) of respondents indicating that they do not use it and 13.6% of respondents indicating at most a low intensity of use.

Table 5-29: The extent of SDMs usage in organisations

N=155	Mean	Std. Dev	Frequencies as percentages					
			1	2	3	4	5	6
Other	5.50	0.707	0.0	0.0	0.0	0.6	0.5	0.0
Scrum	4.58	0.973	1.3	2.6	6.5	29.0	47.7	12.9
Rapid Application Development	3.85	1.144	6.5	4.5	20.6	34.8	32.9	0.6
Rational Unified Process (RUP)	3.68	1.791	23.9	4.5	9.0	20.0	27.7	14.8
Home grown/in-house developed	3.68	1.347	12.3	5.2	18.7	32.9	27.7	3.2
PRINCE2 (Projects In Controlled Environments)	3.65	1.126	6.5	6,5	28.4	32.9	25.8	0.0
Water-Scrum-Fall	3.32	1.329	10.3	15.5	29.0	29.0	9.0	7.1
Microsoft Solutions Framework (MSF)	3.28	1.540	20.6	10.3	20.0	23.9	20.0	5.2
Kanban	3.19	1.221	11.0	14.0	36.1	21.9	14.8	1.3
Structured Systems Analysis and Design Methodology (SSADM)	3.16	1.760	27.7	11.6	16.1	19.4	11.6	13.5
XP (Extreme Programming)	3.10	1.252	13.5	18.1	27.7	27.1	12.8	0.6
Crystal family	2.94	1.323	21.9	12.9	25.8	29.0	9.7	0.6
Jackson Systems Development (JSD)	2.51	1.388	36.1	14.8	20.0	20.0	9.0	0.0
Merise	2.20	1.281	39.4	22.6	26.5	5.2	3.2	3.2
Yourdon Systems Method (YSM)	1.19	0.507	86.5	8.4	5.2	0.0	0.0	0.0

A Cronbach's analysis was conducted on the thirteen items and their Cronbach's alpha level was 0.801. This indicates a good internal consistency reliability within the items (Pallant, 2016).

The suitability of data for PCA was performed. The Kaiser-Meyer-Olkin measure of sampling adequacy (KMO) generated a value 0.832, which is regarded as great (Pallant, 2016; Field, 2013). The results of Bartlett's Test of Sphericity of $\chi^2(78)$ = 475.522, p < 0.0001 indicated that the correlations between the subscale items were adequate (Pallant, 2016; Field, 2013). The Kaiser criterion extracted three components with eigenvalues greater than 1 (Pallant, 2016), that explained a total of 51.4% of variance. The screeplot showed a clear break between the third and the fourth components suggesting the retention of the three components.

The Promax with Kaiser Normalization rotation was performed (Pallant, 2016) on the three components, to generate a pattern of loadings that was easy to interpret without changing the underlying solution. The rotation converged in 6 iterations, resulting in a rotated solution revealing a simple component structure (Tabachnick and Fidell, 2013) presented in Table 5-30. The three components were subjected to Cronbach alpha level analysis and the first component had the highest Cronbach alpha level of 0.757, followed by the second component with Cronbach alpha level of 0.723 and the third component had a low Cronbach alpha level of 0.489. By treating Scrum SDM as a single item component and removing it from the third component, the Cronbach's alpha level of the third component increased to 0.502. The resulting four components were named as: 1) less used known SDMs, 2) Adaptive SDMs, 3) Popular structured SDMs, and 4) Scrum SDM. The Cronbach alpha level for each component is presented at the bottom row of

Table 5-30 and all were above the minimum of 0.5 (Field, 2013), except the fourth component which was constituted by a single item.

Table 5-30: Component structure for SDM intensity of use

SDM intensity of use (F0)	Less used known SDMs (F1)	Adaptive SDMs (F2)	Popular structured SDMs (F3)	Scrum SDM (F4)
All items	XP (Extreme Programming)	Rapid Application Development (RAD)	Structured Systems Analysis and Design Methodology (SSADM)	Scrum SDM
	Kanban	PRINCE2	Water-Scrum-Fall	
	Jackson Systems Development (JSD)	Home grown/in-house developed	Rational Unified Process (RUP)	
	Crystal family			
	Merise			
	Microsoft Solutions Framework (MSF)			
Cronbach α	**0.757**	**0.723**	**0.502**	**-**

5.10.2 Duration of use of SDMs in organisations

The question provided respondents with six options. The first five options were presented in the form of time intervals and the last one catered for those who did not have the information on the usage duration of their current SDM or SDMs. The results are presented in Table 5-31. 12,9% of the respondents indicated that the period of use of their current SDM was less than a year, 9.0% of the respondents indicated the interval to be between a year and two years, 22.6% of the respondents confirmed the period of which their current SDM had been in use in their organisation to be between three and five years. The respondents who indicated their SDM usage period to be between six and ten years were 33.5%. The last interval was the one covering the period above ten years and it was indicated by 15.5% of the respondents. The remaining 6.5% respondents did not know how long their current SDM had been in use.

Table 5-31: The period the current SDM has been in use

Time interval	N	Frequency as a percentage	Cumulative percentage
Less than 1 year	20	12.9%	12.9%
1 - 2 years	14	9.0%	21.9%
3 - 5 years	35	22.6%	44.5%
6 - 10 years	52	33.5%	78.1%
Over 10 years	24	15.5%	93.5%
Not known	10	6.5%	100%

5.11 CHAPTER SUMMARY

This chapter presented the descriptive statistics of the data set generated for the study. The analysis of the measurement subscales demonstrated good reliability and indicated strong and significant item loadings. The characteristics affecting SDM enactments were organised in three levels at which contingent use of SDMs can be analysed. The first level included those factors that were related to the individual systems development practitioner, such as the level of expertise and level of experience; the second level, the systems development project itself, which included factors such as requirements dynamics, available resources, and time constraints; and the third level consisted of the organisation factors such as the culture of the organisation.

More than a third (38.1%) of the respondents held senior positions in systems development projects in their organisations. The majority (79.4%) of the respondents had experience of at least six years in systems development projects. Almost a quarter (24.5%) of organisations that participated in the study had employees that held more than one position or some form of hybrid roles. This may be advantageous in cases where there are shortages of skilled personnel, but when an employee holds more than one role in an organisation, it is challenging to specialise. Perhaps that is the reason why the South African government classified systems development as a critical skills area (Government Gazette, 2014) to encourage learning institutions to develop targeted specialised skills in systems development.

At systems development project level, the systems development project contextual stressors are rarely, if ever, the same for each systems development project case. The SDMs are adopted and adapted to the evolving systems development project contextual stressors during the systems development project execution. A large proportion (77.4%) of the respondents indicated that in practice, SDMs are adopted and adapted on a project-to-project basis. Most (78%) SDMs have been in use for less than 11 years.

Some contextual stressors regarded by respondents as critical in systems development included the systems development project size (98.7%), systems developer experience (93.5%), systems development project criticality (97.4%), systems development stakeholders' politics (56.8%), systems development project costs (78.8%), systems development problem domain (79.9%), compliance with standards (49.7%), and systems development project artefact quality (83.9%). These contextual stressors are also theoretically and empirically important by Marks *et al*. (2017), Rajagopalan and Mathew (2016), Young *et al*. (2016), and Boehm (2006). The systems development project success was measured in terms of keeping within the triple constraints of predetermined schedule, cost, and quality.

Most organisations that participated in the survey classified their SDM adopting culture as market followers. In terms of size, 42.6% were large enterprises, 33.6% were medium enterprises and the rest fell under either micro or small enterprises. All participating organisations had their main area of competency in systems development. The SDM contingency with fixed pattern was the dominant strategy in adopting SDMs among organisations that participated in the survey.

CHAPTER SIX: RESULTS OF INFERENTIAL STATISTICS

6.1 INTRODUCTION

The previous chapter presented descriptive statistics for the study data. This chapter presents the inferential statistics of the study data. The factors influencing the contingent use of SDMs are empirically investigated, using correlation analysis and multiple regression analysis techniques. The empirical investigation of the contingent use of SDMs conceptual model developed in Chapter 3 is also presented. Finally, the hypotheses testing is also performed and presented.

6.2 ADOPTING SDMs WITHIN ORGANISATIONS IN SOUTH AFRICA

Results from the dataset used for the study indicated that the average time allocated to the adoption of an SDM was proportional to 12% of the time allocated to the systems development project (see subsection 5.2.4, Table 5-4). 98.6% (see subsection 5.5.1, Table 5-10) of organisations from the sample considered adopting SDMs or hybrid SDM as important. The majority (78.7%) of the organisations relied on SDM adoption based on best practices and experience (see subsection 5.4.1, Figure 5-3). The SDM adoption based on best practices are a trade-off between the systems development project team's need for ad-hoc or flexible SDM adoption and the policy based SDM adoption enforced by the organisation. The SDM adoption based on best practices allows reflection about the relevance on project-to-project basis and minimises contextual-stressor-free adoption of an SDM for a systems development project.

A crosstabulation analysis revealed that 10% of the systems development managers indicated their systems development team sizes to be less than six members (see subsection 5.3.2). However, 33.9% of systems development managers stated they had less than six people involved in the selection of an SDM for a systems development project (see subsection 5.3.4). For systems developers, 36.6% had less than six members in their systems development teams and 54.5% of the systems developers had less than six people involved in the selection of SDMs. There is strong evidence that fewer people are involved in the selection of SDMs for systems development projects than the whole systems development team members. The evidence was significant χ^2 (df = 4, N=155) = 10.821, $p < 0.05$). The results suggest that SDM selection decision is not devolved to all the members of a systems development project team. There is also higher responsibility and accountability associated with systems development managers and analysts (see subsection 5.3.5, Figure 5-2). Most organisations have a punitive system. Few individuals make decisions and when there is systems failure, blame and punitive measures fall on those individuals.

Significantly, it is not known which SDMs are adopted and used in South Africa. This study investigated the contingent use of SDMs in organisations at both systems development project and organisational levels. The SDMs currently in use in South African organisations are presented in the next section.

6.3 SDMS IN USE WITHIN ORGANISATIONS IN SOUTH AFRICA

This section is core in addressing the first objective of the study. Both the plan-driven SDM class and the agile SDM class have strengths and limitations given specific systems development project contextual stressors. It is not possible to assert that one class of SDMs is superior to another (Rahmanian, 2014; Janes and Succi, 2012). The adoption of SDMs varied across the three dimensions of organisation level, systems development project level and individual level based on the systems development project contextual stressors.

The adoption influence may also come from communities of SDM practice to which the organisation or the team members are affiliated. There are many systems development communities in South Africa where organisations or team members may have allegiance, such as the Institute of Information Technology Professionals South Africa (IITPSA), Agile Alliance, Joburg Centre for Software Engineering (JCSE), ScrumAlliance, PMI, PRINCE2 and Computer Society of the Institute of Electrical and Electronics Engineers. These professional bodies of systems development exert influence in the selection and adoption of specific SDMs for systems development projects, through undertaking training and certification of programs for their members. The JCSE, for instance, organises and hosts the annual Agile Africa Conference where a diversity of South African, African and international professionals from various systems development communities meet to share experiences in systems development (JCSE, 2016).

The Agile Africa conference advances the agile SDM class (JCSE, 2016). In South Africa, 99.4% of the organisations indicated the Scrum SDM (instance of the agile SDM class) to be the dominant SDM during the study. However, evidence indicates that neither the pure plan-driven SDM instances nor the pure agile SDM instances are adopted and used exclusively in practice (see subsection 5.10.1, Table 5-29). Notably, the Scrum SDM (instance of the agile SDM class) was not used as documented and intended by its creators, but it was combined with other SDMs to create hybrid SDMs (Table 6-1).

Due to variability in systems development project contextual stressors, a wide variety of SDMs are used in South African organizations (see subsection 5.10.1, Table 5-29). A significant negative relationship was found between the use of Scrum SDM and its use in its original documented version ($r = -.26$, $p < .01$) as indicated in Table 6-1. This shows that Scrum SDM is not used as-is, but is adapted to the systems development project circumstances. However, there was a significant positive relationship between the Scrum SDM use and the creation of hybrid SDM for a systems development project ($r = .28$, $p < .01$). The results confirmed the assertion made by Martin Fowler (2018), that Scrum is not used as proposed. Similarly, Ken Schwaber (2010) found that what organisations claimed as the use of Scrum SDM, in most cases, was a variation of Scrum SDM and not the version of Scrum SDM as proposed by its creators. A significant moderate positive relationship was found between the creation of alternative SDMs and the use of Scrum SDM ($r = .30$, $p < .01$). This suggests that Scrum SDM was combined with other SDM instances to gain synergies not possible with the implementation of Scrum SDM alone.

The use of SDMs as originally documented (as-is) had a moderate negative relationship with the Adaptive SDMs use ($r = -.34$, $p < .01$), a negative relationship with the less used known SDMs usage ($r = -.25$, $p < .01$), and a weak negative relationship with the usage of popular SDMs ($r = -.19$, $p < .05$). The Scrum SDM, Adaptive SDMs, less used known SDMs, and popular structured SDMs indicated statistically significant negative relationships regarding their rigid use as prescribed (see Table 6-1). This is because systems development is a knowledge intensive process, where systems development project details are progressively adjusted depending on the level of understanding of the systems development project contextual stressors. Each systems development project is unique, and in this regard, SDMs are adopted and adapted according to the demands of the systems development project contextual stressors.

A statistically significant positive relationship was found between SDM tailoring and the use of Adaptive SDMs ($r = .22$, $p < .01$). This suggests that Adaptive SDMs respond continuously to the changing systems development project contextual stressors when the need arises. The creation of alternative SDMs had a moderate positive relationship with the use of popular structured SDMs ($r = .30$, $p < .01$) and a positive relationship with less used known SDMs ($r = .21$, $p < .01$).

Table 6-1: Correlations of the SDMs in use and the variations in their implementation

Variable	Create alternative SDM	Tailor SDM	SDM use as-is
Scrum SDM	.297**	.280**	-.261**
Adaptive SDMs	.084	.218**	-.339**
Less used known SDMs	.210**	.140	-.245**
Popular structured SDMs	.301**	.106	-.185*

* p<.05, ** p<.01, *** p<.001

Results of the contingent use of SDMs within organisations in South Africa and the systems development project-specific contextual stressors are presented in the next section.

6.4 CONTINGENT USE OF SDMs WITHIN ORGANISATIONS IN SOUTH AFRICA

Empirical evidence shows that in most cases, the SDMS are not used as proposed (see section 6.3, Table 6-1). Evidence demonstrates that none of the organisations strictly conforms to the SDMs without considering the systems development project-specific contextual stressors (see subsection 5.9.1, Table 5-28). However, organisations deviate from the standard and documented versions of SDMs (see subsection 6.3, Table 6-1). The SDM adaptation and deviations are contextually framed based on the systems development project-specific contextual stressors. Thus, SDMs evolve during a systems development project life cycle (see subsection 5.6.5, Table 5-20).

It is not common for an SDM to be used rigidly as per its published version (Serrador and Pinto, 2015; Henderson-Sellers *et al*., 2014; Conboy and Fitzgerald, 2010). In practice, each SDM, even the one regarded as the most appropriate, is tailored (Henderson-Sellers *et al*., 2014) or adapted (Diebold *et al.*, 2015) to suit systems development project context (Viljoen, 2016; Sellers, Henderson-Sellers *et al*., 2014; Conboy and Fitzgerald, 2010; Brinkkemper, 1996; Russo *et al.,* 1996). Huisman (2013) defines 'contingency' as a matching of the SDM to the systems development project contextual stressors.

The contingent use of SDMs consists of three important phases of preadoption, adoption and postadoption (see subsection 3.11.3, Figure 3-10). The decision outcome chain is informed by these three phases. The study explored the influence of systems development project contextual stressors on the contingent use of SDMs. Some of the contextual stressors that influence the contingent use of SDMs are presented in the following subsections.

6.4.1 Organisation SDM adopting culture influence

The Competing Values Model (CVM) is used to classify organisations into four organisation culture categories: the group culture, the hierarchical culture, rational culture and the developmental culture (Iivari and Iivari, 2011; Iivari and Huisman, 2007). Iivari and Iivari (2011) state that CVM is a narrow view of organisational culture and that the organisational culture categories are ideal and that there are alternative views of organisational culture conceptualisations. The classification of organisational culture in this study targeted the responsiveness of an organisation in adopting SDMs as contingent innovations (see subsection 5.6.1, Table 5-17). Almost a quarter (24.5%) of the organisations that participated in the study were classified under the market leader SDM adoption culture (see subsection 5.6.1, Table 5-17).

Table 6-2 show results of the correlations of SDM adopting cultures and the SDMs in use. The market leader SDM adopting culture indicates no statistical significant relationship with all the

SDMs in use. The market follower SDM adopting culture indicates a significant negative relationship with Adaptive SDMs ($r = -.28$, $p < .01$).

The late majority SDM adopting culture indicates a negative statistically significant and moderate relationship with popular structured SDMs ($r = -.39$, $p < .05$), a negative relationship with Scrum SDM ($r = -.22$, $p < .01$). One explanation for this may be that the late majority SDM adopting culture avoids the risks of breaking new ground by pragmatically weighing the costs-benefits ratio experienced by both the market leaders and the market follower SDM adopting cultures. This suggests that the late majority SDM adopting culture is risk averse and is a strong barrier in the adoption of SDMs.

There was inconclusive evidence to establish any statistical significant relationship between the laggard SDM adopting culture and the SDMs in use as shown in Table 6-2.

Table 6-2: Correlations of SDM adopting cultures and the SDMs in use

Variable	Less used known SDMs	Popular structured SDMs	Adaptive SDMs	Scrum SDM
Market leader	.195	.051	139	.138
Market follower	-.107	.150	-.281**	-.061
Late majority	-.151	-.386*	-.110	-.216**
Laggard	.025	.087	.242*	.105

* p <.05, ** p <.01, *** p <.001

6.4.2 Influence of SDM adopting approaches

Correlations were used to determine relationships between the SDM adopting approaches and the SDMs in use within organisations as presented in Table 6-3. The results of the correlation indicated a positive weak relationship between adopting SDMs based on organisational policies and the adoption of structured SDMs ($r = .16$, $p < .05$), as well as adopting Adaptive SDMs ($r = .20$, $p < .05$). Both these relationships are statistically significant. Adopting an SDM is not a simple linear problem that can be addressed by the implementation of policies. The challenge with SDM adoption based on policies is that they may not consider contextual stressors and the particularities of each individual systems development project. For instance, an organisational policy may predetermine an SDM for all systems development projects. This violates the assertion that not all systems development projects are the same, and therefore they should not all be developed using the same SDM (Flowler, 2018; Vijayasarathy and Butler, 2015).

There was no credible evidence indicating a relationship between adopting SDM based on organisational policies and the use of less-used known SDMs ($r = .11$. $p > .05$) (Table 6-3). The relationship between adopting SDM based on organisational policies and the use Scrum SDM was also non-statistically significant (r = -.06, p > .05.

The correlation results indicated that there were non-statistically significant relationships between adopting SDMs based on SDM adopting frameworks and the use of less-used known SDMs (r = -.11, p > .05), popular structured SDMs (r = -.07, p > .05), Adaptive SDMs (r = -.04, p > .05), and the Scrum SDM (r = .01, p > .05) respectively.

Adopting SDMs based on best practices indicated non-statistically significant relationships with less-used known SDMs (r = -.03, p > .05), popular structured SDMs (r = -.04, p > .05), Adaptive SDMs (r = .02, p > .05), and the Scrum SDM (r = .12, p > .05) respectively.

Table 6-3: Correlations of SDM adopting approaches and the SDMs in use

Variable	Less used known SDMs	Popular structured SDMs	Adaptive SDMs	Scrum SDM
SDM adoption based on policies	.112	.164*	.201*	-.062
SDM adoption based on SDM adopting frameworks	-.113	-.070	-.044	.012
SDM adoption based on best practices and experience	-.027	-.035	.023	.120

* p <.05, ** p <.01, *** p <.001

6.4.3 Influence of SDM adopting strategies

Adopting SDMs is influenced by the assumptions made on systems development project contextual stressors (Isaias and Issa, 2015; Henderson-Sellers *et al*., 2014; Zhu, 2002). Contextual stressors determine the implementation of SDM adopting strategies to achieve the best results. Correlations were used to determine relationships between the adopting strategies and the SDMs in use. In the dataset used for the study, there was inconclusive evidence to indicate any statistical significant relationship between SDM contingency at outset and the SDMs in use as indicated in Table 6-4. There was no statistically significant relationship between the SDM contingency at fixed pattern and the SDMs in use. There was no credible evidence to indicate the existence of a statistically significant relationship between the SDM situational engineering and the SDMs in use.

