UNISA university of south africa

A FRAMEWORK FOR THE ADOPTION OF AGILE SOFTWARE DEVELOPMENT METHODOLOGIES IN BOTSWANA

by

LAVANYA BALASUBRAMANIAN

submitted per the requirements for
the degree of

DOCTOR OF PHILOSOPHY

in the subject

INFORMATION SYSTEMS

at the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR: PROFESSOR E. MNKANDLA

January 2022

ABSTRACT

Modern software development methodologies (SDMs) such as Agile, Scrum, Extreme Programming, Lean and others are emerging as the de facto software development methodologies around the globe, however, their uptake by Botswana software development companies (BSDCs) have been slow with many opting to use traditional methods such as the Waterfall method. This research aimed to explore the factors that affect the adoption of Agile SDMs by the Botswana software development industry (BSDI) and why the newer methodologies are not considered for adoption, and thereafter develop a guiding framework for the adoption of an appropriate Agile SDM. An interpretivist/constructivist paradigm guided this research study, and the case study methodology was adopted because it allows research of complex phenomena in a specific context to be conducted. A sample of 12 BSDCs using purposive sampling was used for the study, however, only 9 companies participated. Ethical approval for the research was sought from both the University of South Africa and the Government of Botswana. It was found that the adoption and implementation of newer SDMs by the BSDI were hampered by the assumption that this process would attract costs and affect profit margins of the industry, lack of knowledge about newer methodologies such as Agile, shortage of skilled personnel, insufficient resource support, and inefficient knowledge management systems. A framework, guided by the Technology Acceptance Model II, was developed, and validated. This framework provides a starting point for the Botswana software industry, organisations interested in adopting Agile methodologies. The framework emphasises better practices and provides guidance in the use of Agile as a software development methodology.

KEYWORDS

Software Engineering, Software Development Methodology, Agile Software Development Methodology, Methodology Adoption, Software Projects, Business Process, Technology Acceptance Model I and II, Software Development Life Cycle, Traditional Software Development Methodology, Information and Communication Technology, 4th Industry Revolution

DECLARATION

I, *Lavanya Balasubramanian*, declare that *A framework for the adoption of agile software development methodologies in Botswana* is my own work, that it has not been submitted for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged by complete references.

Student signature: ____________________ Date: 31 January 2022

COPYRIGHT

DEDICATION

I dedicate this study to my husband Mr K Balasubramanian for the continuous support and unconditional love he showed me throughout my studies.

ACKNOWLEDGEMENT

I give special recognition to my supervisor, Professor E Mnkandla for his tremendous work and support he gave me. He encouraged me to continue with my studies when I wanted to give up. I have learned a lot from him both academically and socially, he is indeed a man full of wisdom and knowledge.

Special thanks to my best friend Mr. Balasubramanian who encouraged me to continue with my studies. Many thanks go to my parents, in-laws, family, relatives and all my friends. They all supported and encouraged me in their special way.

I also thank my statistician, Claris Siyamayambo (Nee-Mahambo) for her support. Finally, I acknowledge all those whom I have not mentioned, but have contributed in one way or the other for this study to be a success.

TABLE OF CONTENTS

List of Figures

List of Tables

Publications

Balasubramanian, L. and Mnkandla, E. 2016, November. An evaluation to determine the extent and level of Agile Software Development Methodology adoption and implementation in the Botswana Software Development Industry, In *2016 International Conference on Advances in Computing and Communication Engineering (ICACCE)* (pp. 320-325). IEEE

Glossary or List of Acronyms

3D	Three dimensional
4IR	4th Industrial Revolution
AI	Artificial Intelligence
AMDD	Agile Model Driven Development
APLM	Agile Process Lifecycle Management
AR	Agro-robotics
ASD	Adaptive Software Development
ATU	Attitudes Towards Usage
AUP	Agile Unified Process
B2B	Business to Business
BI	Behavioural Intention to Use
BoFiNet	Botswana Fibre Networks
BGCSE	Botswana General Certificate of Secondary Education
BSDC	Botswana Software Development Company(ies)
BSDI	Botswana software development industry
BTC	Botswana Telecommunication Communication
CASE	Computer Aided System Engineering
CEO	Chief Executive Officers
CMMI	Capability Maturity Model Integration
CPS	Cyber Physical Systems
DSDM	Dynamic System Development Methodology
EP	Employee participants
GITR	Global Information Technology Report
GoB	Government of Botswana
ICT	Information Communications Technology(ies)
IDT	Innovation Diffusion Theory
IEEE	Institute of Electrical and Electronics Engineers
IS	Information Systems
ISO	International Organization for Standardization
ISO/IEC 29110	International Organization for Standardization/International Electrotechnical Commission

IT	Information Technology
IoT	Internet of Things
JCM	Job Characteristic Model
JCE	Junior Certificate Examinations
LD	Lean Development
JR	Job Relevance
LITS	Livestock Information Technology System
MPCU	Model of Personal Computer Utilization
OMT	Object Modelling Technique
OOSE	Object Oriented Software Engineering
OOP	Object-Oriented Programming
OP	Organisation participants
PEU	Perceived Ease-of-Use
PU	Perceived Usefulness
PSLE	Primary School Living Examination
PHP	Hypertext Pre-processor
PCI	Perceived Characteristic of Innovating
RAD	Rapid Application Development
RUP	Rational Unified Process
SADC	Southern African Development Council
SDLC	Software development life cycle
SDM	Software Development Methodology
SDM(s)	Software Development Methodology(ies)
SE	Software Engineering
TAM	Technological Acceptance Model
TPB	Theory of Planned Behaviour
UNISA	University of South Africa
VP	Validation participant
XP	Extreme Programming
WEF	World Economic Forum's

CHAPTER 1: SKETCHING THE BACKGROUND

1.1. Research Background

According to Sorooshian and Panigrahi (2020, p.903), an industrial revolution is fundamentally a technological transformation that focuses on new creative and innovative approaches to be productive. The world has witnessed several industrial revolutions, with each spawning and building upon its predecessor. Whereas the first industrial revolution resulted in production being mechanized, the second was based on electrification. The third was exemplified by information communication technology (ICT) and the use of electronics (i.e., digitisation). Schwab describes the fourth industrial revolution as follows:

> *"is characterized by a fusion of technologies that is blurring the lines between the physical, digital, and biological spheres."* (Schwab 2016, p.1).

Furthermore, Schwab (2015) states that:

> *"When compared with previous industrial revolutions, the fourth is evolving at an exponential rate rather than a linear pace."*

Some of the 4th industrial revolution (4IR) technologies include cyber physical systems (CPS), artificial intelligence (AI), robotics, big data analytics, blockchain, cloud computing, internet of things (IoT), three-dimensional (3D) printing, genetic engineering, quantum computing, and virtual reality (VR) (Bagnoli et al. 2018; Cho & Jeong 2019; Oke & Fernandes 2020; Rapanyane & Sethole 2020).

In Africa, the expansion of broadband access is leapfrogging the development and use of ICTs. The recent extension of the broadband undersea fibre optic cables to Africa Song, S (2019) has increased access to the Internet and is advancing the use of ICTs in Africa (Njikam et al. 2019) . In response, Africa has experienced a huge increase in the utilization of the Internet and mobile technologies and these services have become more affordable to the common person in the street (Setimela, MK 2018, p.22)

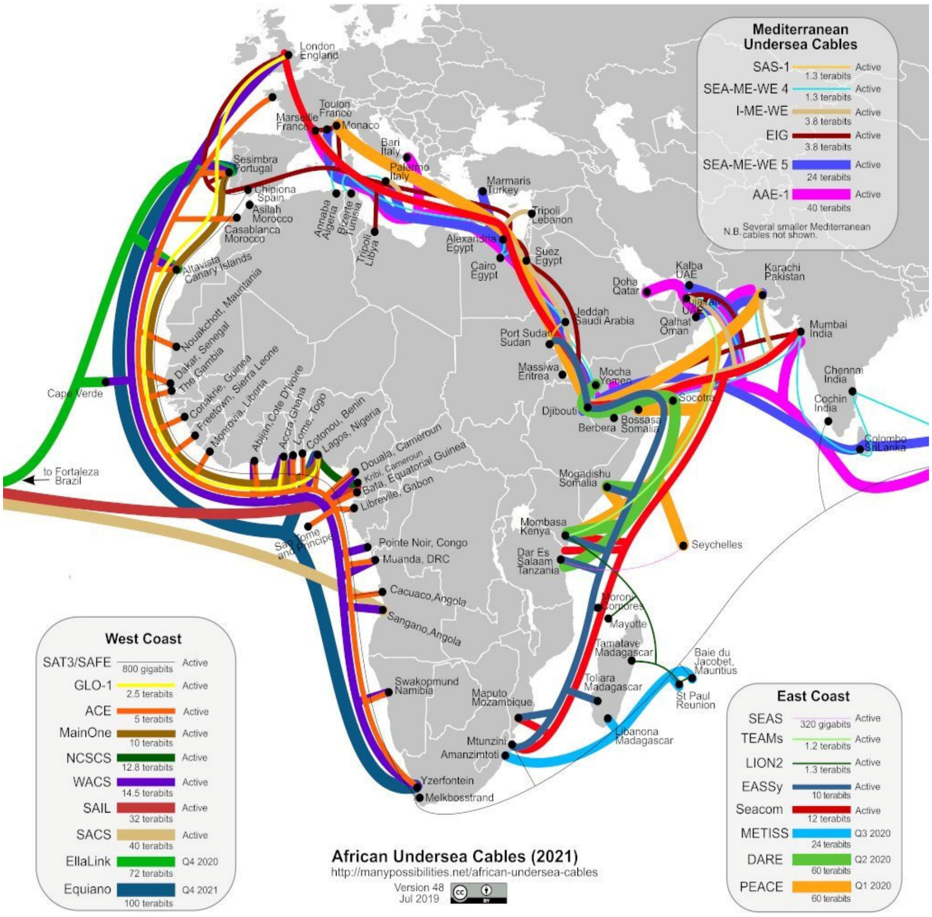


Figure 1: Africa undersea cables (Song 2019)

However, according to Emmanuel et al. (2020, p.8), Africa is still trapped in the 3rd industrial revolution technology, which primarily involves the application of ICTs and digitally-enabled devices for storing, transmission and sharing information. Africa thus needs to develop initiatives to progress beyond the existing dominant industries and promote the digital growth of new industries. A review by Emmanuel et al. (2020, p.8) has pointed out that the incumbent 3rd industrial revolution in Africa is awaiting a metamorphosis into the 4IR where advanced technologies and their applications such as VR, Agro-robotics (AR), IoT, and AI should be adopted and used for enhancing productivity and improving the quality of people's lives. Africa is challenged to introspect and devise appropriate strategies and policies to benefit from the 4IR. However, as demonstrated in Figure 2, the uptake is slow.

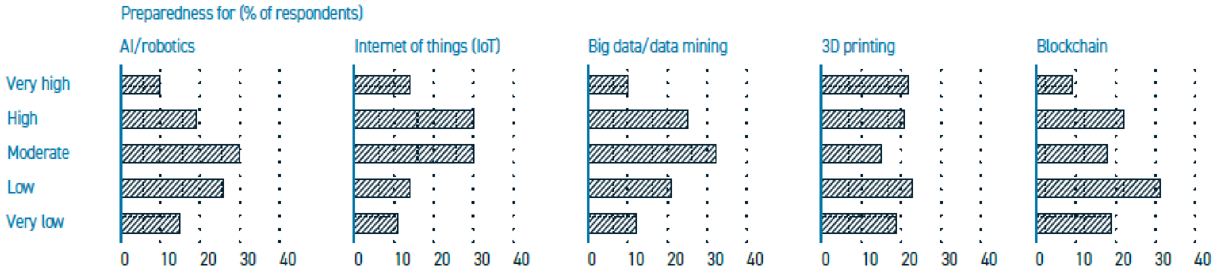


Figure 2: 4th Industrial Revolution Progress in Africa (Monga, Shimeles & Woldemichael 2019)

To be able to be competitive in this fast-changing industrial and technologically-driven environment, Software Development Methodologies (SDMs) will have to be agile. Gupta, M, George and Xia (2019, p.13) have noted that business and technology environments have become increasingly uncertain and dynamic. SDMs such as Agile are being adopted by various software development organisations in response to the failure of the traditional plan-driven SDMs (i.e., traditional SDMs) to manage software development projects.

This study seeks to develop a framework for the adoption of Agile SDMs in a developing country such as Botswana. Botswana is a member state of the Southern African Development Council (SADC) and the country is geographically located in the middle of Southern Africa and situated between South Africa, Namibia, Zambia, and Zimbabwe. Botswana has historically enjoyed strong and stable growth since independence, with sizable fiscal buffers and prudent policies playing a key role in shielding its economy. It is regarded as one of the world's fastest-growing economies in Africa. Botswana is rated as a middle-income country (Mogalakwe & Nyamnjoh 2017, p.2). SADC member states, inclusive of Botswana have acknowledged the need for digital transformation as the global economy is increasingly becoming digital. The terms of reference of a request for expression of interest for the selection of consultants for the development of a SADC 4IR strategy states that:

> *"The digital economy refers to an economy that is based on digital technologies or simply is the economic activity resulting from billions of daily online connections between people, enterprises, electronic devices, data*

> *and processes…The backbone of the digital economy is the growing interconnectedness of people, organisations and machines that results from the Internet, mobile technology and the Internet of things (IoT).*" (SADC 2020).

Botswana with its fast-growing economy and a quest to create a knowledge-based economy requires greater use of quality software applications for services and systems that are 4IR compliant. Already, the government of Botswana has committed itself to the transition from a resource-based economy to a digital-based economy. Then, in his speech Speaking at the World Economic Forum on Africa, Cape Town in September 2020, the president of Botswana, Mr Mokgweetsi Masisi, said:

> *"Through our vision, we aspire to transform Botswana from a resource-based to a knowledge-based economy through digitisation and embracing the 4IR. We are committed to a modern Botswana that is not only open to but is also able to compete with the rest of the world*," (Helmo 2019)

Therefore, quality software applications would be required for the very dynamic changes predicted. To meet these market needs, it is necessary for the Botswana software development community to use Agile SDMs due several major software projects that have failed in Botswana (Mokgoabone 2004; Mphale , Okike & Mogotlhwane 2016; Ramadubu 2021). SDMs provide guidelines for developing quality software applications. De Souza (2020) noted that the decision to select an appropriate software development methodology will either have a positive or negative impact on the quality of the software developed. Thus, it is imperative to understand which SDMs to use, especially in terms of their strengths, limitations, and suitability for the guidance of a project. Furthermore, De Souza (2020) commented that a software development methodology is not a silver bullet for all software projects—it should not be a "*one size fits all"* SDM for developing software. According to MacCormack and Verganti (2003), software development professionals and their organizations should

> *"…evaluate a wide range of contextual factors before deciding on the most appropriate process to adopt for any given project". (p.230)*

Subramanian et al. (2009) also stated that the selected software development methodology should

> *"best fit the conditions, product, talent, and goals of the markets and organisations". (Subramanian et al. 2009, p.118)*

With regards to the Botswana software development community, it is important to know the factors that affect the adoption of new SDMs by the Botswana software industry.

1.2. Research Problem Statement

The problem thus is: How to develop a framework for the adoption of agile software development methodologies in Botswana that can be used by the Botswana software development industry for the selection of an appropriate modern and agile—able to adapt and change when re-assessed— software development methodology.

According to the literature such an inclusive and collaborative reference framework—a framework that can consider situational factors affecting or inhibiting proper selection and use of modern SDMs—would be very useful. Being able to quickly adapt to changing circumstances is not only problematic for the Botswana software industry but is a global phenomenon. The software development environment currently requires the ability to quickly adapt to disruptive technologies that the 4IR would necessitate.

According to Ayalew and Motlhala (2014, p.121), the Botswana software development industry, software development companies and professional software developers have shown a sluggish rate of adoption and utilisation of

Agile SDMs such as, Scrum, Crystal Clear, ICONIX, Agile Unified Process (AUP), Extreme Programming (XP), Lean Development (LD), Dynamic System Development Methodology (DSDM), ICONIX, Agile Model Driven Development (AMDD) and Adaptive Software Development (ASD) etc. In support of this observation (Surya 2018) commended that

> *"It's no use having technology that's responsive to the business if the business can't respond to technology and the demands coming from its customers." (Surya 2018)*

1.3. Aim of the Research Study

This study aims to explore and determine the factors that affect the adoption of Agile SDMs (e.g., Agile, Scrum, and Crystal clear, ICONIX, Agile Unified Process (AUP), Extreme Programming (XP), Lean Development (LD), Dynamic System Development Methodology (DSDM), ICONIX, Agile Model Driven Development (AMDD) and Adaptive Software Development (ASD) and others) by the Botswana software development industry and thereafter develop a framework tool to guide the adoption of Agile SDMs by Botswana Software Development Companies.

1.4. Research Objectives

The objectives of this research are therefore:

- To understand the perceptions of twelve Botswana Software Development Companies (BSDCs) (and their professional software developers) about the adoption and implementation of SDMs
- To explore factors influencing the adoption and usage of SDMs in the Botswana software development industry (BSDI) with specific reference to the twelve BSDC in Botswana
- To develop, in collaboration with the software development community, a framework based on Technology Acceptance Model for assisting and guiding BSDI, BSDC and professional software developers when selecting, adopting, and implementing Agile SDMs in Botswana.

1.5. Research Question and Sub-Questions

The aim and objectives of this research study cannot be addressed in isolation. To this end, the main research question and sub-research questions were formulated to enhance the outcome of this study. To this end, the main research question derived from the problem statement is as follows:

> *What are the factors affecting Botswana's software development companies and software development professionals in adopting and utilising Agile software development methods?*

The following research sub-questions—which resulted from the research objectives of the study—are aimed at addressing the main research question:

- How do the perceptions of Botswana professional software developers and their respective BSDCs influence the adoption and implementation of SDMs in Botswana?
- What are the factors that affect the adoption or non-adoption of Agile SDMs in the BSDI?
- How should an SDM framework be designed for the BSDI to facilitate the use and selection of appropriate and Agile SDMs by BSDC and their professional software developers?

1.6. Significance of the Research

This research was carried out with the aim to identify and understand the factors that inhibit the adoption, selection and use of Agile SDMs in the BSDI. Following which, a framework was developed to assist the BSDI with the selection, adoption and implementation of Agile SDMs.

According to Hazzan and Dubinsky (2008, p.1), a need exists to understand the human behavioural aspects of software development. An empirical analysis by Glass, Ramesh and Vessey (1994, p.89) revealed that software engineering studies were focused on algorithm formulation (58%), descriptive research approach (28%) and evaluative approach (14%). Hazzan and Dubinsky (2008, p.1) and Dybå et al. (2011, p.425) indicated that opportunities abound for undertaking qualitative software engineering research to fully exploit, explore and discover areas not covered by the quantitative research analysis carried out by (Glass, Ramesh & Vessey 1994, p.89). This research study therefore sought to complement the existing quantitative studies by carrying out a qualitative research study to generate new knowledge on SDMs adoption and usage by the BSDI with specific reference to the adoption and implementation of modern and Agile SDMs. The significance of this research study is underpinned by Dybå et al. (2011, p.426)'s observation and contended that software engineering deals with complicated human tendencies in an atmosphere and circumstance that is not well formulated theoretically or empirically. In a nutshell, the authors encourage researchers to use qualitative research to explore and generate new knowledge on the use of software engineering through a qualitative research methodology that seeks perceptions, sentiments, opinions based on their experience and exposure (Dybå et al. 2011, p.426).

This research study is the first of its kind whereby factors that inhibit the adoption of Agile SDMs by BSDI, BSDC and professional software developers are qualitatively analysed. Thus, the research study contributes to the following:

- Theoretical knowledge on the most-used SDMs and in addition, the reasons for selecting types of SDMs used by BCDCs.
- The general mindset of the BSDI with regards to their perceptions of Agile SDMs and state of preparedness in adopting Agile SDMs.
- Identification of the factors that may inhibit or affect the adoption and usage of Agile SDMs in the BSDI.
- Development of an SDM framework that can be used by the BSDI as a tool to assist with the selection, adoption and use of an appropriate Agile SDM.
- The research findings, conclusions, recommendations and the framework provide relevant guidelines for ICT policymakers in the government of Botswana, especially on decisions relating to new aspects on the use of SDMs. This would apply to the BSDI when the government comes with initiatives of growing the economy through the use of appropriate ICT tools

or floating for software development tenders and the selection of typically required SDMs.

- A basis—provided for the government, policymakers and BSDI—to collaborate and share the burden for developing software that benefits the nation and other interested stakeholders.

1.7. Research Limitations and Delimitations

The subjects of the research study were limited to twelve software developer companies in Botswana. These filtered BSDCs are the most proactive companies that develop small to large-scale software applications. The participation of these filtered BSDCs was critical to this research study as they possess rich data and information about practices of software development in Botswana. All the filtered BSDCs that participated in the study are geographically located in the cities of Gaborone and Francistown.

Since this study involved the collection of research data from specific organisations and government departments, issues of privacy needed to be considered. Access to data from the participants, organisations, or even documents was in some cases denied. This unavoidably limited the research findings and affected the progress of the research. Three of the BSDC approached refused to participate in the research.

Personal interviews were done using semi-structured questionnaires and confidentiality and informed consent were assured. The interviewer explained to participants that they had the right to pull out of the interview at any time during the interview. However, during these interviews, some of the participants did not feel free to respond to specific questions. For example, organisations that have previously undertaken government projects were not willing to share data, since they considered such data to be sensitive.

1.8. Research Design and Methodology

This research study adopted a qualitative research approach underpinned by an interpretivist/constructivist philosophy. The case study methodology was adopted (see Figure 3) since Yin (2003) shares the view that a case study design should be considered when: (a) the focus of the study is to answer "how" and "why" questions; (b) the researcher cannot manipulate the behaviour of those involved in the study; (c) the researcher wants to cover contextual conditions because the researcher believes these conditions are relevant to the phenomenon under investigation, and (d) the boundaries are not clear between the phenomenon and context. In this study, the Technology Acceptance Model II was adopted for use in the Botswana software development context to assist identify factors that would inhibit the adoption and usage of a given technology.

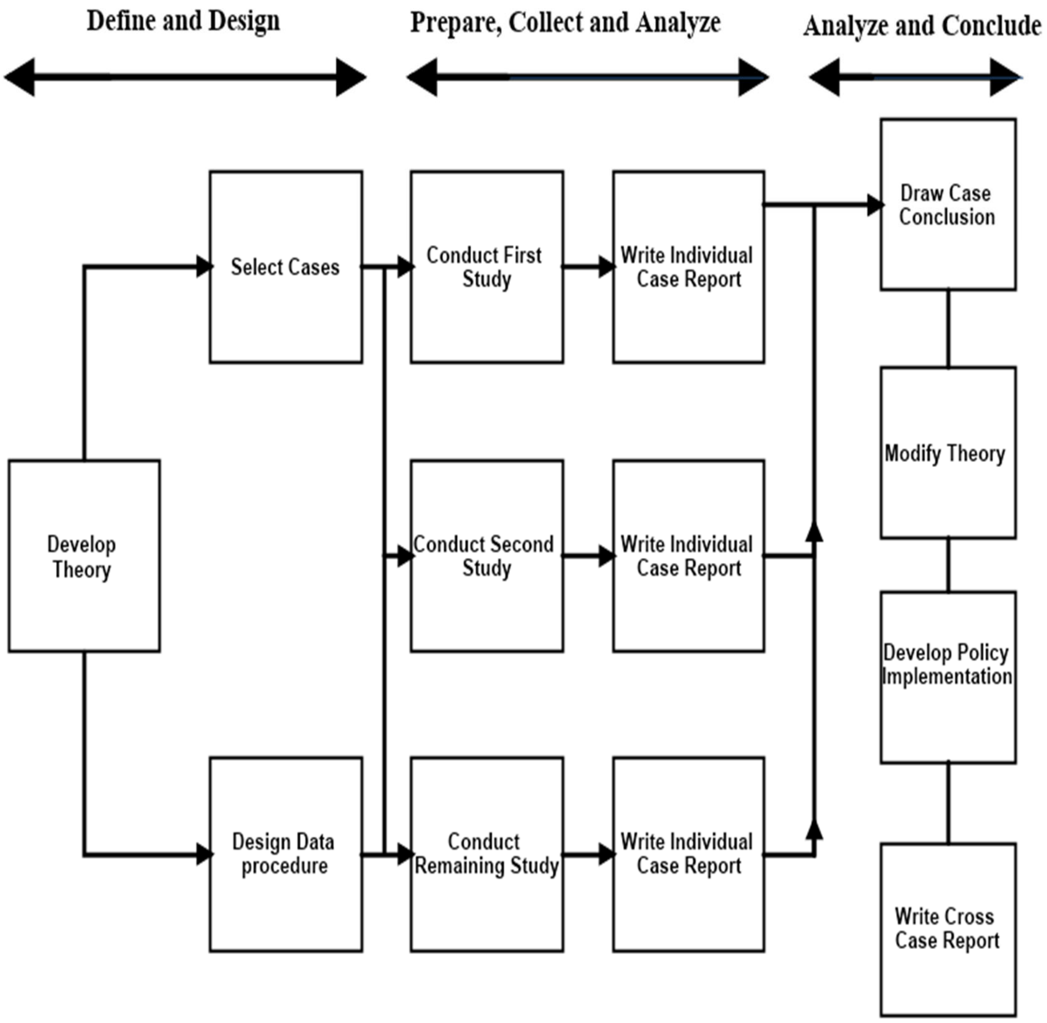


Figure 3 An illustration of the research design process (adopted from Yin, R 2003)

1.9. Layout of the Thesis

This thesis is structured as follows:

Chapter 1: Sketching the Background - presents the background information on the software development methods used by the Botswana software development industry (BSDI) as well as current trends relating to the use of these methods. The research purpose, the problem, research questions, significance and limitations of the study are also explained.

Chapter 2: Literature Review - This chapter critiques the existing literature relating to SDMs, their evolution as well as the adoption of SDMs for the development of software. Furthermore, the chapter deals with literature related to the keywords and key constructs of this study as well as the research questions posed.

Chapter 3: Research Methodology and Design covers the research paradigm, qualitative research, research design, interpretivism/constructivism, the case study research methodology, methods of data collection, methods of data analysis, ethical considerations, and validity and reliability in case study research.

Chapter 4: Findings discusses data analysis, demographic information, and the generation of themes and patterns.

Chapter 5: Interpretation of Results and presents a framework for the adoption of the Agile Software Development Framework as well as the validation.

Chapter 6: Discussion and Conclusion – this chapter presents the practical implications and impact of the use of SDMs in software engineering on the relevant society, limitations of the research, gaps not covered by the research, possible future research, and contribution of the study.

CHAPTER 2: RELATED LITERATURE

2.1. Introduction

In the previous chapter, the motivation for the research and the background to understand the topic of the research were presented. In this chapter, the literature that supports the research is reviewed, the key concepts are discussed, and the research questions are explored. The 4th industrial revolution (4IR) is expected to, in many ways, greatly impact the way we live and work. Disruptive technologies will change our lives, and software will especially play a big role in advancing and achieving these changes. To enable competitiveness in this fast-changing environment, Agile SDMs have been widely adopted in the emerging software industries.

2.2. Disruptive Technologies

Software has contributed immensely to how people—in all walks of life—are living today (Sorooshian & Panigrahi 2020, p.903). For instance, during the COVID-19 pandemic, many people all over the world were able to continue with their work online. This was made possible by broadband Internet services, which provide access to almost every corner of the world. Video conferencing software applications such as Zoom, Skype, Cisco Webex, Google Meets and others are used by many people to keep in contact with friends and family or to attend virtual work-related meetings. According to Blasi et al. (2020, p.1) mobile technologies—such as WhatsApp and Telegram—have also contributed immensely to remote collaboration. During the critical times of the COVID-19 in Italy, data and information were collected on time-related analysis of conversations through applications such as WhatsApp messenger-based group chat and WhatsApp Web. The conclusion of this research showed that:

> *"…WhatsApp instant-messaging system seems to be a useful tool to share news and reactions between medical oncologists to rapidly implement*

> *necessary health measures and answers to most cancer patients' needs and queries in the COVID-19 pandemic scenario"* (Blasi et al. 2020, p.1)

2.2.1. Artificial Intelligence

Sorooshian and Panigrahi (2020, p.903) have mentioned how artificial intelligence (AI) will achieve automation control for monotonous tasks in fields such as healthcare, astronomy, gaming, manufacturing, entertainment, the automotive industry and others. In gaming, AI-based software applications Watson, Deep Blue, and AlphaGo have conquered human competitors at chess games. In the automobile assembly industry, AI software applications have facilitated the auto manufacturing work process by optimizing the production process without manual interventions. The AI software application for automobile assembling traces the real-time motion of automobile products parts being assembled and compute their arrival period by considering factors such as harbour crowding, the impact of natural disaster and other logistical operational factors before assembling a complete automobile.

In robotics, AI software has made a remarkable impact by improving their performance from programmed robots to intelligent robots. The AI-based intelligent robot can learn from its environment of operation without being pre-programmed to tackle the tasks. For instance, the intelligent Humanoid robot, Erica and Sophia can talk to people in a very natural way.

In entertainment, the web- and mobile-based software, Netflix, uses some AI features to interact and track clients and inform them about upcoming movies. Netflix is also able to customize movies to view based on the user's history of movies the user watched. Social media applications such as Twitter, Instagram, and Facebook host so many user accounts and profiles and they require systematic data management systems that include retrieval and security. AI has been used in these applications to efficiently manage the mammoth amount of data accrued and trace patterns and trends in this data to identify thematic issues, hashtags, user requirements and many other things (Alhashmi, Salloum & Mhamdi 2019; Rapanyane & Sethole 2020).