Almost a half (48.4%) of the organisations that formed the study dataset indicated that they used SDM contingency along development dynamics as their SDM adoption strategy (see subsection 5.4.2, Figure 5-4). The SDM contingency along development dynamics indicated a positive statistically significant relationship with Adaptive SDMs (r = .33, p < .01). One explanation for this may be that in SDM contingency along development dynamics, SDMs are regarded as a response to particular configurations of contextual stressors. The SDM contingency along development dynamics is contingent to contextual stressors and allows adaptation of SDMs at any stage of the systems development project, irrespective of the SDM class instance in use. This also confirms that adoption of SDMs, or/and SDM components are contingent to the dynamics of the evolving systems development project contextual stressors.

Table 6-4: Correlations of SDM adopting strategies and the SDMs in use

Variable	Less used known SDMs	Popular structured SDMs	Adaptive SDMs	Scrum SDM
SDM contingency at outset.	-.036	.044	.042	-.032
SDM contingency with fixed pattern.	-.042	.023	-.043	.038
SDM contingency along development dynamics.	.-.028	.044	.333**	.139
SDM Situational Engineering	.015	.026	.059	-.133

* p <.05, ** p <.01, *** p <.001

6.4.4 Influence of SDM adopting standard practices

There is no SDM that fits all systems development project situations. The systems development professionals realised that they are systems development project contextual stressors that can be best addressed by a hybrid of SDM class instances. (Gill *et al.*, 2018; Imani *et al.*, 2017; Rahmanian, 2014). Correlations were used to determine the relationships between the SDM adopting practices and the SDMs in use as shown in Table 6-5. The selection of an SDM from a standard set of SDMs for each systems development project situation had a positive relationship with the less-used known SDMs (r = .26, p < .01), a positive relationship with popular structured SDMs (r = .21, p < .01) and a strong positive with Adaptive SDMs (r = .41, p < .01) respectively. This means that organisations have portfolios of SDMs from which the systems development practitioners choose the SDM that fit the systems development project contextual stressors.

There was no credible evidence indicating any statistically significant relationship between the selection of an SDM from a standard set of SDMs for each systems development project situation and the use of Scrum SDM.

Creation of hybrid SDMs from a set of standard SDMs for each systems development project situation had a significant positive relationship with the less-used known SDMs (r = .22, p < .01), the popular structured SDMs (r = .19, p < .05), the Adaptive SDMs (r = .23, p < .01), and the Scrum SDM (r = .26, p < .05) respectively. The results suggest that systems development projects contextual stressors vary from one systems development project to the other, or within the same systems development project. Therefore, in some cases, the best fit SDM could be a combination of SDM instances from SDM classes.

More than 70% of the respondents (see subsection 5.6.2, Figure 5-7) selected an SDM from in-house developed SDMs as their SDM adopting practice. However, there was no statistically significant relationship between the selection of an SDM from in-house developed SDMs for each systems development project situation and the SDMs in use.

The practice of selecting an SDM from a standard set of SDMs and adapting it for each systems development project situation indicated a significant positive relationship with the popular structured SDMs ($r = .23$, $p < .01$), and the Adaptive SDMs ($r = .19$, $p < .05$) respectively. This means that SDMs are adopted and adapted on a project-to-project basis and within projects to align with the evolving systems development project contextual stressors.

The creation of alternative SDMs had a significant positive relationship with less used known SDMs ($r = .21$, $p < .01$), a significant moderate positive relationship with popular structured SDMs ($r = .30$, $p < .01$) and Scrum SDM ($r = .30$, $p < .01$). There is no organisation that indicated non-use of SDM. Therefore, the statistical analysis could not be performed for this option as indicated in Table 6-5.

Table 6-5: Correlations of SDM adopting standard practices and the SDMs in use

Variable	Less used known SDMs	Popular structured SDMs	Adaptive SDMs	Scrum SDM
Select an SDM from a standard set of SDM for each systems development project situation.	.256**	.210**	.408**	.113
Select SDMs from a set of standard SDMs and combine them for each systems development project situation.	.224**	.193*	.225**	.144
Select an SDM from in-house developed SDMs for each systems development project situation.	.080	.071	.086	-.106
Select an SDM from a standard set of SDMs and adapt it for each systems development project situation.	.098	.225**	.189*	.117
Create an alternative SDM for each systems development project situation.	.210**	.301**	-.084	.297**
Non-use of SDMs in systems development projects.	N/A	N/A	N/A	N/A

* p <.05, ** p <.01, *** p <.001

6.4.5 Influence of SDM adopting cultures to the SDM adopting strategies

Correlations were used to investigate and determine the impact of the SDM adopting cultures in SDM adoption strategies. Table 6-6 shows the results of the correlations between the SDM adopting cultures and the SDM adopting strategies. There was no statistical significant relationship between the market leader SDM adopting culture and the SDM adopting strategies. There was a significant positive and moderate relationship between the market follower SDM contingency with fixed pattern ($r = .23$, $p < .01$). This means the more an organisation tends towards the market follower adopting culture, the more it promotes adjusting, modifying and even creating hybrid SDMs or alternative SDMs at certain stages of the systems development process.

There was a significant negative relationship between the market follower SDM adopting culture and the SDM contingency along development dynamics (r = .16, p < .05). Perhaps the market follower SDM adopting culture allows the systems development practitioners to learn from the market leaders and are aware of what works at different phases of systems development project. The more a market follower SDM adopting culture organisation is, the less its strategy will tend towards the SDM contingency along development dynamics.

There was a significant positive and moderate relationship between the late majority SDM adopting culture and the SDM contingency at outset strategy (r = .32, p < .01). Thus, the late majority base its strategy on what would have been learnt from the market leader SDM adopting culture and the market follower SDM adopting culture. The late SDM adopting culture is averse to evolving the SDM during the systems development project. The late majority SDM adopting culture indicated a significant negative relationship with the SDM contingency at fixed pattern (r = -.18, p < .01). One explanation for this is that the late majority SDM adopting culture selects the SDM from the outset and this is a hindrance to adapting the SDM as the systems development progress.

There was inconclusive evidence to establish statistically significant relationships between the laggard SDM adopting culture and the SDM adopting strategies as shown in Table 6-6.

Table 6-6: Correlations of SDM adopting cultures and SDM adopting strategies

Variable	SDM contingency at outset	SDM contingency with fixed pattern	SDM contingency along development dynamics	Create alternative SDM
Market leader	-.150	-.101	.108	-.077
Market follower	-.100	.229**	-.164*	.153
Late majority	.315**	-.187*	-.129	-.024
Laggard	.043	-.042	.144	-.032

* p <.05, ** p <.01, *** p <.001

6.4.6 Influence of systems development practitioner's experience with systems development projects to the SDM adopting cultures

Correlations were used to determine the influence of the experience of individual systems development practitioner with systems development projects and that of the SDM adopting cultures. The results are presented in Table 6-7. No credible evidence suggested any statistically significant relationship between the systems development practitioner's experience and the SDM adopting culture, except for the late majority SDM adopting culture. The late majority SDM adopting culture indicated a significant negative relationship with the systems development practitioner's experience (r = -.24, p < .01). This suggests that when the individual systems development practitioners gain experience, they may develop more resistance towards the late majority SDM adopting culture. This also means that adopting SDMs for a systems development project by experienced systems development practitioners, may be constrained by the late majority SDM adopting culture.

Table 6-7: Correlations of systems development practitioner experience with systems development projects and SDM adopting cultures

Variable	Market leader	Market follower	Late majority	Laggard
Individual systems development practitioner experience with systems development projects	.124	.005	-.241**	.043

* p <.05, ** p <.01, *** p <.001

6.4.7 Influence of some contextual stressors to the SDM adopting strategies

Results obtained through correlation analysis of some contextual stressors and SDM adopting strategies are presented in Table 6-8. The experience of individual systems development practitioners in systems development projects indicated a positive relationship with the SDM contingency with fixed pattern (r =.24, p < .01), while for SDM contingency along development dynamics indicated a significant positive relationship (r =.27, p < .01). This suggests that when systems development practitioners gain experience in systems development projects, the propensity towards adoption and tailoring of SDMs during a systems development project increases. There was not enough evidence to establish the relationship between systems development practitioner's experience and the SDM contingency at outset. There was significant positive relationship between the experience of the systems development practitioner and the creation of an alternative SDM (r = .24, p < .01). This implies that as the systems development team members gain experience, they gravitate towards in-house developed SDMs.

The size of the systems development project team indicated a negative relationship with the contingency at outset (r = -.18, p < .05) likely because as the systems development project team increases in size, the predetermination and predisposition of an SDM without matching it prior to the systems development project contextual stressors becomes a hindrance.

Table 6-8: Correlations of some contextual stressors and SDM adopting strategies

Variable	SDM contingency at outset	SDM contingency with fixed pattern	SDM contingency along development dynamics	Create alternative SDM
Individual systems development practitioner experience in systems development projects	-.079	.238**	.273**	.235**
The size of the systems development team in terms of the number of employees	-.184*	.018	.060	.053

* p < .05, ** p < .01, *** p < .001

6.4.8 Influence between some contextual stressors

There was a significant positive relationship between the experience of individual systems development practitioners in system development projects and the time spent on selecting an SDM for a systems development project ($r = .26$, $p < .01$) as shown in Table 6-9. This indicates that when the systems development practitioners gain experience, they become more deliberate in selecting SDMs for their systems development projects. They increasingly become aware of the strengths and the limitations of the SDMs in different systems development project contextual stressors.

There was significant positive relationship between the experience of systems development practitioners and the number of systems development projects that can run concurrently in an organisation ($r = .21$, $p < .01$). This is because as the systems development practitioners gain experience, the capacity to develop more systems development projects also increases. The size of an organisation in terms of the number of employees indicated a significant positive relationship with the time spent on SDM selection for a systems development project ($r = .24$, $p < .01$). This means that for bigger organisations more time is required to select an SDM for a systems development project. This is due to the management and control associated with large organisations.

There was a significant strong and positive relationship between the size of an organisation and the number of systems development projects that can run concurrently ($r = .41$, $p < .01$). This suggests that bigger organisations have more capacity in terms of skills base, expertise and other resources. The size of the systems development project team indicated a significant moderate and positive relationship with the number of systems development projects that may run concurrently in an organisation ($r = .37$, $p < .01$).

There was a positive relationship between the number of people involved in the selection of SDM for a systems development project and the time spent on selecting an SDM for a systems development project ($r = .24$, $p < .01$). Similarly, there was a moderate positive relationship with the number of systems development projects that can run concurrently ($r = .32$, $p < .01$). This means that when more people are involved in the selection of an SDM for a systems development project, this delays the decision-making process due to lengthy consultations. The systems development team members involved in the SDM selection may perceive contextual stressors differently and consensus may not be easily reached.

Table 6-9: Correlations between some contextual stressors

Variable	Time spent on SDM selection	Number of systems development projects that can run concurrently
Individual systems development practitioner experience in systems development projects.	.257**	.212**
The size of the organisation in terms of the number of employees.	.241**	.412**
The size of the systems development team in terms of the number of employees.	.208*	.368**
Number of people involved in selecting SDMs.	.237**	.324**

* $p < .05$, ** $p < .01$, *** $p < .001$

6.4.9 Contingent use of SDMs and the SDMs in use

Systems development practitioners adopt and adapt the SDM contingently, because each systems development project is unique. Results show that SDMs are used as they are, adapted, combined with other SDMs or SDM components to create hybrid SDMs on a systems development project-to-project basis. The correlations of some of the contextual stressors and the contingent use of SDMs are presented in Table 6-10. The construct *contingent use of SDMs* is a composite (latent) variable derived from averaging five Likert type items as explained in Chapter 4. A set of 13 candidate critical success factors were selected based on the hypothesised relationships in Chapter 3. Correlations of the candidate critical success factors are presented in Table 6-10. The *contingent use of SDMs* indicated positive significant relationships with SDM measure of success ($r = .31$, $p < .01$), systems development practitioner experience ($r = .36$, $p < .01$), systems development team size ($r = .28$, $p < .01$), horizontal use of SDMs ($r = .17$, $p < .05$), SDM fit assessment ($r = .22$, $p <$ 01), and SDM relative advantage ($r = .28$, $p < .01$).

However, the contingent use of SDMs indicated a significant negative relationship with the market follower SDM adopting culture ($r = -.22$, $p < .01$) and late majority SDM adopting culture ($r = -.20$, $p < .05$). A standard multiple regression was performed on the candidate critical success factors to determine their level of contribution in predicting the contingent use of SDMs. The standard multiple regression is presented in the next section.

Table 6-10: Correlations of candidate contextual stressors and the contingent use of SDMs

	Candidate critical success factors	Contingent use of SDMs
1	SDM success measure	.305**
2	SDM ex-post success	-.089
3	Systems development practitioner experience	.364**
4	Organisational size	.058
5	Systems development team size	.179*
6	Market leader SDM adopting culture	.139
7	Market follower SDM adopting culture	-.218**
8	Late majority SDM adopting culture	-.196*
9	Laggard SDM adopting culture	.129
10	Horizontal use of SDMs	.172*
11	Vertical use of SDMs	.128
12	SDM relative advantage	.283**
13	SDM fit assessment	.217**

* $p < .05$, ** $p < .01$, *** $p < .001$

6.5 CONTINGENT USE OF SDMS CRITICAL SUCCESS FACTORS (CSF)

The adopted SDM is adapted, adjusted or combined with some SDMs to achieve the ideal-fit to the systems development project-specific contextual stressors (see subsection 6.4.4, Table 6-4). However, not all systems development contextual stressors can be considered in every systems development project, nor are equally important or pertinent. Determining the systems development project contextual stressors that have significant influence on contingent use of SDMs was one of the main objectives of this study. Each systems development project contextual stressor that predicted the contingent use of SDMs with a *p*-value less than 0.05 was considered a critical success factor (CSF) for the contingence use of SDMs. The other systems development project contextual stressors that failed to meet the criteria are referred to as success factors.

6.5.1 Testing the assumptions of standard multiple regression

Prior to the application of the standard multiple regression, a preliminary analysis was conducted to ensure no violation of standard linear regression assumptions occurred. The normality of the data was tested by plotting the distributions of the residuals in a histogram as shown in Figure 6-1. The bell curve in Figure 6-1 indicates that the data are normally distributed.

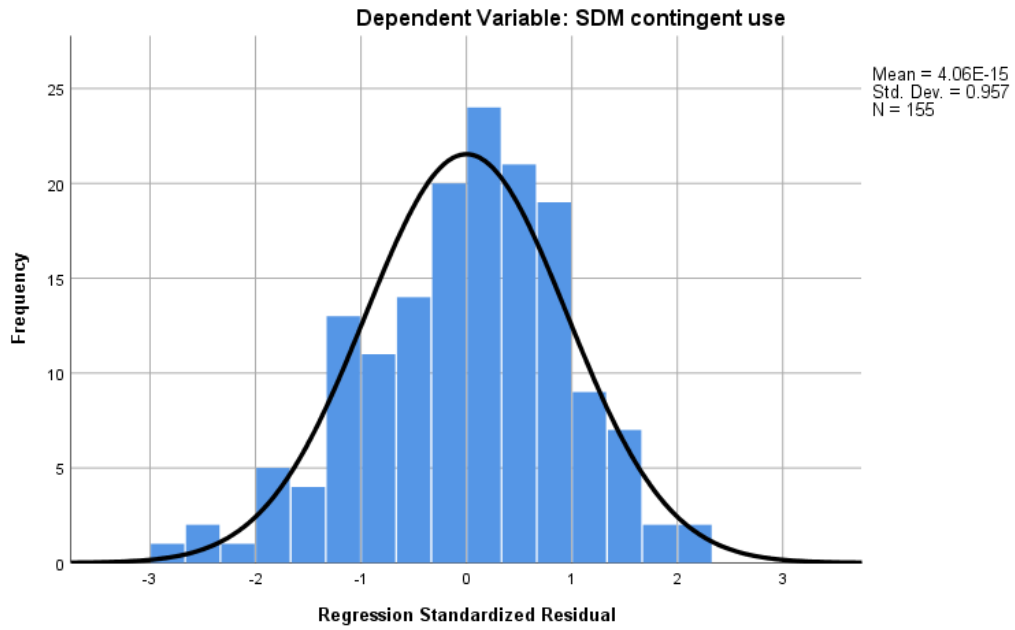


Figure 6-1: Histogram for standardised residual for contingent use of SDMs

The next check was on linearity. The independent variables showed that the data points in the Normal Probability Plot (P-P) of the regression standardised residuals followed an approximately straight diagonal line from bottom left to the top right as indicated in Figure 6-2. Thus, the assumption on linearity was verified.

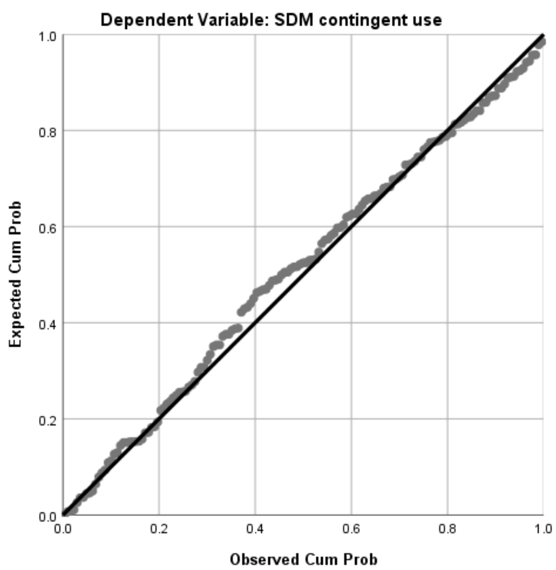


Figure 6-2: Normal Probability Plot (P-P) of the regression standardised residuals

The outliers were checked through the scatter plot shown in Figure 6-3. The points lie between -3 and 3. The Mahalanobis' distance was checked and the largest value was 32.639 corresponding to case number 36 and the Cook's Distance for this case was 0.00089, which is far less than 1. According to Fidell (2013), if the case has a Cook's Distance of less than 1, it does not present a potential problem for the assumption on the outliers. The outlier assumption was verified as valid.

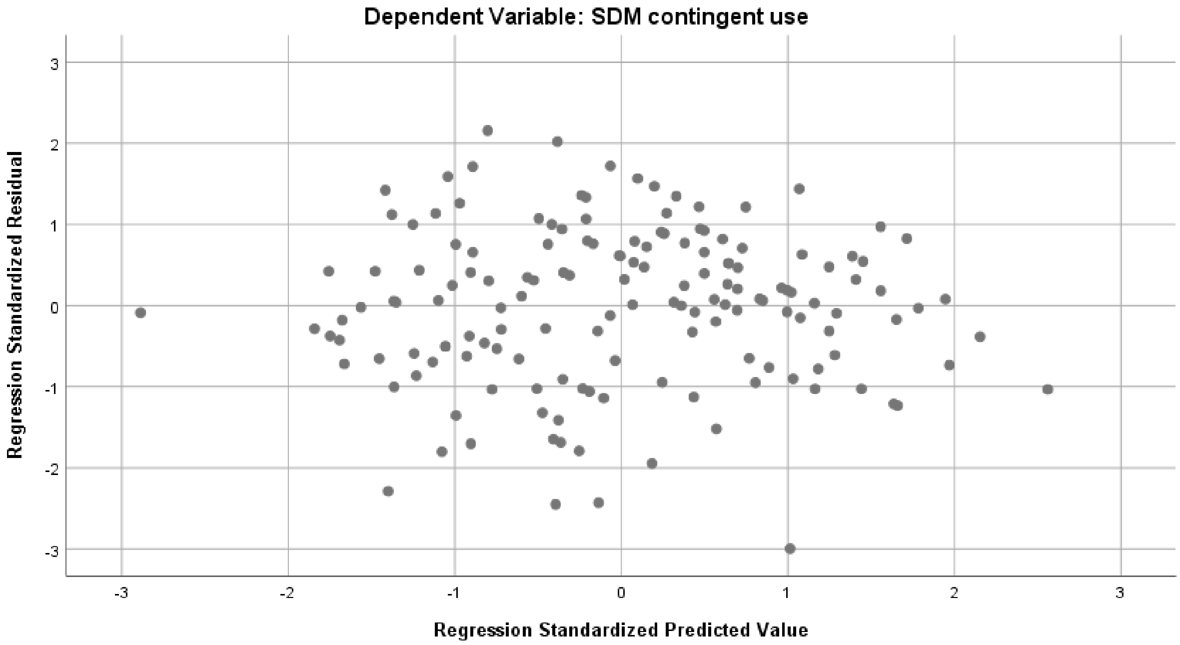


Figure 6-3: Scatter plot for the residual standardised values

The test for multicollinearity was performed through checking the Variance Inflation Factor (VIF) values as indicated in the standardised regression Table 6-10. The VIF were all less than 10, suggesting that the multicollinearity assumption was not a violated.

The assumptions for standard multiple regression were reasonably met to perform standard multiple regression analysis. The testing of hypothesis is presented in the next subsection.

6.5.2 Testing the hypotheses formulated for the study

The hypotheses formulated in Chapter 3 were subjected to statistical processes to establish their validity and the explanatory power with respect to the contingent use of SDMs. The previous subsections tested the validity of the standard multiple regression assumptions. The standard multiple regression was used to determine the extent of the influence of some of the proposed success factors for the contingent use of SDMs. The standard regression was also useful for determining the relative influence of each of the success factors for the contingent use of SDMs.

The model as a whole explained 49.5% of the variance in the contingent use of SDMs in the data set, $F(13,154) = 10.649$, $p < 0.001$. The effect size of $f^2 = 0.810$ is classified as large practical significance according to a method established by Cohen (1988).

Table 6-11: Contingent use of SDMs regression results

Model		Unstandardized Coefficients B	Std. Error	Standardized Coefficients Beta (*β*)	VIF
1	Explanatory variables				
	Intercept	1.130	0.831		
	Organisational level				
	Organisation size (number of employees) (X_1)	-0.121*	0.053	-0.163*	1.452
	Market leader SDM adopting culture (X_2)	-0.660*	0.267	-0.279*	3.563
	Market follower SDM adopting culture (X_3)	-1.001***	0.258	-0.446***	3.694
	Late majority SDM adopting culture (X_4)	-0.829**	0.278	-0.325**	3.341
	Laggard SDM adopting culture (X_5)	-0.556*	0.274	-0.221*	3.310
	Horizontal use of SDMs (X_6)	0.103	0.066	0.099	1.132
	Systems development project level				
	Systems development project team size (X_7)	0.205**	0.066	0.229**	1.515
	SDM success measures (X_8)	0.242**	0.090	0.180**	1.239
	SDM fit assessment (X_9)	0.226*	0.104	0.140*	1.162
	SDM ex-post success (X_{10})	-0.557***	0.128	-0.315***	1.477
	SDM relative advantage (X_{11})	0.536***	0.102	0.386***	1.502
	Individual systems development practitioner level				
	Individual systems development practitioner experience in systems development projects (X_{12})	0.200***	0.054	0.247***	1.262
	Vertical use of SDMs (X_{13})	0.158	0.065	0.149	1.039
	R^2	**.495**			
	Adjusted R^2	**.449**			
	f^2	**0.810**			
	F	**10.649******			

a. Dependent Variable: SDM contingent use * $p < .05$, ** $p < .01$, *** $p < .001$, ****$p < .0001$

The regression Table 6-11 provided the information to answer the question on the critical success factors for the contingent use of SDMs. A standard multiple regression analysis was conducted using 13 explanatory variables. The 13 explanatory variables are identified as X_1 through to X_{13} for clarity when writing the regression equation. The Unstandardised Coefficients corresponding to X_1 is $\boldsymbol{B}_1$ and for the corresponding Standardised coefficient is $\boldsymbol{\beta_1}$ and the explained variable contingent use of SDMs is presented by Y only for the regression model equation. However, in the explanation, the names of the explanatory and explained variable are used.

6.5.2.1 Model based on R^2

The standard regression equation model with Unstandardised Coefficients is presented as in equation 6.1 as a general equation with 13 variables explain Y and the equation 6.2 with the values for Unstandardised Coefficients substituted.

$$\mathbf{Y} = \text{intercept} + \boldsymbol{B}_1(X_1) + \boldsymbol{B}_2(X_2) + \boldsymbol{B}_3(X_3) + \boldsymbol{B}_4(X_4) + \boldsymbol{B}_5(X_5) + \boldsymbol{B}_6(X_6) + {}_7(X_7) + \boldsymbol{B}_8(X_8) + \boldsymbol{B}_9(X_9) + \boldsymbol{B}_{10}(X_{10}) + \boldsymbol{B}_{11}(X_{11}) + \boldsymbol{B}_{12}(X_{12}) + \boldsymbol{B}_{13}(X_{13}) + \boldsymbol{\varepsilon} \quad (6.1)$$

$$\mathbf{Y} = 1.130 + (\mathbf{-0.121})(X_1) + (\mathbf{-0.660})(X_2) + (\mathbf{-1.001})(X_3) + (\mathbf{-0.829})(X_4) + (\mathbf{-.556})(X_5) + (\mathbf{0.103})(X_6) + (\mathbf{0.205})(X_7) + (\mathbf{0.242})(X_8) + (\mathbf{0.226})(X_9) + (\mathbf{-0.557})(X_{10}) + (\mathbf{0.536})(X_{11}) + (\mathbf{0.200})(X_{12}) + (\mathbf{0.158})(X_{13}) + \varepsilon \quad (6.2)$$

The explanatory variables were grouped under three levels of abstraction through which contingent use of SDMs is theorised. These are the organisation, the systems development project, and the individual systems development practitioner levels. There were six explanatory variables considered under the organisational level, four explanatory variables under the systems development project, and three explanatory variables under the individual systems development practitioner level.

6.5.2.1.1 Critical success factors at organisation level

The market leader SDM adopting culture indicated a significant negative association with the contingent use of SDMs ($\boldsymbol{B_2}$ = -0.660, $p < .05$). This means that for every standard deviation unit change in the market leader SDM adopting culture, the contingent use of SDMs is predicted to result in a decrease of 0.66 standard deviation units holding all other explanatory variables constant. This is contrary to the hypothesised relationship in hypothesis *H2a*. The largest absolute value is found on the Unstandardised Coefficient for the market follower SDM adopting culture ($\boldsymbol{B_3}$ =-1.001, $p < .001$), which means that a one standard deviation unit increase in the market follower SDM adopting culture is predicted to result in a decrease of 1.001 standard deviation units in the contingent use of SDMs holding all other explanatory variables constant. The finding provides strong support for hypothesis *H2b* where it is hypothesised that the market follower SDM adopting culture has a negative relationship with the contingent use of SDMs. This confirms previous research findings by Huisman and Iivari (2006) related to the deployment of SDMs in organisations. The late majority SDM adopting culture was significantly and negatively associated with the contingent use of SDMs ($\boldsymbol{B_4}$ = -0.829, $p < .001$). This implies that for every standard deviation unit increase towards the late majority SDM culture orientation, the contingent use of SDMs is predicted to decrease by 0.829 standard deviations units holding all other explanatory variables constant. This confirms the relationship hypothesised in hypothesis *H2c* that the late majority SDM adopting culture has a negative relationship with the contingent use of SDMs. This finding is in line with the findings by McLeod and MacDonell (2011), Siakas and Siakas (2007), Iivari and Huisman (2007). Iivari and Huisman (2007) found a significant positive relationship between the hierarchical and the rational organizations and the deployment of plan-driven SDMs. Their study focused on the relationship between organisational culture and the deployment of SDMs, whereas this study focused on the contingent use of SDMs. The laggard SDM adopting culture indicated a significant negative relationship with the contingent use of SDMs (B_5 = -0.556, $p < .05$). This finding supports hypothesis *H2d,* where the relationship between the laggard SDM adopting culture is hypothesised as negative. The dominant organisational culture, irrespective of the type, indicated a negative relationship with the contingent use of SDMs. Perhaps this is caused by the fact that the contingent use of SDMs is not based on the SDM classes. The agile SDM class favours a certain organisational culture, while the plan-driven SDM class also favours another. However, the contingent use of SDMs is neither agile nor plan-driven oriented. No culture orientation was favourable for the contingent use of SDMs. These results were unexpected since the market leader SDM adopting culture was predicted to have a positive relationship with the contingent use of SDMs. Perhaps this is because the market leader SDM adopting culture embraces the latest innovation and fails to consider specific contextual stressors of a systems development project at hand.

Organisation size indicated a significant negative influence on the contingent use of SDMs (B_1 = -0.121, $p < .05$). This means that for every standard deviation increase in the size of an organisation, there is a predicted corresponding decrease of 0.121 standard deviation units on the contingent use of SDMs. The finding confirms the hypothesised relationship in hypothesis *H3* that organisational size has a negative influence on the contingent use of SDMs. It is also in line with previous findings on the impact of organisational size in the adoption and use of SDMs (Viljoen, 2016; Turner and Zolin, 2012; Barlow *et al*., 2011). Viljoen (2016) found that as an organisation grew, it tended to use the SDM as a guideline. There was no statistical significance in performing standardised regression analysis on the horizontal use of SDMs on the contingent use of SDMs. Therefore, the hypothesised relationship in hypothesis *H8* that the horizontal use of SDMs positively influences the contingent use of SDMs, was not supported.

6.5.2.1.2 Critical success factors at systems development project level

The systems development team size (B_7 = 0.205, $p < .01$) was significantly and positively related to the contingent use of SDMs. Respondents of larger systems development teams indicated a high propensity towards the contingent use of SDMs. This supports hypothesis *H7,* but in the opposite direction. This is likely because when the team increase in size, the SDM is adapted to meet the requirements of team roles assignment and division of systems development tasks. The SDM success measure significantly and positively influenced the contingent use of the SDMs (B_8

= 0.242, $p < .01$). This supports the hypothesised relationship in hypothesis *H5.* The regression result for assessing the appropriateness of an SDM (SDM fit assessment), throughout the course of a systems development project, indicated a significant and positive relationship with the contingent use of SDMs (B_9 = 0.226, $p < .05$). This supports hypothesis *H1*. The finding is consistent with other empirical evidence that SDMs are adapted to specific systems development project situations (Diebold *et al.*, 2015; Conboy and Fitzgerald, 2010; Burns and Deek, 2011; Barlow *et al*., 2011; Turner, 2003). The ex-post success of an SDM indicated a significant negative relationship with contingent use of SDMs (B_{10} = -0.557, $p < .001$). The finding supports hypothesis *H4.* The result is logical, as a history of success of an SDM may result in the users resisting change or adjustment to an SDM that performed successfully on previous occasions. The unstandardised coefficient between SDM relative advantage (B_{11} = 0.536, $p < .001$) and the contingent use of SDMs was statistically significant and positive, and this supported hypothesis *H10*. Respondents indicated that they judged SDMs based on their relative advantage over others given the specific systems development contextual stressors.

6.5.2.1.3 Critical success factors at individual systems development practitioner level

More than 80% of the respondents had experience of more than 5 years in systems development projects. The experience of an individual systems development practitioner act as a guide to adopting, adapting, adjusting, discarding or creating alternative SDMs based on the assessment of the SDM fit to the systems development project-specific contextual stressors. The individual systems development practitioner experience in systems development projects was significantly and positively related to the contingent use of SDMs (B_{12} = 0.200, $p < .001$). This meant that for every one standard deviation unit increase in experience, there is a predicted 0.2 standard deviation units increase in the contingent use of SDMs. That is, respondents who had high levels of experience in systems development projects rated the contingent use of SDMs favourably. The individual systems development practitioner, with high levels of experience in systems development projects, can make detailed evaluation of an SDM in use. The finding supports the hypothesised relationship in *H6.* The relationship between the vertical use of SDM and the contingent use of SDMs was non-significant. Thus, there was no credible evidence to support the hypothesised relationship in *H9*.

6.5.2.2 Model based on Adjusted R^2

The standard regression equation model with Standardised Coefficients is presented in figure 6.3 as a general equation with 13 variables explaining **Y**′ and the equation 6.4 with the values for Standardised Coefficients substituted.

$$\mathbf{Y'} = \text{intercept} + \boldsymbol{\beta}_1(X_1) + \boldsymbol{\beta}_2(X_2) + \boldsymbol{\beta}_3(X_3) + \boldsymbol{\beta}_4(X_4) + \boldsymbol{\beta}_5(X_5) + \boldsymbol{\beta}_6(X_6) + \boldsymbol{\beta}_7(X_7) + \boldsymbol{\beta}_8(X_8) + \boldsymbol{\beta}_9(X_9) + \boldsymbol{\beta}_{10}(X_{10}) + \boldsymbol{\beta}_{11}(X_{11}) + \boldsymbol{\beta}_{12}(X_{12}) + \boldsymbol{\beta}_{13}(X_{13}) + \boldsymbol{\varepsilon} \quad (6.3)$$

$$\mathbf{Y'} = 1.130 + (\mathbf{-0.163})(X_1) + (\mathbf{-0.279})(X_2) + (\mathbf{-0.446})(X_3) + (\mathbf{-0.325})(X_4) + (\mathbf{-0.221})(X_5) + (\mathbf{0.099})(X_6) + (\mathbf{0.229})(X_7) + (\mathbf{0.180})(X_8) + (\mathbf{0.140})(X_9) + (\mathbf{-0.315})(X_{10}) + (\mathbf{0.386})(X_{11}) + (\mathbf{0.247})(X_{12}) + (\mathbf{0.149})(X_{13}) + \boldsymbol{\varepsilon} \quad (6.4)$$

The regression model based on Adjusted R^2 explained 44.9% of the variance in the contingent use of SDMs in the data set is $F(13,154) = 10.649$, $p < 0.001$. The first model with unstandardised coefficients was used to analyse the contribution of each explanatory variable holding all other explanatory variables constant. The model with standardised coefficients was used to determine the relative importance of each explanatory variable in the model with respect to the contingent use of SDMs. The overall model with standardised coefficients focused on both the overall predictive effect of all variables and the relative contribution of each explanatory variable. The market follower SDM adopting culture has the largest unique contribution ($\boldsymbol{\beta}_3 = -0.446$, $p < .001$). Consequently, the market follower SDM adopting culture had a large impact in the contingent use of SDMs based on the study data set.

6.5.2.3 Summary of critical success factors for the contingent use of SDMs

The thirteen factors were organised according to the three levels of contingent use of SDMs abstraction and the standard regression analysis only showed eleven factors as statistically significant. Factor analysis was performed in Chapter 5 and reduced the number of independent variables by grouping them into components. Some independent variables were removed for failing to support the contingent use of SDMs conceptual model developed in Chapter 3. The resulting components supported the contingent use of SDMs conceptual model developed in Chapter 3. The components and other single variables were then used in the regression analysis and reduced to critical success factors. Figure 6-4 presents the critical success factors for the contingent use of SDMs. The list of the success factors does not imply that other factors are not important for the contingent use of SDMs, however, it means that for the study these were rated as the significant critical success factors.

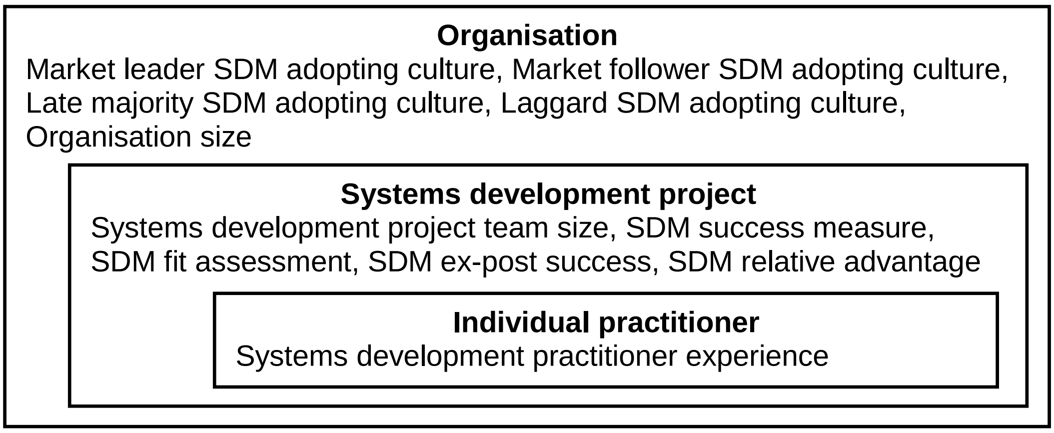


Figure 6-4: Critical success factors for the contingent use of SDMs

6.6 CHAPTER SUMMARY

This chapter presented results from the empirical evidence collected for the study. The chapter also presented inferential statistics. The critical success factors (CSFs) for the contingent use of SDMs were determined through multivariate regression analysis. The next chapter presents findings and conclusions.

CHAPTER SEVEN: FINDINGS AND CONCLUSIONS

7.1 INTRODUCTION

This chapter presents the study findings, discussions, contributions, conclusions, and limitations. It also suggests areas of further research. Further, the research findings are presented and justified in relation to existing literature, and the empirical evidence collected for the study. The chapter also evaluates the aim and the objectives of the study formulated in Chapter 1. The hypothesis testing results are also presented. Additionally, the chapter discusses the implications of the findings for research and practice. Finally, the chapter presents limitations and suggestions for further study.

7.2 REVISTING THE RESEARCH OBJECTIVES

The research question of this study was: Can the contingent use of SDMs conceptual model be developed to investigate the contingent use of SDMs and guidelines proposed for the contingent use of SDMs to assist systems development organisations? Consequently, the aim of the study was to develop a systems development contingent use conceptual model to investigate the contingent use of SDMs and propose guidelines to assist South African system development organisations with the contingent use of SDMs. To accomplish the aim of the study, four research objectives were formulated and the next four subsections evaluate these objectives.

7.2.1 Research objective 1

The first objective was stated in Chapter 1 as follows:
Examine systems development methodologies in use within the systems development organisations in South Africa.

Chapter 5 and Chapter 6 indicate that there is no conclusive evidence to suggest a single SDM class instance in use within the systems development organisations in South Africa. According to empirical findings, the dataset used for the study of SDMs in use cannot be split into two mutually exclusive categories of plan-driven SDM class and agile SDM class in the context of systems development in South Africa. The variations in SDM implementation make it difficult to identify the specific SDM in use, as the instance of an SDM in use may be a temporal hybrid of various SDM components from different SDM classes (see subsection 5.6.5, Table 5-20; section 6.3, Table 6-1).

The findings indicate a hybrid usage of SDMs (see subsection 5.10.1, Table 5-29). Therefore, findings provide strong evidence that both the plan-driven SDM and the agile SDM class coexist (see subsection 5.10.1, Table 5-29) as asserted by Janes and Succi (2012).

The findings show a diverse SDM enactment and deployment trend where different SDMs are implemented either as adapted versions of specific SDMs or hybrid SDMs to address the systems development project specific contextual stressors. (see subsection 5.6.2, Figure 5-5; subsection 6.4.4, Table 6-5). Thus, a base SDM is what an organisation can adopt a priori and adapt it on an ad hoc project-to-project basis (see subsection 5.6.2, Figure 5-5). The base SDM is usually a variant of an instance of ether an agile SDM class or plan-driven SDM class or a hybrid SDM. These findings are consistent with those of Gill *et al*. (2018) and Imani *et al.* (2017) who found evidence that systems development project contextual stressors can be addressed most effectively by a hybrid of SDM instances.

In terms of adoption and use in systems development projects, the Scrum SDM emerged as the most dominant base SDM (see subsection 5.10.1, Table 5-29). However, Scrum SDM was not used as-is, but was adapted and used in combination with other SDMs of either the same agile SDM class or the plan-driven SDM class (see section 6.3, Table 6-1). The findings confirm Martin Fowler's (2018) assertion that Scrum is not used as proposed that is also true for the South African systems development context. The findings are also consistent with those of Schwaber (2010) who found that what organisations claimed as the use of Scrum SDM, in most cases was a version variation of the Scrum SDM as proposed by its creators. Rapid Application Development (RAD) SDM, Rational Unified Process (RUP) SDM and in-house developed SDMs had a high intensity of use.