2.2.2. Cloud Computing

Cloud computing is defined by Amani et al. (2020, p.5326) as an efficient data management approach that allows storing, accessing, and analysing datasets using high-end computing servers. This technology can be applied successfully in many areas such as the mobile business, electronic commerce, and data storage service, which makes revolutionary improvements to the applications of the information technology (IT) industry. Therefore, cloud computing provides infrastructure and platform for storage services, data backup, data retrieval, computing of tasks and data coordinated through cloud software in various arrayed servers (Purnama & Ginardi 2019, p.519). Cloud-computing data-management services are easily accessed from computer networks such as the Internet, intranets or extranets, and are available on-demand anytime and anywhere. The technology provides modern data-management technologies for implementing software web-based applications such as developing an enterprise architecture, developing management dashboards and cloud-based collaboration. To date, many cloud-application services provide services related to real-time navigation and mapping, e-commerce software applications, and online file storage. Amazon is an example of a typical web service-based and cloud-computing application that provides simple storage facilities (Borge & Poonia 2020, p.53).

2.2.3. The Internet of Things

The Internet of Things (IoT) is yet another technology that will become more and more pronounced during the 4IR. IoT has emerged to serve the ever-evolving and dynamic needs of contemporary ways for a modern user's social and business needs. The development of IoT has emerged from the ubiquitous presence of wireless and mobile technologies, miniaturized wireless sensors and provision of scalable data storage facilities such as cloud computing (Guangzhong 2020, p.8).

As defined by Hildayanti and Machrizzandi (2020, p.80), IoT is an advanced technology that provides linkage to a system, physical objects, and service to facilitate communication of an object-to-object by providing means for data sharing and interaction. The IoT allows wireless devices or objects to share information with end-users and provides further integration of technology into users' daily lives. The use of IoT and software-related applications are found in health, utilities, agriculture, security, manufacturing, education, transportation, military, social activities, and many other areas (Baccelli et al. 2016; Vidyarthi & Kumar 2018; Zambonelli 2017).

Hildayanti and Machrizzandi (2020, p.80) talk of smart building technology as a means of using IoTs to manage and control building environmental adjustments such as managing electricity utilisation, personal and home security management, and many others. Most of these technologies can be managed remotely using a smartphone and mobile applications. Smart building technologies can be used to collect data through objects such as wireless sensors throughout the whole house or building. Data collected includes energy usage, space utilization, and occupant productivity thus providing better insight and control over a broad range of systems that span the entire building (Hildayanti & Machrizzandi 2020).

Agricultural IoT—also referred to as smart agriculture—can, for example, share agricultural information on pig breeding, slaughtering, segmentation storage and marketing in China (Guangzhong 2020, p.8). This smart agriculture technology is used to provide traceable reports of products for purposes of marketing and quality control measures.

IoTs are also applied in health. Aceto, Persico and Pescapé (2020, p.1) describe IoT as a technology that has completely transformed eHealth into what is termed Health Care 4.0 technology. The Health Care 4.0 technology, according to Aceto, Persico and Pescapé (2020, p.3), has —through the use of IoTs—offered advanced monitoring of physiological and pathological signals and medication intake. New approaches, processes, and applications such as enhanced living environments, home-based rehabilitation, and personalized healthcare supervision also contribute to the improved health of patients.

2.2.4. Cyber Physical System

A Cyber-physical system (CPS) is a disruptive technology that uses a “computer box” when applied in automobiles. The function of the computers box is to monitor and coordinate the automobile’s key subsystems and functionalities. This new generation digital system uses software to monitor data acquired from the automobile sensors and it provides operational and maintenance information. An automobile computer box runs and detects operational errors, provide diagnostics, and identifies malfunctioning subsystems and components of an automobile. For instance, it can inform on operational issues regarding the car’s fuel consumption, that is, with available petrol at speed x you will travel y km.

2.2.5. Big Data Analytics

Big data analytics is a technology concerned with the data management of large sets of data. The technology involves the integration of various data sets from various sources and facilitates the discovery of trends and patterns within that data. According to Hallikainen, Savimäki and Laukkanen (2020, p.90), the data sets are huge, complex unstructured, and multiple. Examples of typical data sets used for analytics include business records, transactions in mobile banking, online-user-generated content such as images, blog posts, and tweets. Advanced and high-profile organisations are making use of big data analytics to better comprehend their buyers, be competitive in the market, find insights more quickly, accelerate products and services, and improve profits. A Hallikainen, Savimäki and Laukkanen (2020, p.90) study conducted in America reported that 84% of industry-leading companies worldwide have started using big data analytics to manage business-to-business (B2B) customer relationships and to support the personalisation and customisation of sales and customer services.

Modern product management, engineering, and delivery disciplines are needed to be able to keep up with the needs of the 4IR (Brummelen & Slenders 2019). 4IR technologies are critiqued in the literature as having the potential for advancing technology and will most probably promote new ways of doing business using AI, augmented reality, virtual reality, big data analytics, IoT, and automated robots (Aceto, Persico & Pescapé 2020, p.3; Borge & Poonia 2020, p.53; Guangzhong 2020, p.8; Hildayanti & Machrizzandi 2020, p.80; Li et al. 2020, p.325; Marnewick & Marnewick 2019, p.314; Purnama & Ginardi 2019, p.519). There is thus a need to develop high-value software applications that can be used anywhere, anytime and on any platform.

2.3. Software Development and Methodologies

Software is a collection of computer instructions that are logically and sequentially arranged and are executed by a computer to achieve a predefined task. Software is intangible, invisible, extremely logical and abstract (Kittlaus & Clough 2009, p.5). To create software requires time, money, and several other resources. Software developers need to apply and understand the theory and practice of software development and in particular the discipline of software engineering (SE). SE is a discipline that has emerged and evolved for software development. Wang and Patel (2000, p.3), defined software engineering as:

> *"…a discipline that adopts engineering approaches, such as established methodologies, processes, tools, standards, organization methods, management methods, quality assurance systems, and the like, in the development of large-scale software seeking to result in high productivity, low cost, controllable quality, and measurable development schedule"* Wang and Patel (2000, p.3)

To develop a quality software product that meets required business standards and development constraints, the software product must be developed using engineering methods that ensure the quality of the resulting product (Akbar et al. 2017, p.4811).

2.4. The Software Development Life Cycle

The software development life cycle (SDLC) is an established methodology used for developing software. The stages specified in the SDLC framework provide a descriptive overview of the generic procedures to be followed by software project teams when developing software (Usman & Ogwueleka 2018, p.52). The choice of environment and setting is largely dependent on the user's needs (Mishra & Dubey 2013, p.64).

Kramer (2018, p.77) defines SDLC as a framework that provides an abstract guideline for the development of software applications. Olszewska and Allison (2018, p.1) and Georgiou, Rizou and Spinellis (2019, p.1) state that the SDLC is composed of task-oriented stages, and for each stated stage, a clear task decomposition is described and is split into sub-tasks.

The common stages specified by the SDLC Swersky (2018). include eliciting user requirements for the software to be developed, developing the software based on agreed user requirements, designing the models for developing the software, developing or coding the software, testing the functionality of the developed software and deployment and maintenance of the developed software. These tasks need to be planned, scheduled, budgeted for, monitored, and communicated by adopting an appropriate SDM for each different software development setting. There are many variants of SDMs to suit specific software development needs. SDMs are also referred to as SDLC models in many research studies (Adanna & Nonyelum 2020; Alshamrani, Adel & Bahattab 2015; Anureet & Kulwan 2015; Okesola et al. 2020; Usman & Ogwueleka 2018).

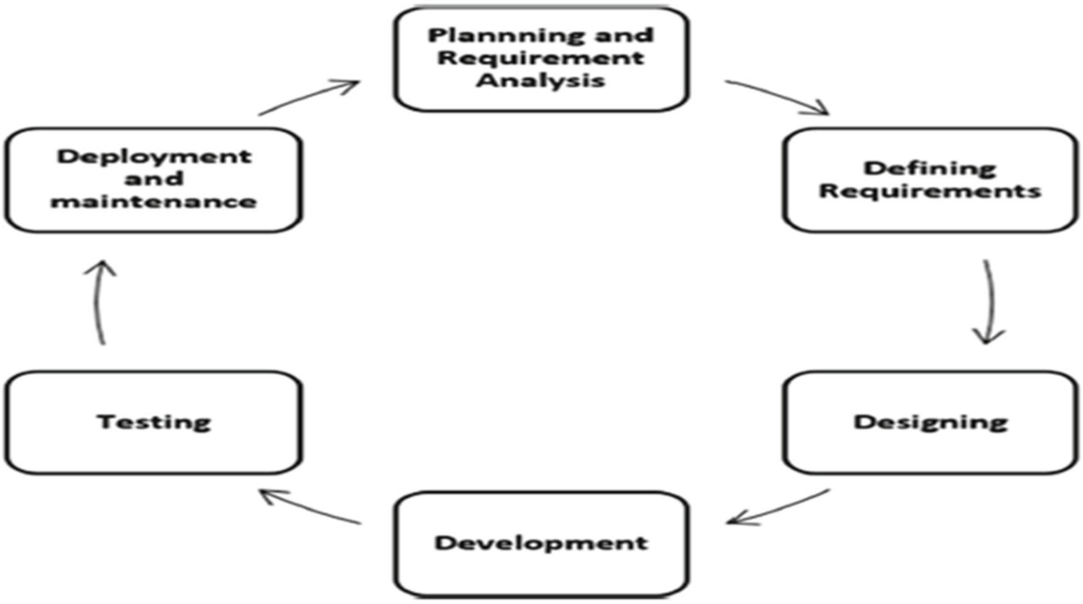

Plannning and Requirement Analysis
Defining Requirements
Designing
Development
Testing
Deployment and maintenance

Figure 4: SDLC common phases (adapted from (Okesola et al. 2020))

Whilst the SDLC provides a **descriptive** approach in developing software applications, SDMs provide a **prescriptive** approach in developing software applications (Aaen 2002, p.379).

The SDLC framework provides a wide and generic explanation as to how software should be developed but there is a need for a specific context and setting that requires special and appropriate approaches. SDMs were developed by various software development experts: universities, industry and Standard Organisations with the intent to address specific issues related to existing problems noted in given circumstances (Akbar et al. 2018, p.8066; Vijayasarathy & Butler 2015, p.86). The many designed and created SDMs guide about time, budget and general resources estimate required based on a specific technical approach of the SDM model.

Aggarwal and Dhir (2014, p.476) define SDM as a controlled strategy for carrying out the steps in the life cycle of a software application in a foreseeable, effective, and repeatable way. The software development process can be articulated as a systematic method for conducting the stages in the development of a software application program in a predictable, successful and iterative way (Vasantrao 2011, p.96). Munassar and Govardhan (2010, p.96) describe SDM as an abstract representation describing the process of developing software from scratch to maintenance (Munassar & Govardhan 2010, p.96). Pressman (2010, p.65) defined SDM as a group of methods, skills, equipment, and documentation aids that will assist systems developers in their efforts to design and develop software. The overall purpose of an SDM is to guide a software project for the software construction in any given circumstance and setting, and that a high-quality software application with requisite user functionalities is produced in time, developed within an affordable budget limit and avoids waste of critical resources.

Subramanian et al. (2009, p.118) describe four key critical success factors for evaluating the success of a software project when using an SDM. These focal points are project schedule, project scope, project budget and finally the software product quality. These four focal points or metrics can be used to effectively appraise the success of a software project. The adoption of a proper SDM should focus on these four metrics to deliver quality software. These key metrics consider logistical arrangements such as: how to get what software clients want as solutions to an existing problem, what tools to use for developing the software, the documentation of the software development process, which software models to use, validating and verifying the developed software functionality, deploying and maintaining the developed software within a given time frame and budget, and the delivery of intended functions (Vasantrao 2011, p.97).

2.5. Common types of Software Development Methodologies

Parveen, Khan and Sadiq (2014, p.535) and Aggarwal and Dhir (2014, p.476) refer to the two common types of SDMs, namely: traditional SDMs and Agile SDMs. These authors define traditional SDMs as a heavyweight SDM approach and the agile SDM as a lightweight SDM approach. The term heavyweight describes the rigidity and firmness experienced by the traditional software development approach as they follow sequential and fixed stages. There is thorough and detailed planning and documentation of each software stage. Upon completion of each stage, there is no going back until the cycle is completed. The software development is not responsive to new and dynamic changes during software development.

In contrast, the lightweight term implies the agility and flexibility of software development. Software development is done by planning and developing through short cycles called *'iterations'*, which allow for rapid production and constant revision that involves the software client throughout the process. The lightweight SDM is specially focused on constant communication with clients and developers. This interaction facilitates the management of new emergent and dynamic software requirement functions needed by the client market or business. The constant interaction between developer and client facilitates early releases of the software with prime functional requirements. There is less documentation as more effort is directed in developing the application on a real-time basis with confirmation from the client of implemented functional requirements. Chan and Thong (2009, p.804) also distinguish between the traditional SDMs and the Agile SDMs, the difference of which is based on the software client involvement in the process and how the expectations of the software client are fulfilled during the software development process.

In the traditional SDM approach, the software developer does not have continuous contact with the software client, it only ends at elicitation of software requirements stage only. Therefore traditional SDMs limited have limited contact with the software client during software development and occasionally involves meeting with the software client during the software requirements stage only (Stoica, Mircea & Ghilic-Micu 2013, p.64).

In the agile SDM approach, it is assumed that the software client is more knowledgeable on the functional requirements than the software developer, and client participation throughout the software development process is essential. By putting the software client as the focus of the software development process, the agile SDM facilitates constant interaction with the client throughout the software development process. Mnkandla (2008, p.8) is of the opinion that Agile SDMs consider the way people relate to one another to produce the appropriate software product. According to Mnkandla (2008, p.8), involving the software client in the development team will allow the dynamic needs of the software client to be met, with specific emphasis being placed on producing a quality software product.

Over five decades, software development methods and concepts have transformed into new designs and perceptions to suit market-oriented and technology-oriented software applications and solutions, which are called modern SDMs (Vohra & Singh 2013, p.23). These modern SDMs have emerged with various pros and cons as well as weaknesses and strengths. The multitude of these modern SDMs has emerged to correspond to different software development organisations' requirements that are based on the goals, needs, structures, and backgrounds of the organisation. Thus, the modern SDLC models are adequately flexible to be used across different types of businesses, products and services (Tegegne, Seppänen & Ahmad 2019, p.1).

In this research study, the term “modern software development methodology” is used to refer to SDMs other than the traditional SDMs. The literature also conforms to this approach where Agile SDMs and component-based SDMs are referred to as modern SDMs (Kumar & Bhatia 2014, p.189; Vohra & Singh 2013, p.23). For instance, Scheller et al. (2010, p.253) describe modern SDM as “iterative development” or “iterative and incremental development” techniques. The modern SDMs use iteration cycles during software construction (Scheller et al. 2010, p.253). Each iteration formulates a discrete mini-project, consisting of defined and selected requirements for its model design, coding, testing, assessment, and documentation (Scheller et al. 2010, p.254). A single iteration is done in 2 to 8 weeks and therefore facilitates the early release of a functional part of working software.

Golfarelli, Rizzi and Turricchia (2011, p.66) define modern SDMs as approaches that focus on working with the client as well as with the software developers through the concept of self-organising teams. These teams are highly dynamic and are responsible for developing the software by including the client as part of the development team and owning the responsibility and ownership. The ultimate goal of adopting this approach is to reduce the risk of expressing ambiguous requirements and making software validation easier and more effective, thus allowing the early release of the software functions during each iteration (Golfarelli, Rizzi & Turricchia 2011, p.74).

For purposes of clarity and objectivity, this research study considered SDM variants of agile and traditional SDMs, which are considered to be the "leaders" in the multitude of many SDMs (Version One 2013). These two variants of SDMs have acquired the status of a "standard" for a significant part of the software development industry. All the remaining SDMs are considered as "followers" and are therefore not discussed in this study even though their existence is acknowledged. These "follower" SDMs are categorised based on the principle that they follow the standard SDM leaders in their number of references, users, and implementations both in research and the software industry.

2.5.1. Traditional Software Development Methodologies

According to (Despa 2014, p.37; Rao, Naidu & Chakka 2011, p.35; VersionOne 2013), the most used traditional SDMs include the Waterfall, Prototype, Incremental and Spiral methodologies.

2.5.1.1. Waterfall SDM

The Waterfall SDM comprises sequential and logical cascading stages. The software development stages are executed linearly and fixedly. Naderuzzaman, Rabbi and Beg (2011, p.36) list these stages as planning, software requirements analysis, software design plans, coding and testing, deployment, and implementation, and finally maintenance (see Figure 5). Completion of each

Waterfall SDM stage leads to the closure of the current phase activities before the next phase tasks are started. Each Waterfall SDM stage involves thorough conceptualisation and documentation of the stages. This approach is good for large-scale software development projects or predictive software development projects. However, the Waterfall SDM's greatest risks are the possibility of exceeding the project budget cost, deviating from the project scope and heavy-duty documentation before delivery of the software application (Munassar & Govardhan 2010, p.96).

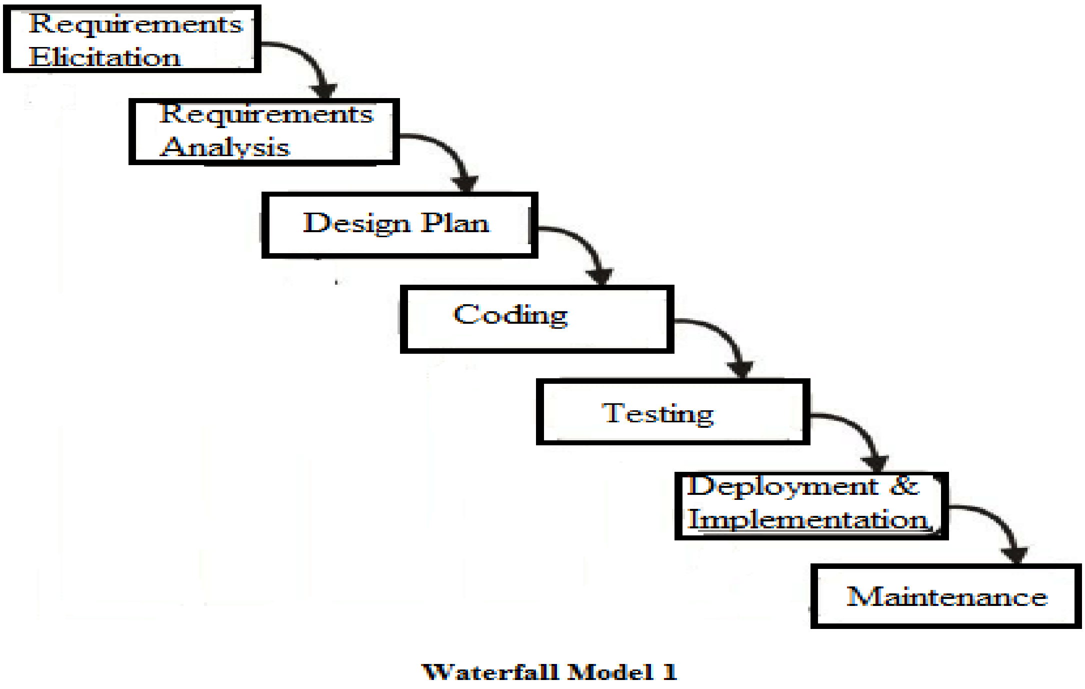


Figure 5: Waterfall models Stages (Munassar & Govardhan 2010)

2.5.1.2. The Prototype Methodology

The Prototype SDM is another variant of the traditional SDM (see Figure 6**Error! Reference source not found.**). It is not a complete standalone SDM approach but a way to drive particular parts of a bigger, more traditional SDM such as the Incremental, Spiral, or Rapid Application Development (RAD) approaches (Naderuzzaman, Rabbi & Beg 2011, p.37). The two important stages of the Prototype SDM are the prototype development stage and the final software development stage. The final development stage is when other traditional SDM stages are co-opted to develop the final software application (Naderuzzaman, Rabbi & Beg 2011, p.37). The Prototype stage is used as an SDM tool that is nested within another SDM or meta-model SDM. The focus of the Prototype stage is to ensure that the software meets the specified functional requirements of the client through a constant interaction process indicated in Figure 6. The prototype stages are iterative through stages of rapid design of the software application plans, development of the prototype, customer's interaction to evaluate the software and adjustment of the prototype based on software client inputs. The cycle begins again until the software client is content with the software application functionality.

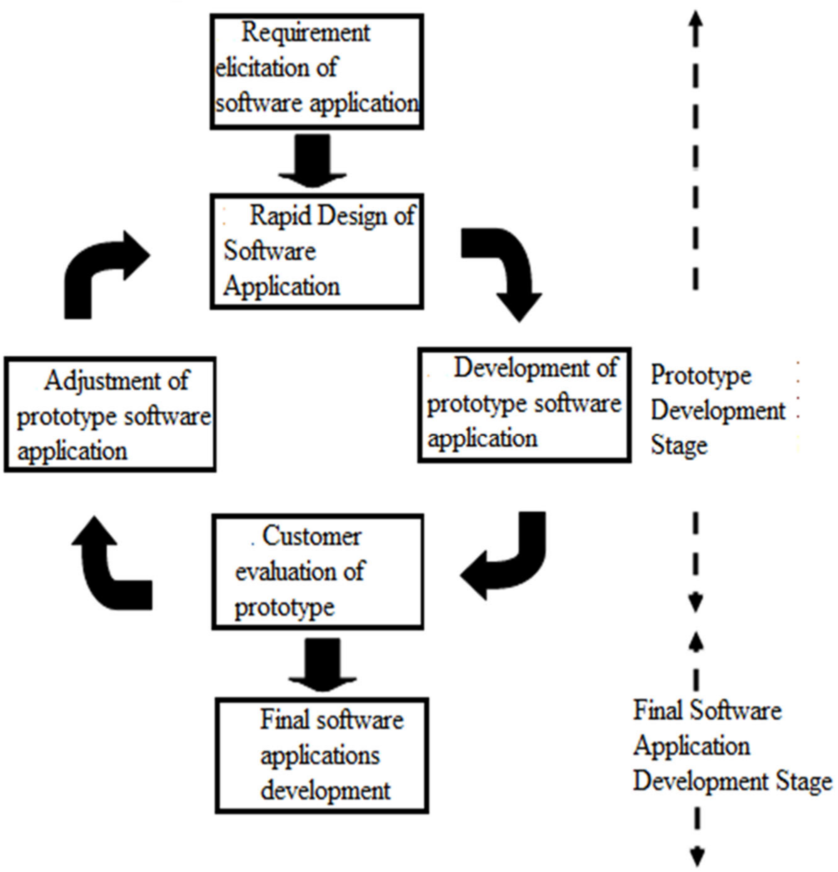

Requirement elicitation of software application
Rapid Design of Software Application
Development of prototype software application
Prototype Development Stage
Customer evaluation of prototype
Adjustment of prototype software application
Final software applications development
Final Software Application Development Stage

Figure 6: Prototype SDM (Naderuzzaman, Rabbi & Beg 2011)

2.5.1.3. The Incremental SDM

Incremental SDM is a traditional SDM that effect changes in the software requirements much more effectively through constant interaction with the software client. This SDM can accommodate the software client's ever-changing functional requirement because "software customers often do not know what they need during the software development session" (Naderuzzaman, Rabbi & Beg 2011, p.36). Alshamrani, A and Bahattab (2015 p.106) described the Incremental SDM as an iterative and incremental SDM because the SDM brings together components of the waterfall model in a repetitive way. The iterative processes are indicated in Figure 7, whereby the incremental software application developments are from Build 1 to Build N. The client feedback is used to adjust the software application until the software client is satisfied with the software's functional requirements. The first Build 1 is a prototype and is initially created to resemble the basic functionality of the intended system; Naderuzzaman, Rabbi and Beg (2011, p.38) call it a "*demo system*".

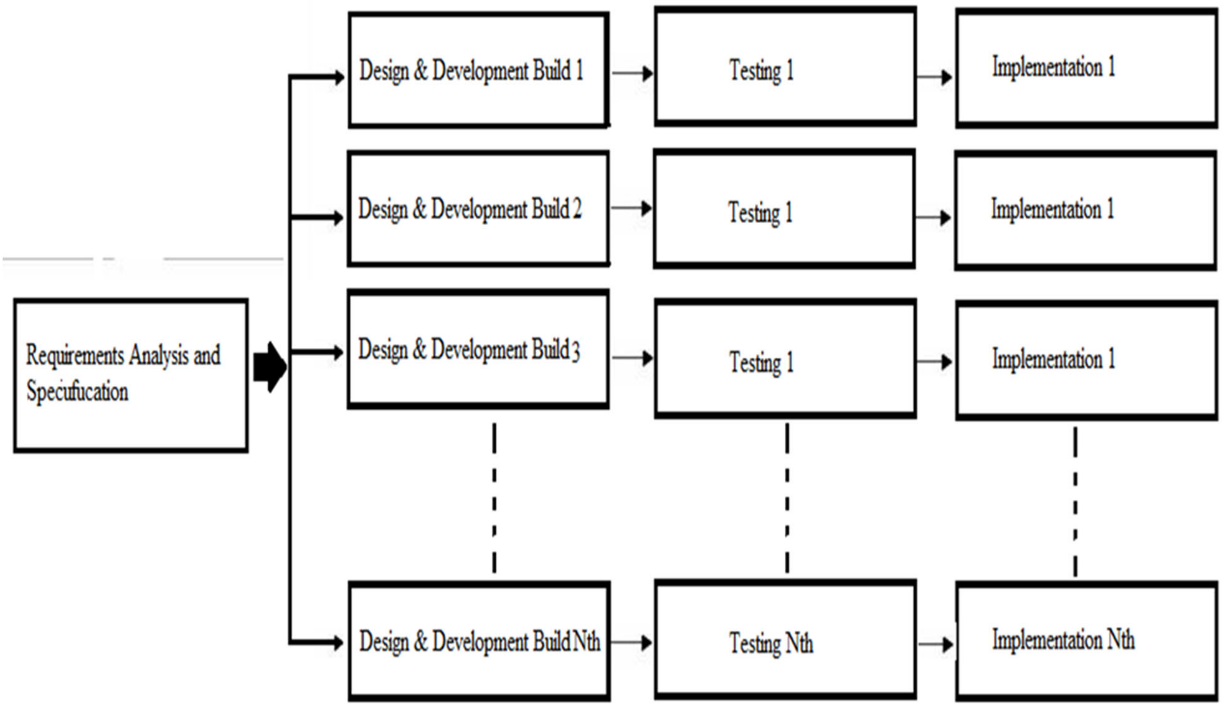

Requirements Analysis and Specufucation
Design & Development Build 1
Testing 1
Implementation 1
Design & Development Build 2
Testing 1
Implementation 1
Design & Development Build 3
Testing 1
Implementation 1
Design & Development Build Nth
Testing Nth
Implementation Nth

Figure 7: Incremental SDM (Naderuzzaman, Rabbi & Beg 2011)

2.5.1.4. The Spiral SDM

Alshamrani, A and Bahattab (2015 p.107) refer to this SDM as a meta-model, which can be nested within another SDM during software development. The Spiral SDM’s greatest strength lies in managing risk during software development (Akbar et al. 2017, p.4811; Alshamrani, A & Bahattab 2015 p.107). The Spiral SDM is conducted in four distinct phases, which are depicted Figure 8 (Despa 2014, p.43). The first software development stage begins with a minimum set of needs and thereafter examines all the formation stages (excluding installation and maintenance) for these set of needs (Alshamrani, A & Bahattab 2015 p.109). This phase involves planning, and it includes an understanding of the system requirements. The second phase involves an assessment of the threats, which have possibilities of affecting the software project through the lifecycle (Krishnan 2015, p.302). The general software development process is vulnerable to threats, and these can be easily identified by understanding the problems, risks and causes of the software development process (Hijazi et al. 2014, p.213). The third spiral stage is the development phase whereby the software is produced and tested. The last stage of the spiral is the evaluation stage, and it allows software clients to appraise the product of the software project before continuation to the next spiral phase (Despa 2014, p.43).

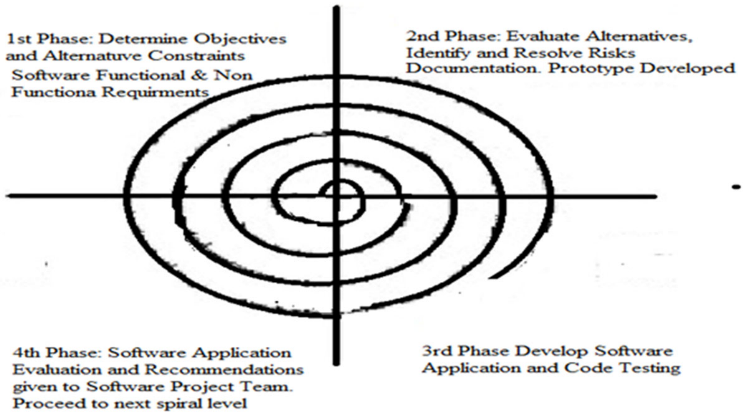

1st Phase: Determine Objectives and Alternatuve Constraints Software Functional & Non Functiona Requirments
2nd Phase: Evaluate Alternatives, Identify and Resolve Risks Documentation. Prototype Developed
3rd Phase Develop Software Application and Code Testing
4th Phase: Software Application Evaluation and Recommendations given to Software Project Team. Proceed to next spiral level

Figure 8: Spiral SDM (Despa 2014)

The Spiral SDM deliverables put together several operations for more requirements in spirals, which will continue to increase until the software is ready for deployment (Alshamrani, A & Bahattab 2015 p.107).

2.5.2. Modern Software Development Methodologies

The 4IR is expected to result in a rapid and increased growth of digitization the world over. Existing technologies will struggle to keep pace with this revolution’s demands; thus, the development of modern product management and engineering systems is needed to keep pace with the fast-changing environment. However, it is important to ensure quality across the value chain. Technology-developing organisations—including software houses—have been noted to move towards adopting agile frameworks and techniques to combine IT functions and business. Agile frameworks entail a productive platform that entails collaborative, cross-functional, highly automated, innovative, self-managed, and productive practices through techniques such as value stream mapping, Design Thinking, Lean, Agile, and DevOps. According to Brummelen and Slenders (2019), these approaches of working are proven principles and mindsets that empower organisations to realise value through improved performance, profitability and market share. Since Agile approaches as proposed by a group of professional software developers when they published the Agile Manifesto Declaration in February 2001 (Beck et al. 2001, p.4; Kiv et al. 2018) do encompass most practices of modern SMDs, this research will focus on Agile SDMS.

2.5.3. The Agile Manifesto Declaration

The professional software developers—in their response to the shortcomings of Traditional SDMs and their impact on software projects and software development success—decided on a Manifesto to improve software development (Beck et al. 2001, p.4). According to the literature, there are several key challenges in software development, these are:

- A lack of user involvement when developing the software which can lead to unclear statements of requirements and specifications, which in turn can lead to the development of wrong software; and

- The inability to cater for changes in requirements during the development of the software or requirements that only become apparent during a later stage of development. This can lead to costs exceeding the budget as well as exceeding the project duration scope, all of which can fail to meet deadlines, the inflation of budgets and the loss of work quality (Conger 2011, p.66; Dearle 2007, p.1; Jayaswal & Patton 2006; Kaur & Sengupta 2011, p.1; Krishnan 2015, p.301; Usman & Ogwueleka 2018, p.52; Weigers 2005).

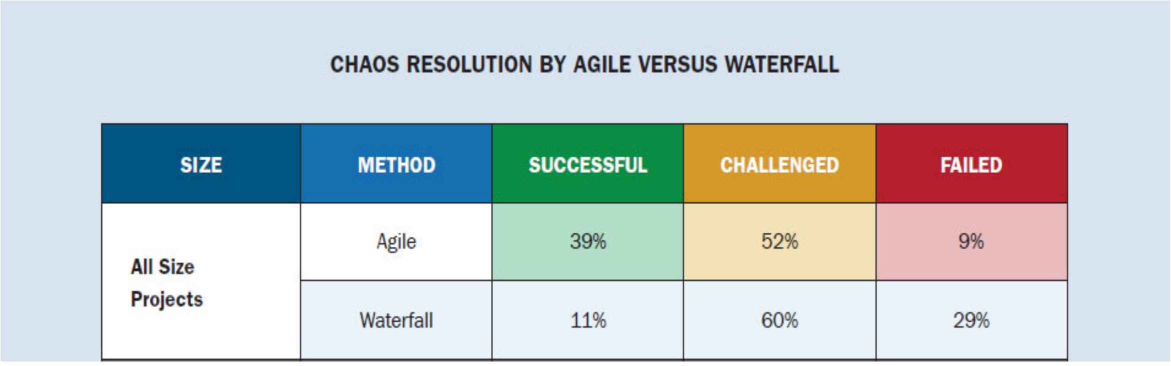
The Agile Manifesto group declared that for any agile SDM to be created or developed it should conform to the following philosophical views:

- Highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even in the late stages of software development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with preference given to the shorter timescale.
- Businesspeople and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.

- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

Ever since the launch of the Agile Manifesto Declaration, several modern SDMs have been released. The agile SDM variants include Agile Unified Process (AUP), Extreme Programming (XP), Lean Development (LD), Dynamic System Development Methodology (DSDM), ICONIX, Agile Model Driven Development (AMDD) and Adaptive Software Development (ASD) (Despa 2014, p.43; Rao, Naidu & Chakka 2011, p.38; VersionOne 2013).

The CHAOS Report 2015—an empirical analysis of the traditional SDMs versus modern SDMs—evaluated 10 000 completed software projects from 2011 to 2015 in four major areas of the world, namely: North America, Europe, Asia and the Rest of the World (see Figure 9). Their findings, which were based on several software development factors (i.e., *"OnTime, OnBudget, OnTarget, OnGoal, Value, and Satisfaction"*), revealed that agile SDM fared better than traditional SDM (Waterfall).



CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%

Figure 9: Comparison of Agile and Traditional SDMs from 2011–2015 (Standish Group 2015)

The research of the Standish Group established that projects conducted using modern SDMs were 39% successful whereas those using traditional SDM were only 11% successful (Standish Group 2015).

The Version One 2015 State of Agile Survey based on 3800 respondents the world over, informed that the Agile SDM as a modern SDM has grown increasingly because of the following factors: software productivity delivery is guaranteed early, as confirmed by 62% of respondents; managing changes in the expected requirements (confirmed by about 87% of the respondents); team productivity is 85% and project visibility 84% (VersionOne 2015).

Based on the principles enunciated by the agile manifesto, the literature reveals that software development organisations and software developers have been greatly empowered and motivated by the working concept of agile ideologies (Agile Alliance 2014). The aspect of embodied technical merit coupled with simple designs and shared creativity has created a new business value, which renders the delivery of software products at regular and short periods. More importantly, is the element of constant interactions and team build-up, which develops a self-organising team with members who own responsibility and accountability of what they are doing compared to the orthodox management style and that includes the owner of the client of the software.

Many SDMs are regarded as agile. The key software developmental stages in Agile SDMs are brainstorming (requirement gathering); analysis and modelling; design (design document and prototyping); development (coding which entails, iterations, demo and feedback); quality assurance (code testing, identify defects and resolving bugs); deployment (production and technical support); feedback from the client; and finally, after completing all the iterations, the release stage (see Figure 10).

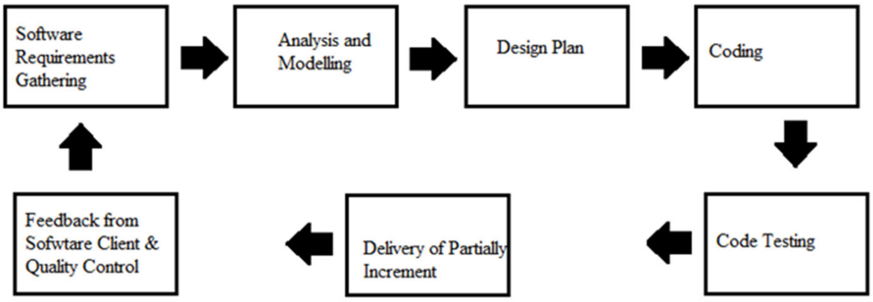

Software Requirements Gathering
Analysis and Modelling
Design Plan
Coding
Code Testing
Delivery of Partially Increment
Feedback from Sofwtare Client & Quality Control

Figure 10: Modularised phases in the agile SDM (Sharma, Sarkar & Gupta 2012, P.893).

The key philosophy of the agile SDM is the iterative, incremental developmental approach, which can be regarded as modularisation (Chan & Thong 2009; Davison & Fitzgerald 2006; Gupta, D, Sharma & Sarkar 2012; Kiv et al. 2018). The word agility, according to Tolfo et al. (2011, p.423), reflects both the ability to create and react to a change in a turbulent business IT environment. The iterative process involves developing, testing, and getting feedback from the software client. All these processes (iterative and incremental) are driven by the self-organisation philosophy where the agile software development team is always in constant interaction with one another and the client thus ensuring high-quality software development (Sani et al. 2013, p.43). The software project manager heads the software project development team, and the servant-leader facilitator heads the agile development teams. The software project development team is more into administrative, financial, technical and logistical management of the software development team. The agile development team is the implementation team which actual develops the software product and reports back to the software development team. The team also comprises stakeholders like the software clients who is part and parcel of the Agile implementation team.

The most important features of Agile SDMs that distinguish it from the traditional SDM cycle is that each stage of software development has feedback from clients, delivery is in partial increments, and the agile team is focused on the coordination of the clients and the software project team. The involvement of the software client during software application development guarantees ownership of the software application and enhances accurate and high-quality software development (Stoica, Mircea & Ghilic-Micu 2013, p.70). The concept behind modular or iterative development involves the full implementation of the software while taking into consideration the initial set of requirements and eventually culminating into an incremental release of the developed software (Abrahamsson et al. 2017, p.9). The development of software in incremental iterative stages caters for the involvement of the software client and in turn their dynamic requirements (Sani et al. 2013, p.43). This client-focused software development approach has made Agile SDMs the preferred SDM when compared to other approaches (Iivari & Iivari 2011, p.511).

Another characteristic of Agile SDMs is the software development group. The software team has the responsibility to manage their knowledge and organise the group dynamics as the development of the software progresses with the software clients who play an incredible guidance role according to Sharma, Sarkar and Gupta (2012, p.892), because they are persistently and constantly providing feedback to the software product being developed until it is fully completed. Software development tools and techniques are not predetermined, but emergent software tools and techniques are introduced during regular meetings (Mnkandla 2008, p.71) (Sharma, Sarkar & Gupta 2012, p.892). Naderuzzaman, Rabbi and Beg (2011, p.35) articulates that Agile SDMs pay particular attention to people it is communication aligned, changeable, fast, lean (aims to reduce the time and cost and enhance quality), reactive (responding properly to anticipated and unexpected alterations)) and learning (targets product enhancement in the time of and after formulation).

2.5.4. The Agile Software Development Methodologies

Some of the most commonly used Agile SDMs include variants such as Agile Unified Process (AUP), Extreme Programming (XP), Lean Development (LD), Dynamic System Development Methodology (DSDM), ICONIX, Agile Model

Driven Development (AMDD), and Adaptive Software Development (ASD) (Despa 2014, p.43; Rao, Naidu & Chakka 2011, p.38; VersionOne 2013).

2.5.4.1. Agile Unified Process (AUP)

The Agile Unified Process (AUP) is dependent on the Rational Unified Process (RUP) (Edeki, 2013, p.13). AUP employs the agile SDM conceptual approach that focuses on applying larger life-cycle software development phases and iterations within each step to deliver incremental releases over time. The roots of AUP emanate from the Rational Unified Process, and SDM developed by Rational® Software (Edeki, 2013, p.13). The key tenet of RUP was that the software development team worked closely with stakeholders, software clients, software developers, Rational's end-result teams as well as Rational's consultant company. This type of teamwork intends to ensure procedure revision, recent experiences sharing and acceptable best practices during the execution of software projects.

As indicated in Figure 11, the AUP cycle involves four distinct phases, namely: inception, elaboration, construction, and transition (Edeki, 2013, p.14). During the inception phase, the development team comprising the key stakeholders explores the possibilities of implementing the project. The AUP also determines the resources required for supporting the software project (more precisely team production) by giving each team member easy access to information with guidelines, templates, and equipment trainers for all major development activities. The elaboration phase defines the project's structure and additional resources that are required for further assessment.

The construction phase involves the development of the software application using user stories and iteratively reworked to show the growing understanding of the domain of the problem with the progression of the project (Edeki, 2013, p.15. This is followed by the software testing stage, which determines software errors and the reliability of the developed application. The last aspect involves the release of the software to the software client during the transition phase. Software client's experience on the use of the software application facilitates final changes or updates to the released software application; for example, changes such as iterative improvements such as modelling, implementation, testing, and management activities are still possible. The key development element of AUP is that it prevents wastage of resources and limits unexpected development costs.

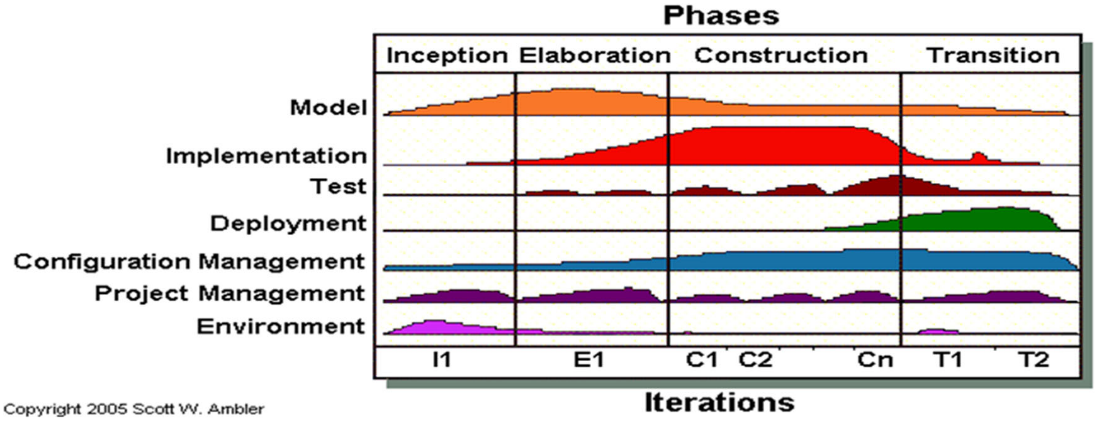


Figure 11:Agile Unified Process (AUP) process model (Edeki, C 2013, p.14)

2.5.4.2. Extreme Programming (XP)

Extreme Programming (XP) is a typical agile SDM that organises people to create high-quality software in a more productive manner using the five phases depicted in Figure 12 (Geambaşu et al. 2011, p.485). These phases or stages are explorations, iteration planning, iteration sessions, customer approval and small releases phases. Extreme Programming attempts to enable cost reduction through adjustments and these are done to meet the software requirements expressed by the software client for the period of system development (Munassar & Govardhan 2010, p.97).

During the exploration phase, the software client and developer explore all possible software requirements at that moment, and an initial set of software requirements is generated and baseline software architecture is produced. A major issue observed in XP programming is the need for a software client who is willing to spend time in the office until a project ends. The team (software developer and client) builds focus on developing relevant software for specific business processes. This is done through an initial team build-up exercise with the software client so that collective ownership is maintained throughout the development process. According to Amir et al. (2013, p.80), the strength of XP programming is communication, simplicity, feedback, respect and courage between the client and software development team.

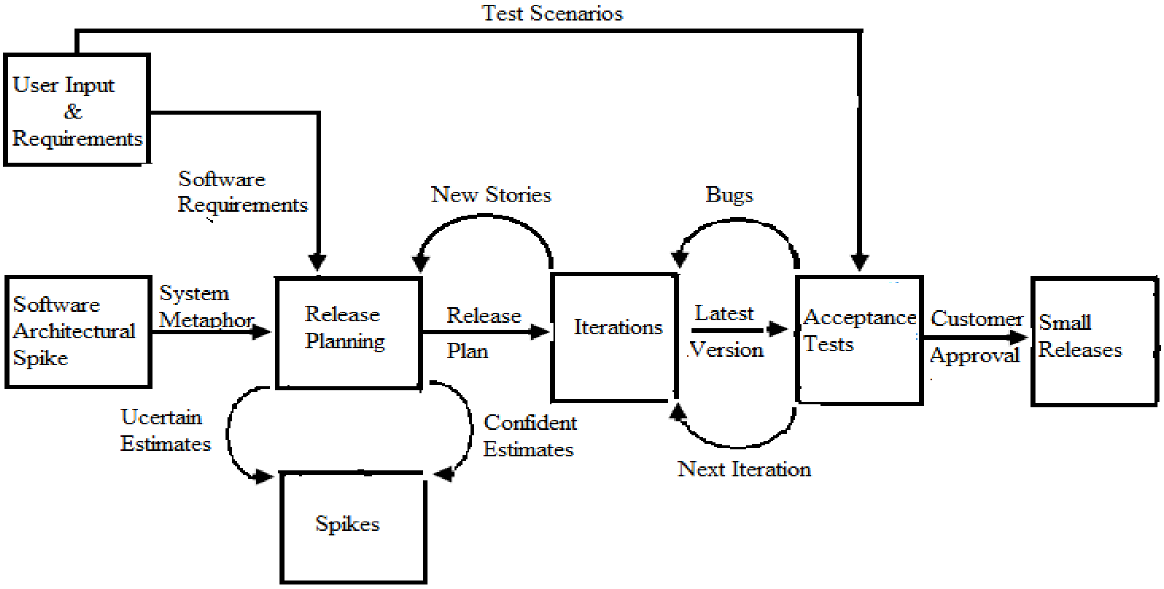


Figure 12:Extreme Programming Agile SDM (Geambaşu et al. 2011)

A set of iteration meetings are organised based on a release plan for developing the software application. The iteration process involves the development of the software application in pair programming and conducting acceptance tests. The concept of pair programming according to Amir et al. (2013, p.80) and Mnkandla (2008, p.71) is to improve the quality of the software application being developed and ensure technical mastery of programming. Amir et al. (2013, p.80) described the philosophy of XP programming as being focused on steady progress team meetings at a frequency rate of two or more a day and project-driven development. If the customer is happy, about the developed or attained requirement, the software is released for that developed functionality and a second iteration for programming is resumed based on the iteration planning.

Extreme Programming is an agile SDM that is widely accepted in the industry (Amir et al. 2013, p.80). Mnkandla (2008, p.70) seems to be also of the view that XP is the most adopted and experienced agile SDM. Extreme Programming (XP) considers the software client and provides high-quality applications in a more favourable way (Geambaşu et al. 2011, p.485).

2.5.4.3. Lean Development (LD)

The Lean Development (LD) philosophy is based on the Lean Thinking approach by Lean Production that allows clients to take as long as required to decide on their specific needs and when it is needed (Mnkandla 2008, p.68). The Lean Development SDM (see Figure 13) adopts strict leadership for software development that is risk management-oriented. It does not allow for any errors while attempting to produce the most correct and precise initial software application release and it is a methodology that tries to avoid waste (Woods 2010).

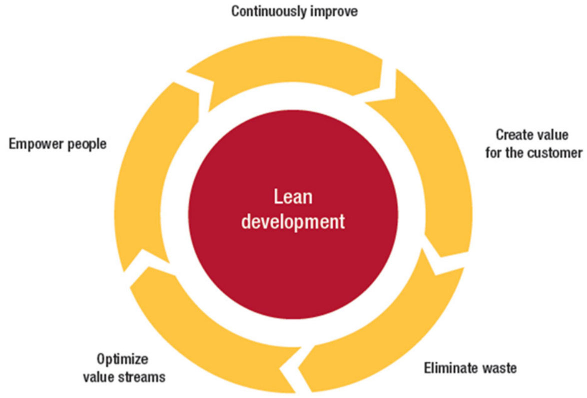


Figure 13: Lean Agile SDM (Ebert, Abrahamsson & Oza 2012, p.22)

In addition, LD limits time, allows changes to be done, ensures the use of noted changes to be made with one–a third of the workforce, a third of the production time, a third of the investment in equipment and procedures, and a third of the effort to adjust to a current market atmosphere. However, this type of software development approach needs the support of the top management of the organisation. Policies that encourage LD needs to be accepted to bring about changes in an organisation (Mnkandla 2008, p.68).

The key stage of the LD SDM is the conceptualisation stage when software requirements are elicited from the software client. During the conceptualisation stage, factors such as what value to add and how to eliminate processes that could affect the software delivery need to be considered. The product must be delivered at the correct time, with precise functionality and within a given budget scope.

The next stage is developing a prototype based on the baseline software requirements. The developed software application is then reviewed by top-level senior developers and management to ensure that the required software functionalities are attained, and any software bugs are removed or fixed. The prototype is then presented to the software client for acceptance. The first software release is deployed, and feedback is used for the development of the next iteration (see Figure 13). The key advantages of the LD SDM are to: add value in developing high-quality software, which meets the required business process functionalities using limited resources; remove any waste process when developing the software application; ensure customer satisfaction, and retain software developers (employees) satisfaction.

2.5.4.4. Dynamic System Development Methodology (DSDM)

The Dynamic System Development Methodology (DSDM) philosophy is premised on the software developers and programmers constituting a development team that develops software in an incremental approach at the earliest possible time. The development is iterative and evolves until the software is complete. The focus is to prioritise critical features valuable to the business processes’ functionality. Craddock et al. (2012, p.2) think that DSDM is an established structure for agile project administration and presentation. It assists to develop the end product rapidly and successfully. The DSDM focuses on systematic objectives and incremental presentation of actual business advantages while controlling time, value, threat and quality (Anwer et al. 2017, p.1; Craddock et al. 2012, p.2).

The main development features of the DSDM are the feasibility study of transforming or automating the business process (see Figure 14). The feasibility study is followed by an agreement between the software developer and client and a functional model of the software application is created and a software prototype is developed.

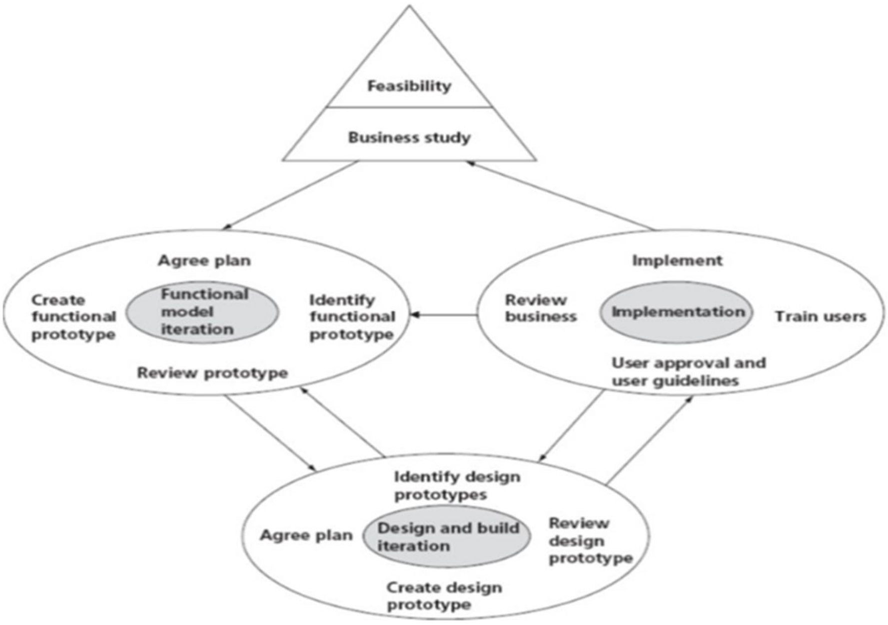

Feasibility
Business study
Agree plan
Create functional prototype
Functional model iteration
Identify functional prototype
Review prototype
Implement
Review business
Implementation
Train users
User approval and user guidelines
Identify design prototypes
Agree plan
Design and build iteration
Review design prototype
Create design prototype

Figure 14: DSDM Agile SDM (Zafar, Nazir & Abbas 2017)

The software prototype, which functions as early delivery, serves as a platform for enhancing the design of the software and progressing towards a better implementation of the software application through iterations amongst the functional model, software design and implementation stages.

2.5.4.5. Scrum

According to Denning (2015), Scrum is an agile SDM that comprises a set of well-organised activities for change that deals with work conducted in self-organising groups. It is iterative to enhance rapid software development. The software application development is driven by software client participation (Denning 2015).

Work objectives are stated and prioritised before starting each sprint cycle and a product backlog is completed. During sprint meeting sessions, tasks, as well as the resources needed per team, are allocated. Specific and short deadlines (24hr to 30 days) are set for attaining the defined tasks (see Figure 15). The development team works on the tasks and sprint review sessions are conducted daily to determine challenges and progress attained. If there are any deliverables, they are shipped to the software client.

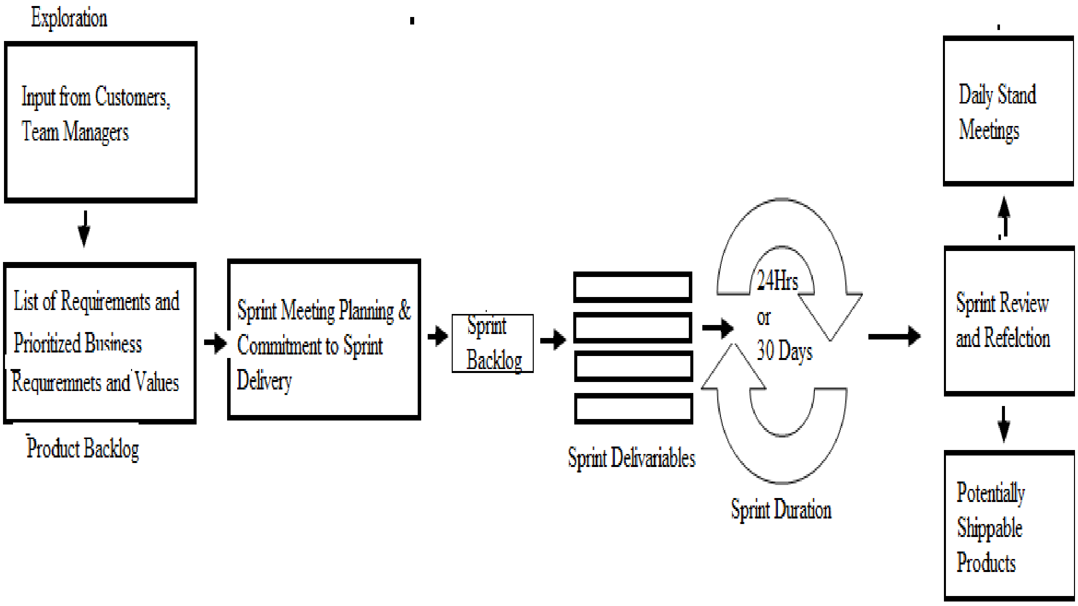

Exploration
Input from Customers, Team Managers
List of Requirements and Prioritized Business Requremnets and Values
Product Backlog
Sprint Meeting Planning & Commitment to Sprint Delivery
Sprint Backlog
Sprint Delivariables
24Hrs or 30 Days
Sprint Duration
Daily Stand Meetings
Sprint Review and Refelction
Potentially Shippable Products

Figure 15: SCRUM Agile SDM (Denning 2015)

The lessons learnt from the iterative sprint review sessions are recorded for future iterations. In addition, management takes overall charge of removing risks and obstacles in the team's work. This approach reflects management's commitment to the software development process and avoids management having to wait for a faulty product at the end of the development process. The production group prioritises progress reports to the client to ensure the delivery of correct software to the client instead of the manager. The client's preferences are strategically examined and included in the SCRUM team plans. The development team itself estimates the scheduling of work and what is needed to be covered rather than the manager.

SCRUM is a lightweight SDM that is appropriate for handling and monitoring the development of software applications in a changing environment. This SDM is teamwork-based and manages the software application development through constant meetings where obstacles and challenges are noted early thus avoiding risks. It ensures maximum productivity by setting realistic goals within short periods called sprints. Scrum is used by popular software development houses such as Microsoft, Yahoo, Google, Amazon, Siemens, Nokia, Phillips, and many other famous organisations.

2.5.4.6. ICONIX

The ICONIX process is use case-driven (Morteza et al. 2011, p.278). The use case diagrams are generated with the help of the software client and a preliminary design is coded. It initiates the iteration process between the developers and the software clients as progress is made towards a detailed design using domain models. The ICONIX process can be considered an SDM located between the Rational Unified Process (RUP) and Extreme Programming SDMs (Rosenberg 2001, p.12). This SDM's philosophy was developed as a hybrid of the popular Object Oriented modelling skills, Booch's Object Modelling Technique (OMT), and Object Oriented Software Engineering (OOSE) (Rosenberg 2001, p.12).

The ICONIX process puts more significance in obtaining the requirements first before implementing the code. As shown in Figure 16, the process of requirements elicitation and confirmation is implemented by modelling user requirements through illustrative graphics and use-case diagrams. This is done to ensure that any vagueness in the requirements is unearthed and clarified.

ICONIX is an agile SDM used chiefly as a tool to quickly implement an initial prototype. Iterations are used to refine the models before the application is developed (Putra et al. 2020, p.1).

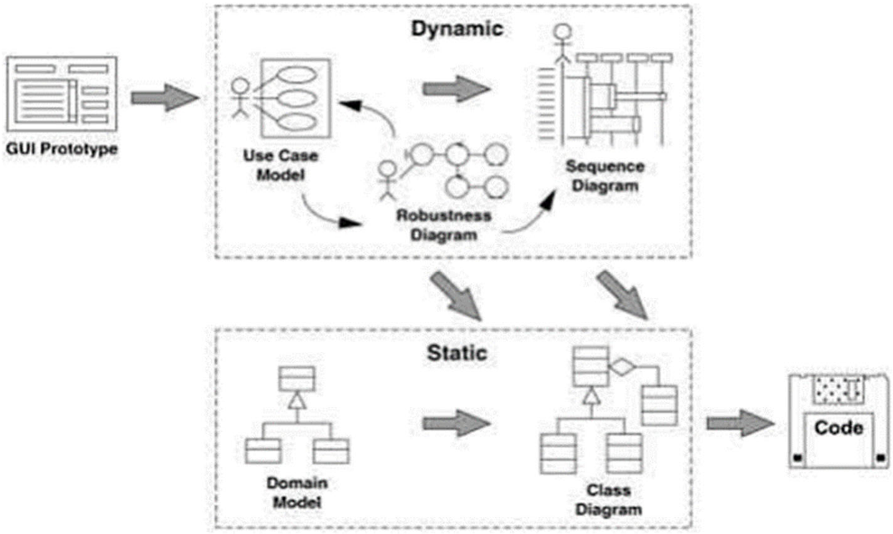


Figure 16: The ICONIX Process (Rosenberg 2001)

2.5.4.7. Agile Model Driven Development (AMDD)

The Agile Model Driven Development (AMDD) considers the functioning software as the main objective for modelling (see Figure 17).