Therefore, South African organisations enact, adopt and adapt SDMs or SDM components contingently to address specific contextual stressors (see subsection 5.6.5, Table 5-20; subsection 5.10.1, Table 5-29). This addresses the systems development project-specific contextual stressors, rather than the naming of the SDM class or the SDM instance in use. The findings show that most SDMs are used with some level of variation from their original documentation and are also combined with other SDMs or SDM components of either the same SDM class or a different SDM class (see subsection 5.6.5, Table 5-20, Figure 5-5; section 6.3, Table 6-1). The findings also indicate that the relationship between perceived strengths and limitations of SDMs is more important than their class based discrimination. In fact, SDM tailoring is contingent to the limitations (misfits or misalignments) of an SDM to the perceived systems development project contextual stressors. That is SDMs are adopted, adapted and used based on their fit with the systems development project-specific contextual stressors. This is consistent with the findings of Gill *et al*. (2018) and Isaias and Issa (2015) that different SDMs provide alternative or complementary ways of dealing with contextual stressors of the system development project.

7.2.2 Research objective 2

The second objective was stated in Chapter 1 as follows:
Identify the contingent use of systems development methodologies in organisations in South Africa.

Chapter 5 and Chapter 6 provided findings that addressed the second study objective. Each systems development project is unique, and in this regard, the systems development practitioners adopt and adapt the SDM contingently (see subsection 5.6.5, Table 5-20). According to the empirical evidence, SDMs are used as they are, adapted, combined with other SDMs or SDM components to create hybrid SDMs on a systems development project-to-project basis (see subsection 5.6.2, Figure 5-5; section 6.3, Table 6-1). The significant correlations of some key contextual stressors and the (latent construct) contingent use of SDMs indicate the level of contingent use of SDMs in systems development organisations in South Africa (see subsection 6.4.9, Table 6-10). Evidence suggests that contingent use of SDMs is part of the systems development practices in South Africa.

7.2.3 Research objective 3

The third objective was stated in Chapter 1 as follows:
Identify the critical success factors of the contingent use of systems development methodologies.

Based on the literature, the conceptual model and the hypothesised relationships in Chapter 3, a set of 13 candidate critical success factors for the contingent use of SDMs was identified. The candidate critical success factors were organised into three levels of abstraction; the organisation level, the systems development project level and the individual systems development practitioner level (see subsection 6.5.2, Table 6-11). The formulated research hypotheses tested the significance of each critical success factor and empirically validated the contingent use of SDMs conceptual model. A standard multiple regression was conducted to determine which candidate critical success factors were predictors of the contingent use of SDMs. The standard regression model containing all the predictors was statistically significant at $F(13, 154) = 10.649$, $p < 0.001$ (see subsection 6.5.2, Table 6-11). The effect size was $f^2 = 0.810$ and this indicates a large practical significance according to Cohen (1988) (see subsection 6.5.2, Table 6-11). The standard multivariate regression results indicated an overall model of eleven critical success factors that predict the contingent use of SDMs. The eleven significant independent variables (critical success factors) shown to predict the contingent use of SDMs in this research study were; 1) SDM success measure (β = 0.180, p < .01), 2) SDM ex-post success (β = -0.315, p < .001), 3) systems developer practitioner experience (β = 0.247, p < .0001), 4) organisation size (number of employees)(β = -0.163 p < .05), 5) systems development team size (β = 0.229, p < .01), 6) market leader SDM adopting culture (β = -0.279, p < .05), 7) market follower SDM adopting culture (β = -0.446, p < .05), 8) late majority SDM adopting culture (β = -0.325, p < .05), 9) laggard SDM adopting culture (β = -0.221, p < .05), 10) SDM relative advantage (β = 0.386, p < .01), and 11) SDM fit assessment (β = 0.140, p < .05). The model based on Adjusted R^2 explains 44.9% of the variance in the contingent use of SDMs. The Cohen's (1988) states that f^2 =0.810 indicates a large practical significance. This implies that in a systems development project, SDM success measure, SDM ex-post success, systems developer practitioner experience, organisation size, systems development team size, organisation SDM adopting culture, SDM relative advantage, and SDM fit assessment have the most impact on influencing the contingent use of the SDMs. These critical success factors can help system development practitioners to understand factors that are likely to affect the contingent use of SDMs in their organizations.

The summary of hypotheses testing results is presented in the next subsection.

7.2.3.1 Research study hypotheses testing results

A total of thirteen hypotheses were formulated in Chapter 3 and were empirically tested in Chapter 6 using empirical evidence from systems development industry in South Africa. A standard multiple regression model assessment demonstrated the predictive power of the contingent use of SDM model based on the empirical evidence for the study. The empirical validation indicated that only eleven hypotheses were supported. A summary of hypotheses testing results is presented in Table 7-1. The final model is presented in Figure 7.1.

Table 7-1: Summary of hypotheses testing results

Hypothesis tested	Results
H1: The SDM fit assessment positively influences the contingent use of SDMs.	*Supported*
H2a: There is a positive relationship between the market leader SDM adopting culture and the contingent use of SDMs.	*Supported but opposite direction*
H2b: There is a negative relationship between the market follower SDM adopting culture and the contingent use of SDMs.	*Supported*
H2c: There is a negative relationship between the late majority SDM adopting culture and the contingent use of SDMs.	*Supported*
H2d: There is a negative relationship between the laggard SDM adopting culture and the contingent use of SDMs.	*Supported*
H3: The organisation size negatively influences the contingent use of SDMs.	*Supported*
H4: The SDM ex-post success negatively influences the contingent use of SDMs.	*Supported*
H5: The SDM success measure positively influences the contingent use of SDMs.	*Supported*
H6: The individual systems development practitioner's experience negatively influences the contingent use of SDMs.	*Supported*
H7: The systems development project team size negatively influences the contingent use of SDMs.	*Supported but in the opposite direction*
H8: The horizontal use of SDM positively influences the contingent use of SDMs.	*Not supported*
H9: The vertical use of SDM positively influences the contingent use of SDMs.	*Not supported*
H10: The SDM relative advantage to the systems development project contextual stressors positively influences the contingent use of SDMs.	*Supported*

7.2.4 Research objective 4

The fourth objective was stated in Chapter 1 as follows:
Develop a contingent use of SDMs conceptual model and the guidelines for the contingent use of SDMs.

Chapter 3, Chapter 6 and Chapter 7 address the requirements of the fourth objective. This objective is the main contribution of this research study, which is to develop a conceptual model for the contingent use of SDMs to describe and explain the contingent use of SDMs in organisations. The original contingent use of SDM model was validated against empirical evidence from systems development industry in South Africa. The contingent use of SDMs model evolved after validation to the final contingent use of SDMs model shown in Figure 7.1. The second part of the objective entails the proposal of contingent use of SDM guidelines and is presented in the next subsection.

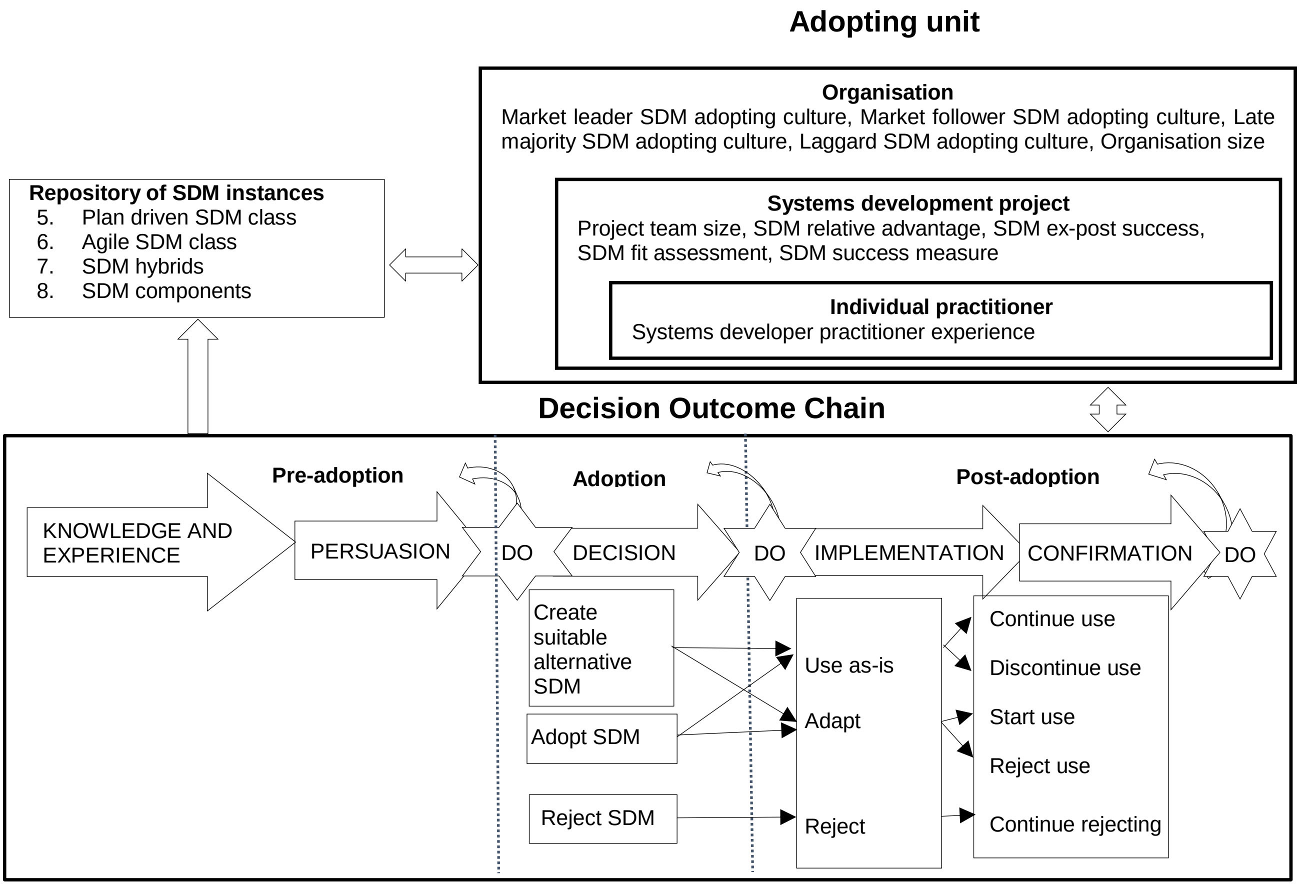


Figure 7-1: The final contingent use of SDM conceptual model

7.2.4.1 Guidelines for the contingent use of SDMs

The guidelines on the contingent use of SDMs are based on literature reviewed, hypotheses formulated in Chapter 3, and the critical success factors for the contingent use of SDMs. The basic assumption is that systems development project contextual stressors evolve and have different levels of importance at different stages of the project. The adoption of an SDM is not a once-off event, but a continuous evaluation for an ideal fit between the SDM in use and the evolving systems development project contextual stressors. Changes that emerge from the interaction between the SDM and the contextual stressors require constant assessment of prior and current decisions on the appropriateness of the SDM to the contextual stressors. Contingent use of SDMs attempts to minimize the discrepancy between the state of practice of the SDM initially adopted and the trajectory of the system development project. The continuous assessment of the state of alignment between the systems development project contextual stressors and the SDM is the focus of the contingent use of SDMs. The implementation of the guideline is important. For example, creating an alternative SDM may involve components from both plan-driven and agile SDM classes. Therefore, the assignment of roles to members of the systems development team may also need tailoring. However, the guidelines presented in Figure 7-2 form the first version of the proposal for the contingent use of SDMs. The contributions of the research are presented in the next section.

CONTINGENT USE OF SDMS GUIDELINES

1. Identify organisation size, organisation SDM adopting culture, systems development team size, systems development project artefact success criteria, and individual systems developer experience.
2. Evaluate the interaction between the adopted SDM, systems development project contextual stressors, and the development team for strain or misfit.
3. In case of ideal fit use SDM as-is and keep referring to step 6.
4. If there are minor misfits; tailor and deploy and keep referring to step 6.
5. Otherwise create an alternative SDM and deploy and keep referring to step 6.
6. Assess the SDM fit to the systems development contextual stressors continuously during the systems development project (systems development project contextual stressors evolve over time and their relevance might change) and go to step 2.

Figure 7-2: Contingent use of SDMs guidelines

7.3 RESEARCH CONTRIBUTIONS

The main research contributions are classified into three aspects: the theoretical, the practical and the methodological. The next subsection discusses the research contributions under these three aspects.

7.3.1 Theoretical Contributions

The main theoretical contribution of this study is the development of a contingent use of SDMs conceptual model for describing and explaining the evolving contextual stressors and their impact on the SDM during a systems development project life cycle. The contingent use of SDMs conceptual model draws on complementary insights from three theoretical models, namely DOI (Rogers, 2003) TAM (Venkatesh and Davis, 1996) and TTF (Goodhue, 1995). The SDM conceptual model presents a unique combination of the three Gregor (2006) Type IV theories that provide the necessary constructs to guide the investigation into the contingent use of SDMs. Not only does the study contribute towards the Gregor (2006) Type IV theories, but also addresses Alter's (2017) call to consider conceptual artefacts in Computer Science and Information Systems research. The conceptual artefacts are addressed by developing the contingent use of SDMs conceptual model itself, the formulation of conceptual definition of contingent use of SDMs and the development of guidelines for the contingent use of SDMs.

The innovation decision process in DOI (Rogers, 2003) is viewed as a sequence of decisions, actions and events in a stable community of adopting units. It is perhaps viewed as a sequential process because it is easy to understand it that way. The decision stage of the innovation process is expected to produce a single outcome in the end. The study contributes by explicitly proposing the adoption outcome chain to reflect DOI's iterative nature. The adoption outcome chain reorganises the innovation decision process into a three-phase iterative process of pre-adoption, adoption and post-adoption. The adoption outcome chain is not linear, but has multiple decision points and multiple outcomes. The emergence of a decision on the use of an SDM is guided by the evolving contextual stressors.

The DOI in its original application assumes a single instance of a contingent innovation at a time (Rogers, 2003). In this study, the DOI is extended to focus on the contingent use of SDMs (various instances of contingent innovations). Again, DOI assumes the same class of adopting units at a time (Jeyaraj and Sabherwal, 2014). If the adopting unit class is composed of individuals, then all that is considered as the unit of analysis should be individuals. If systems development projects are considered, then the unit of analysis should be systems development projects, and if organisations are considered only organisations will constitute the unit of analysis. The study considered a nested approach to adopting units (Viljoen, 2016). The adopting units are organised into three levels, the individual systems development practitioner, the systems development project and the organisation which results in three units of analysis.

The DOI does not explicitly describe the complex process of contingent use of SDMs in detail. It indicates that one either adopts or rejects a contingent innovation. The decision is not so dichotomous, straightforward and simplistic. This creates the need to propose an alternative way to describe the contingent use of SDMs. The alternative way is presented as the modification of DOI and using it as an iterative multi-phased and multi-staged decision process referred to as the decision outcome chain. The decision outcome chain spans through three levels, the organisation, the systems development project and the individual systems development practitioner level. The systems development project contextual stressors evolve and have different levels of importance at different times during the systems development project life cycle. Changes that emerge from the interaction between the SDM, the contextual stressors and the adopting unit, requires constant evaluation of prior and present decisions on the appropriateness of an SDM to the contextual stressors.

Roger’s (2003) view of the implementation stage assumes a pure technical perspective. Rogers (2003) considered the implementation stage as a technical process, such as installing a software package and starting to use it, customise it or reject its use. However, implementation is neither a pure technical process nor a once-off event. The empirical evidence shows that there are several steps to be taken during implementation that may involve decisions on fine-tuning, tailoring, adjusting, adapting, and creation of suitable alternatives or creating workarounds. The decisions impact changes along the course of the systems development project (Zhu *et al.*, 2016).

The TTF has not been fully investigated in the field of SDMs. The study presents evidence on the importance of fitting the SDM to the contextual stressors. The research developed a contingent use of SDMs conceptual model to conceptualise the contingent use of SDMs as a dynamic process of a continuum of fits. The study also developed and measured the contingent use of SDMs construct and empirically tested the contingent use of SDMs conceptual model using a survey data collected from systems development organisations.

7.3.2 Methodological Contributions

The definition proposed by Huisman and Iivari (2006) is a comparative framework that can be used by researchers to analyse, evaluate, and compare SDMs. The definition of contingent use of SDMs provides a frame of reference for comparing studies of similar nature. The theoretical and empirical evidence show that selecting one SDM over another does not imply superiority of one SDM over another, but indicates the emphasis of aspects which are prioritised based on the appropriateness of the selected SDM to the contextual stressors at any given moment.

The development of a contingent use of SDMs conceptual model presented in Chapter 3 is one of the major methodological contributions of the study. The contingent use of SDMs conceptual model assists researchers to investigate the relevant aspects of contingent use of SDMs in organisations. In general, the contingent use of SDMs conceptual model is a means that can be used by researchers to investigate the contingent use of SDMs in organisations. The empirical evidence provides a guideline to both the researcher and the practitioner on systems development process as a value centred process (Biffl *et al*., 2006), which emphasises and prioritises tasks based on the configuration of contextual stressors at each moment of the systems development project. The study demonstrates the appropriateness of the survey method in the investigation into the contingent use of SDMs in organisations. A three-tier perspective was adopted to provide an integrated understanding of the contingent use of SDMs across the organisation level, the systems development project level and individual level.

7.3.3 Practical Contributions

The study provides empirical evidence against the value of contrasting plan-driven SDM class and the agile class for superiority. This can be useful to practitioners in terms of considering the relevance of contextual stressors when selecting an SDM or emphasising the different aspects of an SDM for a systems development project. Notwithstanding the fact that some philosophical assumptions may be incompatible, they may be also complementary to each other. Each systems development project is unique, and in this regard, an SDM that provides an ideal fit to its contextual stressors is necessary. The findings indicate that in some cases, instances of both SDM classes may be combined to form a hybrid SDM that provides the necessary fit or interleaved in the same systems development project.

The research builds a foundation for understanding the contingent use of SDMs through identifying a set of critical success factors. These critical success factors constitute the archetypes that are relevant within the context of systems development projects depending on their configuration. A first draft of guidelines for the contingent use of systems development methodologies is proposed. A comprehensive survey data generation instrument was developed and validated and can be used by researchers for data collection to engage in further research in this area.

7.4 STUDY LIMITATIONS

One of the limitations of this study is that it was difficult to establish the total population of systems development companies in South Africa. This was because the Department of Trade and Industry (DTI) does not allocate a separate economic sector for the systems development industry in its economic classification, so systems development companies may be placed within any sector such as Finance sector, Business sector, or Other Services sector (MICT SETA, 2017). The second limitation was that the prospective participants in the identified systems development companies were mostly unwilling to participate in the completion of the data collection instrument. The third limitation relates to the inconsistences and conceptual incongruences about SDM definitions. For instance, it is not clear whether an organisation that adopts a practice such as pair programming, can be considered as having XP as its SDM in use. The roles assigned to systems development practitioners varied and this constrain shared perspectives on the contingent use of SDMs as views are shaped by the role-task interaction.

7.5 FURTHER RESEARCH

This research could be expanded by further empirical testing of this contingent use of SDM model, after ascertaining the number of systems development companies in South Africa or replicating it in another country. Data can also be collected from more than one participant per organisation and cover all roles and responsibilities in a systems development project. The analysis based on hierarchical multiple regression model may also be a way of extending the research to expose the effects of the order, where different levels of the contingent use of SDMs can be considered. The study may also include organisations whose main area of competence is not systems development, to investigate the extent to which the contingent use of SDMs construct can explain the practice in organisations whose main area of competence is not systems development.

7.6 CONCLUSIONS

This study indicates that one SDM does not fit all systems development situations. It extends the idea based on the contingency theory that fit is temporal and should be assessed continuously during a systems development project based on the contextual stressors. The study set out to achieve four objectives. The first one investigated the SDMs in use in South Africa. Results reveal that SDMs are being used by South African systems development organisations. The second objective identified the contingent use of systems development methodologies in South African organisations. The third objective identified the critical success factors for the contingent use of SDMs. The last objective developed a conceptual model for the contingent use of SDMs. The literature review provided this researcher with critical success factors and gave the theoretical support for the development of both the conceptual framework for the study and the questionnaire for data collection. The critical success factors in previous studies are in relation to the selection and adoption of SDM and not the contingent use of SDMs. Using standard multiple linear regression, a proposed set of eleven critical success factors was determined. Organisational culture, especially the market follower SDM adopting culture, provides the largest unique contribution to the model of contingent use of SDMs.

7.7 CHAPTER SUMMARY

This chapter unifies the research performed in this study. The chapter also discussed the study findings, and evaluated the formulated objectives and the posed research hypotheses. The chapter also highlighted important aspects of the research findings. The research contributions are classified into three perspectives, the theoretical, the practical and the methodological. Finally, the chapter outlines the study limitations and discusses the directions for further research.

REFERENCES

Abrahamsson, P., Salo, O., Ronkainen, J., and Warsta, J. (2017) Agile software development methods: Review and analysis, [online] Retrieved from: https://arxiv.org/ftp/arxiv/papers/1709/1709.08439.pdf, Date accessed: 25 August 2018.

Aitken, A., and Ilango, V. (2013) A comparative analysis of traditional software engineering and agile software development, *In System Sciences (HICSS), 2013, 46th Hawaii International Conference*, pp. 4751-4760.

Albers, M.J. (2017) Introduction to Quantitative Data Analysis in the Behavioral and Social Sciences, Hoboken: John Wiley and Sons.

Alter, S. (2017) Nothing is more practical than a good conceptual artefact which may be a theory, framework, model, metaphor, paradigm or perhaps some other abstraction, *Information Systems Journal*, Vol. 27, No. 5, pp.671-693.

Ansari, S.M., Fiss, P.C., and Zajac, E.J. (2010) Made to fit: How practices vary as they diffuse, *Academy of Management Review*, Vol. 35, No. 1, pp. 67-92.

Avison, D. and Fitzgerald, G. (2006) Information Systems Development Methodologies, Techniques and Tools, London: McGraw Hill.

Avison, D. E., and Fitzgerald, G. (2003) Where Now for Development Methodologies?, *Communications of the ACM* , Vol. 46, No. 1, pp. 78-82.

Avison, D. E., and Taylor, V. (1997) Information Systems development methodologies: A classification according to problem situation, *Journal of Information Technology*, Vol. 12, No. 1, pp. 73-81.

Bahari, S. F. (2010) Qualitative versus quantitative research strategies: contrasting epistemological and ontological assumptions, *Sains Humanika,* Vol. 52, No. 1, pp. 17-28.

Baker, J. (2012) The technology-organization-environment framework, In Information systems theory: Explaining and Predicting Our Digital Society, vol. 1, New York: Springer.

Bannon Jr, W. (2015) Missing data within a quantitative research study: How to assess it, treat it, and why you should care, *Journal of the American Association of Nurse Practitioners*, Vol. 27, No. 4, pp. 230-232.

Barlow, J. B., Giboney, J., Keith, M. J., Wilson, D., Schuetzler, R. M., Lowry, P. B., and Vance, A. (2011) Overview and guidance on agile development in large organizations, *Communications of the Association for Information Systems*, Vol. 29, No. 2, pp. 25-44.

Beck, K. (2000) Extreme programming explained: Embrace change, Boston: Addison-Wesley.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001). Manifesto for Agile Software Development, [online] Retrieved from: http://agilemanifesto.org/, Date accessed: 25 August 2018.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., & Fowler, M. (2016) 12 principles behind the agile manifesto, Agile Alliance [online] Retrieved from:

https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/, Date accessed: 4 August 2018.

Berente, N., Hansen, S. W., and Rosenkranz, C. (2015) Rule Formation and Change in Information Systems Development: How Institutional Logics Shape ISD Practices and Processes, *48th Hawaii International Conference on System Sciences*, pp. 5104-5113.

Bertram, C., and Christiansen, I.M. (2014) Understanding research: an introduction to reading research, Pretoria: Van Schaik.

Bhattacherjee, A., Limayem, M., and Cheung, C. M. (2012) User switching of information technology: A theoretical synthesis and empirical test, *Information and Management*, Vol. 49, No. 7, pp. 327-333.

Bhattacherjee, A., Perols, J., and Sanford, C. (2008) Information technology continuance, A theoretical extension and empirical test, *Journal of Computer Information Systems,* Vol. 49, No. 1, pp. 17-26.

Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., and Grünbacher, P. (2006) Value-based software engineering, Berlin: Springer Science and Business Media.

Boehm, B. (2006) A View of 20th and 21st Century Software Engineering, *ACM Proceedings of the International Conferences on Software Engineering*, pp. 12-30.

Boehm, B. W. (1988) A spiral model of software development and enhancement, *IEEE Computer*, Vol. 21, No. 5, pp. 61-72.

Boehm, B., and Turner, R. (2004) Balancing Agility and Discipline: A Guide for the Perplexed, Boston: Addison-Wesley.

Boehm, B., and Turner, R. (2005) Management challenges to implementing agile processes in traditional development organizations, *IEEE Software*, Vol. 22, No. 5, pp. 30-39.

Boehm, B., Turner, R. (2003) Using risk to balance agile and plan-driven methods, *IEEE Computer Society,* Vol. 36, No. 6, pp. 57-66.

Booch, G. (1994) Object-Oriented Design with Applications, Redwood City: Benjamin Cummings.

Boone, H. N., and Boone, D. A. (2012) Analyzing Likert data, *Journal of extension*, Vol. 50, No. 2, pp. 1-5.

Börner, R. (2011) Towards construction of situational methods for service identification, *In Working Conference on Method Engineering*, pp. 204-218.

Bossini, J., and Fernández, A.R. (2013) Using agile methodologies in people management, *RPM,* Vol. 10, No. 1, pp. 33-42.

Brechner, E. (2015) Agile project management with Kanban, Washington: Pearson Education.

Brinkkemper, S. (1996) Method engineering: engineering of information systems development methods and tools, *Information and software technology*, Vol. 38, No. 4, pp. 275-280.

Brix, J. (2017) Exploring knowledge creation processes as a source of organizational learning: A longitudinal case study of a public innovation project, *Scandinavian Journal of Management*, Vol. 33, No. 2, pp. 113-127.

Brooks, F. (1987) No silver bullet: Essence and accidents of software engineering, *IEEE Computer*, Vol, 20, No. 4, pp. 10-19.

Brown, S. A., Dennis, A. R., and Venkatesh, V. (2010) Predicting Collaboration Technology Use: Integrating Technology Adoption and Collaboration Research, *Journal of Management Information Systems*, Vol. 27, No. 2, pp. 9-53.

Brown, S. A., Venkatesh, V., and Goyal, S. (2014) Expectation confirmation in information systems research: A test of six competing models, *MIS Quarterly*, Vol. 38, No. 3, pp. 729-A9.

Bryman, A., and Bell, E. (2011) Business Research Methods, Oxford: Oxford University Press.

Bui, Q. (2015) A review of innovation diffusion theories and mechanisms, *In Proceedings of the Twentieth DIGIT 2015 Workshop*, [online] Retrieved from: https://pdfs.semanticscholar.org/e344/53e8f747f2fc93617d62e16016b ce6a6007c.pdf, Date accessed: 5 November 2018.

Burns, N., and Grove, S.K. (2003) The practice of nursing research: Conduct, critique and utilization, Toronto: WB Saunders.

Burns, R.N., and Dennis, A.R. (1985) Selecting the appropriate application development methodology, *ACM SIGMIS Database*, Vol. 17, No. 1, pp. 19-23.

Burns, T. J., and Deek, F. (2011) A methodology tailoring model for practitioner based information systems development informed by the principles of general systems theory, *Journal of Information Systems Applied Research*, Vol. 4, No. 2, pp. 28-37.

Bustard, D., Wilkie, G., & Greer, D. (2013) The diffusion of agile software development: insights from a regional survey, *In Information Systems Development*, pp. 219-230.

Bygstad, B., Ghinea, G., and Brevik, E. (2008) Software development methods and usability: Perspectives from a survey in the software industry in Norway, *Interacting with Computers*, Vol. 20, No. 3, pp. 375-385.

Byrne, B.N. (2001) Structural Equation Modeling with AMOS: Basic Concepts, Applications and Programming, Mahwah: Lawrence Erlbaum.

Carroll, J. (2003) The process of ISD methodology selection and use: a case study, *Proceedings of the 11th European Conference on Information Systems,* pp. 379-392.

Carson, D., Gilmore, A., Perry, C., and Gronhaug, K. (2001) Qualitative Marketing Research, London: Sage.

Cavaye, A.L. M. (1995) User participation in systems development revisited, *Information and Management*, Vol. 28, No. 5, pp. 311-323.

Chan, F.K.Y., and Thong, J.Y.L. (2009) Acceptance of agile methodologies: A critical review and conceptual framework, *Decision Support Systems*, Vol. 46, No. 4, pp. 803-814.

Charvat, J. (2003) Project Management Methodologies: Selecting, implementing and supporting methodologies and processes for projects, New York: John Wiley and Sons.

Chatzoglou, P.D., and Macaullay, L.A. (1996) Requirements capture and IS methodologies, *Information Systems Journal,* Vol. 6, No. 3, pp. 209-225.

Check, J., and Schutt, R. K. (2011) Research methods in education, London: Sage.

Checkland, P.B. (1981) Systems Thinking, Systems Practice, Redwood City: John Wiley.

Checkland, P.B. (2000) Systems thinking, systems practice: includes a 30-year retrospective, *Journal-Operational Research Society*, Vol. 51, No. 5, pp. 647-647.

Chow, T., and Cao, D. B. (2008) A survey study of critical success factors in agile software projects, *Journal of Systems and Software*, Vol. 81, No. 6, pp. 961-971.

Clarke, P., and O'Connor, R. V. (2015) Changing situational contexts present a constant challenge to software developers, *European Conference on Software Process Improvement,* pp. 100-111.

Clarke, P., and O'Connor, R.V. (2012) The situational factors that affect the software development process: Towards a comprehensive reference framework, *Journal of Information and Software Technology*, Vol. 54, No. 5, pp. 433-447.

CMMI, P.T. (2010) CMMI for Development, Version 1.3, (CMU/SEI-2010-TR-033). Pittsburgh: Software Engineering Institute, [online] Retrieved from: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=9661, Date accessed: 20 July 2018.

Cockburn, A. (2002) Agile software development, Agile software series, Boston: Addison-Wesley.

Cockburn, A. (2004) Crystal clear: A human-powered methodology for small teams: A human-powered methodology for small teams, Upper Saddle River: Pearson Education.

Coe, R. (2002) It's the effect size, stupid: what effect size is and why it is important, *Paper presented at the Annual Conference of the British Educational Research Association*, held 12–14 September 2002, at the University of Exeter, UK, [online] Retried from: www.leeds.ac.uk/educol/documents/00002182.htm, Date accessed: 17 June 2017.

Cohen, J. (1988) Statistical power analysis for the behavioral sciences, (2nd edn), N.J.: Lawrence Erlbaum.

Cohn, M. (2004) User stories applied: For agile software development, Boston: Addison-Wesley.

Collyer, S., and Warren, C. (2009) Project Management Approaches for Dynamic Environments, *International Journal of Project Management*, pp. 355-364, [online] Retrieved from: http://dx.doi.org/10.1016/j.ijproman.2008.04.004, Date accessed: 23 July 2017.

Conboy, K. (2009) Agility from first principles: Reconstructing the concept of Agility in information systems development, *Information Systems Research*, Vol. 20, No. 3, pp. 329-354.

Conboy, K., and Fitzgerald, B. (2010) Method and developer characteristics for effective agile method tailoring: A study of XP expert opinion, *ACM Transactions on Software Engineering and Methodology (TOSEM),* Vol. 20, No. 1, pp.1-30.

Conger, S. (2013) Software Development Life Cycles and Methodologies: Fixing the old and adopting the new, *All Sprouts Content*., [Online] Retrieved from: http://aisel.aisnet.org/sprouts_all/428, Date accessed: 2 November, 2018.

Creswell, J. W. (2009) Research design: Qualitative, quantitative, and mixed methods approaches, Thousand Oaks: Sage.

Cuellar, M. (2010) Assessing project success: Moving beyond the triple constraint, *International Research Workshop on IT Project Management*, pp. 19-28.

Davis, F. D. (1993) User acceptance of information technology: system characteristics, user perceptions and behavioral impacts, *International journal of man-machine studies*, Vol. 38, No. 3, pp. 475-487.

Davis, F. D. (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology, *MIS Quarterly*, Vol. 13, No. 3, pp. 319-339.

Davis, F. D., Bagozzi, R. P., and Warshaw, P.R. (1989) User acceptance of computer technology: a comparison of two theoretical models, *Management Science*, Vol 35, No. 8, pp. 982 -1003.

De Vaus, D. (2001) Research design in social research, London: Sage.

De Vos, A. S., Delport, C. S. L., Fouché, C. B., and Strydom, H. (2011) Research at grass roots: A primer for the social science and human professions, Pretoria: Van Schaik.

DeLone, W. H., and McLean, E. R. (2003) The DeLone and McLean model of information systems success: a ten-year update, *Journal of management information systems*, Vol. 19, No. 4, pp. 9-30.

Dhurka, D.K. (2015) Agile Transformation 2015 and the Gartner's Hype Cycle. [online] Retrieved from: http://deepadhurka.blogspot.co.za/2015/08/agile-transformation2015-and-gartners.html, Date accessed: 20 November 2018.

Diebold, P., Ostberg, J. P., Wagner, S., and Zendler, U. (2015) What do practitioners vary in using scrum?, *In International Conference on Agile Software Development*, pp. 40-51.

Dijkstra, E. W. (1970) Structured Programming, *In Software Engineering Techniques, B. Randell and J.N. Buxton, (Eds.), NATO Scientific Affairs Division*, pp. 84-8.

Dillman, D. A., Smyth J. D., Christian L. M. (2014) Internet, phone, mail, and mixed-mode surveys: The tailored design method, Hoboken: John Wiley and Sons.

Dima, A. M., and Maassen, M. A. (2018) From Waterfall to Agile software: Development models in the IT sector, 2006 to 2018, Impacts on company management, *Journal of International Studies*, Vol. 11, No. 2, pp. 315-326.

Dishaw, M.T., and Strong, D.M. (1999) Extending the technology acceptance model with Task-Technology Fit constructs, *Information and Management*, Vol. 36, No. 1, pp. 9-21.

DTI. (2008) Annual Review of Small Businesses in South Africa 2005-2007. In Industry, D. O. T. A. (edn.), Pretoria: DTI.

Dwolatzky, B. (2008) CMMI making inroads in South Africa, [online] Retrieved from http://www.ee.co.za/article/cmmi-making-inroads-in-south-africa.html, Date accessed: 15 October 2017.

Dybå, T. (2005) An Empirical Investigation of the Key Factors for Success in Software Process Improvement, *IEEE Transactions on Software Engineering*, Vol. 31, No. 5, pp. 410-424.

Dybå, T., and Dingsøyr, T. (2009) What Do We Know about Agile Software Development? *IEEE Software*, Vol. 26, No. 5, pp. 6-9.

Dybå, T., Sjøberg, D, I, K., and Cruzes, D. S. (2012) What Works for Whom, Where, When, and Why? On the Role of Context in Empirical Software Engineering, *Proceeding of the ACM IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 19-28.

Enders, C. (2010) Applied Missing Data Analysis, New York: Guilford Press.

Erlank, S., Pelteret, D., and Meskin, M. (1991) An analysis of the usage of system development methods in South Africa, *South African Computer Journal,* No. 4, pp. 68-77.

Eva, M. (1994) SSADM Version 4: A User's Guide, Maidenhead: McGraw-Hill.

Eva, M., and Guilford, S. (1996) Committed to a Radical approach? A survey of systems development methods in practice. *In Lessons Learned from the Use of Methodologies: Proceedings of the Fourth Conference of the British Computer Society Information Systems Methodologies Specialist Group*, pp. 87-96.

Eveleens, J. L., and Verhoef, C. (2010) The Rise and Fall of the Chaos Report Figures, *IEEE Software*, Vol. 27, No. 1, pp. 30-36.

Fichman, R. G. (2004) Going beyond the dominant paradigm for information technology innovation research: Emerging concepts and methods, *Journal of the Association for Information Systems*, Vol. 5, No. 8, pp 314-355.

Field, A. (2013) Discovering statistics using IBM SPSS Statistics (4th ed.). London: Sage.

Fitzgeral, B., Russo, N., and O'Kane, T. (2000) An Empirical Study of System Development Method Tailoring in Practice, *In Proceedings of 8th European Conference on Information Systems,* pp. 187-194.

Fitzgerald, B. (1994) Whither systems development: time to move the lamppost, *In Proceedings of Second Conference on Information Systems Methodologies,* pp. 371-380.

Fitzgerald, B. (1996) Formalized systems development methodologies: A critical perspective, *Information Systems Journal*, Vol. 6, No. 1, pp. 3-23.

Fitzgerald, B. (1997) The use of systems development methodologies in practice: a field study, *Information Systems Journal*, Vol. 7, No. 3, pp. 201-212.

Fitzgerald, B. (1998) An empirical investigation into the adoption of systems development methodologies, *Information and Management*, Vol. 34, No. 6, pp. 317-328.

Fitzgerald, B., Hartnett, G., and Conboy, K. (2006) Customising Agile Methods to Software Practices at Intel Shannon, *European Journal of Information Systems*, Vol. 15, No. 2, pp. 200-213.

Fitzgerald, B., Russo, N. L., and O'Kane, T. (2003) Software development method tailoring at Motorola. *Communications of the ACM*, Vol. 46, No. 4, pp. 65-70.

Flowler, M. (2018) The State of Agile Software in 2018, The 10th Annual Agile Astralia Conference, 18-19 June, Melboune [online] Retrieved from: https://martinfowler.com/articles/agile-aus-2018.html, Date accessed: 12 June 2019.

Foddy, W. (1994) Constructing Questions for Interview and Questionnaires. Cambridge: Cambridge University Press.

Fowler, M (2004) "Is Design Dead?", Software Development, [online] Retrieved from: http://www.martinfowler.com/articles/designDead.html, Date accessed: 23 July 2018.

Fowler, M. (2005) The New Methodology [online] Retrieved from: http://www.martinfowler.com/articles/newMethodology.html, Date accessed: 23 July 2018.

Fowler, M., and Highsmith, J. (2001) The agile manifesto, *Software Development*, Vol. 9, No. 8, pp. 28-35.

Geambasu, C. V., Jianu, I., Jianu, I., and Gavrila, A. (2011) Influence factors for the choice of a software development methodology. *Accounting and Management Information Systems*, Vol. 10, No. 4, pp. 479-494.

Geels, F. W. (2010) Ontologies, socio-technical transitions (to sustainability), and the multi-level perspective, *Research policy*, Vol. 39, No. 4, pp. 495-510.

Gill, A.Q., Henderson-Sellers, B., Niazi, M. (2018) Scaling for agility: A reference model for hybrid traditional-agile software development methodologies, *Information Systems Frontiers: a journal of research and innovation*, Vol. 20, No. 2, pp. 315-341.

Goodhue, D. L. (1995) Understanding user evaluations of information systems, *Management Science*, Vol. 41, No. 12, pp. 1827-1844.

Goodhue, D. L., and Thompson, R. L. (1995) Task-Technology Fit and Individual Performance, *MIS Quarterly*, Vol. 19, No. 2, pp. 213-236.

Government Gazette (2014) Republic of South Africa, Vol. 588, No. 37716, Pretoria: Government Printing Works (GPW).

Gravetter, F., and Wallnau, L. (2014) Essentials of statistics for the Behavioral Sciences (8th edn.) Belmont: Wadsworth.

Greener, S. (2008) Business Research Methods. London: Ventus.

Gregor, S. (2006) The nature of theory in information systems, *MIS Quarterly*, Vol. 30, No. 3, pp. 611-642.

Griffin, A. S., and Brandyberry, A. A. (2008) System development methodology usage in industry: a review and analysis, *Journal of information system applied research (JISAR)*, Vol. 3, No. 19, pp. 1-18.

Grossman, M., Aronson, J. E., and McCarthy, R. V. (2005) Does UML make the grade? Insights from the software development community, *Information and Software Technology*, Vol. 47, No. 6, pp. 383-397.

Gruver, G., and Mouser, T. (2015) Leading the transformation: Applying agile and devops principles at scale, Portland: IT Revolution.

Guba, E. G., and Lincoln, Y. S. (1994) Competing paradigms in qualitative research, In N. K. Denzin and Y. S. Lincoln (Eds.), Handbook of qualitative research, Thousand Oaks: Sage.

Gupta, S. K., Gunasekaran, A., Antony, J., Gupta, S., Bag, S., and Roubaud, D. (2019) Systematic literature review of project failures: Current trends and scope for future research, *Computers and Industrial Engineering*, Vol. 127, No. 1, pp. 274-285.

Hair, J.F., Black, W.C., Babin, BJ., and Anderson, R.E. (2018) Multivariate Data Analysis, (8th edn.), London: Cengage.