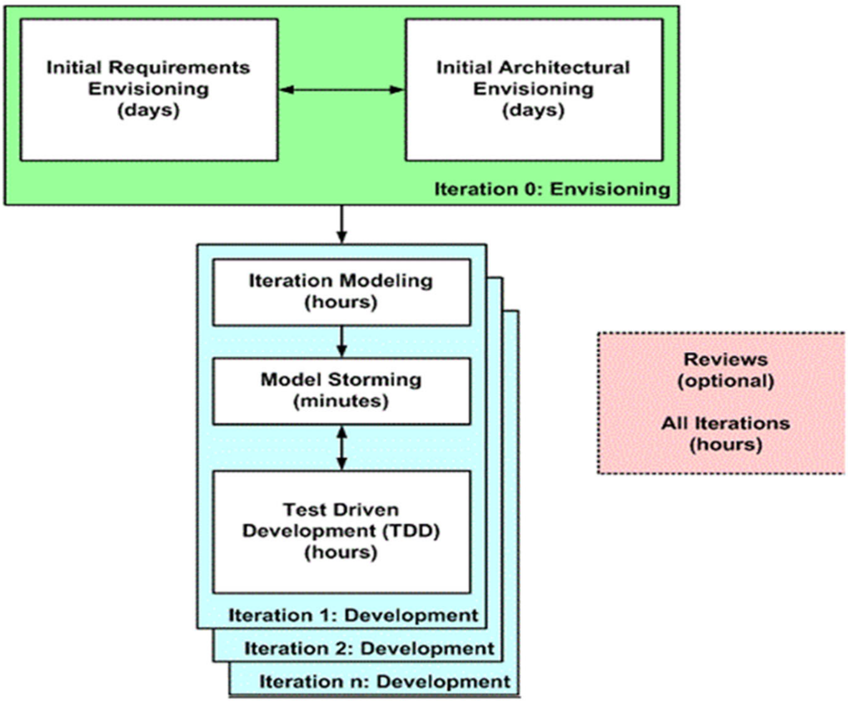


Figure 17: AMDD (Ambler 2012)

The golden rule is that the number of models and documentation produced should be limited but should have a firm basis to encourage serious planning and modelling of the software (Mnkandla 2008, p.58).

The philosophy of AMDD is that documentation and planning—done by the software development team—should not distort the prime inputs and expectations of the software client, that is, the primary stakeholder. Agile Model Driven Development SDM has been noted by general literature review for the following important benefits; ensuring quality development of the product through increasing transparency, team empowerment and ownership of the development process by all stakeholders (Ameller et al. 2019; Cabot 2020; Siddiqui 2019). The works of Alfraihi et al. (2018) mention that AMDD's other great contribution is the ability to minimise risk in the development process and therefore develop the product within the budget and in time of the project of these are the most critical factors for a successful project management

2.5.4.8. *Adaptive Software Development (ASD)*

Highsmith (2002, p.7) defined the Adaptive Software Development (ASD) philosophy in terms of five key objectives of software management. These objectives include "*adaptive culture*", which views companies as complicated adaptive systems and formulates emergent order from a group of people. The second objective is "*adaptive frameworks*", which provide several structures or models to assist a company to engage in adaptive principles. The third objective, "*adaptive collaboration*", enables collaboration (i.e., the association of individuals with similar and sometimes dissimilar interests to corporately formulate and innovate) and becomes the company drive for coming up with emergent answers to product development challenges. Collaboration is handled concerning interpersonal, cultural, and structural associations. The "*adaptive scale*" being the fourth objective, provides an alternative for companies to employ the approach on bigger projects. The fifth and last objective, "*adaptive management*", exchanges the culture of command-control management with a flexible supervision approach that allows for many people in decision-making and empowerment. This means that "*leadership*" replaces "*command*" and "*collaboration*" replaces "*control*" Highsmith (2002, p.7).

One fundamental attractive feature of this agile SDM is the early delivery of software products with prime requirements implemented. Important requirements are implemented early during the time of development of the application and a review is conducted with software clients during each iteration or cycle of software development. This type of approach is referred to by Sharma, Sarkar and Gupta (2012, p.892) as a modularisation development approach. The iteration caters for consultation with software clients to ensure the development of the correct product and to review any risks.

2.6. Adoption and Use of Modern Software Development Methodologies

The development of software should be directly linked to a proper SDM and this has a direct bearing on the success of developed software (Parveen, Khan &

Sadiq 2014, p.534). The literature indicates that there are several reasons for the lack of the adoption of SDMs, namely: developers simply ignore newly introduced SDMs; SDMs treat the process of systems development as an orderly rational process, and SDMs are assumed to be universally applicable, however, they should be adjusted across different development situations (Agile Alliance 2014; Ahmad et al. 2018; Chan & Thong 2009; Edeki, C. 2015; Roses, Windmöller & Carmo 2016; Totten 2017; Tsoy & Staples 2020). Therefore, acceptance, adoption, and use of SDMs amongst many organisations and practitioners are still a problem.

SAS—a renowned software development organisation for software and business intelligence applications, which had more than 13 000 workers in 55 countries assisting clients in 135 countries at the time, conducted research to determine the importance of agile SDM usage (Al-Kautsar et al. 2013; Arthur 2013, p.2). Their research findings revealed that Agile SDMs were adopted by most organisations and that the most used agile SDM variant was Scrum (Arthur 2013, p.19).

An evaluation by Ambler (2013) to measure the success rate when using Agile SDMs in developing software applications revealed that the adoption and use of Agile SDMs were gaining momentum due to metrics such as timely completion of the software application; developing software project within the limit of the project budget; staying within the prescribed project scope; and the ability to fulfil the client's requirements. In 2013, VersionOne (2013) listed the following top three reasons for adopting Agile SDMs: accelerated delivery of the product to market; increased productivity, and easy management of ever-changing user functional requirements and priorities.

Bhadoriya, Mishra and Malviya (2014, p.1656) collected data from a software firm incubation centre on the adoption and usage of Agile SDMs to develop software for various multinational companies. The company had 30 teams with each comprised of ten members. The main function of each team was to develop software using hypertext pre-processor (PHP) or Java based on the rapid changing requirements of their clients. The research concluded that Agile SDMs were the most used in small- and medium-sized business applications. They found that Agile SDMs are suitable for developing software applications, which have frequent changing requirements, and that the iteration and repetition features catered for this purpose. However, several drawbacks were noted for using Agile SDMs, namely, the rate of software fluctuations was huge and affected the scheduling of the delivery of results. They concluded that Agile SDMs are suitable for software project teams with experienced and senior developers in the team (Bhadoriya, Mishra and Malviya (2014, p.1656).

Vijayasarathy and Butler (2015, p.86), in their study titled "*Choice of Software Development Methodologies, Do Organizational, Project, and Team Characteristics Matter*" surveyed 11 countries in Europe, America and Asia. The counties included the United States, India, China, Hong Kong, the UK, Germany, Romania, Sri Lanka, France, Singapore, and Saudi Arabia. The study found that software companies used SDMs in the following manner: a combination or hybrid SDMs (45.3%), Agile SDMs (33.1%), traditional SDMs (13.8%) and iterative SDMs (7.7%) (Vijayasarathy & Butler 2015, p.90). This study considered iterative SDMs and traditional SDMs in the same category and their combined usage score of 21.5% is still lower than the Agile SDMs adoption and usage. The 153 surveyed participants had different occupational roles, namely: project managers; team leaders; analysts; architects or designers; and testers.

Roses, Windmöller and Carmo (2016, p.439), conducted a study in Brazil to determine the adoption of SDMs within a Brazilian public financial institution (the Bank). Their study was focused on determining which type of SDM can be adopted based on the Bank's perception of behaviour, practice, and mental models. The study scoped three critical areas related to SDM selection, namely: being knowledgeable; administration; and processes, and considered these as factors to consider when selecting a traditional or agile SDM. The knowledge perspective consisted of acquired knowledge, perceived knowledge, and knowledge to manage and handle the phases of the software development. The administration factor was concerned with the nature of the organisation's administration orientation towards managing a traditional SDM or agile SDM. The software process perspective was considered in terms of how many people should be involved in the development, must have application domain specialization, and their capacity for innovation. Their research concluded that there was a slight favourability towards the adoption of Agile SDMs practices in the institution (Roses, Windmöller & Carmo 2016, p.439) Licorish et al. (2016, p.369) conducted a study involving three countries Brazil, Finland, and New Zealand. They aimed to understand the adoption and use of SDM and practices. One hundred and eighty-four practitioners were involved in the study. One of the key research study questions was:

> *"What software development methods and practices do practitioners adopt for their development portfolio?"* (Licorish et al. 2016, p.369).

The response elicited from this question indicated that many software developers were using traditional SDMs, namely: the Waterfall, RAD and Spiral SDMs. The study outcomes contrasted with the Versionone 2014 study, which indicated that 94% of software developers were using Agile SDMs.

Abdalhamid and Mishra (2017, p.817) conducted a systematic literature review on the adoption of Agile SDMs and practices in software development organisations. The keywords used in their research study to search for literature from several electronic databases were *agile*, *adopting*, and *software development organizations*. The AND operator was used to join *adopting* and *software development organizations* as well. The following five electronic databases (DB) were searched with these keywords: ACM Digital Library IEEE Xplorer, Springer, Google Scholar and the Web of Science. The questions for the research study were: *What motivates organisations to adopt Agile SDMs? Are*

Agile SDMs limitations in practice? Are there any guidelines for adopting Agile SDMs? and *What are the organisation's culture on Agile SDMs adoption and use*? According to the literature, the adoption of agile SDM is vital for organisations to deliver software to the market on time, it is the best approach to elicit software requirements, and it is inexpensive. Furthermore, the organisation's culture has a bearing on the successful implementation of Agile SDMs in their institutions. Hobbs and Petit (2017, p.3) reported:

> *"In recent years, agile methods have become highly prevalent in the software industry, and today it is one of the hottest topics in project management"* (Hobbs & Petit 2017, p.3)

Tegegne, Seppänen and Ahmad (2019) conducted a systematic literature review of software start-ups. Software start-ups are an emerging approach for the development of cutting-edge software products under highly uncertain conditions, overcoming fast-growing markets under multiple influences (Tegegne, Seppänen & Ahmad 2019, p.1). Software start-up entities such as Facebook, WhatsApp, Twitter, and others have produced software products that have gained popularity in use. The research study of Tegegne, Seppänen and Ahmad (2019, p.1) investigated software start-ups to identify and determine what SDMs were adopted and what practices were used by these entities. The study validated the findings with 14 real-life software start-ups in Finland, Italy and Norway and it identified 37 relevant primary studies out of 1982 papers. The researchers found that Agile SDMs such as Lean were the methodologies most commonly used by software start-ups due to their flexible nature and easy tailoring (Tegegne, Seppänen & Ahmad 2019, p.1). The selection and adoption of an appropriate methodology is a difficult process and requires deep analysis of the company itself; the nature of the company; the team size and behaviour; and the type and stage of the product concerned. Agile SDM adoption and use depended on the experience of the owners or/and project managers, the maturity level of the start-ups, the team size, and the availability of resources.

2.7. Drivers of Usage and Adoption of Agile Software Development Methodologies

According to Highsmith and Cockburn (2001, p.120),

> *"…what is new about agile methods is not the practices they use, but their recognition of people as the primary drivers of project success, coupled with an intense focus on effectiveness and manoeuvrability. This yields a new combination of values and principles that define an agile world view."* (Highsmith & Cockburn 2001, p.120)

The traits discussed in the sub-sections that follow are noted as main factors for adopting and use of modern software development, Agile SDMs and have been regarded as success factors in various research studies.

2.7.1. People Oriented

Agile SDMs places value on people who develop the software, namely: software practitioners; customers; managers; administrators; software clients; directors; and chief executive officers of software development houses. They are regarded as vital stakeholders for each phase of software development. According to Cockburn and Highsmith (2001, p.131),

> *"The most important implication to managers working in an agile manner is that it places more emphasis on people factors in the project: amicability, talent, skill, and communication".* (Cockburn & Highsmith 2001, p.131)

2.7.2. Software Prototypes

Agile SDMs are mostly used as a tool to quickly implement a prototype to realise the basis for iterations when developing finer models for application. Prototype development is highly dynamic until a steady state is achieved, and it is for this reason that the software design and implementation are undertaken amongst software developers and programmers.

2.7.3. Iterations of Software Development

Iterations are engaged as software development evolves until and up to completion, and the focus is to prioritise critical features valuable to the business processes' functionality. It is known that system requirements in an agile methodology are decomposed into smaller iterative segments and each iterative segment needs to be completed by team members within the iteration period (Al-Kautsar et al. 2013).

2.7.4. Adaptive Interaction

Change is second nature to stakeholders of Agile SDMs. They embrace change as a vital part of software development because change facilitates continuous software development and allows consideration of new requirements that were not visible before. Through the adaptive and iterative process, these functional features become "*visible"*. Gagel (2017) cited Goldman et al 1995 who described organizational agility or adaptive interaction as:

> "*… a comprehensive response to the business challenges of profiting from rapidly changing, continually fragmenting, global markets for high-quality, high-performance, customer-configured goods and services*." Gagel (2017, p.9)

2.7.5. Adaptive management

Agile SDMs, through the organisation, practice adaptive management. Adaptive management replaces the culture of command-control management with a flexible supervision approach that empowers many people in decision-making. In a study that considered the social factors affecting the adoption and use of Agile SDMs, 271 Pakistani software professionals were interviewed face-to-face. The interviewed software professionals included: portfolio/program/project managers; Scrum masters; and product owners. Their findings indicated that visionary leadership was one of the key success factors for the adoption and use of Agile SDMs (Riaz, Mahboob & Buriro 2018, p.94).

2.7.6. Conformance to Actual

The nature of Agile SDMs is to embrace change and be adaptive, and this, therefore, leads to conformance to the actual functional requirements and values during software development. There is no other way to conform to reality and to derive 100% functionality than through constant interaction with software clients, software developers and organisation management. Agile projects are not *plan controlled* but are *actioned controlled* in line with the requirements of the business value. Agile SDMs consider issues such as the functioning software as the main objective of modelling hence confirmation to actuality (Ahimbisibwe, Daellenbach & Cavana 2017, p.401).

2.7.7. Practical and Flexibility Planning

Planning is essential but planning for Agile SDMs is detailed and strategised over a few weeks, very rough plans for a few months, and extremely crude plans beyond that (Al-Zewairi, Malek et al. 2017, p.75).

2.7.8. Communication and Collaboration

Agile SDMs require client feedback regularly. The client collaborates and work with the software development team frequently and through the software development stages. Collaboration is conducted by interpersonal, cultural, and structural association levels. According to Al-Zewairi, Malek et al. (2017, p.76), collaboration encourages discussion and knowledge exchange at all levels of communication, that is. interpersonal and intrapersonal levels. The importance of collaboration also enables access to various levels of communication within the software development house or within the profile of the software client. As Martin Fowler describes:

> *"Agile teams cannot exist with occasional communication. They need continuous access to business expertise"* (Fowler 2018, p.193)

2.7.9. Self-organising teams

An agile team is made up of dynamic and small self-organising teams Kropp et al. (2018, p.112). Each team is allocated tasks with short and constant meetings in which results for the assigned task are reported. The teams are self-organising in the sense that they own the responsibility of the tasks to execute and are committed to its success. Agility is a key functional characteristic of the self-organising team. Agile teams tend to be self-organising and are provided with a high degree of freedom by their senior management to organise and commit to their team's goals (Al Kautsar et al. 2013, p.6). This type of working and approach is not common in traditional SDMs.

2.7.10. Deliver on time and ensure quality

According to (Brummelen & Slenders 2019), the noted advantage of Agile SDMs is the ability to deliver on time, ensure the quality of the product, adopting the best software engineering practices, have good team cooperation, and understanding of matters at hand. Agile development ensures customer satisfaction and improved relationships.

> *Modern delivery approaches for software development focus on the entire value chain, combining a mixture of Design Thinking, Lean, Agile and DevOps practices* (Brummelen & Slenders 2019, p.12)

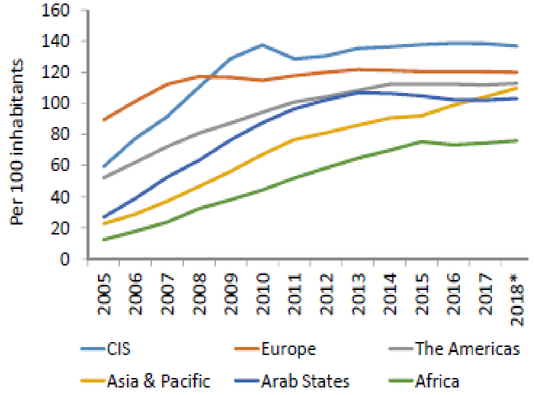
2.7.11. Detection of possible issues

One critical element of software development is detecting problems inherent in the developed software from the onset of coding of the software, rather than waiting until millions of lines of code have been written and errors crop up at the end of coding. This *'to the right'* phenomenon—typical of traditional SDMs—leads to poor quality software and expensive fixes. Instead, modern SDMs have ushered in the *"shift to the left"* concept—where software is delivered in bits and pieces and testing and discovery of issues is immediate (Brummelen & Slenders 2019).

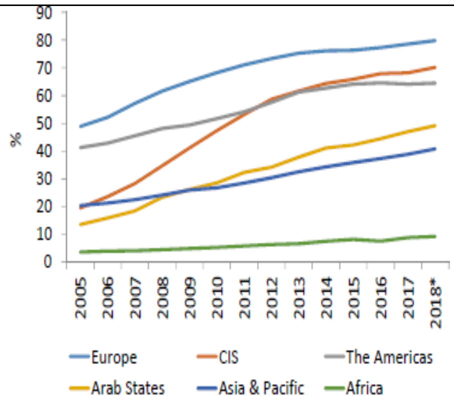
2.8. ICT Development Trends and Patterns in Africa

The extension of the broadband undersea fibre optic cables to Africa has vastly enhanced access to the Internet and mobile technology and has as such leapfrogged the development and use of information and communication technologies (ICTs) (Njikam et al. 2019, p.577; Song , S 2019). This remarkable development has seen ICT spatial infrastructure deployed as satellites, inland fibre cables, online databases, mobile technology and the Internet. This has promoted the use of social media platforms, electronic commerce, mobile commerce, electronic learning, video conferencing, telemedicine, and many others. The structures have altered the African community's behavioural approach in terms of communicating and collaboration, social interaction, doing business as well as political engagements.

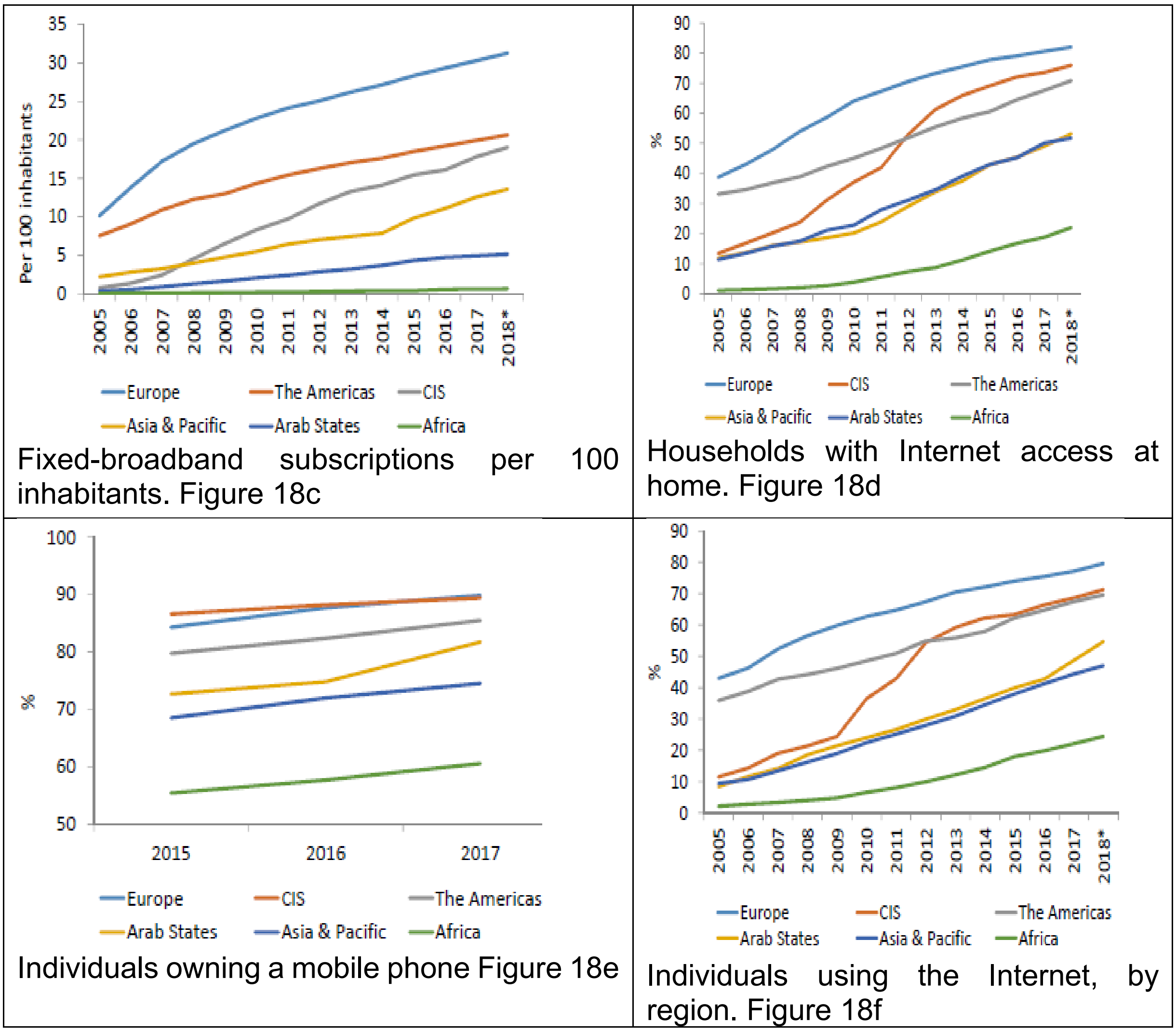
The empirical evaluation of Africa's position is reported best by the 2018 ITU Report, which annually gives the status of ICT developments the world over (see Figure 18 a-f).



Mobile-cellular subscriptions per 100 inhabitants. Figure 18a



Percentage of households with a computer. Figure 18b



Fixed-broadband subscriptions per 100 inhabitants. Figure 18c

Households with Internet access at home. Figure 18d

Individuals owning a mobile phone Figure 18e

Individuals using the Internet, by region. Figure 18f

Figure 18: ICTs in Africa by 2018 ITU Report (International Telecommunication Union 2018)

Mobile Internet access can thus easily be integrated into IoT applications and as well as Cloud computing. Very few people in Africa—about 15 people per 100—own a computer. Computer ownership has been overtaken by smartphones—these currently have almost similar functions to a computer and is more portable. This level of ICT access and ownership shows the impact of ICTs at an individual level and is also a sign of its utilisation. The fast-growing information society is characterized by the development of a digital economy and a need for software to support this trend.

2.9. ICT Development Trends and Patterns in Botswana.

Botswana is a member state of the Southern African Development Community (SADC) and the country is geographically located in the middle of Southern Africa, situated between South Africa, Namibia, Zambia, and Zimbabwe. Botswana has historically enjoyed strong and stable growth since its independence, with a sizable fiscal buffer and prudent policies playing a key role in shielding the economy. It is regarded as one of the fastest-growing economies in Africa. Suffice to say that Botswana, is now rated as a middle-income country (Mogalakwe & Nyamnjoh 2017, p.2).

Botswana is also highly rated for its ICT policies and regulatory procedures. The policy position on ICT implementation has allowed various services to be introduced into the ICT market, thus offering new services, lowering prices for ICT services and introducing competition (Kerretts-Makua 2014, p.28)

Infrastructurally, Botswana has implemented numerous ICT activities to support its strategic goal towards transforming the country from a resource-based economy to a knowledge-based economy. Botswana's progress towards this is based on its National Strategic Plan and ICT policies:

Telecommunication facilities constitute a key infrastructure for any ICT infrastructure to be set up. The telecommunications infrastructure is responsible for data and information transmission using various technologies. Botswana Telecommunications Corporation (BTC) is mandated with the administration of broadband connections and fixed connections. Botswana Fibre Networks (BoFiNet) has been delegated to administer the East Africa Submarine System (EASSy) and West African Cable System (WACS) undersea fibre cables and provide high-capacity digital data transmission. The two undersea fibre cable networks interconnect African countries with the rest of the world. The West African Cable System (WACS) provides additional broadband and connects to Europe (Setimela, M, K. 2018). Internet coverage in the country is largely attributed to these two international fibre optic cables and the earth satellite connection administered by BTC (Isaac et al. 2017, p.737).

Table 1 shows key parameters of ICT adoption and utilization in Botswana and their rankings or comparison with Africa and the world. The first row shows ownership of fixed telephone line subscription with the record showing about 13 people per 100 inhabitants of the world have fixed telephone lines and Botswana has 6 people per 100 inhabitants which is almost a ratio of 1:2 when compared with the world ownership statistics. For mobile cellular subscription, Botswana overrides the world ownership statistics as it shows that for every 100 inhabitants there is a subscription of 141 mobile cellular ownership, meaning that half of the population in Botswana has at least two cell phones.

Table 1:Key indicators for Botswana (International Telecommunication Union 2018)

Indicators for Botswana 2017	**Botswana**	**Africa**	**World**
Fixed-telephone sub. per 100 inhabitants.	6.2	0.9	13.0
Mobile-cellular sub. per 100 inhabitants.	141.4	74.4	103.6
Active mobile-broadband sub. per 100 inhabitants.	66.9	24.8	61.9
3G coverage (% of the population)	84.0	62.7	87.9
LTE/WiMAX coverage (% of population)	65.0	28.4	76.3
Individuals using the Internet (%)	*41.4*	22.1	48.6
Households with a computer (%)	*31.2*	8.9	47.1
Households with Internet access (%)	*45.7*	19.4	54.7
International bandwidth per Internet user (kbit/s)	26.5	11.2	76.6
Fixed-broadband sub. per 100 inhabitants	2.1	0.6	13.6
Fixed-broadband sub. by speed tiers, % distribution			
-256 kbit/s to 2 Mbit/s	84.0	38.7	4.2
-2 to 10 Mbit/s	15.5	37.2	13.2
-equal to or above 10 Mbit/s	0.4	24.1	82.6

This is a very high subscription rate. Africa mobile cell phone subscription stands at 74.4 per 100 inhabitants. Active mobile broadband utilisation per 100 inhabitants is also more than that of Africa. Botswana uses the latest mobile communication technologies (i.e., 3G to LTE) compared to the rest of Africa and is almost comparable to the world. In addition, utilisation of the Internet is also comparable to that of the world. In a nutshell, Botswana's ICT infrastructure, utilisation and appropriateness is comparable to some developed countries and is much greater compared to that of a majority of its peer countries in Africa—it is ranked 6th in Africa and 105th in the world (International Telecommunication Union 2018). Mobile communication and related mobile infrastructure is run by three service providers namely: Bemobile, Mascom and Orange. They all run 2G to 4G and Wi-Fi mobile broadband technologies that support the ever-growing mobile commerce in the country (Setimela, MK 2018).

The government of Botswana in partnership with the local Botswana Software Development Industry (BSDI) have developed a range of software applications in their endeavour to develop a digital economy. These are software applications are:
e-Legislation, e-Government, e-Passport, 3-Education, e-Health, e-Commerce, e-Agriculture and e-Tourism. According to the Ministry of Transport and Communication, e-Government services shall:

> *"...help raise public sector service quality to higher levels, allowing citizens to access government information and services (anywhere, anytime)"* (Setimela, M, K. 2018, p.21).

According to Ayalew and Motlhala (2014, p.121), 90% of software development organisations around the globe in 2014 were companies that employed 50 employees or less. Similarly, most of the software developers in Botswana are small companies. The surveyed companies felt that SMDs are only applicable to big software companies (Ayalew & Motlhala 2014, p.121). In addition, Ayalew and Motlhala (2014, p.122) have pointed out that, in 2014, BSD companies were using untested software engineering methods and process applications were mainly ad-hoc (Ayalew & Motlhala 2014, p.135).

Several Botswana software projects have failed as reported by Mphale , Okike and Mogotlhwane (2016, p.966), and the following refer to three such projects:

- The MALEPA system, a web-based examination system designed to process and release the Botswana General Certificate of Secondary Education (BGCSE), Junior Certificate Examinations (JCE) and Primary School Living Examination (PSLE). The MALEPA system's failure is attributed to the fact that appropriate technical procedures were not implemented (Mphale , Okike & Mogotlhwane 2016, p.966).
- The Livestock Information Technology System Agricultural Project (LITS) is yet another software project that failed owing to customer requirements not being understood and poor project management (Mphale , Okike & Mogotlhwane 2016, p.966)
- The Botswana Telecommunication Communication (BTC) billing system is yet another software project that collapsed due to improper project planning techniques and failure to adapt to business change factors (Mokgoabone 2004). BTC lost their market reputation since customers failed to use the billing system, which caused irregular statements and inaccurate information. The system cost BTC a budget of P60 million to implement (Mokgoabone 2004).