Harb, Y., Noteboom, C., and Sarnikar, S. (2015) Evaluating Project Characteristics for Selecting the Best-fit Agile Software Development Methodology: A Teaching Case, *Journal of the Midwest Association for Information Systems (JMWAIS),* Vol. 1 No. 1. pp 33-52.

Hardgrave, B. C., Davis, F. D., and Riemenschneider, C. K. (2003) Investigating Determinants of Software Developers' Intentions to Follow Methodologies, *Journal of Management Information Systems*, Vol. 20, No. 1, pp. 123-151.

Hardy, C.J., Thompson, J.B., and Edwards H.M. (1995) The use, limitations and customization of structured systems development methods in the United Kingdom, *Information and Software Technology,* Vol. 37, No. 9, pp. 467-477.

Henderson, D. (2006) The Impact of Systems Development Expertise on Information SDM Use, *AMCIS 2006 Proceedings, pp. 46.*

Henderson-Sellers, B., and Ralyté, J. (2010) Situational Method Engineering: State-of-the-Art Review, *Journal of Universal Computer Science*, Vol. 16, No. 3, pp. 424-478.

Henderson-Sellers, B., Ralyté, J., Ågerfalk, P. J., and Rossi, M. (2014) Situational method engineering, Heidelberg: Springer.

Highsmith, J. A. (2000) Adaptive Software Development: A Collaborative Approach to Managing Complex Systems, New York: Addison Wesley.

Highsmith, J. (2010) Agile Project Management: Creating Innovative Products, Boston: Addison-Wesley.

Highsmith, J., and Cockburn, A. (2001) Agile software development, the business of innovation, *Computer*, Vol. 34, No. 9, pp. 120-127.

Hirschheim, R., and Klein, H. K. (1994) Realizing Emancipatory Principles in Information Systems Development: The case for ETHICS, *MIS Quarterly,* Vol. 18, No. 4, pp. 83-109.

Hohl, P., Klünder, J., van Bennekum, A., Lockard, R., Gifford, J., Münch, J., Stupperich, M., and Schneider, K. (2018) Back to the future: origins and directions of the "Agile Manifesto"-views of the originators, *Journal of Software Engineering Research and Development*, Vol. 6, No. 1, pp. 15.

Hoppenbrouwers, S., Zoet, M., Versendaal, J., and van de Weerd, I. (2011) Agile service development: a rule-based method engineering approach. *In Ralyte´ J, Mirbel I, Denecke`re R (eds) Engineering methods in the service-oriented context. Proceedings of the 4th IFIP WG8.1 working conference on method engineering,* pp 184-189.

Howard, G. S., Bodnovich, T., Janicki, T., Liegle, J., Klein, S., Albert, P., and Cannon, D. (1999) The efficacy of matching information systems development methodologies with application characteristics-an empirical study, *Journal of Systems and Software*, Vol. 45, No. 3, pp. 177-195.

Hudson, L., and Ozanne, J. (1988) Alternative Ways of Seeking Knowledge in Consumer Research. *Journal of Consumer Research*, Vol. 14, No. 4, pp. 508-521.

Huisman, M. (2004) Factors that affect the use and acceptance of systems development methodologies by system developers, *ACIS 2004 Proceedings,* [online] Retrieved from: http://aisel.aisnet.org/acis2004/57, Date accessed: 17 June 2019.

Huisman, M. (2013) The influence of organisational culture on the use of systems development methodologies, *European, Mediterranean and Middle Eastern conference on information systems*, pp. 1-14.

Huisman, M., and Iivari, J. (2002) The individual deployment of systems development methodologies, *In International Conference on Advanced Information Systems Engineering*, pp. 134-150.

Huisman, M., and Iivari, J. (2003) SDM Use in South Africa, *AMCIS 2003 Proceedings 129*, pp 1040-1052.

Huisman, M., and Iivari, J. (2006) Deployment of systems development methodologies: Perceptual congruence between IS managers and systems developers, *Information and Management*, Vol. 43, No. 1, pp. 29-49.

Iivari, J. (1989) A methodology for IS development as organizational change: A pragmatic contingency approach, *In Klein and K. Kumar (eds), Systems Development for Human Progress,* pp. 197-217.

Iivari, J., and Huisman, M. (2007) The Relationship Between Organisational Culture and the Deployment of Systems Development Methodologies, *MIS Quarterly*, Vol. 31, No. 1, pp. 35-58.

Iivari, J., and Iivari, N. (2011) The relationship between organisation culture and the deployment of agile methods, *Information and Software Technology*, Vol. 53, Vol. 5, pp. 509-520.

Iivari, J., and Maansaari, J. (1998) The usage of systems development methods: are we stuck to old practices?, *Information and Software Technology*, Vol. 40, No. 9, pp. 501-510.

Iivari, J., Hirschheim, R. and Klein, H.K. (2004) Towards a distinctive body of knowledge for information systems experts: coding ISD process knowledge in two IS journals, *Information Systems Journal,* Vol. 14, No. 4, pp. 313-342.

Iivari, J., Hirschheim, R., and Klein, H. K. (1999) Beyond methodologies: keeping up with information systems development approaches through dynamic classification, *In Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences, HICSS-32, IEEE, pp. 10-pp.*

Iivari, J., Hirschheim, R., and Klein, H.K. (2001) A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches, *Journal of Management Information Systems*, Vol. 17, No. 3, pp. 179-218.

Ikonen, M., Kettunen, P., Oza, N., and Abrahamsson, P. (2010). Exploring the sources of waste in kanban software development projects, *In 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications*, *IEEE*, pp. 376-381.

Imani, T., Nakano, M., and Anantatmula, V. (2017) Does a Hybrid Approach of Agile and Plan-Driven Methods Work Better for IT System Development Projects?, *International Journal of Engineering Research and Application*, Vol. 7, No. 3, pp. 39-46.

Ionel, N. (2009) Agile software development methodologies: An overview of the current state of research, *Annals of Faculty of Economics*, Vol. 4, No. 1, pp. 381-385.

Isaias, P., and Issa, I. (2015) High Level Models and Methodologies for Information Systems, New York: Springer.

ISO 9001 (2000) Quality Management Systems- Requirements, International Organisation for Standardization: Geneva.

ISO/IEC 12207 (2008) Systems and Software Engineering-Software Life Cycle Processes, in ISO/IEC. 2008: Geneva.

Jackson, M.A. (1983) System Development, Englewood Cliffs: Prentice Hall.

Jacobson, I., Booch, G., and Rumbaugh, J. (1999) The Unified Software Development Process, Reading: Addison Wesley Longman.

Janes, A. A., and Succi, G. (2012) The dark side of agile software development, In *Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software*, pp. 215-228.

Jayaratna, N. (1994) Understanding and evaluating methodologies: A systemic framework, London: McGraw Hill.

Jayaswal, B. K., and Patton, P. C. (2006) Design for trustworthy software: Tools, techniques, and methodology of developing robust software, Upper Saddle River: Prentice Hall.

JCSE (2015) 10-Year Review, Johannesburg Centre for Software Engineering, Johannesburg: JCSE.

JCSE (2016) The transformation issue - JCSE 2015-2016 Annual Report, [online] Retrieved from: https://www.jcse.org.za/wp-content/uploads/2018/05/JCSE_AR_2016.pdf, Date accessed: 28 October 2018.

Jeners, S., Clarke, P., O'Connor, R. V., Buglione, L., and Lepmets, M. (2013) Harmonizing Software Development Processes with Software Development Settings-A Systematic Approach, *20th European Conference on Systems, Software and Services Process Improvement (EuroSPI 2013)*, pp. 167-178.

Jenkins, A. M., Naumann, J. D., and Wetherbe, J. C. (1984) Empirical investigation of systems development practices and results, *Information and Management,* Vol. 7, No. 2, pp. 73-82.

Jeyaraj, A., and Sabherwal, R. (2014) The bass model of diffusion: recommendations for use in information systems research and practice, *JITTA: Journal of Information Technology Theory and Application*, Vol. 15, No. 1, pp. 5-30.

Jiang, L., and Eberlein, A. (2008) Towards a framework for understanding the relationships between classical software engineering and agile methodologies, *In Proceedings of the 2008 international workshop on Scrutinizing agile practices or shoot-out at the agile corral*, pp. 9-14.

Johnson, J., and Mulder, H (2016) CHAOS Chronicles, focusing on failures and possible improvements in IT projects, *Proceedings of The 20th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI 2016),* pp. 209-213.

Jørgensen, M. (2013) Myths and over-simplifications in software engineering. Lecture Notes *Software Engineering*, Vol. 1, No. 1, pp. 7-11.

Judy, K.H. (2012) Agile Values, Innovation and the Shortage of Women Software Developers, *In 2012 45th Hawaii International Conference on System Sciences. IEEE,* pp. 5279-5288.

Jun, L., Qiuzhen, W., and Qingguo, M. (2011) The effects of project uncertainty and risk management on IS development project performance: A vendor perspective, *International Journal of Project Management*, Vol. 29, No. 7, pp. 923-933.

Kalus, G., and Kuhrmann, M. (2013) Criteria for software process tailoring: a systematic review, *In Proceedings of the 2013 International Conference on Software and System Process*, pp. 171-180.

Kaplan, A. (1964) The Conduct of Inquiry, San Francisco: Chandler.

Karahanna, E., Agarwal., R., and Angst, C. M. (2006) Reconceptualizing Compatibility Beliefs in Technology Acceptance Research, *MIS Quarterly*, Vol. 30, No. 4, pp. 781-804.

Karahanna, E., and Straub, D. W. (1999) The psychological origins of perceived usefulness and ease-of-use, *Information and Management*, Vol. 35, No. 4, pp. 237-250.

Karlsen, J., Andersen, J., Birkely, L., and Odegard, E. (2005) What characterizes successful it projects, *International Journal of Information Technology and Decision Making*, Vol. 4, No. 4, pp. 525-540.

Keith, T.Z (2015) Multiple Regression and Beyond: An Introduction to Multiple Regression and Structural Equation Modeling (2nd edn.), New York: Routledge

Kendall, J. E., and Kendall, K.E. (1993) Metaphors and Methodologies: Living Beyond the Systems Machine, *MIS Quarterly*, Vol. 17, No. 2, pp. 149-171.

Kerzner, H. (2017) Project management: a systems approach to planning, scheduling, and controlling, New York: John Wiley and Sons.

Kettunen, P., Laanti, M. (2005), How to steer an embedded software project: tactics for selecting the software process model, *Information and Software Technology,* Vol 47, No. 9, pp. 586-608.

Khalifa, M., and Verner, J.M. (2000) Drivers for Software Development Method Usage, *IEEE Transactions on Engineering Management*, Vol. 47, No. 3, pp. 360-369.

Khoza, L., and Marnewick, C. (2020) Waterfall and Agile information system project success rates-A South African perspective, *South African Computer Journal*, Vol. 32, No. 1, pp. 43-73.

Kloppenborg, T. J., Tesch, D., and Manolis, C. (2014) Project success and executive sponsor behaviors: Empirical life cycle stage investigations, *Project Management Journal*, Vol. 45, No. 1, pp. 9-20.

Komus, A. (2014) Status Quo Agile - Second Study on Success and Forms of Usage of Agile Methods, [online] Retrieved from: http://www.komus.de/fileadmin/downloads/2014-10-DoSE-Status-quo-agile.pdf, Date accessed: 15 June 2017.

Krafcik, J. F. (1988) Triumph of the lean production system, *MIT Sloan Management Review*, Vol. 30, No. 1, pp. 41.

Kruchten, P. (2000) The Rational Unified Process: An Introduction, Reading: Addison-Wesley.

Kuhn, S., and Muller, M. J. (1993) Participatory design, *Communications of the ACM*, Vol. 36, No. 6, pp. 24-29.

Kuhn, T.S. (1970) The Structure of Scientific Revolutions, (2nd edn.), Chicago: University of Chicago Press.

Kuhn, T.S. (1977) The essential tension: Selected studies in scientific tradition and change, Chicago: University of Chicago Press.

Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., Trektere, K., McCaffery, F., Linssen, O., Hanser, E., and Prause, C. R. (2017). Hybrid software and system development in practice: waterfall, scrum, and beyond. *In Proceedings of the 2017 International Conference on Software and System Process*, pp. 30-39.

Kuhrmann, M., Lange, C., and Schnackenburg, A. (2011) A survey on the application of the V-modell XT in German government agencies, *In European Conference on Software Process Improvement*, pp. 49-60.

Lacey, M. (2012) The scrum field guide: Practical advice for your first year, Indianapolis: Addison-Wesley.

Lane, D. C.M, and Oliva, R. (1998) The greater whole: Towards a synthesis of system dynamics and soft methodology development, *European Journal of Operations Research*, Vol. 107, No. 1, pp. 214-235.

Larsen, T.J., Sørebø, A.M., and Sørebø, Ø. (2009) The role of task-technology fit as users' motivation to continue information system use, *Computers in Human Behaviour,* Vol. 25, pp. 778-784.

Leau, Y. B., Loo, W. K., Tham, W. Y., and Tan, S. F. (2012) Software development life cycle AGILE vs traditional approaches, *In International Conference on Information and Network Technology*, Vol. 37, No. 1, pp. 162-167.

Lech, P. (2013) Time, Budget, and Functionality? IT Project Success Criteria Revised, *Information Systems Management*, Vol. 30, No. 3, pp. 263-275.

Lee, G. and W. Xia. (2010) Toward agile: an integrated analysis of quantitative and qualitative field data, *MIS Quarterly*, Vol. 34, No. 1, pp. 87-114.

Leffingwell, D. (2011) Agile software requirements: Lean requirements practices for teams, programs, and the enterprise: Boston: Addison-Wesley.

Levin, D. M. (1988) The opening of vision: Nihilism and the postmodern situation, London: Routledge.

Lim, W. M. (2018) Dialectic Antidotes to Critics of the Technology Acceptance Model: Conceptual, Methodological, and Replication Treatments for Behavioural Modelling in Technology-Mediated Environments, *Australasian Journal of Information Systems*, Vol. 22, [online] Retrieved from: http://dx.doi.org/10.3127/ajis.v22i0.1651, Date accessed: 9 September 2019.

Limayem, M., Hirt, S. G., and Cheung, C. M. (2007) How Habit Limits the Predictive Power of Intention: The Case of Information Systems Continuance, *MIS Quarterly*, Vol. 31, No. 4, pp. 705-737.

Lin, W. T., and Shao, B. B. (2000) The relationship between user participation and system success: a simultaneous contingency approach. *Information and Management,* Vol. 37, No. 6, pp. 283-295.

Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., Tesoriero, R., Williams, L., and Zelkowitz, M. (2002) Empirical findings in agile methods, *In Extreme Programming and Agile Methods-XP/Agile Universe 2002,* pp. 197-207.

Loftus, C., and Ratcliffe, M. (2005) Extreme programming promotes extreme learning? *ACM SIGCSE Bulletin*, Vol. 37, No. 3, pp. 311-322.

Lyytinen, K. (1987) A taxonomic perspective of information systems development: theoretical constructs and recommendations, Critical issues in information systems research, Redwood City: John Wiley.

Mahanti, R., Neogi, M.S., and Bhattacherjee, V. (2012) Factors Affecting the Choice of Software Life Cycle Models in the Software Industry-An Empirical Study, *Journal of Computer Science*, Vol 8, No. 8, pp. 1253-1262.

Maree, K. (2007) First Steps in Research, Pretoria: Van Schaik.

Marks, G., O'Connor, R. V., and Clarke, P. M. (2017) The impact of situational context on the software development process–a case study of a highly innovative start-up organization, *In International Conference on Software Process Improvement and Capability Determination*, pp. 455-466.

Markus, K.A., and Borsboom, D. (2013) Frontiers of Test Validity Theory: Measurement, Causation, and Meaning, New York: Routledge.

Marshall, C., and Rossman, G. B. (2014) Designing qualitative research, London: Sage.

Martin J. (1991) Rapid Application Development, New York: Macmillan Publishing.

Martin, J. (1990) Information Engineering: Introduction, Englewood Cliffs: Prentice Hall.

Maruping, L. M., Venkatesh, V., and Agarwal, R. (2009) A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research*, Vol. 20, No. 3, pp. 377-399.

Massey, V., and Satao, K. (2012) Comparing various SDLC models and the new proposed model on the basis of available methodology, *International Journal of Advanced Research in Computer Science and Software Engineering,* Vol. 2, No. 4, pp. 170-177.

McKenna, S., Richardson, J., and Manroop, L. (2011) Alternative paradigms and the study and practice of performance management and evaluation, *Human Resource Management Review*, Vol. 21, No. 2, pp. 148-157.

McKerchar, M. (2008) Philosophical paradigms, inquiry strategies and knowledge claims: Applying the principles of research design and conduct to taxation. *EJournal of Tax Research*, Vol. 6, No. 1, pp. 5-22.

McLean, E. R., and Sedera, D. (2010) The Measurement of Information System Use: Preliminary Considerations, *In AMCIS*, p. 255.

Mcleod, L., and Macdonell, S. G. (2011) Factors that Affect Software Systems Development Project Outcomes: A Survey of Research, *ACM Computing Surveys*, Vol. 43, No. 4, pp. 1-56.

Mendoza, A., Miller, T., Pedell, S., and Sterling, L. (2013) The role of users' emotions and associated quality goals on appropriation of systems: two case studies, *In 24th Australasian Conference on Information Systems (ACIS),* pp. 1-11.

Meso, P., and Jain, R. (2006) Agile software development: Adaptive systems principles and best practices, *Information Systems Management*, Vol. 23, No. 3, pp. 19-30.

MICT SETA. (2017) MICT SETA Sector Skills Plan for 2018-2023, Midrand: Government Print Works (GPW).

Middleton, P., and McCollum, B. (2001) Management of process improvement by prescription, *Journal of Systems and Software*, Vol. 57, No. 1, pp. 9-19.

Mingers, J., and Brocklesby, J. (1997) Multimethodology: Towards a framework for mixing methodologies, *Omega*, Vol. 25, No. 5, pp. 489-509.

Mirza, M. S., and Datta, S. (2019) Strengths and Weakness of Traditional and Agile Processes-A Systematic Review, *Journal of Software*, Vol. 14, No. 5, pp. 209-219.

Misra, S.C., Kumar, V., and Kumar, U. (2009) Identifying some important success factors in adopting agile software development practices, *Journal of Systems and Software*, Vol. 82, No. 11, pp. 1869-1890.

Mitchell, S. M., and Seaman, C. B. (2009) A comparison of software cost, duration, and quality for waterfall vs. iterative and incremental development: A systematic review, *In 2009 3rd International Symposium on Empirical Software Engineering and Measurement, IEEE*, pp. 511-515.

Mnkandla, E., and Dwolatzky, B. (2007) Agile methodologies selection toolbox, *In International Conference on Software Engineering Advances, ICSEA 2007*, pp. 72-72.

Mohan, K., and Ahlemann, F. (2011) A theory of user acceptance of IS project management methodologies: Understanding the influence of psychological determinism and experience, *In 10th International Conference on Wirtschaftsinformatik*, pp. 912-922.

Moody, D. L., Jacob, M. E., and Amrit, C. (2010) In Search of Paradigms: Identifying the Theoretical Foundations of the IS Field, *18th European Conference on Information Systems, ECIS 2010*, pp. 1-13.

Morgan, D. (2007) Paradigms lost and pragmatism regained: Methodological implications of combining qualitative and quantitative methods, *Journal of Mixed Methods Research,* Vol. 1, No. 1, pp. 48-76.

Müller, R., and Jugdev, K. (2012) Critical success factors in projects: Pinto, Slevin, and Prescott-The elucidation of project success, *International Journal of Managing Projects in Business,* Vol. 5, No. 4, pp. 757-775.

Mumford, E. (1983) Designing Participatively: A Participative Approach to Computer Systems Design, Manchester: Manchester Business School Press.

Munassar, N., and Govardhan, A. (2010) Comparison between five models of software engineering, *International Journal of Computer Science Issues*, Vol. 7, No. 5, pp. 94-101.

Nagel, R. and Dove, R. (1991) 21st Century Manufacturing Enterprise Strategy, Iacocca Institute, Bethlehem: Lehigh University.

Naumann, J. D., and Palvia, S. (1982) A Selection Model for Systems Development Tools, *MIS Quarterly*, Vol. 6, No. 1, pp. 39-48.

Nerur, S., Mahapatra, R., and Mangalaraj, G. (2005) Challenges of migrating to agile methodologies, *Communications of the ACM*, Vol 48, No. 5, pp. 72-78.

Neuman, W.L. (2011) Social Research Methods: Qualitative and Quantitative Approaches, (7th edn.), Boston: Pearson.

Noruwana, N., and Tanner, M. (2012) Understanding the structured processes followed by organisations prior to engaging in agile processes: A South African Perspective, *South African Computer Journal*, Vol. 48, No.1, pp. 41-58.

Oates, B. (2006) Researching Information Systems and Computing, London: Sage.

Office of Government Commerce (OGC). (2009) Managing Successful Projects with PRINCE2, London: TSO.

OMG, O. (2010) Unified Modeling Language (OMG UML), superstructure, version 2.3, Technical Report Document Number formal/2010-05-05, OMG, [online] Retrieved from: http://www.omg.org/spec/UML/2.3/Superstructure/PDF/ (2010), Date accessed: 25 August 2018.

OMG, O. (2013) Unified Profile for DoDAF and MODAF (UPDM), Version 2.1, Technical Report Document Number formal /2013-08-04, [online] Retrieved from: http://www.omg.org/spec/UPDM/2.1/PDF, Date accessed: 20 July 2018.

Pagani, M. (2006) Determinants of adoption of high speed data services in the business market, Evidence for a combined technology acceptance model with task technology fit model, *Information and Management*, Vol. 43, No. 7, pp. 847-860.

Pallant, J. (2016) SPSS Survival manual: A step by step guide to data analysis using IBM SPSS 6th (edn.), Maidenhead: McGraw-Hill.

Paredes, D. (2015) 10 things CIOs need to know about agile development. CIO, [online] Retrieved from: http://www.cio.co.nz/article/578741/10-things-cios-need-know-about-agile-development/, Date accessed: 28 September 2017.

Petersen, K. (2011) Is lean agile and agile lean?: a comparison between two software development paradigms, *In Modern software engineering concepts and practices: Advanced approaches, IGI Global,* pp. 19-46.

Pietri, W. (2011) Agile's Second Chasm, [online] Retrieved from: http://agilefocus.com/2011/02/21/agiles-second-chasm-and-how-we-fell-in/ Date accessed: 10 January 2019.

PMI. (2013) A Guide to the Project Management Body of Knowledge (PMBOK® Guide) (5th edn.), Newtown Square: Project Management Institute.

Poo, D.C.C. and Chung, M.K. (1998) Software engineering practices in Singapore, *Journal of Systems and Software*, Vol. 41, No. 1, pp. 3-15.

Poppendieck, M., and Cusumano, M. A. (2012) Lean software development: A tutorial, *IEEE software*, Vol. 29, No. 5, pp. 26-32.

Poppendieck, M.; Poppendieck, T. (2003) Lean Software Development: An Agile Toolkit, Boston: Addison-Wesley.

Qumer, A., and Henderson-Sellers, B. (2007) Measuring agility and adoptability of agile methods: A 4-dimensional analytical tool, *In Proceedings of IADIS International Conference Applied Computing*, pp. 503-507.

Rahim, M., Seyal, A.H., Raham, N. (1998) Use of software systems development methods: An empirical study in Brunei Darussalam, *Information and Software Technology,* Vol. 39, No. (14-14), pp. 949-963.

Rahmanian, M. (2014) A Comparative Study on Hybrid It Project Managment, *International Journal of Computer and Information Technology*, Vol. 3, No. 5, pp.1096-1099.

Rahmany, N. (2012) The differences between life cycle models-Advantages and disadvantages, [online] Retrieved from: http://narbit.wordpress.com/2012/06/10/the-differences-between-life-cyclemodels-advantages-and-disadvantages/, Date accessed: 4 August 2018.

Rajagopalan, S. and Mathew, S. K. (2016) Choice of Agile Methodologies in Software Development: A Vendor Perspective, *Journal of International Technology and Information Management*, Vol. 25, No. 1, pp. 39-54.

Ramsin, R., and Paige, R. F. (2008) Process-centered review of object oriented software development methodologies, *ACM Computing Surveys (CSUR)*, Vol. 40, No. 1, pp. 1-89.

Rashmi, J., and Anithashree, C. (2009) Rapid System Development (RSD) Methodologies: Proposing a Selection Framework, *Engineering Management Journal*, Vol. 21, No. 4, pp. 30-35.

Reed, R. (1996) Methodology for real time systems, *Computer Networks and ISDN Systems*, Vol. 28, No. 12, pp. 1685-1701.

Riemenschneider, C. K., Hardgrave, B. C., and Davis, F. D. (2002) Explaining software developer acceptance of methodologies: a comparison of five theoretical models, *IEEE transactions on Software Engineering*, Vol. 28, No. 12, pp. 1135-1145.

Rogers, E. M. (2003) Diffusion of innovations, New York: Simon and Schuster.

Rogers, E.M. (1995) Diffusion of Innovations, New York: Free Press.

Royce, W. W. (1970) Managing the development of large software systems, *Proceedings of IEEE WESCON*, pp. 328-388.

Ruparelia, N.B. (2010) Software development lifecycle models, *ACM SIGSOFT Software Engineering Notes,* Vol. 35, No. 3, pp. 8, [online] Retrieved from: http://portal.acm.org/citation.cfm?doid=1764810.1764814, Date accessed: 20 September 2017.

Russo, N. L., Fitzgerald, G., and Shams, S. (2013) Exploring Adoption and Use of Agile Methods: A Comparative Case Study, *Proceedings of the Nineteenth Americas Conference on Information Systems*, pp. 1-8.

Russo, N. L., Hightower, R., and Pearson J.M. (1996) The failure of methodologies to meet the needs of current development environments, *In Jayaratna, N., and Fitzgerald, B., (eds.) Lessons learned from the Use of Methodologies: Fourth Conference on Information Systems Methodologies,* pp. 387-393.

Saeki, M. (1998) A meta-model for method integration, *Information and Software Technology,* Vol. 39, No. (14-15), pp. 925-932.

Saunders, M.N.K., Lewis, P., and Thornhill, A. (2015) Research methods for business students, (7th edn.), Harlow: Pearson.

Schach, S. R. (2010) Object-oriented and classical software engineering, New York: McGraw Hill.

Schein, E. H. (2009). The corporate culture survival guide (Vol. 158), San Francisco: Jossey-Bass.

Schlauderer, S., Overhage, S., and Fehrenbach, B. (2015) Widely Used but also Highly Valued? Acceptance Factors and Their Perceptions in Water-Scrum-Fall Projects, *Thirty Sixth International Conference on Information Systems*, [online] Retrieved from: https://pdfs.semanticscholar.org/25e9/47f09c12b5b08e29fab15087d033d7e64f97.pdf, Date accessed: 25 August 2018.

Schofield, A. and Dwolatzky, B.(2019) 2019 JCSE-IITPSA ICT Skills Survey [online] Retrieved from: https://www.iitpsa.org.za/wp-content/uploads/2019/09/2019-JCSE-IITPSA-ICT-Skills-Survey-v1.pdf, Date retrieved: 28 November 2019.

Schön, D. (1995) The reflective practitioner: how professionals think in action, Hants: Arena.

Schwaber, K. (2010) On Shoulders of Giants, Interview, Agile Collab, [online] Retrieved from: http://mattcallanan.blogspot.de/2010/02/ken-schwaber-on-scrum.html, Date accessed: 10 August 2018.

Schwaber, K., and Beedle, M. (2002) Agile software development with Scrum (Vol. 1), Upper Saddle River: Prentice Hall.

Schwaber, K., and Sutherland, J. (2012) Software in 30 days: how agile managers beat the odds, delight their customers, and leave competitors in the dust, Hoboken: John Wiley and Sons.

Schwaber, K., and Sutherland, J. (2017) The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game, [online] Retrieved from: https://www. scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US. Pdf, Date accessed: 5 November 2019.

Schwandt, T. A. (2007) The Sage Dictionary of Qualitative Inquiry (3rd edn), Thousand Oaks: Sage.

Sekaran, U., and Bougie, R. (2010) Research methods for business, A skill-building approach (5th edn.) Haddington: John Wiley and Sons.

Selic, B. (2009) Agile documentation, anyone?, *IEEE Software*, Vol. 26, No. 6, pp. 11-12.

Serrador, P., and Pinto, J. K. (2015) Does agile work? A quantitative analysis of agile project success, *International Journal of Project Management,* Vol. 33, No. 5, pp. 1041-1051.

Sheffield, J., and Lemétayer, J. (2013) Factors associated with the software development agility of successful projects, *International Journal of Project Management*, Vol. 31, No. 3, pp. 459-472.

Shenhar, A., and Dvir, D. (2007) Reinventing Project Management: The Diamond Approach to Successful Growth and Innovation, Boston: Harvard Business School Press.

Siakas, K.V., and Siakas, E. (2007) The agile professional culture: a source of agile quality, *Software Process, Improvement and Practice*, Vol. 12, No. 6, pp. 597-610.

Siau, K., and Rossi, M. (2011) Evaluation techniques for systems analysis and design modelling methods-a review and comparative analysis, *Information Systems Journal*, Vol. 21, No. 3, pp. 249-268.

Snowden, D. J., and Boone, M. E. (2007) A leader's framework for decision making. *Harvard business review*, Vol. 85, No. 11, pp. 68-76.

Somerville, I. (2016) Software Engineering, 10th (edn.), London: Pearson.

Špundak, M. (2014) Mixed agile/traditional project management methodology-reality or illusion?, *Procedia-Social and Behavioral Sciences*, Vol.119, No. 1, pp. 939-948.

Standish Group. (1995) The Chaos Report, Boston, [online] Retrieved from: http://www.projectsmart.co.uk/docs/chaos-report.pdf, Date accessed: 30 September 2017.

Standish Group. (2016) CHAOS Summary Report 2016 [online] Retrieved from: http://www.standish/reports/reports.php, Data accessed: 25 August 2017.

Stober, T., and Hansmann, U. (2010) Agile Software Development: Best Practices for Large Software Development Projects, Berlin: Springer.

Stoica, M., Mircea, M. and Ghilic-Micu, B. (2013) Software Development: Agile vs. Traditional, *Informatica Economica, Vol.* 17, No. 4, pp. 64-76.

Strode, D. E., and Huff, S. L. (2015) A coordination Perspective on Agile Software Development, *In S. Gao, and L. Rusu, Modern Techniques for Successful IT Project Management*, pp. 64-96.

Sun, H. (2013) A Longitudinal Study of Herd Behavior in the Adoption and Continued Use of Technology, *MIS Quarterly,* Vol. 37, No. 4, pp. 1013-1041.

Tabachnick, B.G., and Fidell, L.S. (2013) Using multivariate statistics 6th (edn.) Boston: Pearson.

Tekin, A. K., and Kotaman, H. (2013) The epistemological perspectives on action research, *Journal of Educational and Social Research*, Vol. 3, No. 1, pp. 81-91.

Tornatzky, L. G., and Klein, K. J. (1982) Innovation characteristics and innovation adoption-implementation: A meta-analysis of findings, *IEEE Transactions on engineering management,* Vol 29, No. 1, pp. 28-45.

Tourangeau, R. (1984) Cognitive sciences and survey methods. *In Jabine T, Straf M, Tanur J, Tourangeau R (eds), Cognitive Aspects of Survey Methodology: Building a Bridge Between the Disciplines,* pp. 73-100.

Tung, F.C., Chang, S.C., and Chou, C. M. (2008) An extension of trust and TAM model with IDT in the adoption of the electronic logistics information system in HIS in the medical industry, *International Journal of Medical Informatics*, Vol. 77, No. 5, 324-335.

Turk, D., France, R., and Rumpe, B. (2014) Limitations of agile software processes, [online] Retrieved: https://arxiv.org/pdf/1409.6600, Date accessed: 25 August 2018.

Turner, M., Kitchenham, B., Brereton, P., Charters, S., and Budgen, D. (2010) Does the Technology Acceptance Model Predict Actual Use? A Systematic Literature Review, *Information and Software Technology*, Vol. 52, No. 5, pp. 463-479.

Turner, R., and Zolin, R. (2012) Forecasting Success on Large Projects: Developing Reliable Scales to Predict Multiple Perspectives by Multiple Stakeholders Over Multiple Time Frames, *Project Management Journal,* Vol. 43, No. 5, pp. 87-99.

Vakoc, M., and Buchalcevova, A. (2017) Becoming Predictably Adaptable in Software Development, Journal of systems integration, Vol 8, No. 4, pp. 21-28.

Valacich, S, J., George, F, J., and Hoffer, A, J. (2012) Essentials of Systems Analysis and Design, (5th edn), New York: Prentice Hall.

Van de Ven, A.H. and Drazin, R. (1985) The Concept of Fit in Contingency Theory, *Administrative Science Quarterly*, Vol. 30, pp. No. 4, pp. 514-539.

Van Slooten, K., and Hodes, B. (1996) Characterizing IS development projects, *In Brinkkemper, S., Lyytinen, K., and Welke, R. (eds) Proceedings of IFIP TC8 working conference on method engineering: principles of method construction and tool support*, pp. 29-44.

Vavpotič, D., and Hovelja, T. (2012) Improving the Evaluation of Software Development Methodology Adoption and its Impact on Enterprise Performance, *Computer Science and Information Systems*, Vol. 9, No. 1 pp. 165-188.

Vavpotič, D., and Vasilecas, O. (2011) An Approach for Assessment of Software Development Methodologies Suitability, *Electronics and Electrical Engineering: Systems Engineering, Computer Technology*, Vol 8, No 114, pp. 107-110.

Venkatesh, V., and Bala, H. (2008) Technology acceptance model 3 and a research agenda on interventions, *Decision sciences*, Vol. 39, No. 2, pp. 273-312.

Venkatesh, V., and Davis, F.D. (1996) A model of the antecedents of perceived ease of use: development and test, *Decision Sciences,* Vol. 27, No. 3, pp. 451-481.

Venkatesh, V., Morris, M. G., Davis, G. B., and Davis, F. D. (2003) User Acceptance of Information Technology: Toward a Unified View, *MIS Quarterly*, Vol. 27, No. 3, pp. 425-478.

Venkatesh, V., Thong, J.Y. L., and Xu, X. (2012) Consumer Acceptance and Use of Information Technology: Extending the Unified Theory of Acceptance and Use of Technology, *MIS Quarterly* Vol. 36, No. 1, pp. 157-178.

Venkatraman, N. (1989) The Concept of Fit in Strategy Research: Toward Verbal and Statistical Correspondence, *Academy of Management Review*, Vol. 14, No. 3, pp. 423-444.

Versionone (2018) The State of Agile Survey, [online] Retrieved from: https://explore.versionone.com/state-of-agile/12th-annual-state-of-agile-report-overview, Date accessed: 28 October 2019.

Vijayasarathy, L. R., and Butler, C. W. (2015) Choice of software development methodologies: Do organizational, project, and team characteristics matter?. *IEEE software,* Vol. 33, No. 5, pp. 86-94.

Vijayasarathy, L. R., Turk, D. (2008) Agile Software Development: A survey of early adopters, *Journal of Information Technology Management*, Vol XIX, No. 2, pp. 1-8.

Vijayasarathy, L., and Turk, D. (2012) Drivers of agile software development use: Dialectic interplay between benefits and hindrances, *Information and Software Technology*, Vol. 54, No. 2, pp. 137-148.

Viljoen, J. (2016) The selection and use of system development methodologies, Potchefstroom: North West University. (Dissertation - MSc).

Vlasblom, G., Rijsenbrij, D., and Glastra, M. (1995) Flexibilization of the methodology of system development, *Information and Software Technology*, Vol. 37, No. 11, pp. 595-607.

Walden, D. D., Roedler, G. J., Forsberg, K., Hamelin, R. D., and Shortell, T. M. (2015) Systems engineering handbook: A guide for system life cycle processes and activities, New York: John Wiley and Sons.

Walters, S., Broady, J., and Hartley, R. (1994) A review of Information Systems Development Methodologies, *Library Management*, Vol. 15, No. 6, pp. 5-19.

Wang, X., Conboy, K., and Pikkarainen, M. (2012a). Assimilation of agile practices in use, *Information System Journal*, Vol. 22, No. 6, pp. 435-455.

Wang, Y.S., Wu, S.C., Lin, H.H., Wang, Y.M., and He, T. R. (2012b) Determinants of user adoption of web ''Automatic Teller Machines': an integrated model of 'Transaction Cost Theory' and 'Innovation Diffusion Theory', *The Service Industries Journal*, Vol. 32, No. 9, pp. 1505-1525.

Weber, R. (2004) The rhetoric of positivism versus interpretivism: A personal view. *MIS, Quarterly*, Vol. 28, No. 1, pp. 3-12.

Weigel, F. K.; Hazen, B. T.; Cegielski, C. G.; and Hall, D. J. (2014) Diffusion of Innovations and the Theory of Planned Behavior in Information Systems Research: A Metaanalysis, *Communications of the Association for Information Systems*: Vol. 34, No. 31, pp. 619-636.

Wendler, R. (2016) Dimensions of Organizational Agility in the Software and IT Service Industry: Insights from an Empirical Investigation, *Communications of the Association for Information Systems*: Vol. 39, No. 21, pp. 439-482 , [online] Retrieved from: http://aisel.aisnet.org/cais/vol39/iss1/21, Date accessed: 20 February 2017.

West, D. (2011) Water-Scrum-Fall Is the Reality of Agile for Most Organizations Today, Forrester Research [online] Retrieved from: https://www.verheulconsultants.nl/water-scrum-fall_Forrester.pdf, Date accessed: 16 March 2018.

Williams, L., and Cockburn, A. (2003) Agile software development: it's about feedback and change, *IEEE Computer*, Vol. 36, No. 6, pp. 39-43.

Wistrand, K., and Karlsson, F. (2004) Method Components-Rationale Revealed, *In Advanced Information Systems Engineering 16th International Conference, CAiSE,* pp. 189-201.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012) Experimentation in software engineering, Heidelberg: Springer.

Womack, J. P., Jones, D. T., and Roos, D. (2007) The machine that changed the world: The story of lean production-Toyota's secret weapon in the global car wars that is now revolutionizing world industry, New York: Simon and Schuster.

Wynekoop, J. L., and Russo, N. L. (1993) System Development methodologies: unanswered questions and the research-practice gap, *In ICIS* , pp. 181-190.

Wynekoop, J. L., and Russo, N.L. (1995) Systems development methodologies: unanswered questions, Journal *of Information Technology,* Vol.10, No.2, pp. 65-73.

Wynekoop, J.L., and Russo, N.L. (1997) Studying system development methodologies: an examination of research methods, *Information Systems Journal*, Vol. 7, No. 1, pp. 47-65.

Wysocki, R. K. (2011) Effective project management: traditional, agile, extreme, Indianapolis: John Wiley and Sons.

Yaghini, M., Bourouni, A., and Amiri, R. H. (2009) A framework for selection of information systems development methodologies, *Computer and information science*, Vol. 2, No. 1, pp. 3-11.

Yen, D. C., Wu, C. S., Cheng, F. F., and Huang, Y. W. (2010) Determinants of users' intention to adopt wireless technology: An empirical study by integrating TTF with TAM, *Computers in Human Behavior*, Vol. 26, No. 5, pp. 906-915.

Young, D. K., Beebe, N.L., Dietrich, G., and Liu, C. Z. (2016) An Empirical Examination of an Agile Contingent Project/Method Fit Model, *Communications of the Association for Information Systems*: Vol. 39, No. 1, pp. 267-296.

Young, R. R. (2002) Recommended requirements gathering practices, CROSSTALK, *The Journal of Defense Software Engineering*, Vol. 15, No. 4, pp. 9-12.

Yourdon. (1993) Yourdon Systems Method: Model-Driven Systems Development, Upper Saddle River: Yourdon Press.

Youssef, M. A. (1992) Agile manufacturing: a necessary condition for competing in global markets*, Industrial Engineering*, Vol. 24, No. 12, pp. 18-20.

Yu, T. K., and Yu, T. Y. (2010) Modelling the factors that affect individuals' utilisation of online learning systems: An empirical study combining the task technology fit model with the theory of planned behaviour, *British Journal of Educational Technology*, Vol. 41, No. 6, pp. 1003-1017.

Yusof, M.M., Shukur, Z., and Abdullah, A.L. (2011) CuQuP: A Hybrid Approach for Selecting Suitable Information SDM, *Information Technology Journal*, Vol. 10, pp. 1031-1037.