The proliferation of ICTs coupled with emerging industrial technologies witnessed a demand and growth in use of ICTs by the society and the economy in Botswana. For instance, use of social media applications by the society in Botswana was due to internet access to Europe and western countries through undersea fibre cables, this increased mobile device ownership and subscription to mobile services (International Telecommunication Union 2018; Njikam et al. 2019). In turn this also prompted further usage of mobile and electronic commerce applications which are cloud based or served based. Mobile banking, use of electronic government services that included e-Legislation, e-Passport, e-

Education, e-Health, e-Agriculture and e-Tourism have been launched (MmegiOnline 2021).

However, in coping with demand and need for developing effective software application, it was noted in some cases that the BSDI was facing challenges of developing software applications which met the required or expected functionalities, developing the software applications in time and within budget (Mokgoabone 2004; Mphale , Okike & Mogotlhwane 2016). Typical software applications developed by the BSDI and had challenges include; The MALEPA web-based examination system designed to process and release the Botswana General Certificate of Secondary Education (BGCSE), Junior Certificate Examinations (JCE) and Primary School Living Examination (PSLE), Mphale , Okike and Mogotlhwane (2016, p.966); Livestock Information Technology System Agricultural Project (LITS), Mphale , Okike and Mogotlhwane (2016, p.966); Botswana Telecommunication Communication (BTC) billing system, Mokgoabone (2004). The latest software development failure was the Department of Tertiary Education Financing (DTEF) online student registration system which failed to meet its expected functionalities (Ramadubu 2021). Based on the general and aggregated failure attributes of the developed software applications, it could be concluded that use of Agile SDMs could provide a better solution to further development of software in Botswana by BSDI.

2.10. Technology Adoption Models

Many theories on acceptance of IT exist for examining the acceptance of IT tools or concepts. These include the Innovation Diffusion Theory (IDT) (Rogers 1995), Technological Acceptance Model (TAM) (Davis 1986), Technology Acceptance Model 2 (TAM2) Venkatesh and Davis (2000, p.186), Theory of Planned Behaviour (TPB) (Ajzen 1991, p.179), Theory of Reasonable Action (TRA) Fishbein and Ajzen (1975), Decomposed Theory of Planned Behaviour,Taylor (1995, p.144), and Technology Acceptance Model 3 (TAM3) (Venkatesh & Bala 2008, p.273).

2.10.1. Technology Acceptance Model

The Technology Acceptance Model (TAM) was introduced by Fred Davis in 1986 Davis (1986) (see Figure 19). The theory is derived from the theory of the Reasoned Action (TRA) and the Theory of Planned Behaviour (TPB) (Venkatesh & Davis 2000, p.186). TAM is meant to model and explain a users' acceptance of information systems or technologies.

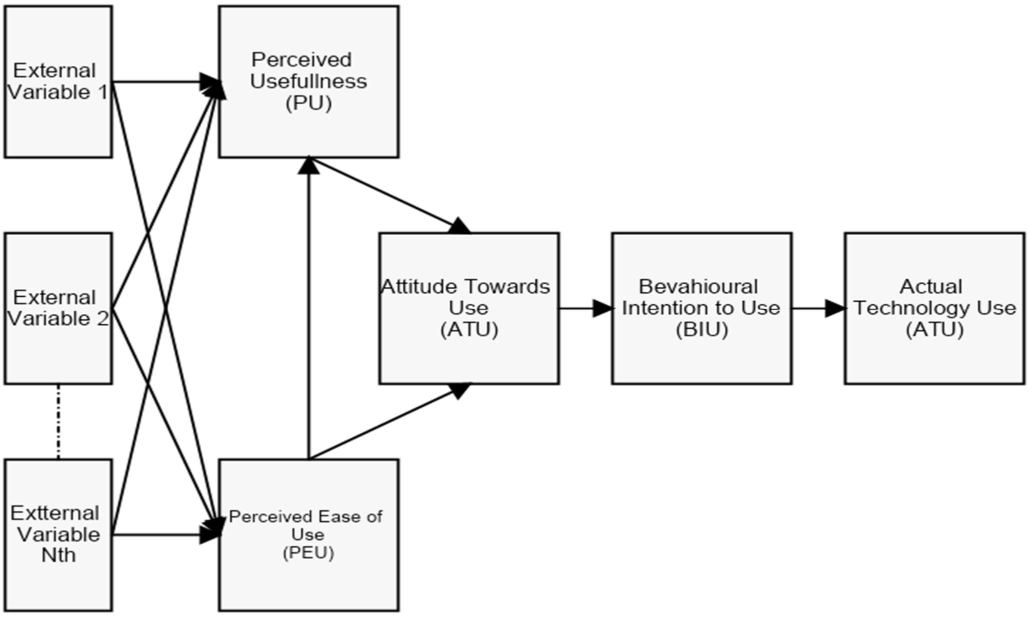

External Variable 1
External Variable 2
Extternal Variable Nth
Perceived Usefullness (PU)
Perceived Ease of Use (PEU)
Attitude Towards Use (ATU)
Bevahioural Intention to Use (BIU)
Actual Technology Use (ATU)

Figure 19:Technology Acceptance Model (Adopted from Wixom & Todd 2005)

According to Ng, Shroff and Lim (2013, p.355), the success of any given technology or system is determined by user acceptance, measured by four TAM constructs or factors as defined by Davis (1986), refer figure 19. These factors are namely: Perceived Usefulness (PU), Perceived Ease-of-Use (PEU), Attitudes Towards Usage (ATU) of the system and Behavioural Intention to Use (BI). The belief comprises factors PU and PEU of the user towards a system or technology and can be influenced by other external factors referred to as external variables (see Figure 2-16). The belief then shapes the attitude or feeling (ATU) to do the task using the IT or system. The attitude attained influences the behaviour (BI) whether to accept and adopt and use the IT or system. In summary, the TAM theory tends to explain why individuals adopt or decide not to adopt a particular IT or system. The TAM identifies the contributory relations among system design features, perceived usefulness, perceived ease of use, attitude toward using, and actual usage behaviour.

The TAM theory proposes that the point of view towards usage of the IT technology is jointly determined by the functionality and the ease of use and not the user's attitude towards using the system (Al-Gahtani, Hubona & Wang 2007, p.681). According to Park (2009, p.150), TAM accounts for 40-50% of user preference and it is a model that has continuously evolved and has matured with time leading to the development of the new version TAM2.

2.10.2. Extended Technology Acceptance Model

The TAM theory proposes that the point of view towards usage of the IT technology is jointly determined by the functionality and the ease of use and not the user's attitude towards using the system (Al-Gahtani, Hubona & Wang 2007, p.681). In contrast, Technology Acceptance Model 2 (TAM 2) hypothesises that the users' perceptual evaluation of the usefulness of an innovation or technology is related to the degree of comparison or match between tasks to be done at work using innovation or technology and the outcome attained. The outcome is, therefore, a function for determining the basis for deriving perceptions regarding whether that innovation or technology is useful or not (Venkatesh & Davis 2000, p.186). The major enhancements made to TAM2 (Figure 20) are two theoretical constructs involving **social influence processes** (subjective norm, voluntariness, and image) and **cognitive instrumental processes** (job relevance, output quality, result demonstrability, and perceived ease of use) (Cobelli 2020, p.21).

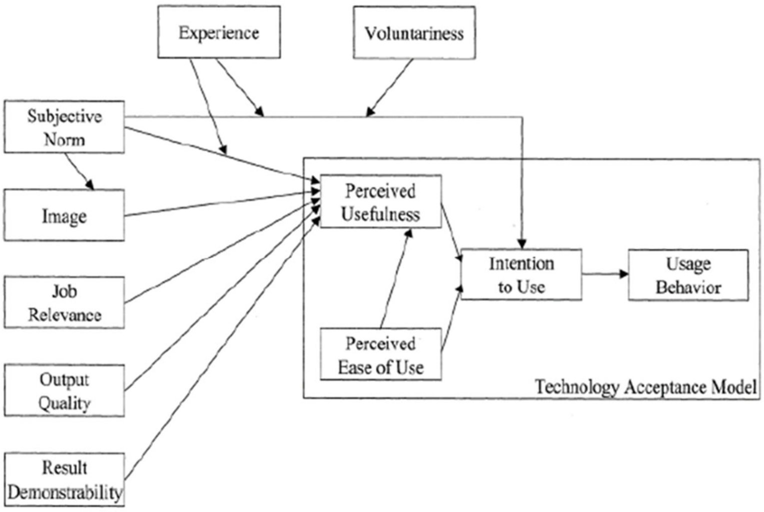
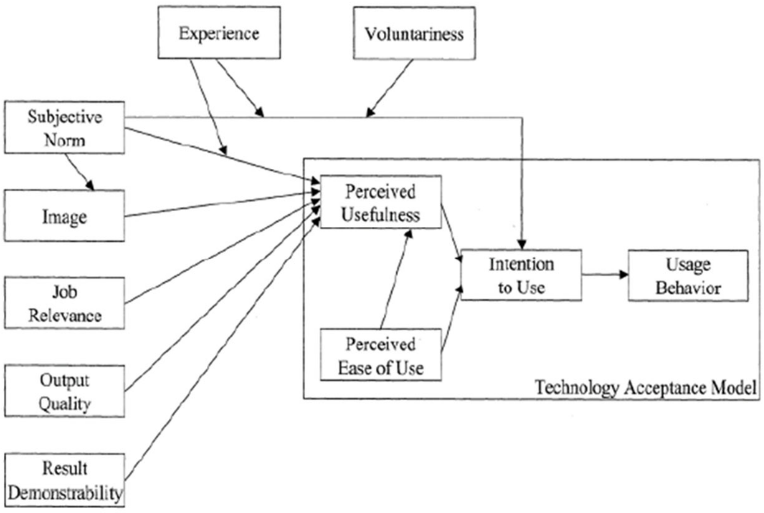


Figure 20: Technology Acceptance Model 2 (Adopted from Venkatesh & Davis 2000, p.188)

These two constructs qualifiers as external factors that affect the perceived usefulness (PU) of the original TAM model (Cobelli 2020, p.22). The moderators, voluntariness (mandatory usage) and experience (during early stages) have links between subjective norm and behavioural intention. The TAM2 model (see) was tried in both optional and fixed settings, and according to Park (2009, p.150), the results explained the user adoption 60% correctly better than the TAM, which accounted for 40-50%.



Although there is a further extension to TAM 2 (i.e., the Technology Acceptance Model 3 (TAM 3), TAM3 has further exogenous factors that qualifiers the PEU. The factors for PU are similar to those in TAM 2 and the factors for PEU are anchor variables (Computer Self-Efficacy, Perceptions of External Control, Computer Anxiety, and Computer Playfulness) and adjustment variables (Perceived Enjoyment and Objective Usability) (Venkatesh & Bala 2008, p.274).

TAM 2 has been widely used for acceptance and adoption of technologies from a broader perspective of organizations to people according to literature view.

Table 3 gives an overview of the additions from TAM to TAM 2 and TAM 3 constructs. TAM has the basic constructs PU, PEU, ATU and BI. In TAM 2, the PU construct is further qualified by the exogenous factors social influence and cognitive instrumental. In TAM 3, the PU is also qualified by cognitive instrumental and cognitive instrumental the PEU is qualified by Anchor variables and Adjustment variables.

Table 2: Definition of different TAM Model constructs

Model	Construction	Definition
TAM basic construction	*Perceived Ease of Use (PEU)*	The degree to which a person believes that using a particular system would be free of effort (Isaac et al. 2017; Zainab , Awais & Alshagawi 2017)
	Perceived Usefulness (PU)	The degree to which a person believes that using a particular IT or system would enhance his or her job performance (Mezhuyev et al. 2019)
	Attitudes Towards Usage (ATU)	Defined as the feeling and thought by user expressing an intention to utilise an information technology or system being positive or negative (Weng et al. 2018).
	Behavioural Intention to Use (BI)	The behavioural intention to use the tool or system (Alhashmi, Salloum & Mhamdi 2019)
TAM2 additional constructs	Subjective Norm	Individual's perception based on peers' views on him that he should or should not perform a behaviour in question

Model	Construction	Definition
	Image	The extent of influence for a particular innovation is perceived to extend one's image in their environment of operation.
	Job Relevance	An individual's perception is relative to the level of influence of technology or tool in one’s area of performance regarding his or her job area.
	Output Quality	The degree to which completed work satiates and gives credible value to job goals or tasks.
	Result Demonstrability	The demonstration and effectiveness of attained results using the innovation and translated through observation and communication.
TAM3 additional constructs	Computer Self-efficacy	The ability of the user to use a computer in doing a particular task
	Perceptions of External Control	The degree to which an individual believes that organisational and technical resources exist to support the use of the system.
	Computer Anxiety	The level of anxiety and fear in using a computer to complete a given task
	Computer Playfulness	The level of cognitive spontaneity in interaction with computers
	Perceived Enjoyment	The level of excitement and completeness in using a computer to achieve and complete a task
	Objective Usability	The level of measurement of a task done using a computer and that not done using a computer (i.e., level comparability and effectiveness).

TAM 2 as an upgrade of TAM, addressed the inadequacies of the original TAM, i.e., inability to evaluate social influence of technology beyond work place Lai (2017, p.963), the need to add external variables to TAM to ensure reliable prediction on adoption of technology and finally the cognitive assessment of the technology whether it can produce the desired product of quality and relevance of the technology (Putra 2019, p.26). TAM3 improved TAM2 by adding external variables to the construct PEU such as: perceived usability, computer anxiety, efficacy and playfulness, perception of external control, perceived enjoyment and objective usability and are explained as in table 2 (Venkatesh & Bala 2008).

2.10.3. Application of Technology Acceptance Models

Since its inception in 1986, the TAM theory has been used and applied in various research studies of different and various domains of IT. In Software Engineering (SE), TAM was employed to check the acceptance of Computer-Aided Software Engineering (CASE) systems (Chau 1996, p.269), object-oriented programming (Hardgrave & Johnson 2003, p.322), software measures (Wallace & Sheetz 2014, p.249), factors affecting the meta-modelling acceptance (Mezhuyev et al. 2018, p.49476), acceptance of search-based SE techniques (Mezhuyev et al. 2019, p.101073), among many others.

Researchers in SE have proved that TAM could explain not only the acceptance of an information system (IS) or a particular software system but also the processes involved under these systems. For instance, TAM has been applied to the software process improvement initiatives because the reasons for accepting a new initiative is similar to that for accepting a new technology (Umarji & Seaman 2005, p.1). The theory of TAM has been widely used for examining how individual users accept, adopt and use IT innovations and concepts such as the World Wide Web, spreadsheets and many other IT tools (Chan & Thong 2009, p.805). Furthermore, other researchers such as Evans et al. (2014, p.6) have used the TAM theory to determine individual user acceptance of social media tools.

Chan and Thong (2009, p.807) used the TAM theory to determine acceptance of SDMs based on a four-point merit system and observations. Firstly, the TAM theory provides clear and distinct constructs for measuring characteristics of SDMs such as user's perceptions, attitudes, ease of use, perceived usefulness on adopting and utilisation of SDMs for the development of the software (Chan & Thong 2009, p.807). Secondly, the TAM theory has the capability to factor in non-technical issues on selection and adoption of SDMs based on the individual and organisations. The third factor mentioned by Chan and Thong (2009, p.805) recognises organisational management's orientation to accept Agile SDMs. Alhashmi, Salloum and Mhamdi (2019, p.27) mention some of the factors (technological partnerships, training and local skill on new technologies, democratic management), which encourages self-organisation and self-sense. According to these authors, organisations must adopt a cooperative type of management that encourages self-organisation, communication and collaboration and development of responsibility in self-sense amongst project team members (Al-Mamary, Shamsuddin & Aziati 2014a; Aldahmash, Gravell & Howard 2017; Forrester Research 2011; Krishnan 2015)

TAM theories have been used in many empirical research studies to predict and forecast the adoption and use of various IT tools and systems (Georgiou, Rizou & Spinellis 2019, p.1; Lee & Low 2017, p.150; Lee, Hsieh & Hsu 2011, p.124; Mezhuyev et al. 2018, p.49476; Mezhuyev et al. 2019, p.101073; Middlemass, Vos & Siriwardena 2017, p.1; Morteza et al. 2011, p.274; Rigopoulos & Askounis 2007, p.1; Rönnby et al. 2018, p.1; Wallace & Sheetz 2014, p.249; Wu & Chen 2017, p.221). The empirical research studies have collected data analysing TAM constructs using quantitative research methods. However, the research also noted that PU, PEU and ATU are also regarded as cognitive factors, where ATU represents individuals' evaluative feelings (positive or negative) for a particular behaviour in using an IT tool or a system, PU regards the ease of use, and finally, PEU is the effortless input in using the IT tool or the system. These are all beliefs that are not enumerative but can also be expressed qualitatively. Therefore, data can equally be collected using the qualitative approach and analysed to determine the usefulness and applicability of an IT tool to use a system.

Research studies that have used the TAM theory and conducted qualitative research methods to collect experiential data based on PU, PEU and ATU TAM are known (Lee & Low 2017, p.150; Middlemass, Vos & Siriwardena 2017, p.1; Ng, Shroff & Lim 2013, p.355; Rönnby et al. 2018, p.1; Song & Kong 2017, p.865; Vogelsang, Steinhüser & Hoppe 2013; Yucel & Gulbahar 2013, p.89). These studies have analysed the data qualitatively and used qualitative analytical methods to determine experiential knowledge on beliefs, attitudes and behaviour towards the adoption, acceptance, and usage of IT tools. The TAM model constructs can facilitate qualitative explanation of non-cognitive skills on adoption and use of IT or systems (Lee & Low 2017, p.150; Middlemass, Vos & Siriwardena 2017, p.1; Ng, Shroff & Lim 2013, p.355; Rönnby et al. 2018, p.1; Song & Kong 2017, p.865; Vogelsang, Steinhüser & Hoppe 2013; Yucel & Gulbahar 2013, p.89). The user's experiential knowledge on ease of usage, attitude, and actual usage behaviour of that technology can be evaluated qualitatively to gain a deeper insight understating that is beyond numbers and expressions

2.11. Identified gap

The contribution of software in accomplishing mammoth and complex technologies cannot be underestimated in the current socio-economic order we are living in. These complex technologies have paved new ways for automating and enhancing business activities such that anything is possible, and the limit is a person's ingenuity, creativity and innovativeness (Brummelen & Slenders 2019). Countries that have adopted and realised this gap has started to engage knowledge-based economies as a means of production. These countries have developed "smart technologies" that have found use in all walks of life. In smart homes, smart electrical meters are used. In huge organisations, automatic telephone answering, and monitoring are done by bots. Internet bots are used to manage the security and interaction of some social media applications and many others. However, the development of modern software must reflect these complex technical aspects of the IT evolution. Modern SDMs—which focus on the entire value chain—would therefore be appropriate for software development by combining a mixture of Design Thinking and the use of Lean, Agile and DevOps practices (Brummelen & Slenders 2019).

Botswana, a middle-income country that has invested and set up an ICT infrastructure that is rated by the World Economic Forum's (WEF) and the Global Information Technology Report (GITR)—the most respected research periodicals—to be 6th in Africa in terms of ICT readiness (MmegiOnline 2021). The GITR measures the degree to which economies leverage ICT to enhance competitiveness by assessing ten pillars ranging from policy, regulations, infrastructure, affordability, skills, usage and impact. According to the website news report by MmegiOnline (2021), Botswana is powering ahead with its e-government strategy, which will ultimately result in the provision of most government services online thus enhancing efficiency, lowering costs and boosting access. The BSDI, the constituent software development companies, and software development professionals should be seen to embrace the new modern SDM to ensure the success of the country's e-government strategy, realise the strategic plan of implementing a knowledge-based economy and avoid the pitfalls that were witnessed during the development of software applications such as the MALEPA system, Livestock Information Technology System Agricultural Project(LITS) and Botswana Telecommunication Communication (BTC) billing system.

Therefore, there is a need to develop a framework for the adoption of modern SDMs in Botswana. The framework can be used by the BSDI as a tool to assist with the selection, adoption and use of an appropriate modern SDM for modern business processes. This could contribute to the efforts of the BSDI to produce quality software and manage the process of software projects. This study therefore selected TAM2 model as the most relevant theory to use.

2.12. Summary

In this chapter, the literature that relates to the key concepts of the research were reviewed. The review revealed a need for a study to be conducted that provides insights into the adoption and usage of modern SDMs in Botswana, selection of appropriate theory to be used for developing an Agile SDM framework in collaboration with the software development community, a framework based on Technology Acceptance Model for assisting and guiding BSDI, BSDC and professional software developers when selecting, adopting, and implementing

Agile SDMs in Botswana. The TAM2 model was used as a theory to underpin the study.

CHAPTER 3: RESEARCH METHODOLOGY AND DESIGN

3.1. Introduction

In Chapter 2, related literature based on the keywords defined as well as the key concepts of the study was reviewed. In Chapter 3, the research methodology and design will be discussed. Specifically, the research epistemology, theoretical perspective, methodology and methods will be discussed. The data sources used, data analysis that was implemented, ethical considerations of the research, and validation and reliability issues will also be considered.

3.2. Research Design

According to (Crotty 1998), four elements define the research design (see Figure 21). These key elements depict the researcher's epistemological stance, theoretical perspective, methodology, and methods to be used (Crotty 1998). Each of the elements will be discussed in the subsections that follow.

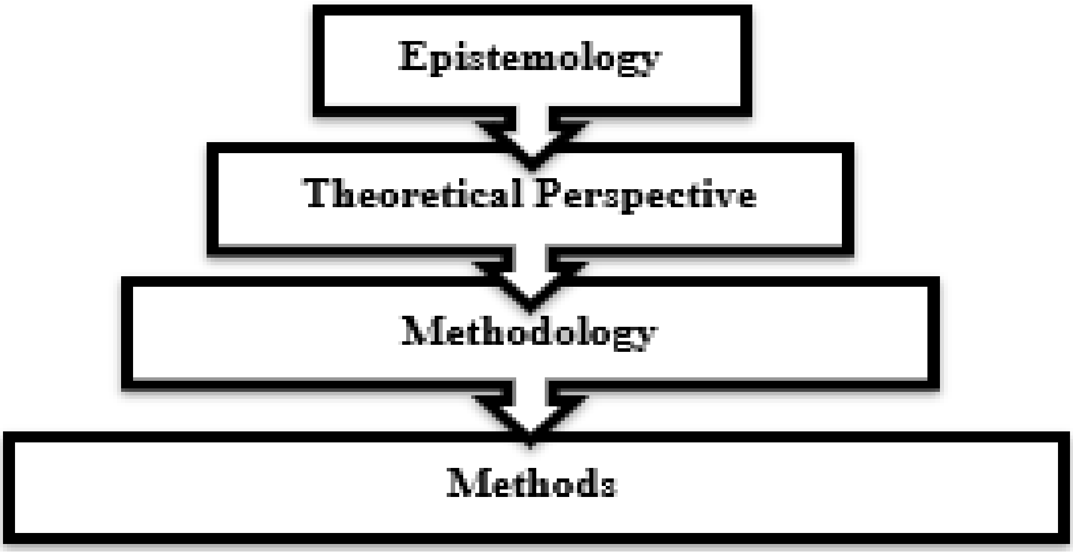


Figure 21: Four elements of the research process (Adopted from Crotty 1998, p.4)

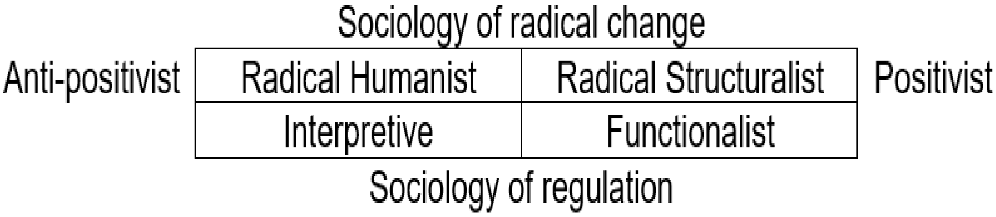
3.2.1. Epistemology

Most epistemological beliefs range from objectivism to subjectivism. Crotty (1998) considers three main epistemological assumptions, namely: *objectivism*, *constructivism, and subjectivism*. Objectivism is the belief that research can determine the objective truth and meaning while the constructivist approach takes into account a person's perceived social reality (Crotty 1998). Subjectivism has, on the other hand, been ascribed to the following comment: "*our own mental activity is the only unquestionable fact of our experience*" (Stanek 2017, p.2).

The approach of the constructivist research paradigm, as articulated by Given (2008, p.119), encourages interaction of the research and participants when collecting data intended to understand an occurrence from the point of view of the participants. The researcher's understanding is linked with that of the participants through their complementary association within the inquiry environment and dialogic association through the investigator's data generation practices such as interviewing (Given 2008, p.119).

3.2.2. Theoretical paradigm

Burrell and Morgan (1979) define four paradigms that range from the anti-positivist to positivist and from radical change to regulation (see Figure 22).



Sociology of radical change

Anti-positivist	Radical Humanist	Radical Structuralist	Positivist
	Interpretive	Functionalist	

Sociology of regulation

Figure 22: Four paradigms for the analysis of social theory (Burrell & Morgan 1979, p.22).

The radical humanist paradigm is concerned with the subjective world and the need to change it, whereas the interpretivist paradigm aims to explain situations (Cronje 2012; Roode 1993). The functionalist paradigm believes that stronger structures and rules will improve situations. The radical structuralist view considers and describes the current situation and is based on an objective world view (Burrell & Morgan 1979; Cronje 2012; Roode 1993).

An interpretivist perspective assumes that truth is only attained *via* social links such as perception, verbal communication and research tools (Myers 2009). This research is thus assuming an interpretivist theoretical perspective.

3.2.3. Methodology

A methodology explains the strategy and actions that need to be taken to execute the research. It is the research design and the rationale behind the choice of methods adopted (Crotty 1998). The research framework for this research will follow a case study methodology.

As previously mentioned, Yin , R (2003) is of the opinion that a case study design should be followed when: (a) “how” and “why” questions are to be answered; (b) the behaviour of participants in the study cannot be manipulated; (c) contextual conditions relevant to the phenomenon under study should be highlighted, and (d) boundaries between the phenomenon and context are not clear.

Although the case study approach tends to provide an opportunity to explore the research problem, ask related questions and grasp the richness of company behaviour, conclusions drawn from the case study may be specific to a particular organisation researched and can as a result not be generalised (Gable 1994, p.112). This is especially true when a single case study research method is adopted. It is for this reason that Yin (2009, p.19) advocates for a multiple-instance case research strategy spawning many organisations that guarantees generalisation of the derived results. Darke, Shanks and Broadbent (1998, p.277) seem to be in support of a multiple-case research strategy. This research study used a research method that involves the multiple-case study approach. The selected eligible software development organisations in the Botswana software development industry (BSDI) was based on a predefined criterion.

A qualitative exploratory research design was used to investigate the adoption and implementation of relevant modern software development methodologies (SDMs). An exploratory research study is used when a new topic is considered or a new angle of looking at the topic is used (Guetterman, Fetters & Creswell 2015, p.554). A qualitative exploratory approach using the case study methodology was used to explore explanations and comprehension and describe the adoption and implementation of relevant modern SDMs in Botswana.

3.2.3.1. Methodology Case Study

Qualitative researchers are interested in understanding the meaning constructed by people, that is, how people make sense of their world and the experiences they have in the world (Cohen, Manion & Morrison 2007, p.47). In addition, qualitative investigators focus on socially-built truths by addressing solutions to questions that look at how social experience is developed and given meaning (Denzin & Lincoln 2008, p.8).

The case study methodology enabled extensive information gathering over a given period from identified Botswana software development companies and their software development professionals (Creswell 2009, p.97). It helped the researcher to conduct a cross-case analysis of these companies and their professionals with the view to build theory or knowledge based on the factors that determine the adoption of modern SDMs to develop software applications for contemporary business processes.

3.2.4. Sampling

Purposive sampling was used to head-hunt specific Botswana Software Development companies (BSDCs). The research only considered companies that were actively tendering for software application development tenders—in both the private and public domain—and was also the most active and consistent software developing companies in Botswana. To find these companies, tendering information was requested from the tender board and other organisations. A total of twelve BSDCs were identified and initially contacted telephonically. An e-mail explaining what the research entailed was then sent to these companies, and they were asked if they would be prepared to participate in the study. Only nine companies volunteered to participate in the research. An interview schedule was developed and shared with the relevant companies before agreeing with the researcher on the time of the interview. The goal of the interviews was to gain a good understanding and perspective of software development in Botswana with a specific emphasis being placed on the SDMs the BSDCs are using.

The semi-structured interview questions, one for managers i.e., organisation participants (OP) and one for software developers i.e., employee participants (EP) were prepared and sent to respective participants. The semi-structured interview questions were meant to gather information on the demographic profile of sample population participants, their perception, experiential exposure and comprehension on issues of SDMs, implementation and utilization of SDMs based on the study research questions (Table 3).

Table 3: Interview respondents.