Zhang, Z., Yuan, K. (2016) Robust coefficients alpha and omega and confidence intervals with outlying observations and missing data: Methods and software, *Educational and Psychological Measurement*, Vol. 76, No. 3, pp. 387-411.

Zhou, F., Tong, Y., Teo, H. H., and Tan,S. S. (2016) Understand the System Fit Challenge at the System Fit Challenge at the Initial Post-Adoption Stage: The Roles of Emotions in Users' Adaptation Behaviors, *Thirty Seventh International Conference on Information Systems*, pp. 1-20.

Zhu, L., Bass, L., and Champlin-Scharff, G. (2016) DevOps and its practices, *IEEE Software*, Vol. 33, No. 3, pp. 32-34.

Zhu, Z. (2002) Evaluating contingency approaches to information systems design, *International Journal of Information Management*, Vol. 22, No. 5, pp. 343-356.

Zwanenburg, S. P. (2015) How to Tie a Construct to Indicators: Guidelines for Valid Measurement, *Thirty Sixth International Conference on Information Systems,* Fort Worth, [online] Retrieved from: https://pdfs.semanticscholar.org/6238/48d0fe256ab0c8ce7f1919de6b62bd3c9e64.pdf, Date accessed: 20 February 2018.

Zwikael, O., Pathak, R. D., Singh, G., and Ahmed, S. (2014) The Moderating Effect of Risk on the Relationship between Planning and Success, *International Journal of Project Management*, Vol. 32, No. 3, pp. 435-441.

APPENDIX A: ETHICS CLEARANCE

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

Faculty of Natural and Agricultural Sciences
Ethics Committee
Private Bag x6001, Potchefstroom,
2520, South Africa
Web: http://www.nwu.ac.za

To: Moyo B

RE: Approval of your application by the FNAS Ethics Committee

Ethics number: NWU-01214-19-S9

Study title:The contingent use of systems development methodologies in South Africa

Study leader: HM Huisman

Student: Moyo B

You are kindly informed that after review by the FNAS Ethics committee, North-West University, your ethics approval application has been successful.

Your study has been approved as a **No Risk** project.

Yours sincerely,

Prof. Oriel Thekisoe

Acting Chairperson

FNAS Ethics Committee

APPENDIX B: INFORMED CONSENT LETTER

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
POTCHEFSTROOM CAMPUS

Private Bag X6001, Potchefstroom
South Africa 2520

School of Computer Science and Information Systems

Dear Prospective participant

RE: INFORMATION LEAFLET AND INFORMED CONSENT

My name is Benson Moyo and I am a PhD student conducting a study entitled "The contingent use of systems development methodologies in South Africa". The Faculty of Natural and Agricultural Sciences Ethics Committee of the North West University has granted ethics approval for the study and the ethics number is NW-01214-19-S9.

You are hereby invited to participate by completing a questionnaire that aims at collecting information related to the contingent use of systems development methodologies in South Africa. Participation is completely voluntary. The questionnaire consists of nine parts. It is estimated that it will take approximately 15-25 minutes to complete the questionnaire. You have the right to withdraw from completing the questionnaire at any time without providing a reason for doing so. All information obtained during the course of this study is strictly confidential. The study data will be coded in such a way that it will not be linked to any name, be it an organization or an individual. No identity will be revealed when research results are reported in scientific journals or presented at local or international conferences.

Definition of Terms

The following terms are defined according to the way they are used in the research study.
System development methodology is defined as a combination of the following:

i) Systems development approach/approaches for example the agile or the structured approach.
ii) Systems development process model/models for example a sequential, iterative or incremental process.
iii) Systems development technique/techniques for example ER diagrams, use case diagrams.

A **systems development project** is defined as an effort that has a clear-cut beginning and an explicit end that is undertaken to create a systems development product or service, or enhance an already existing system development product or service.

Thank you very much for your participation.

If you have any queries regarding this research work, please contact either me or my study leader.

The main findings of the study can be obtained from my promoters or myself on request.

Prof. H M Huisman(Study leader/promoter)	magda.huisman@nwu.ac.za
Prof. L Drevin (promoter)	lynette.drevin@nwu.ac.za
Benson Moyo(researcher)	benson.moyo@univen.ac.za

APPENDIX C: CODE OF CONDUCT FOR RESEARCHERS

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

Research and Innovation

CODE OF CONDUCT FOR RESEARCHERS

This code of conduct is applicable to all NWU researchers.

As a researcher of the North-West University (NWU), I subscribe to the rules of the NWU Institutional Research Ethics Regulatory Committee (IRERC), all applicable policies of the NWU as well as all national and international laws and regulations applicable to my field of study. Furthermore, I commit myself to abide by the ethical principles and responsibilities as set out in the Singapore statement on Research Integrity (22 September 2010), in any and all research endeavours that I undertake as a researcher of the NWU.

The four major principles of research integrity to which I will adhere and that will guide my research are:

- Honesty in all aspects of research;
- Accountability in the conduct of research;
- Professional courtesy and fairness in working with others;
- Good stewardship of research on behalf of others.

Consequently I will also adhere to the following ethical responsibilities:

1. I will take responsibility for the originality and trustworthiness of my research.

2. I will stay abreast of and adhere to all institutional, national, and international laws, regulations, and policies applicable and related to my research.

3. I will at all times employ appropriate research methods, base my conclusions on critical analysis of the evidence and report my findings and interpretations fully and objectively.

4. I will keep clear and accurate records of all research that I have conducted in a manner that will allow verification and replication of my work by others, if applicable.

5. I will, where applicable, share my data and findings openly and promptly, in line with external funding rules. This will be done as soon as possible after I have had an opportunity to establish priority and ownership claims.

6. I will take responsibility for my own contributions to publications, funding applications, reports and other representations of my research. I will also and only include authors who meet valid authorship criteria.

7. I will acknowledge the names and roles of those who made significant contributions to my research in publications, including writers, funders, sponsors, and others, but do not meet authorship criteria.

8. In my peer reviews, I will provide fair, prompt and rigorous evaluations and I will respect confidentiality when I review others' work.

9. I will disclose all conflicts of interest (financial and other) that could compromise the trustworthiness of my work in research proposals, publications, public communications, and in review activities.

10. When I publically address a community in the spirit of academic freedom, I will in all stages base my professional comments on research findings (if applicable) and my expertise. I will distinguish between professional comments and opinions based on personal views.

11. Should any irresponsible research practices and/or research misconduct become known to me or brought under my attention, I will report such irresponsible research activities to the appropriate authorities.

12. I will respond to irresponsible research practices or conduct, by taking prompt actions as set out in the procedures of the university. I will also protect those who report misconduct in good faith, to the best of my abilities.

13. I will endeavour to create and sustain an environment that encourage research integrity through education of students, research teams and peers, as well as abide by policies, and reasonable standards for advancement.

14. I will at all times weigh societal benefits against the risks inherent in my work.

Name: BENSON MOYO   Signature: Date: 20-04-2019

Original details: (11664754) P:\9. Research and Post-graduate Education\9.1 Implementation of the research strategy\9.1.5 Ethics\9.1.5.1.3_Code_Conduct_2017.docm
18 July 2017

File reference: 9.1.5.1.3

APPENDIX D: PUBLICATIONS

Based on the first objective of the study which examined systems development methodologies in use within the South African systems development organisations, a research article titled: “The state of systems development methodologies use within the systems development organisations in South Africa”, was submitted to the *South African Computer Journal* (SACJ) and is currently under review.

Moyo, B., Huisman, H.M. and Drevin, L. Systems development methodologies used in South African systems development industry, *South African Computer Journal*. (Under Review).

APPENDIX E: STUDY QUESTIONNAIRE

The questionnaire document

Dear respondent

Thank you for accepting to participate in the survey. Please carefully go through the nine parts of the questionnaire and assist with the information requested.

In Part 1, Part 2 and Part 3, please indicate your choice by ticking the corresponding box or boxes.

Part 1: Personal Information

1. What is the primary role you hold in systems development projects in your organisation?

☐ Manager

☐ Developer

☐ Analyst

☐ Designer

☐ Other, specify……………………………………………………………………………

2. What other role do you assume in systems development projects in your organisation?

☐ Manager

☐ Developer

☐ Analyst

☐ Designer

☐ Other, specify……………………………………………………………………………

3. How many years of experience do have in systems development projects?

☐ less than 5 years

☐ 5 to less than 10 years

☐ 10 years to less than 15 years

☐ 15 years to less than 20 years

☐ more than 20 years

4. What percentage of the systems development project time is spent on selecting an SDM?

…………………………………………………………………………………………………

Part 2: Organizational Information

1. What is the size of your organization in terms of the number of employees?

☐ 1 to 10 employees

☐ 11 to 50 employees

☐ 51 to 100 employees

☐ 101 to 250 employees

☐ 251 or more employees

2. What is the size of your systems development project team in terms of the number of systems employees?

☐ 1 to 5 employees

☐ 6 to 15 employees

☐ 16 to 30 employees

☐ 31 to 50 employees

☐ 51 or more employees

3. How many systems development projects may run concurrently in your organization?

....

4. How many of the personnel in your organization are actively involved in selecting a SDM for a project?

☐ 1 to 5 employees

☐ 6 to 15 employees

☐ 16 to 30 employees

☐ 31 to 50 employees

☐ 51 or more employees

5. When things go wrong (project fails), what is the distribution of blame? Choose what you think is the proportion of blame for each position in a project by ticking the corresponding cell in each case.

i	Manager	Less than 20%	20 to less than 40%	40 to less than 60%	60 to less than 80%	More than 80%

ii	Developer	Less than 20%	20 to less than 40%	40 to less than 60%	60 to less than 80%	More than 80%

iii	Analyst	Less than 20%	20 to less than 40%	40 to less than 60%	60 to less than 80%	More than 80%

iv	Designer	Less than 20%	20 to less than 40%	40 to less than 60%	60 to less than 80%	More than 80%

v	Other, specify ………………………………	Less than 20%	20 to less than 40%	40 to less than 60%	60 to less than 80%	More than 80%

Part 3: Experience with the selection of systems development methodologies for a systems development project

1. When do you select an SDM or a combination of SDMs, for a systems development project?

☐ Only once prior to the start of the systems development project.

☐ Prior to commencement of the systems development project, and during each phase of the Systems Development Life Cycle.

☐ At any point during project development when situations emerge that require change of the SDM.

☐ We do not select, but create an SDM for each systems development project.

☐ Other, specify…………………………………………………

2. How do you adopt an SDM or a combination of SDMs for a systems development project in your organization?

☐ By following guidelines and organisational policies.

☐ By using SDM adoption frameworks.

☐ By deploying best practices and experience from previously developed systems development projects.

☐ Other, specify...

Part 4: Systems development methodologies selection criteria

1. What is the rating of the following statements considering the criteria used to select an SDM for a project in your organisation? Please indicate your choice by ticking the number corresponding to your level of agreement or disagreement with the statement.

a)	Systems development project **type** influenced the selection of SDM.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
b)	System **criticality** influenced the selection of SDM.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
c)	Systems developer **expertise** influenced the selection of SDM.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
d)	Systems developer **experience** influenced the selection of SDM.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
e)	Systems development **team size** influenced the selection of SDM.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
f)	Systems development **team structure** influenced the selection of SDM.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
g)	Systems development **problem domain** influenced the selection of SDM.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
h)	Stakeholder **culture** influenced the selection of SDM.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
i)	Stakeholder **politics** influenced the selection of SDM	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
j)	System development methodology **familiarity** influenced its selection for a project.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
k)	SDM's ability to **early discover** problems influenced its selection.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6

l)	Emphasized systems **quality factors** influenced the selection of system development methodology.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
m)	System post release **defect** history influenced the selection of SDM.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
n)	The degree of meeting the **user requirements** influenced the selection of system development methodology.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
o)	The development **cost** influenced the selection of system development methodology.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
p)	The system's **market window** influenced the selection of SDM.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
q)	The systems development standards **compliance** influenced the selection of SDM.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6

2. How important is the selecting of an SDM or a combination of SDMs for the success of systems development projects in your organisation? Please indicate your choice by ticking the corresponding number.

Extremely unimportant	Quite unimportant	Unimportant	Important	Quite Important	Extremely Important
1	2	3	4	5	6

3. How do you rate the following statements based on the practices followed by your organisation to justify the selection of an SDM from a repository of available SDMs? Please indicate your choice by ticking the number corresponding to your level of agreement or disagreement with the statement.

a)	Different systems development methodologies have different failure histories.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
b)	Different systems development methodologies have different success histories.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
c)	Different systems development methodologies identify critical project stakeholders differently.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
d)	Different systems development	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree

	methodologies disseminate common project vocabulary to all identified stakeholders differently.	1	2	3	4	5	6
e)	Different systems development methodologies facilitate delivery of projects within same schedule constraints differently.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
f)	Different SDMs facilitate delivery of projects within same budgetary constraints differently.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
g)	Different SDMs detect defects differently.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
h)	Different SDMs match with project problem space with varying degrees.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
i)	Different SDMs break down work differently.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
j)	Different SDMs comply with standards differently.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
k)	Different SDMs have different cost implications in a project.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6

4. If you have other reasons for selecting a systems development methodology not captured above, you may state them in the space provided below.

   ....

Part 5: Selection, adoption and use of systems development methodologies

1. How responsive is your organization in selecting and adopting a new SDM? Please indicate your choice by ticking in one box only.

☐ Market leader

☐ Market follower

☐ Conservative

☐ Static

2. Which option describes the standard practice on SDM selection in your organization? Please indicate your choice by ticking in a box corresponding to your choice (you may select more than one option).

- [ ] We select one SDM from a standard set of SDMs for each project situation.
- [ ] We select a set of SDMs from a set of standard SDMs for each project situation.
- [ ] We select a SDM from in-house developed systems development methodologies for each project situation.
- [ ] We select a SDM from a standard set of SDMs and adapt it to each project situation.
- [ ] We develop a new SDM for each project situation.
- [ ] We do not use a SDM to develop systems.

3. What is the proportion of projects that have been developed using the SDM knowledge in your organization so far? Please indicate your choice by ticking the number corresponding to the percentage range shown.

0 %	1-20%	21-40%	41-60%	61-80%	Over 80%
1	2	3	4	5	6

4. What is the proportion of people who apply SDM knowledge in your organization? Please indicate your choice by ticking the number corresponding to the percentage range shown.

0 %	1-20%	21-40%	41-60%	61-80%	Over 80%
1	2	3	4	5	6

5. To what extent did you deviate from the prescription of the SDM in use in your organisation? Please indicate your choice by ticking the number corresponding to your level of agreement or disagreement with the statement.

a)	No deviation at all.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
b)	Deviation from the SDM prescription was caused by the need to address some missing components.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
c)	Deviation from the SDM prescription was caused by the need to remove some irrelevant components.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
d)	Deviation from the SDM prescription was caused by the need to reconfigure some components.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
e)	We created an alternative SDM by combining different SDMs.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6

Part 6: Success measure of selected SDM/methodologies

1. How do you rate the following statements when evaluating the success of a selected SDM on a systems development project in your organisation? Please indicate your choice by ticking the number corresponding to your level of agreement or disagreement with the statement.

a)	Success is achieved when the selected methodology helps to complete the project within budget.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
b)	Success is achieved when the selected methodology helps to complete the project within small deviations from the budget.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
c)	Success is achieved when the selected methodology helps to complete the project on time.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
d)	Success is achieved when the selected methodology helps to complete the project within small deviations from the schedule.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
e)	Success is achieved when the selected methodology helps to complete the project with all the features and functionality as initially specified.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
f)	Success is achieved when the selected methodology helps address evolving requirements.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
g)	Success is achieved when the selected methodology helps to complete the project with minimum restarts.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
h)	Success is achieved when the selected methodology helps to detect defects early during development.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6

2. How do you rate your success on your current selected SDM for systems development project in your organisation? Please indicate your choice by ticking the number corresponding to the best percentage estimate for your case.

0 %	1-20%	21-40%	41-60%	61-80%	Over 80%
1	2	3	4	5	6

Part 7: Systems product quality measurement after the use of selected SDM/SDMs

1. With reference to practices in your organisation, how do you rate the following statements relating to the quality of a system developed using a SDM? Please indicate your choice by ticking the number corresponding to your level of agreement or disagreement with the statement

a)	Quality was achieved when the system developed using the SDM provided functions which met stated requirements.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
b)	Quality was achieved when the system developed using the SDM provided functions which met implied requirements.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
c)	Quality was achieved when the system developed using the SDM had response time that met user expectations.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
d)	Quality was achieved when the system developed using the SDM provided accurate output for users.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
e)	Quality was achieved when the system developed using the SDM was user friendly.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
f)	Quality was achieved when the system developed using the SDM maintained its level of performance under stated conditions for a stated period of time.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
g)	Quality was achieved when the system developed using the SDM was stable and unlikely to fail.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
h)	Quality was achieved when the system developed using the SDM recovered data and restored optimal functioning during failures.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
i)	Quality was achieved when the system developed gave consistent results when used.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
j)	Quality was achieved when the system developed accommodated modification to remove defects.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
k)	Quality was achieved when the	Totally	Disagree	Partially	Partially	Agree	Totally

	system developed using the SDM was easily adapted to new specifications or changed environment.	Disagree		Disagree	Agree		Agree
		1	2	3	4	5	6
l)	Quality was achieved when the system developed accommodated changes suggested by the user.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
m)	Quality was achieved when the system developed using the SDM had a long life expectancy.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6

Part 8: Strategies for the use of SDMs in projects in your organization.

1. How do you rate the following statements when matching the current system development methodology with the problem domain characteristics of the system development project in your organisation? Please indicate your choice by ticking the number corresponding to your level of agreement or disagreement with the statement.

a)	SDM in use complies with fitness for purpose (doing things right) in each project in my organization.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
b)	SDM in use complies with fitness of purpose (doing the right things) in each project in my organization.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
c)	Specific use of a SDM is grounded on realities of each situation in my organization.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
d)	SDMs complement each other in a project in my organization	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
e)	The selected SDM or SDMs components suit contextual characteristics of each project situation in my organization.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
f)	The SDM in use matches with all systems development project situations.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
g)	The SDM is reconfigured as the systems development project situation dictates.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6
h)	The SDM in use is configured in each phase of the systems development life cycle.	Totally Disagree	Disagree	Partially Disagree	Partially Agree	Agree	Totally Agree
		1	2	3	4	5	6

Part 9: The extent of usage of SDMs in your organization.

1. To what extent is your organization currently using the following SDMs? (1= Not at all, 6 = Higher intensity) . Please indicate your choice by ticking the number corresponding to the extent of SDM use in your organization. You may indicate more than one SDM.

a	Rational Unified Process (RUP)	Not at all					Higher intensity
		1	2	3	4	5	6
b	Microsoft Solutions Framework (MSF)	Not at all					Higher intensity
		1	2	3	4	5	6
c	Structured Systems Analysis and Design Methodology (SSADM)	Not at all					Higher intensity
		1	2	3	4	5	6
d	Home grown/in-house developed	Not at all					Higher intensity
		1	2	3	4	5	6
e	Merise	Not at all					Higher intensity
		1	2	3	4	5	6
f	Yourdon Systems Method (YSM)	Not at all					Higher intensity
		1	2	3	4	5	6
g	Jackson Systems Development (JSD)	Not at all					Higher intensity
		1	2	3	4	5	6
h	Rapid Application Development	Not at all					Higher intensity
		1	2	3	4	5	6
i	XP (Extreme Programming)	Not at all					Higher intensity
		1	2	3	4	5	6
j	Scrum	Not at all					Higher intensity
		1	2	3	4	5	6
k	Crystal family	Not at all					Higher intensity
		1	2	3	4	5	6
l	PRINCE2 (Projects In Controlled Environments)	Not at all					Higher intensity
		1	2	3	4	5	6
m	Water-Scrum-Fall	Not at all					Higher intensity
		1	2	3	4	5	6
n	Kanban	Not at all					Higher intensity
		1	2	3	4	5	6
o	Other specify:	Not at all					Higher intensity
	………………………………………	1	2	3	4	5	6

2. How long has the current SDM been in use in your organization?

Less than 1 year	1
1 to less than 2 years	2
2 years to less than 5 years	3
5 years to less than 10 years	4

10 years and above	5
I don't know	6

If you would like to be informed about the findings of this research you may provide the email address that will note identify you or your organization:

………

End of questionnaire

Thank you very much for your time and cooperation