Company	Employee participant (EP)	Organisation manager participant (OP)
A	1	1
B	1	1
C	1	1
D	1	1
E	1	1
F	1	1
G	1	1
Total	9	9

For the semi-structured interviews, specific persons who are proficient and knowledgeable in software development could communicate effectively on issues regarding their experiences and perceptions in a clear, expressive, and thoughtful manner, and were willing to participate in this research study were identified based on organization human resources selection. Semi-structured interview questions interviews allow a deeper understanding of conditions and situations relating to software development. Nine interviews with OP participants and nine interviews with EP participants were successfully conducted. For each company, one software developer employee and one employee occupying a managerial or senior position were interviewed (i.e., a total of two people were interviewed in each company).

Semi-structured interview questions /probes facilitate the collection of information that is broad and rich in content. The probes allow the respondent to answer openly without any specific guide. Typical questions asked are: ***"**What do you think about the agile software development methodology?***"**

Even though the researcher gave the OP respondents the option to fill in the research questionnaire or opt for an oral interview, all the nine OP respondents elected to participate in an oral interview. Before the commencement of the interview, permission was asked from the respondents to record the oral interviews. The semi-structured interview questioning allowed different opinions, perceptions and viewpoints from the different companies and their staff. The semi-structured interview questions allowed exploration of grey areas in implementation, adoption, utilization, challenges and proposal for adoption of modern SDMs.

The researcher worked with the identified BSDCs with the view to becoming familiar with software developers' approaches to software development, that is, their thoughts, knowledge processing and praxis. As an observer, the researcher explored the influence of BSDCs on the selection and use of appropriate SDM for software development. Understanding a BSD perspective on the selection of appropriate and pertinent SDMs for the development of high-quality software is very important as it determines resource allocation and distribution. The researcher wanted to understand the application of appropriate SDMs to develop business processes and to explore the company's agility and dynamics when confronted with these scenarios. The researcher observed and played the role of a passive participant rather than an active participant.

3.3. Data Gathering Methods

Data collection methods depending on the research approach, research design, the type of data collected during the research and the methods of data analysis technique to be employed (Etikan, Musa & Alkassim 2016, p.1). Data collection methods can be categorised as qualitative or quantitative. In this research study, the qualitative method was adopted (i.e., qualitative data collection methods were used). According to Johnston (2017, p.619) and Walliman (2017, p.69), data can also be classified as primary or secondary. Similarly, data collection methods can be divided into primary or secondary. The research study employed research instruments and standards that are specific to a research case study methodology for collecting primary data from the Botswana Software Development companies (BSDCs) and software developers. Three specific data collection methods were used in this research study, namely case study, multiple case study and document analysis through literature review. The research methods used for collecting data were thus semi-structured interviews (Ritchie & Lewis 2003, p.138).

3.3.1. Interviews

Interviews can be classified as structured, semi-structured and unstructured (Coolican 2017). In this research study, face-to-face semi-structured interviews were conducted using a list of probes. These types of interviews are considered focused but rich. They allow for a comparative analysis to be conducted since they are guided by a script or probes. Interviews are also appropriate for exploring sensitive subjects or issues, where participants have freedom of expression (Gill et al. 2008, p.292).

Therefore, the open-ended probes were aimed at establishing a person's opinion, feelings, viewpoints and perceptions with regards to the issue under investigation (Mitsuhashi 2013, p.1; Schmidt, Gummer & Roßmann 2020, p.3). From the researcher's side, open-ended questions facilitate a deeper probe into the respondents' answers thus gaining valuable information and insight into the subject at hand. Schmidt, Gummer and Roßmann (2020, p.3) mentioned that, as much as open-ended questions elicit rich information and content, they also have disadvantages such as being time-consuming.

The semi-structured interview questions, one for managers i.e., organisation participants (OP) and one for software developers i.e., employee participants (EP) were prepared and sent to respective participants. The semi-structured interview questions were meant to gather information on the demographic profile of sample population participants, their perception, experiential exposure and comprehension on issues of SDMs, implementation and utilization of SDMs based on the study research questions (Table 4).

Table 4: Interview respondents.

Company	**Employee participant (EP)**	**Organisation manager participant (OP)**
A	1	1
B	1	1
C	1	1
D	1	1
E	1	1
F	1	1
G	1	1
Total	9	9

For the semi-structured interviews, specific persons who are proficient and knowledgeable in software development could communicate effectively on issues regarding their experiences and perceptions in a clear, expressive, and thoughtful manner, and were willing to participate in this research study were identified based on organization human resources selection. Semi-structured interview questions interviews allow a deeper understanding of conditions and situations relating to software development. Nine interviews with OP participants and nine interviews with EP participants were successfully conducted. For each company, one software developer employee and one employee occupying a managerial or senior position were interviewed (i.e., a total of two people were interviewed in each company).

Semi-structured interview questions /probes facilitate the collection of information that is broad and rich in content. The probes allow the respondent to answer openly without any specific guide. Typical questions asked are: ***"What do you think about the agile software development methodology?"***

Even though the researcher gave the OP respondents the option to fill in the research questionnaire or opt for an oral interview, all the nine OP respondents elected to participate in an oral interview. Before the commencement of the interview, permission was asked from the respondents to record the oral interviews. The semi-structured interview questioning allowed different opinions, perceptions and viewpoints from the different companies and their staff. The semi-structured interview questions allowed exploration of grey areas in implementation, adoption, utilization, challenges and proposal for adoption of modern SDMs.

The researcher worked with the identified BSDCs with the view to becoming familiar with software developers' approaches to software development, that is, their thoughts, knowledge processing and praxis. As an observer, the researcher explored the influence of BSDCs on the selection and use of appropriate SDM for software development. Understanding a BSD perspective on the selection of appropriate and pertinent SDMs for the development of high-quality software is very important as it determines resource allocation and distribution. The researcher wanted to understand the application of appropriate SDMs to develop business processes and to explore the company's agility and dynamics when confronted with these scenarios. The researcher observed and played the role of a passive participant rather than an active participant.

3.4. Data Analysis Methods

This research study applied a qualitative content analysis technique for documentation provided by the BSDCs related to their software development processes. The aim was to discover trends and patterns when performing a particular task, which indicates the resident knowledge, perceptions and understanding of how tasks are performed. Documentation also captures an organisation's perceptions, thus leading to a broader understanding of the company's thought processes, which often have a bearing on the individuals who are being directed. Content analysis revealed patterns and trends of BSDCs how their software development professionals choose SDMs and what factors influences these choices.

A cross-case analysis was used to analyse the data collected from the BSDCs and their software development professionals to determine the relevant patterns and trends. Data collected from the different software development companies were coded using NVivo (and Microsoft Excel) for each of the companies. Related patterns were grouped and further coding of related groups was conducted to identify generated themes or summarised categories.

These trends and patterns formed the factors that were used to construct the framework for the adoption of modern SDMs for the BSDI and related companies

3.4.1. Content analysis

Cho and Lee (2014, p.4) quoted by Moretti et al. (2011, p.420) defined qualitative content analysis as an analysis method that categorises literature or verbal material into identified classes of similar content. An open-coding approach considers questions such as why, how, where when, and what. This process then generates many loose arrays of concepts and categories that can be labelled and tagged into a coherent pattern of events or concepts. Cho and Lee (2014) are of the view that content analysis can be adopted for:

> *"…narrative responses, open-end survey questions, interviews, focus groups, observations, printed media such as articles, books, or manuals"*

(p.4)

Elo and Kyngäs (2008, p.107) claim that content analysis may be used with either qualitative or quantitative data and in an inductive or deductive way. Inductive content analysis can be employed in cases where there is no history of studies dealing with the situation or when history is partitioned; in contrast, the deductive approach is useful for comparing groups at different time intervals (Elo & Kyngäs 2008, p.111). Content analysis is a way of analysing written, verbal, or visual information. It helps to summarise words and group them into fewer classes. It is believed that, when grouped into the same categories, words and phrases can have the same meaning (Elo & Kyngäs 2008, p.109). Content analysis is a research procedure for making iterative and valuable assumptions from data in its context to give information, new ideas, a sample of facts, and a practical action guide (Elo & Kyngäs 2008, p.109). The goal is to obtain a deep and wide description of the occurrence, and the result of the analysis is concepts or categories explaining the occurrence. The concepts or categories are mainly meant for building a model, conceptual system, conceptual map or classes of concepts (Elo & Kyngäs 2008, p.109).

According to Hays (2007, p.44), the small statements are contextual units, a group of small statements make analytic units, and a group of analytic units make the coding units. Categorisation of data leads to data analysis and interpretation, and conclusions are thereafter drawn. Data is then analysed according to themes that arrange and describe the data in rich detail and frequently interpret various aspects of the research that emerge from the collected data (Cruzes et al. 2015, p.1634). Content analysis briefly organises, describes the data set in rich detail, and frequently interprets various aspects of the research topic (Cruzes et al. 2015, p.1634). The thematic analysis involves the identification of the main, recurrent or most important issues or themes arising from a body of evidence (Cruzes et al. 2015, p.1634; Forman & Damschroder 2007, p.39).

3.4.2. Cross Case Analysis

Cross case analysis is defined by Khan and VanWynsberghe (2008, p.1) as a research procedure to facilitate information collected from individual case studies. It has been reported that mobilisation of case knowledge occurs when researchers collect case knowledge, compare and contrast cases, and in doing so, produce new knowledge (Khan & VanWynsberghe 2008, p.1).

The cross-case analysis is also regarded as a research method that enables a researcher to conduct a comparative study of the given cases by generating new questions, pointing at new dimensions, producing alternatives, generating models and developing new ideas and perceptions (Khan & VanWynsberghe 2008, p.1).

Cruzes et al. (2015, p.1634) recounted that case studies are flexible, allow for several forms of data collection, and are mostly informed by qualitative data to build a body of knowledge from individual cases. Case studies have a special ability to provide a deep understanding of the phenomena under study. The cross-case analysis is sometimes used as a general umbrella term for the analysis of two or more case studies to produce a synthesised outcome (Khan &

VanWynsberghe 2008, p.1). It enables a comparison of events, tasks and procedures, and the elements of analyses in case studies to be carried out.

Furthermore, Cruzes et al. (2015, p.1634) have elaborated that cross-case analysis can be used as a way of conducting data analysis by arranging data from individual cases in tables and graphs. McGuiggan et al. (2008, p.2) indicated that case-study methodology is well established and widely accepted in management information systems and is suitable for exploratory, descriptive, and explanatory research. Cross-case studies explain the causal connections in real-life circumstances that are too difficult for a single survey or experiment. McGuiggan et al. (2008, p.3) cited (Eisenhardt 1989, p.532) who explained that patterns within cross-case studies help in establishing dimensions or constructs from the literature followed by looking for within-group similarities and inter-group differences. Content analysis and cross-case analysis were used in this research for purposes of analysing the generated data. The NVivo software analysis program coded related information and developed associations among the constructs by thematic coding to identify similarities and differences.

3.4.3. Interpretative, Structural and Reflective Analyses

The interpretational analysis includes a systematic set of methods to code and categorise qualitative data to ensure that valuable constructs, themes, and patterns emerge (Cohen, Manion & Morrison 2007, p.461). The structural analysis involves a correct set of methods for analysing qualitative data that do not need to be assumed from the data but are inherent features of the discourse, text, or occurrences that the investigators are studying (Cohen, Manion & Morrison 2007, p.462). Reflective analysis refers to a process in which qualitative researchers depend mostly on their intuition and individual evaluation to examine the data that have been collected; it can also involve critical appreciation (Cohen, Manion & Morrison 2007, p.461).

3.5. Research ethics

All the relevant ethical considerations were considered by the researcher before the undertaking of this research study. A clearance letter from the University of South Africa (UNISA) approving work with software development organisations was issued (see Appendix A). The approval letter was obtained to comply with the legislative requirements of the government of Botswana that regulates ethics policies relating to the conducting of research in the country. Other requirements relating to conformance to confidentiality and anonymity with all stakeholders involved in the research were also complied with. Consent forms for the targeted study population were used for purposes of obtaining information from the respondents and gathering primary data. Contributions in the form of gathered data were kept confidential and can only be released to stakeholders after signing a legal agreement.

3.6. Research Assumptions, Limitations and Scope

Dimitrios and Antigoni (2019) describe research limitations as a possible weakness that is totally out of the researcher’s control. These limitations are related to adopted research design, sampling constraint, access to respondents of the research study and many others. In this research study, the following were constraints experienced by the research. Out of the twelve selected and sampled research respondents, only nine (75%) participated in the study. The three respondents could no longer participate in the study due to other commitments or they felt it is a sensitive issue to discuss. Another limitation that confronted the research study is that all the respondents were located in Gaborone and no respondents could be located in other cities in Botswana. Gaborone is the capital city of Botswana where most government offices and major industries are located. Therefore, the study is limited to a single city in Botswana (i.e., Gaborone) and can therefore not be generalised to the rest of the country.

Delimitations or scopes are essentially limitations of t the research is in total control of (Dimitrios & Antigoni 2019, p.155). The scope of the study is therefore limited to the Botswana software industry and companies that have developed software and are registered with the registrar of companies in Botswana. Twelve companies that constitute a sample population of the Botswana software industry were selected to participate in the research. In this study, the Botswana software companies and their employees are considered software developers. The selected research methodology in terms of theoretical background, objectives, research questions, variables under study and study sample were set by the nature of knowledge to be created, that is, the knowledge that is based on practice and belief. Therefore, the study adopted the constructive/interpretivist philosophy approach for decoding and formulating the meaning and intentions of practices when selecting modern SDMs in the Botswana software development industry for software development.

The assumptions being made are things that are accepted as true, or at least plausible, by researchers and peers who will read the dissertation or thesis (Dimitrios & Antigoni 2019, p.155). In this study, it was anticipated that there would be many software developments companies that could be found in several cities of Botswana; however, this was not the case. Instead, the software development companies that were identified were located only in the capital city of Gaborone thus limiting the generalisability of this study to Gaborone only. The key assumption that was initially made was to have all the twelve BSDCs participating in the study. However, this was not the case since only nine (i.e., 75%) BSDCs elected to participate in the study. For the research to be substantive and authentic, the researcher believes that all the respondents should provide answers to asked questions with honesty because any dishonest responses could have serious flaws in the framework to be developed.

3.7. Validity and reliability in case study research

3.7.1. Internal validity

In a case study that involves qualitative research, internal validity is achieved by applying tools such as case analysis, cross-case analysis, pattern matching, assurance of internal flow of results, expert peer review, the formulation of diagrams, illustration, and data matrices to demonstrate the internal consistency of the information collected (Christie et al. 2000, p.17). According to Yin (2009, p.36), internal validity is applied to uniquely distinguish the unit of analysis, linking the analysis to prior theory identified in a literature review, and presentation and analysis of pilot case studies. In this research study, the units of analysis were identified using the theory of extended technology acceptance model (TAM2), which is deliberated in detail in the literature review chapter.

Another aspect of internal validity that can be done in concurrence is a research article presentation on the key subject matter of the research study where the researchers write a research article and use peer feedback provided for correcting some concepts of the research. Other approaches that are mentioned by Christie et al. (2000, p.17) include peer debriefing, discussion of the findings and conclusions with other academic investigators or research supervisors.

3.7.2. External validity

Leung (2015, p.325) describes qualitative research validity as the suitability of equipment, procedures and information. Mohajan (2017, p.15) thinks that validity can be either internal (i.e., looks at the usefulness of the outcomes due to data collection or data processing) or external (i.e., looks at how the results can be generalized to other different situations). In contrast, Devroe (2016, p.5) articulates external validity as a term used to refer to how the findings of a given study can be applied to other different situations, groups and results. In the case of this research study, multiple-case studies in various BSDCs were adopted to establish a general replication pattern or trend so that similarity of results attains a level of external validity (Yin 2009, p.36).

3.8. Conceptual framework

The conceptual framework for this research study is shown in Figure 23. It displays the primary constructs of the conceptual framework as individual factors, namely: software developer factor, social factor (subjective norm of the BSD software developer) and organisational factor (BSD companies and resources disposition). According to the TAM2 theory, constructs that characterise information technology (IT) include perceived usefulness (PU), perceived ease of use (PEU), attitude and the aim to use (ATU) (Evans et al. 2014, p.8).

When presenting classical research on Theory of Planned Behaviour, Ajzen commented that:

> *"the aim to do certain actions of various kinds can be forecasted with high accuracy from positions toward the action, subjective norms, and perceived behavioural control; and these goals, together with views of behavioural control, explain the considerable difference in actual behaviour."* Ajzen (1991, p.179)

The constructs of perceived ease of use and perceived usefulness are all attributes of perceived behavioural control, and all contribute towards the attitude that has a direct effect on the intention to use IT. The research used an existing theory to study the adoption, selection and use of modern SDMs in BSDI. The selected research theory guided the exploration of the nature, meaning and challenges of the research phenomena. The research, therefore, considered PU, PEU, ATU and behavioural intention to use (BI) as very critical constructs of the TAM2 theory needed for investigating their influence on the beliefs, attitude, and behavioural intention and ultimately usage of modern SDMs by BSDI, BSDC and software practitioners. As indicated in the theory of TAM2, these constructs are affected by circumstantial or external factors within the occurrence environment of the phenomenon.

The factors depicted in Figure 23, which have been identified as external variables that affect PU based on literature review and the researcher's exposure to the environment of study and according to the TAM2 theory, are discussed in more detail below.

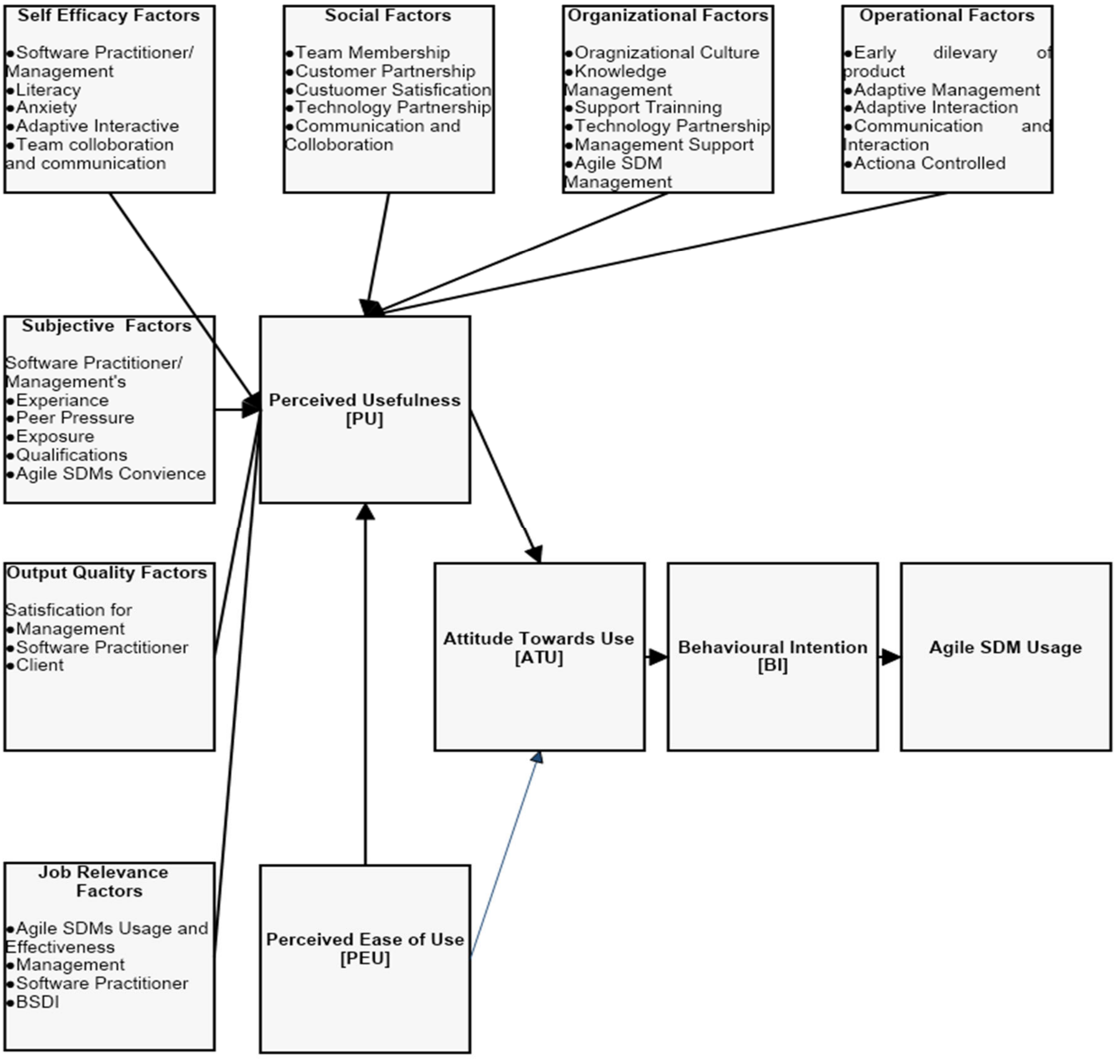


Figure 23: Research Study Conceptual Framework.

External Variables for TAM2 Constructs: Parkman, Litz and Gromik (2018, p.1253) showed that expanding the range of external theoretical factors utilized in the technology acceptance model (TAM) may increase the model's predictive value. Lim (2018, p.1) has also argued that TAM requires external factors that need to be integrated to extend and contextualize motivational influences that explain emerging realities around users and their interaction with technology. These external factors are perceptions, behavioural control, personal factor, social factors, descriptive factors and security (Lim 2018, p.1).

- **Organisational factor:** According to Park (2009, p.150), the context of a company or organisation influences both PU and PEU use of IT technology. When designing variables to measure the adoption of Agile SDMs, Chan and Thong (2009, p.804) considered the external variable organisation to qualify the constructs of PEU. Typical data values considered for the variable were top management support, organisational culture, communication, knowledge management and arduous relationships. Alhashmi, Salloum and Mhamdi (2019, p.27) linked organisational factors to like: managerial support equip software developers with the appropriate skill sets and training, global partnerships with leading technology providers. Navimipour and Charband (2016, p.730) observed that appropriate training programs provided by organizations assist employees to accept advanced technologies because they enhance their skills. Relevant expertise within the workplace also increases the possibility that new technology will be accepted by target users (Venkatesh , Thong & Xu 2016, p.328). The variables or constructs; training, management support, communication, capacity building, technical partnerships facilitated PU and PEU of new technologies or systems. These organisational factors or constructs were used in the conceptual framework to investigate the position of BSDC and how the influence to select SDMs.

- **Social factor:** The social context variable considers issues such as teamwork, shared understanding, external support, arduous relationships. Kukreja, Ahuja and Singh (2018, p.1236) consider the social factor as being affected by customer participation in decision making, team membership with great motivation, customer satisfaction, and cross-functional teams with flexible and skilled people. Alhashmi, Salloum and Mhamdi (2019, p.27) viewed the social factor as an organization's strategic factor because it is the way an organization handles its clientele to achieve clients' satisfaction. Their research found that stakeholder satisfaction had a direct influence on PU and PEU. In this research study, instead of using the social context variable as the abstract variable, the research adopted subjective norm as the substitute. In general, subjective norm implies the perceived social coercion to involve or not to adopt a construed certain behaviour. Subjective norm has a strong bearing on the behaviour of software developers given what other software developers do or what are the most adopted SDMs in the software development community.

- **Subjective Norm factor:** The subjective factor is related to an individual factor construct Park (2009, p.150), and it provides information about an individual's experience, training, career consequences, voluntariness and knowledge management as being attributes that affect subjective factor. Many studies have used subjective norms to determine acceptance of information technology or systems and their opinions varied on its influence on PU and BI. Park (2009, p.150) measured acceptance of e-learning as an IT technology used by students. The student's confidence in utilising e-learning for acquiring knowledge and communicating with a mentor showed a high level of e-learning self-efficacy in utilising the technology. The students experienced the value and convenience of using e-learning technology and therefore expressed their voluntariness and willingness to adopt the e-learning system. In this research study, SDM efficacy is defined as PEU and PU of Agile SDM in developing effective and efficient quality software for modern business processes. This research study, therefore, further defined a set of attributes such as qualifications, experience training and teamwork, convenience as variables to measure the construct.

- **Operational Factor:** Operational factors are defined by Al Kurdi et al. (2020, p.19) as desired services put in place to facilitate the provision of a service or product in a timeous manner to the client. According to the Agile manifesto, Agile SDMs have laid down operational guidance for all Agile SDMs and these are early delivery of software through software prototypes, adaptive interactions to facilitate changes, adaptive management to facilitate growth and self-sense in employees, controlled action plan and lastly communication and collaboration. Song and Kong (2017, p.865) described operational factors as facilitating factors because they play a role in the adoption of technologies by end-users. In the context of this study, the software clients would like to see their product delivered early and be involved in the development of the software as well. The software development practitioners would like to be supported through adaptive management that develops employee self-confidence, growth and appreciation of contribution. Al Kurdi et al. (2020, p.19) and Song and Kong (2017, p.865) found that operational or facilitating factors that influence PU and PEU TAM constructs

- **Self-Efficacy:** Self-efficacy is defined as a belief in one's ability to organize and implement actions to carry out designated types of performance and tasks (Bandura 1977, p.191). Several studies have found that self-efficacy has a direct bearing on TAM constructs of belief, PU and PEU in cultivating an individual's capability to organize and implement actions related to the adoption and use of information technology or system (Isaac et al. 2017, p.737; Joo, Park & Lim 2018, p.48; Zainab , Awais & Alshagawi 2017, p.1261). In this study, determining the self-efficacy of both the BSDCs and software practitioners is vital because it builds into a positive attitude and hence the behavioural intention to adopt and use Agile SDMs in the prevailing environment.

Job Relevance Factor: Job relevance (JR) was considered a factor that directly impacts PU according to the TAM2 theory. Job relevance reflects the users' perception of how technology is applicable and important to their jobs (Venkatesh & Davis 2000, p.186). Siyam (2019, p.2035) and Nadri et al. (2018, p.238) have established that JR has a positive effect on PEU. In the context of this study, job relevance is regarded as the perceptions of BSDI, BSDCs and software practitioners towards the use of modern SDMs like Agile in developing software and its influence on PU. Do BSDI, BSDCs and software practitioners follow the modern SDMs like Agile SDMs process, do they do project management planning, follow the guidelines of modern SDMs like Agile SDM, configuration management, does the project management team follow the guidelines do they stick to the budget, cost, schedule and quality embodiment

- **Experience Factor:** The TAM2 theory considers experience in combination with the use of information technology or system as a factor that can affect users' consequent intentions to use that technology. Several researchers advocate this point of view (Hornbæk & Hertzum 2017, p.1; Shea, Pickett & Li 2005, p.1; Ulmer, Watson & Derby 2007, p.59; Vaziri et al. 2016, p.1). In a study conducted by Shea, Pickett and Li (2005, p.1), 90% of over 900 faculty members surveyed after being exposed to concepts of teaching online courses were content with teaching online courses. Ulmer, Watson and Derby (2007, p.59), revealed that 137 faculty members surveyed on the use of online education had increased and positive attitudes towards conducting online education. Faculties with more experience on teaching online had significantly more positive perceptions on the overall effectiveness of instructor-student interaction and the ability to increase student performance in online courses (Ulmer, Watson & Derby 2007, p.60)

- **Output Quality:** TAM 2 theory engages the output quality external factor, which specifies how well the technology facilitates the accomplishment of a task thus ensuring delivery of a quality product (Venkatesh & Davis 2000, p.186). A review of literature has shown that there is a strong bearing of the output quality factor on PU (Bui et al. 2020, p.3; Purnama & Ginardi 2019, p.519; Wingo, Ivankova & Moss 2017, p.15). Purnama and Ginardi (2019, p.519) have observed that the output quality has a positive and significant effect on PU. Bui et al. (2020) in their study noted that the lecturers' usage of the learning management system (LMS) was perceived by the contract's perceived ease of use (PEU), subjective norm (SN) and perceived use (PU).

3.9. Summary

This chapter outlined the research approach, research design and framework suitable for understanding a study of this nature. A summary of the research methods used during the application of the methodology is shown in Table 5: The research questions addressing each study area.

Table 5: The research questions addressing each study area

Case Study	Aim	Research strategy/methods	Theoretical perspective	Analytical techniques
A framework for the adoption of agile software development methodologies in Botswana	Explore and determine what factors affect the adoption of modern software development methodologies by the Botswana Software Development Industry	Inductive /Qualitative	Interpretive	Content Analysis Cross Case Analysis Interpretative, structural and reflective analyses

CHAPTER 4: RESEARCH FINDINGS

4.1. Introduction

In Chapter 3, the research approach was presented as well as the design of this research effort. In this chapter, the findings based on the content cross-case analysis is presented. In addition, the chapter explores issues relating to data analysis and interpretation. To develop a framework for the adoption of modern software development methodologies (SDMs) in Botswana, interviews were conducted in Gaborone. Out of the twelve organisations initially scheduled for an interview, nine interviews were successfully carried out (i.e., 75% response rate). Case studies of the nine organisations informed the cross-case analysis that is presented in this chapter in terms of generated themes and patterns. The analysis considered an understanding of the perceptions of nine Botswana software development companies (BSDCs) (and their professional software developers) about the adoption and implementation of SDMs. Other than exploring the perceptions of BSDCs towards the adopted and implemented SDMs about Agile SDMs, the analysis also scrutinised the factors influencing the adoption and usage of SDMs in the Botswana software development industry (BSDI), with specific reference to the nine BSDCs. This information is gathered to guide the development of a framework in collaboration with the software development community for assisting and guiding BSDI, BSDCs and professional software developers to adopt and implement Agile SDMs in Botswana.

4.2. The Study and its Characteristics

The research targeted organisations that are actively and consistently tendering for software application development tenders in both the private and public domains in Botswana. In addition, tendering information from the tender board and some other organisations assisted to ascertain the qualification of the organisations for this research. Having used purposive sampling (explained in greater detail in section 3.2.4.2), the sample was composed of employees involved in software development at management and lower levels of the relevant organisations. The sampling was done such that from each organisation there was at least a single participant at each of the management and software development levels with relevant experience in development being a key requirement. In total, eighteen interviews were conducted in all the nine BSDCs that were available.

In accordance with the research ethics requirements, the researcher asked for consent from each of the participants to ensure that the participants would make time for the interview with an understanding that they were free to exit the interview at any time if they felt the need to do so.

4.3. Results Approach

As explained in section 3.2 of Chapter 3, this research study adopted the qualitative research approach since it seeks to develop a framework for the adoption of SDMs in Botswana. This approach was informed by the constructivist research paradigm, which is appropriate for exploring and explaining new knowledge. This is key to the research because the researcher seeks to conduct an exploration of the BSDI to understand the underlying circumstances that influence the SDMs choices of the industry. This paradigm is aligned with the research aim, which is basically to establish the position of the BSDI software developers in terms of their SDM choices and their perceptions on the adoption or non-adoption of traditional and Agile SDMs.

A qualitative exploratory research design was used to conduct the research study to develop a framework for the adoption of modern SDMs in Botswana. This was supported by a multiple case-study research strategy that allows replication of results. Case studies of study companies allow a cross-case analysis method to explore the working approaches and strategies adopted in selecting appropriate SDMs for software development in BSDCs.

Purposive sampling was used to select the sample of the study following studies conducted by Lambert & Lambert (2012, p.255) and Etikan, Musa & Alkassim (2016, p.1). This is because not many of the BSDCs are deemed active in tendering. Structured interviews that use open-ended questions were adopted for data collection purposes. To complement the case study approach, an additional data-collection technique in the form of the desk-document approach (also known as Qualitative Content Analysis) was used. Such as duplication of the data-collection instrument ensures and establishes research data reliability. In addition, it is envisaged that the multiplicity of the data collection techniques would lead to data convergence. Data analysis tools involved the NVivo 12 and Excel spreadsheet, which were used for summarising and coding data to allow further analysis of the data.

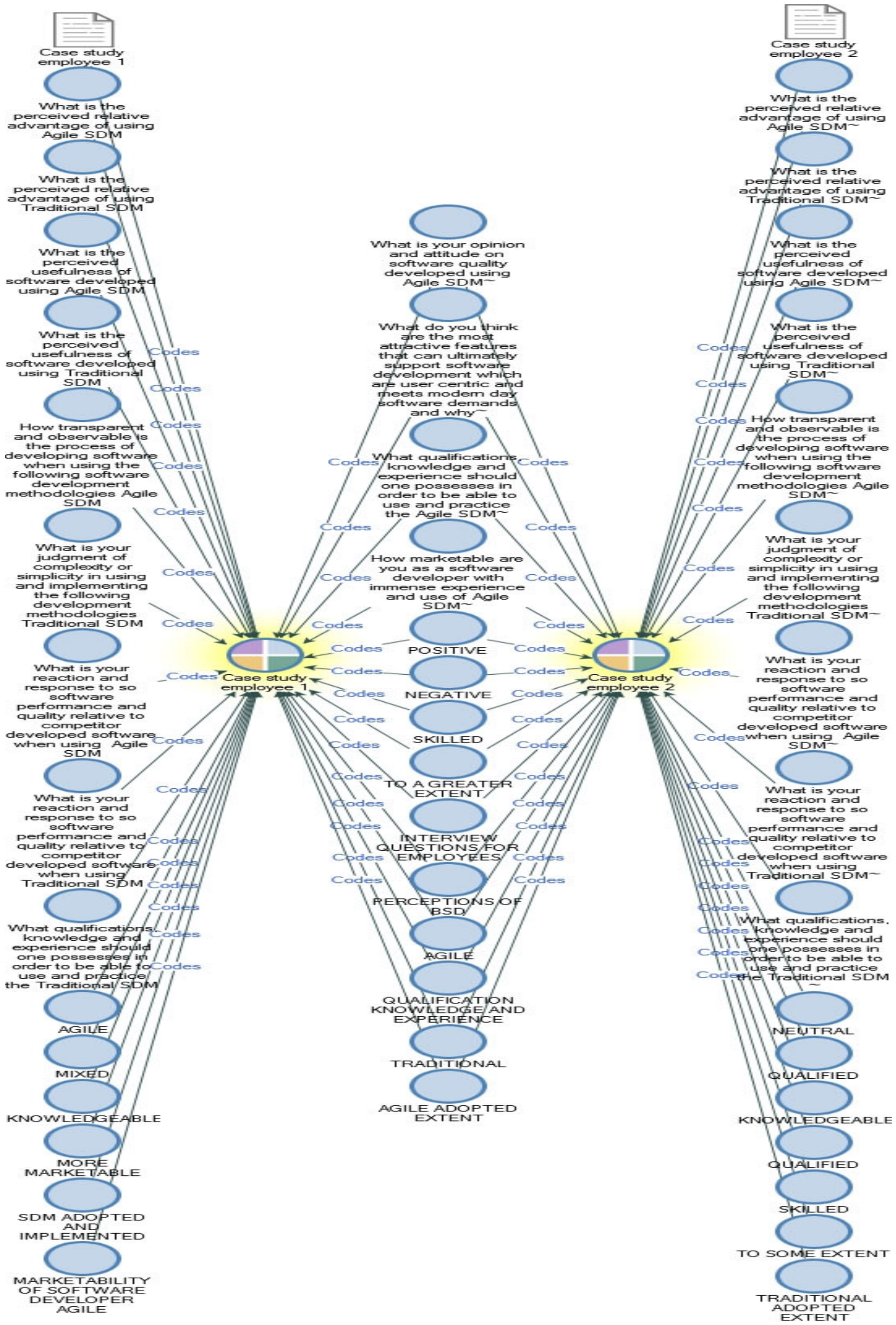

Case study employee 1
What is the perceived relative advantage of using Agile SDM
What is the perceived relative advantage of using Traditional SDM
What is the perceived usefulness of software developed using Agile SDM
What is the perceived usefulness of software developed using Traditional SDM
How transparent and observable is the process of developing software when using the following software development methodologies Agile SDM
What is your judgment of complexity or simplicity in using and implementing the following development methodologies Traditional SDM
What is your reaction and response to so software performance and quality relative to competitor developed software when using Agile SDM
What is your reaction and response to so software performance and quality relative to competitor developed software when using Traditional SDM
What qualifications, knowledge and experience should one possesses in order to be able to use and practice the Traditional SDM
AGILE
MIXED
KNOWLEDGEABLE
MORE MARKETABLE
SDM ADOPTED AND IMPLEMENTED
MARKETABILITY OF SOFTWARE DEVELOPER AGILE
What is your opinion and attitude on software quality developed using Agile SDM~
What do you think are the most attractive features that can ultimately support software development which are user centric and meets modern day software demands and why~
What qualifications, knowledge and experience should one possesses in order to be able to use and practice the Agile SDM~
How marketable are you as a software developer with immense experience and use of Agile SDM~
Case study employee 1
POSITIVE
NEGATIVE
SKILLED
TO A GREATER EXTENT
INTERVIEW QUESTIONS FOR EMPLOYEES
PERCEPTIONS OF BSD
AGILE
QUALIFICATION KNOWLEDGE AND EXPERIENCE
TRADITIONAL
AGILE ADOPTED EXTENT
Case study employee 2
Case study employee 2
What is the perceived relative advantage of using Agile SDM~
What is the perceived relative advantage of using Traditional SDM~
What is the perceived usefulness of software developed using Agile SDM~
What is the perceived usefulness of software developed using Traditional SDM~
How transparent and observable is the process of developing software when using the following software development methodologies Agile SDM~
What is your judgment of complexity or simplicity in using and implementing the following development methodologies Traditional SDM~
What is your reaction and response to so software performance and quality relative to competitor developed software when using Agile SDM~
What is your reaction and response to so software performance and quality relative to competitor developed software when using Traditional SDM~
What qualifications, knowledge and experience should one possesses in order to be able to use and practice the Traditional SDM~
NEUTRAL
QUALIFIED
KNOWLEDGEABLE
QUALIFIED
SKILLED
TO SOME EXTENT
TRADITIONAL ADOPTED EXTENT
Codes

Figure 24: Comparative analysis for case study employee 1 and case study employee 2 (NVivo 12)

The responses were populated into the summary nodes according to possible responses per question or theme (or node). Each interview script with questions and responses for a particular respondent was also coded as a case and the demographic information was coded as case classification in NVivo 12 Pro. At this point, the data were considered ready for analysis. For example, Figure 24 shows a comparative analysis of two cases, namely case study employee 1 and case study employee 2. Responses shown in the middle of Figure 24 are common to both cases and responses on either side of the common response are specific to each of the cases. Table 1 in Appendix A shows the demographic profile of employees interviewed from the companies under investigation.

4.4. Data Analysis

As defined by Khan and VanWynsberghe (2008, p.1438), a cross-case study is a research method that can gather knowledge from individual case studies. The case knowledge and information were collected and thereafter compared to produce new knowledge. In this research, employers and employees identified from nine BSDCs that actively tender for software development in Botswana were compared using information generated during the interviews. Cross-case and content analysis were adopted in this research for purposes of analysing the generated data. The NVivo software analysis programs in combination with Microsoft Excel were used to analyse employee and employer interview content. Interview data were analysed by identifying common themes and patterns portrayed from the data for each interview and all the employee interviews grouped. A similar method was implemented for the analysis of data generated from interviews with employers. Thereafter, a comparative analysis of employee and employer responses was undertaken. Specific patterns emerging from cross-case studies helped categorise constructs from the literature review and recognising within-group and between-group similarities. Related patterns were grouped and further coding of related groups was carried out to identify summarised categories. The NVivo software-generated relationships among the constructs by thematic coding to identify similarities and differences between the constructs. Data analysis findings were presented as content-response summaries, bar charts and hierarchy charts. Other diagrams such as comparison diagrams, word trees, word cluster diagrams, word cloud and several others were used to confirm the content analysis findings (please refer to appendix F for sample diagrams).

4.4.1. Demographic Profile Information

In this section, the demographic profile information of the respondents' explored specific attention paid to BSDC's owner/manager and software developer positions, duration of working or operation, gender, levels of education, and ages of respondents. All the respondents were males, reflecting that this is a male-dominated field in Botswana. Managers/owners who were holding more than one portfolio represented BSD organisations (employers). Some of the managers/owners held positions such as director, project manager, and consultant at the same time.

Figure 25 depicts the positions of the employers and the duration their organisations have been developing software. This diagram is significant for showing available skills in organisations and their experiences. Figure 25 reflects that some of the employers (managers) had dual roles as directors, chief executive officers (CEOs) or consultants. However, very few of the employers were holding joint portfolios (e.g., CEO and director). Many of the employers were also consultants. Furthermore, Figure 25 shows that despite the positions of the employers, many of the organisations have been developing software for 5 – 10 years in Botswana.

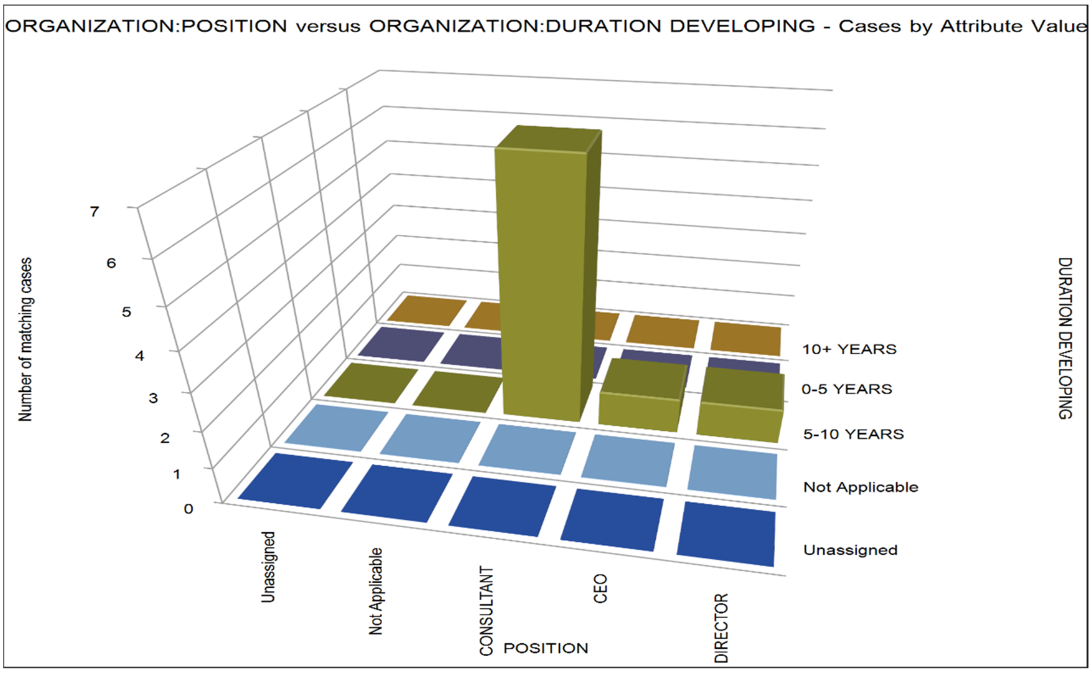


Figure 25: Employer positions and the organisations period of developing software (NVivo 12)

Figure 26 shows the number of software applications that had so far been developed by the companies against the time they have been operating. As shown in Figure 26, all the BSDCs have been developing software for between 5 and 10 years. The majority of the companies have developed 11 – 50 software applications and a few have developed over 100 software products. This indicates that the software development throughput of BSDCs is, by international standards, still low. This diagram is significant for showing the levels of software development in Botswana. These findings corroborate a Deloitte report on science and technology human resources that was commissioned by the Ministry of Communications, Science and Technology of Botswana (Final Report May 2009). The report established a challenge of limited innovation in Botswana owing to a lack of skilled personnel such as software engineers and finance and technology resources.

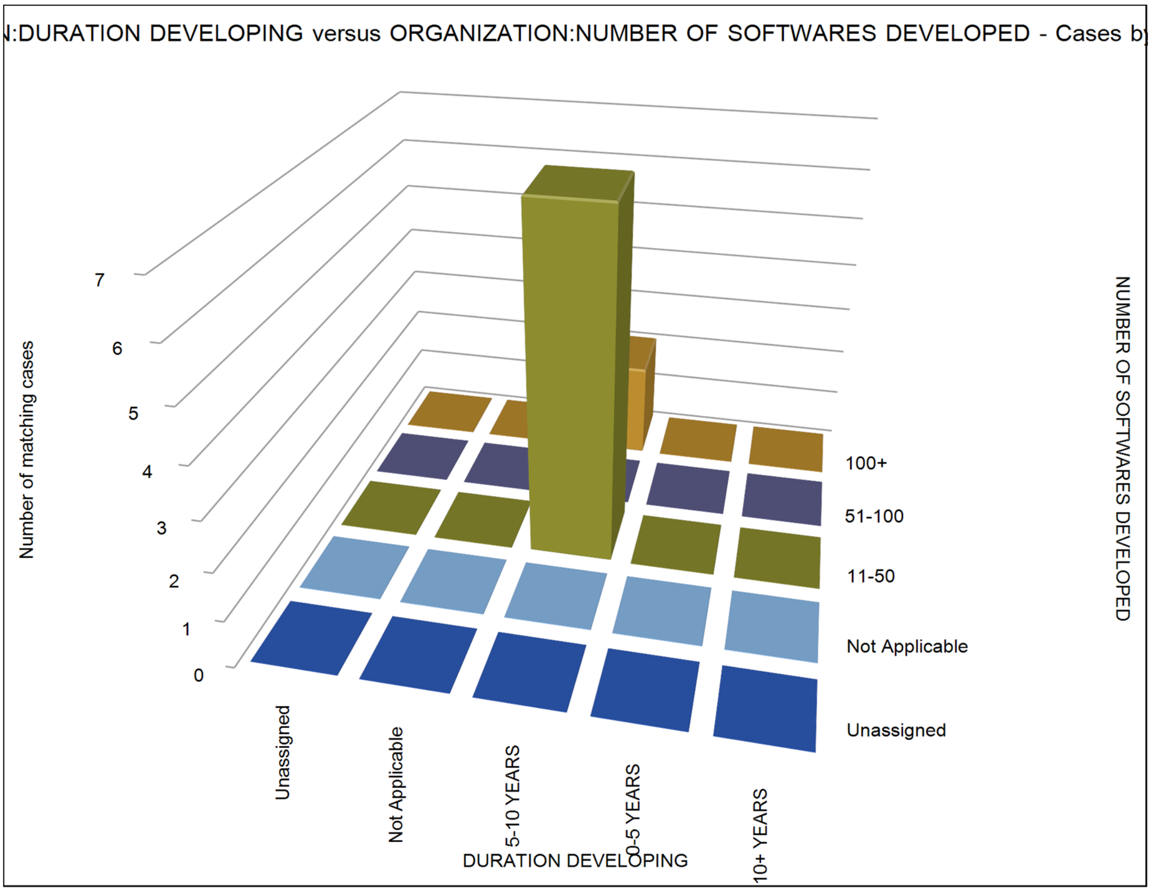


Figure 26:Software development in organisations with respect to time of operation (NVivo 12)

Figure 27 illustrates the most preferred SDM categories concerning employer (organisational) position. This diagram is significant for showing clearly softwares developed in organisations with respect to employer capabilities. As depicted in Figure 27, most BSDCs are led by managers who double as consultants and they prefer traditional SDMs. Consultants to some extent prefer the Agile SDMs and to a lesser extent the Ad-hoc SDMs. However, the managers who double as directors or CEOs prefer Agile SDMs. These findings suggest that the traditional SDMs are still dominant in Botswana.

:

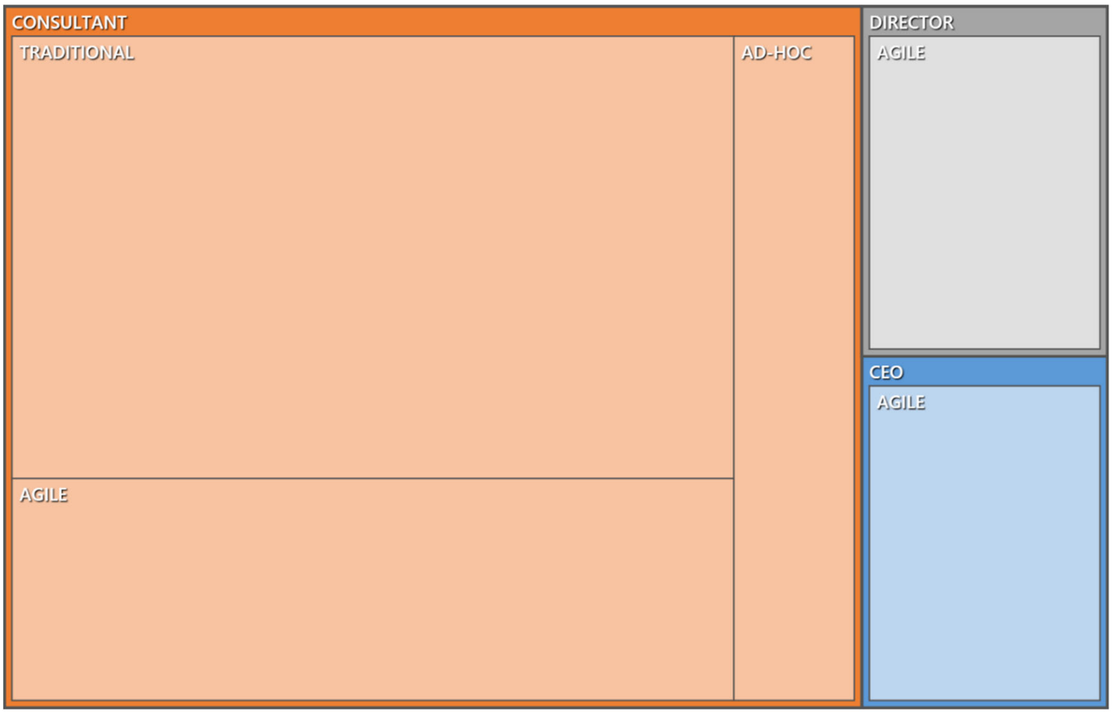


Figure 27: Preferred SDMs categories with respect to organisational position (NVivo 12)

Figure 28 shows a comparative analysis of the number of software developed in BSDCs about the preferred SDMs. This diagram is significant for showing the preferences of SDMs by organisations based versus quantities developed. It can be observed that the two BSDCs that prefer Agile SDMs have developed 11 – 50 software applications whilst the four BSDCs that use Traditional SDMs developed more than 100 software applications. Furthermore, the BSDCs that preferred Ad-hoc SDMs have developed 11 – 50 software products. These findings concur with earlier findings that suggest that the rate of software development in Botswana is very low since most of the BSDCs preferred using Traditional SDMs.

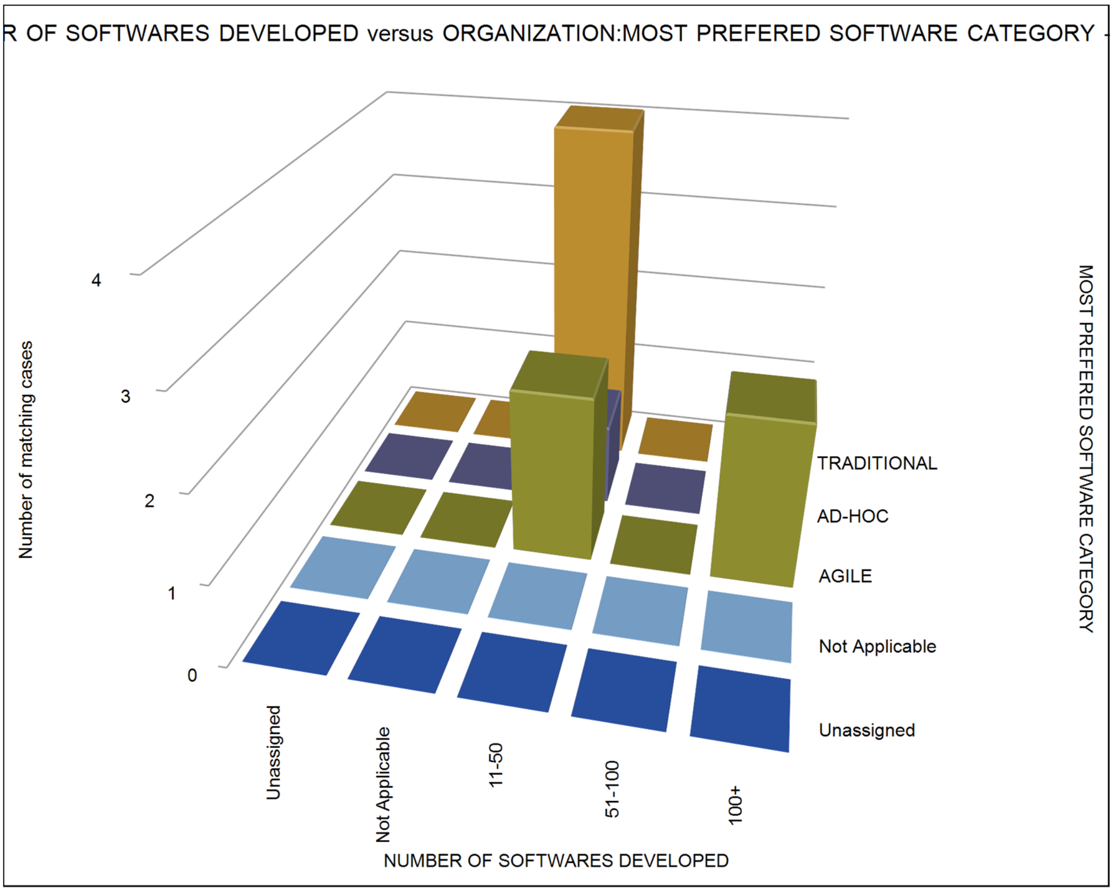


Figure 28:Software developed in organisations compared to most SDMs (NVivo 12)

Figure 29 shows the positions of the employees working in Botswana software development. This diagram is significant for showing the available skills of the employees in the BSDCs. It is evident from Figure 29 that the types of positions occupied by employees in the software development sector are varied. These positions range from developers or programmers, analysts, project managers, executives and others such as technical directors, software architectures and software testers. However, employees working as project managers were found to dominate the sector.

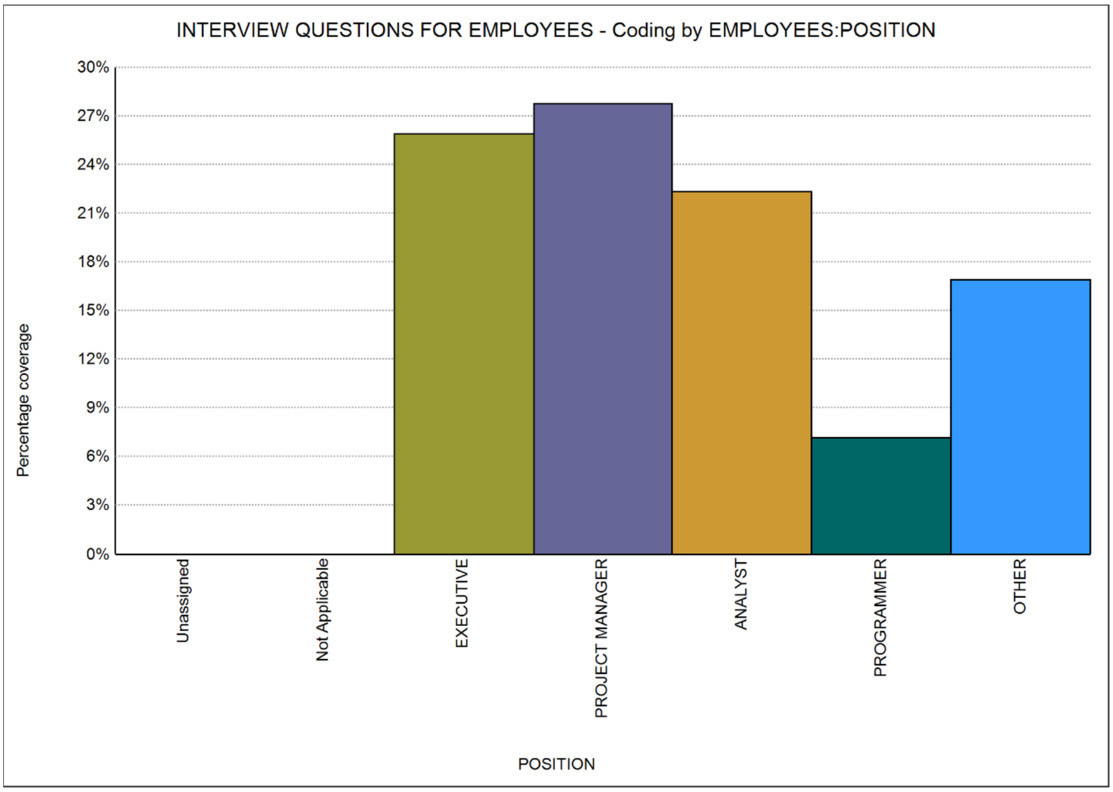


Figure 29: Employee positions (NVivo 12)

Figure 30 was significant for showing the distribution of employees' age groups in BSDCs. An age distribution analysis of the employees and their respective years working in their organisations developing software (see Figure 30) shows that most of the employees (78 %) are in their thirties, with very few (11 %) being over 50 years of age. The majority (89%) of the employee respondents had worked for their organisations for more than 10 years developing software applications. It was found that very few (11 %) of the employees had been working in their organisations for 5 years or less. These findings seem to suggest that most employee respondents were made up of young adults who are well experienced because they have been working in the BSD industry for many years.

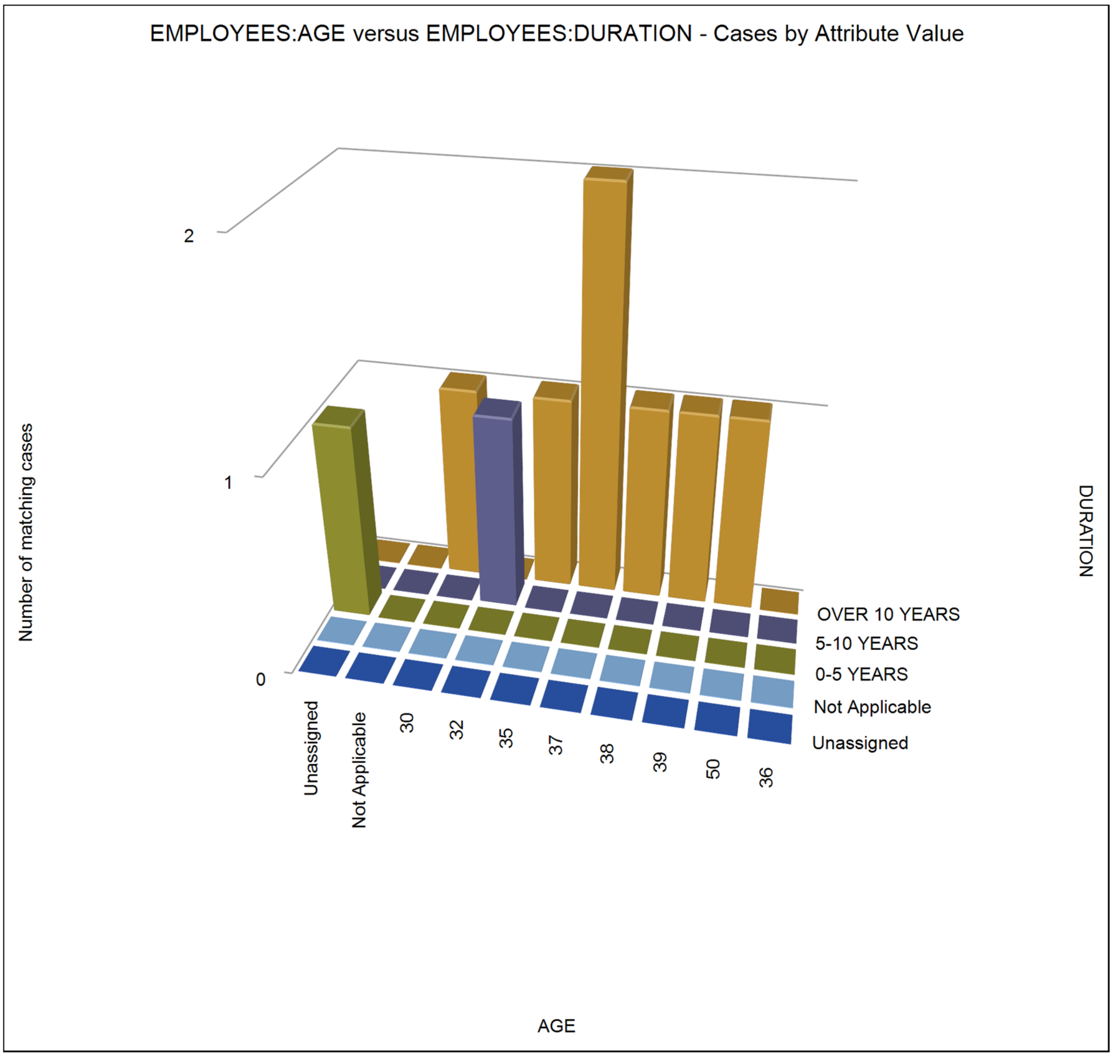


Figure 30:: Employees' ages and years of working in BSD companies (NVivo 12)

Figure 31 is significant for illustrating which positions have the most experienced employees. It is quite clear that the executive employees, project managers, programmers and others tended to be the most experienced employees in the sector because they typically work for their BSDCs for over 10 years. Analysts were found to have been working for their BSDCs for 10 years or less, and even for less than 5 years in some cases. From Figure 4.8, it was established that most employees had worked for their BSDCs for over 10 years.

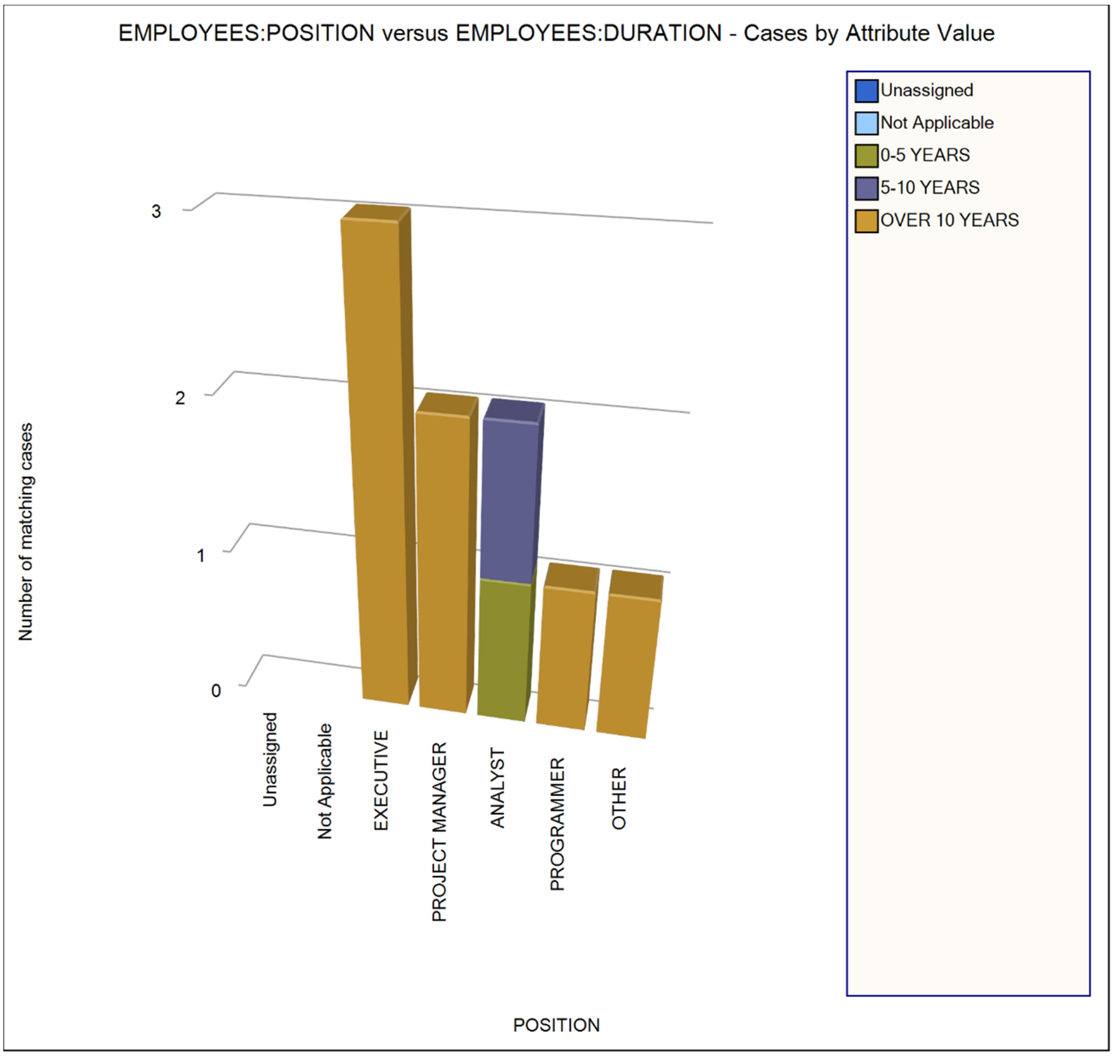


Figure 31: Employee positions and duration with organisations (NVivo 12)

4.4.2. Generated Themes and Patterns

Since the research aimed to understand the factors that inhibit the adoption, selection and use of modern SDMs in the BSDI, the study sought to develop a framework to assist the BSDI with the selection, adoption and implementation of modern and Agile SDMs in Botswana. Software development is a process that follows ordered phases. Swersky (2018), Tiky (2016, p.7), and Ruparelia (2010, p.8) supported this viewpoint by proposing a software development life cycle (SDLC) that can be adopted for software development. For purposes of this research study, the researcher organised the collected data according to the implied phases of SDLC as proposed by the authors. Thus, a step-by-step evolution of the desired framework informed the thematic-content analysis based on the cross-case analysis findings. The questions from the employers and employees were coded into nodes. Table 7 shows the nodes and the corresponding files and references for a particular node. The files give the number of respondents per node and the references show the different responses given by a particular respondent to a particular question (node). The initial nodes generated from the interview questions were further grouped into common themes or related statements according to the implied phases of SDLC. A summary of the employee interview nodes was generated by combining the initial nodes from the individual question nodes.

Table 6: Initial nodes summarised for employee interviews

NODES	FILES	REFERENCES
A. Interview questions for employees	9	280
1. Perceived relative advantage of using Agile SDMs	8	8
2. Perceived relative advantage of using Traditional SDMs	8	8
3. Perceived usefulness of Agile SDM during software development.	8	8
4. Perceived usefulness of Traditional SDMs during software development.	8	8
5. Opinion and attitude on software quality developed using Agile SDMs	9	9
6. Opinion and attitude on software quality developed using Traditional SDMs	5	5
7. Transparency and observability of the process of developing software when using Agile SDMs.	4	4
8. Transparency and observability of the process of developing software when using Traditional SDMs.	3	3

NODES	FILES	REFERENCES
9. Judgment of complexity or simplicity in using and implementing the Agile SDM	9	9
10. Judgment of complexity or simplicity in using and implementing the Traditional SDM	8	8
11. Reaction and response to so software performance and quality relative to competitor developed software when using Agile SDM	9	9
12. Reaction and response to so software performance and quality relative to competitor developed software when using Traditional SDM	9	9
13. The most attractive features that can ultimately support software development which is user-centric and meets modern-day software demands and with respect to Agile SDM	9	9
14. The most attractive features that can ultimately support software development which is user-centric and meets modern-day software demands with respect to Traditional SDM	9	9
15. Qualifications, experience and knowledge one should possess to use and practice the Agile SDM	9	9
16. Qualifications, experience and knowledge one should possess to use and practice Traditional SDM	9	9
17. Marketability of a software developer with immense experience and use of Agile SDM.	9	9
18. Marketability of a software developer with immense experience and use of Traditional SDM.	3	3

Table 8 is a summary of nodes obtained from the employee interview nodes determined according to the research objectives and other identified themes and patterns. The files give the number of respondents per node or node response and the references show the different responses given by a particular respondent to a particular node or node response.

Table 7: Summaries of nodes derived from employee interviews

NODES	FILES	REFERENCES
A. Interview questions for employees	9	280
SDM adopted and implemented		
1. Agile adopted and implemented extent	7	14
a) To a greater extent	6	11
b) To some extent	2	2
c) To a lesser extent	1	1
2. Traditional adopted and implemented extent	8	11
a) To a greater extent	1	2
b) To some extent	2	3
c) To a lesser extent	6	6
3. Perceptions of BSDCs towards adopted and implemented SDMs about Agile	9	96
a) Mixed	8	19
b) Negative	5	7
c) Neutral	6	9
d) Positive	9	61
4. Levels of knowledge management and support to software developers	0	0
5. Challenges in adopting Agile	9	9
6. Marketability of software developers with respect to Agile	5	8
a) Marketable	0	0
b) More marketable	5	8
c) Not marketable	0	0
7. Qualification, Knowledge and experience required	9	26
a) Agile	8	16
i) Knowledge	4	4
ii) Qualifications	3	3

NODES	FILES	REFERENCES
iii) Skills	7	9
b) Traditional	7	10
i) Knowledge	2	2
ii) Qualifications	3	3
iii) Skills	5	5

4.4.2.1. Perceptions of Respondents Towards Adoption or Non-Adoption of Modern Agile SDMs

Looking at the views of BSDCs on the use of SDM technologies to provide guidance to project progress, communication, achievement, results, identifying and managing risks, project cost control, project technical needs and project flow, engaging workmanship relationship and teamwork, several views were provided by the respondents. In this regard, organisation participant 1 (OP1) said that concerning Agile SDMs their company has a well-defined project management methodology for projects in the form of Project Management Body of Knowledge (PMBOK). PMBOK covers a collection of best practices, terminologies, project progress, communication, and relevant guidelines, which are accepted by the project management industry as standard. The respondent gave views that indicate that Agile SDM is just one of the SDMs being used by the company and the company is not entirely dependent on Agile SDMs. Agile SDMs are usually customised for specific situations. These sentiments were also expressed by OP4, who elaborated that it is difficult to plan and prepare for the future when using traditional SDMs. OP4 explained that with traditional SDM, it will be difficult to predict any additional future efforts or resources that will be needed in the project since the resources will only be known in the final stage of the project, that is, during acceptance when testing with the client.

However, OP2 shared views that show that their company prefers Traditional SDMs over Agile SDMs because an Agile SDM is viewed as a complicated methodology to follow. OP2 justified his views by explaining that with Agile SDMs, sometimes the project manager can lose control over the project and that this will lead to time and budget constraints and the scope of the project will not be well understood by the client as it may never end because of these changes. However, OP2 indicated that Traditional SDM is a fixed methodology whereby the project manager has control over the above-mentioned points and can finish the project in time because of crystal-clear procedures and it is budget free since the organisation will be in control of the cost. These views were supported by OP3 who appraised the Traditional SDMs.

Interestingly, OP6 seem to have positive views about Agile SDM as he indicated that their company uses Agile and Ad-hoc SDMs. Similarly, the following positive sentiments about Agile were expressed by OP7:

> *"We have a well-defined project management methodology for the Agile SDM. As a result, the organisation has experienced the following about Agile SDM, focus on user: Agile usually employs user stories with business-targeted acceptance methods to define product features. By focusing features on the requirements of real users, each feature incrementally delivers value, not just an IT component. This also gives the opportunity to beta test software after each sprint, giving good feedback early in the project and providing the ability to make alterations as needed. Improves quality: by breaking down the project into smaller units, the project team can aim for high-quality production, testing, and collaboration. Also, by producing frequent developments and doing testing and reviews during each stage, quality is enhanced by establishing and fixing irregularities and identifying errors quickly.*
>
> *We have seen solutions being delivered on time and with a higher degree of client satisfaction. By incorporating the ability to change, we have been able to incorporate feedback from demos, usability testing, and client and customer feedback.*

> *The Waterfall methodology stresses meticulous record keeping. Having such records allow for the ability to improve upon the existing program in the future. With the waterfall methodology, the client knows what to expect. They will have an idea of the size, cost, and timeline for the project. They will have a definite idea of what their program will do in the end. In the case of employee turnover, Waterfall's strong documentation allows for minimal project impact."*

Such a response shows that as much as the respondent has positive views about Agile SDMs, Traditional SDMs are still thought of as being very good.

However, OP8 expressed both positive and negative views about the use of traditional and Agile SDMs Furthermore, the respondent offered the following comparative and differential analysis of Agile and Traditional SDMs:

> *"Concerning Agile, stakeholders are indirectly involved in development. It has operational management procedures. We are doing project development through outsourcing. The SDM does not guide us to follow certain procedures. Agile is a methodology, which does not help the third party and needs more interaction with the customers. We have ISO procedures. Traditional SDM is pre-defined whilst in Agile a customer can inject new procedures. It will change the project. Conventional SDMs have boundaries whereas Agile has no boundaries. Agile can take a long time than Traditional. Agile is expensive for customers compared to traditional."*

In this case, this response shows that the respondent is talking about what they have heard or seen but had no first-hand experience as they outsource project development. OP9 said:

> *"Concerning Agile SDM, no comment as our organisation hasn't done anything on Agile SDM and concerning Traditional SDM, our organisation views it with respect and above all, it is manageable and quite straightforward."*

In general, the respondents demonstrated mixed perceptions about the adoption or non-adoption of Agile SDMs. The Traditional SDMs continue to be viewed positively even when the Agile SDMs are adopted by their company.

4.4.2.1.1. Perceptions On Software Development

The respondents were also probed to share their perceptions on the adoption or non-adoption of modern Agile SDMs to meet the demands of modern-day software. Perceptions of the respondents on software development were revealed through different targeted themes and patterns, which were generated from the gathered information.

i. The most attractive features that support software development

Employee respondents (EPs) gave several responses on the most attractive features that can ultimately support software development, which is user-centric and meets the demands of modern-day software for Agile SDMs. They said Agile SDMs are more adaptive and iterative, so they are more client-centric as they can ultimately support the dynamic environment using rapid applications to meet the changes in the business environment. In addition, Agile SDMs facilitate further improvements starting from the beginning of the first prototype. Since Agile SDMs allow further changes, it is possible to add new features that comply with the requested software functionalities of the user. The Traditional SDMs are viewed as being more supplier centric as they set the boundaries of scope.

EP6 concurred with EPs 1, 2, and 3 on Agile being adaptive and iterative, and the participant went on to say:

> *“Traditional SDM necessitate the user to deliver a detailed knowledge of the exact software functional requirements concerning the intended software and business analysis (gathering requirements from users before implementation) procedures. If the business analysis were done properly then implementation would be properly done to suit the user requirements. The procedures executed in the analysis and design are the main stages*

> *in the traditional methodological design. Of course, the focus of the whole project is there and therefore the effort at these stages is very important."*

This shows that the respondent has positive perceptions about both Agile and Traditional SDMs. More so, EP7 said:

> *"Traditional SDM works well where requirements are clearly articulated."*

In this case, the respondent is supporting the point of view that Traditional SDMs support straightforward instructions. On the same token EP9 said:

> *"Agile SDM's attractive features are that it is user involvement, easily noticeable progress on development, easy to change whilst Traditional SDM's attractive features are that it has a clear design and it has a fixed deadline to deliver."*

All these sentiments support the notion that the Agile and Traditional SDMs have some good elements about them thus suggesting that they are good for software development. However, some of the respondents feel comfortable with adopting Traditional SDMs for large projects and Agile SDMs for small projects. In contrast, other respondents are more comfortable with the sole adoption of Traditional SDMs.

ii. Perceptions on quality of software projects developed using Traditional SDMs

Responding to opinions and attitudes on the quality of software developed using Traditional SDM, most of the employee respondents did not give their opinions and attitudes, but rather gave general comments and repeated what they have said before. However, EP1 believes that the Traditional SDM is more based on design and analysis than user requirements and, owing to this, the Traditional SDM should be implemented on small projects without complex requirements. Quality is affected by the choice of development methodology. Interestingly, EP3 believes that Traditional SDM is flexible, does not have time constraints, and minimises budget because of structured procedures. However, EP7 seemed to differ, believing that SDM does not affect quality.

Moreover, EP9 is of the view that:

> *"Software quality developed using Traditional SDM is very stable and won't get bugs quite often."*

In addition, EP8 said:

> *"Software quality is based on the budget."*

These perceptions do not reveal much about the quality of software products that are developed using Traditional SDMs. However, the given perceptions were inclined towards using Traditional SDMs. Some of the respondents believe that software quality is based on budget whilst the others were of the view that software quality has nothing to do with SDMs used.

iii. Perceptions on quality of software projects developed using Agile SDMs

Most of the employees believed that Agile SDMs are flexible and easy to change as they are released in phases. Testing is factored in during the entire lifecycle of software development using Agile SDM thus facilitating regular examination of the working process and as the software is being developed. This allows the software client to provide any necessary adjustments to the Agile SDM project team for corrections hence leading to an improvement of the quality of the software being developed.

Agile SDMs can easily be modified to suit user requirements and flexibility. However, EP7 still believes that SDMs do not affect quality. EP8 believes that the quality of software projects developed using Agile SDMs is better and is evolving. These findings do not indicate any real experience relating to quality that was obtained by the respondents following the testing of Agile-developed software. They are just mere comments emanating from the general knowledge of the respondents that shows how casual the respondents or the BSDCs can be regarding their products.

iv. Perceptions on complexity or simplicity when using and implementing the Agile SDMs

Judging from the complexity or simplicity of using and implementing the Agile SDMs, EP1 gave a detailed description of the opportunities that Agile SDMs present to assess the direction of a project throughout the development lifecycle. The respondent also indicated that the agile methodology is described as iterative and incremental. However, EP2 was more specific and went on to state that:

> *"Agile SDM is more complex, there is confusion because there are a lot of discussions during development to apply and control. The project manager has to monitor and control compared to traditional."*

This comment suggests that EP2 found the use and implementation of Agile to be difficult or complex.

In contrast, EP3 explained that change is expected when using Agile. This means requirements emerge and evolve as the product is being developed and it is important to have an actively-involved stakeholder who understands and can make necessary decisions.

Similarly, EP4 maintained a positive outlook on the complexity or simplicity when using and implementing the Agile method and said:

> *"It is all about light manoeuvrability and sufficiency to facilitate future development. However, Agile methods involve planning what one wants and then adapting these plans to achieve the results."*

In line with EP4's subscription to the simplicity of Agile SDMs, EP7 pointed out that Agile helps teams to respond to unforeseen circumstances through sprints. EP7 delineated the ability of Agile SDMs to assist teams to achieve deliverables that are aligned with the client's expectations and the project team's targets. In agreement, EP8 and EP9 alluded to the straightforward and easy to follow nature of Agile. However, the respondents pointed out that Agile can be a disaster if it is not properly controlled because the iterations can protract the process until the user is satisfied.

Most of the views and comments of the interviewed software developers accentuate their mixed feelings about Agile use and implementation. While some respondents view Agile as a simple procedure, others view it as a complex or delicate procedure needing extra effort to produce good results. It suffices to say that some of the respondents see Agile as a non-ending process to follow which aligns well with the non-ending nature of software.

v. Perceptions on complexity or simplicity in using and implementing the Traditional SDMs

As much as most respondents have indicated an inclination towards Traditional SDMs, their perceptions of the complexity or simplicity of Traditional SDMs was mixed. This was particularly evident in EP1's response who proclaimed that they do not find Traditional SDM to be simple to use and implement but rather strict or harsh in terms of opportunities. In contrast, EP2 deviated from this view by mentioning that Traditional SDMs are simpler, more structured and easier to apply.

In addition, EP3 discussed said:

> *"In traditional development projects, we write a big spec up-front and then tell business owners how expensive it is to change anything, particularly as the project goes on. In fear of scope creep and a never-ending project, we resist changes and put people through a change control committee to keep them to the essential minimum. Agile development principles are different."*

It is clear from the above comment that the respondent has good knowledge and understanding of their operations when using Traditional SDMs. However, the respondent was unable to say much about Agile (save for saying that Agile principles are different). Simultaneously, the response brings to attention the fears they have when using Traditional SDMs without necessarily stating whether this complicates things or make things better for them.

Furthermore, EP4 explained that:

> *"Software engineers under Traditional SDM came along with the well-known engineering methods of controlling the processes. These methods are applying a disciplined approach where the stages of design and building are well predictable. Detailed stages of analysis and design precede the stage of building the software. These methodologies are well documented and thus are quite complex to apply."*

This respondent unmasks the documentation challenges experienced by software developers when implementing Traditional SDMs. The software developers may experience challenges in following the strict and non-flexible instructions associated with Traditional SDMs. The notion of the strict and non-flexible nature of Traditional SDMs was corroborated by EP7. EP7 said the following:

> *"As this process is sequential, once a step has been completed, developers can't go back to a previous step i.e., not without scratching the whole project and starting from the beginning. There's no room for change or error, so a project outcome and an extensive plan must be set in the beginning and then followed carefully."*

EP8 agreed with EP4 on the issue of Traditional SDMs being a complex procedure attached to a stringent process flow to be followed with regards to documentation. Similarly, EP 9 also concurred with EP4 and EP8 that Traditional SDM is a complex procedure to follow as one needs to design the whole product before development. However, EP9 found Traditional SDMs to be simple to manage and complete the product.

The employee respondents indicated that, although most of their companies use Traditional SDMs, these types of SDMs are not simple because they provide limited opportunities to develop the product. Extensive documentation associated with Traditional SDMs and limited opportunities that are available to effect changes before the software application is delivered to the client make Traditional SDMs complex to adopt and implement. However, very few of the respondents found the instruction of Traditional SDMs easy to follow.

vi. Perceived relative advantage of using Agile SDMs

Several advantages of using Agile SDMs such as flexibility of this methodology have been identified. The flexibility of the Agile SDM provides enough room to effect changes to the software being developed thus leading to increased customer satisfaction and customer retention, which are often and ultimately drivers of customer lifetime value. Customer lifetime value is a currency that can be directly linked to the same company, and can therefore be exploited to generate revenue for the company. Agile SDM is people-oriented, that is, it considers customers, developers, other stakeholders and end-users placing value and importance on experts in Agile SDM, issues that impact the software project’s amicability, and the required talent and skills. Agile is adaptive, that is, change is good at all stages of the project. In addition, Agile SDMs is in conformance to actual, empirical process, has fewer risks, and it gives quick results. Agile is easy for software projects development where the software client’s scope is not specific. Agile has less documentation. In addition, some of the respondents said:

> *“We can keep on billing the customer as the cost is controlled in terms of customers and it is best compared to traditional in business.”*

Furthermore, EP3 gave a long explanation by saying the following:

> *"The Agile methodology allows for changes to be made after the initial planning. Re-writes to the program, as the client decides to make changes, are expected. Because the Agile methodology allows you to make changes, it is easier to add features that will keep you up to date with the latest developments in your industry. At the end of each sprint, project priorities are evaluated. This allows clients to add their feedback so that they ultimately get the product they desire. The testing at the end of each sprint ensures that the bugs are caught and taken care of in the development cycle. They will not be found at the end. Because the products are tested so thoroughly with Agile, the product could be launched at the end of any cycle. As a result, it's more likely to reach its launch date."*

This response provides evidence of an agreement among the respondents' perceived relative advantages of using Agile SDMs. EP9 further added,

> *"Developers / Analysts have a clear understanding, little rework is needed in case of any change, the product can be phased out as soon as it is tested and accepted, and progress can be noted easily."*

These responses reveal that the respondents have positive perceptions about using Agile SDMs. The respondents demonstrated that although they may not be using Agile that much, they know the benefits of using Agile.

vii. Perceived relative advantage of using Traditional SDM

Traditional SDMs have several advantages that have been reported by different players in the field. Some of these advantages include getting a clear picture of the product before development (such a picture enables developers to think of all possible scenarios) and having a fixed date of delivery and a scope that is fixed. In addition, some of the respondents mentioned that a Traditional SDMs has more structures, has risks that are easy to control, and encourages a standardised program structure. The respondents went on to mention that the Waterfall SDM places emphasis on good record or document creation and management. The created documentation facilitates further software product maintenance and improvement in the future.

With the Waterfall methodology, the client will have an idea of the size, cost, and timeline of the project. The client will have a definite idea of what their program will do in the end. In the case of a high employment rate of turnover, Waterfall SDM's strong record management allows for minimal project impact. In addition, employee respondents went on to state that Traditional SDMs are easy to manage, especially with regards to software projects where the client scope is specific. These responses support the position that although the respondents find Traditional SDMs complex to adopt and implement, they have also had first-hand knowledge and experience of the advantages of Traditional SDMs.

viii. Perceived usefulness of software developed using Agile SDMs

Several related and varying views were given concerning the use of software developed using Agile SDMs. Respondents stated that Agile SDMs are interactive processes that result in better-quality projects. Agile SDMs are useful when early production of the software product is urgent versus the total quality of the software product. The Agile SDM assists the clients who do not want to alter the scope of their software project but do not have a clear picture of what functional requirements should the software application have for the intended task. In addition, respondents agreed that Agile SDMs are useful in cases where transparency and early predictable delivery are needed. For example, EP1mentioned that,

> *"Agile is useful as it gives a better focus on customers, better prioritisation of development and focuses on the product, improved productivity, increased morale (often tied to continuous integration with deliverables at the end of each sprint), and more reliance on test-driven development."*

This shows that the respondent has positive perceptions about the usefulness of Agile SDMs.

ix. Perceived usefulness of software developed using Traditional SDMs

Software developed using Traditional SDMs have different uses known to and identified by employees interviewed. Employee respondents asserted that software developed using Traditional SDMs is useful in that it is very stable and often bugs-free, allows control of structure processes, and is useful when there is a clear and specific outline of the functional requirements of the software to be developed. In addition, a Traditional SDM is especially useful when software clients do not have the comprehension and capacity to change the latitude of the software project. Once a Traditional SDM has started, issues such as definition, speed, and vital to success are not a priority.

x. Mixed perceptions on the Agile and Traditional SDMs by employees

Some of the employee respondents had mixed perceptions towards the Agile and Traditional SDMs and this necessitated a comparative analysis of views on the SDMs. Some of the respondents believe that Traditional SDMs are based more on design and analysis than Agile, which is more based on user requirements. Owing to this discrepancy between the two SDMs, Traditional SDMs should be implemented on small projects without complex requirements. On one hand, other respondents are of the view that Traditional SDMs are more supplier centric since they set the boundaries of scope than the Agile SDMs. On the other hand, although the Traditional SDMs are simple to manage, their extensive and non-flexible documentation processes are extensive and difficult to follow as opposed to Agile, which allows changes at every stage of the software development process.

Important to note was that the company participants concurred with the employee participants in that Agile SDMs are better suited to small projects and are mainly used when the client is not sure about the detailed requirements of the application to be developed. However, the company respondents went on to indicate that, in some projects, they use SDMs that are recommended by technology vendors or by clients.

xi. Negative perceptions on using Agile and Traditional SDMs

As much as the respondents had generally positive perceptions about the Agile and Traditional SDMs, others openly gave negative perceptions about the SDMs. For example, OP3 commented that:

> *"In terms of the Agile SDM, the project manager should monitor carefully the development of the project at every stage. There is a need for more team coordination and constant interaction between customer and developer and this is difficult for the project manager. Because of continuous interaction between customer and development team, the project manager sometimes loses his control over the project, and this led to time constraints, budget constraints and the scope of the project becomes difficult to understand for the client. Thus, if the project manager loses control of the process, project failure will be the result."*

OP2 agreed with what OP3 said and further explained that:

> *"Agile doesn't have a start and an end process because it is iterative."*

As an additional comment, OP8 asserted that:

> *"Agile is expensive for customers as compared to traditional."*

Pointing out the negative aspect of Traditional SDM, EP1 said:

> *“What is negative about the traditional SDM is that in Waterfall, development teams only have one chance to get each aspect of a project right.”*

Furthermore, EP2 agreed with OP2 and OP3 on the need for many discussions to take place during the development process of Agile SDMs to control and monitor the processes. However, EP2 went on to say that the Traditional SDMs need to be replaced with a new method.

EP6 and EP7 concurred on the extensive documentation required for the traditional methodologies and their complexity in respect of their application. However, EP7 elaborated further with regards to traditional methodologies by saying:

> *“The process is strictly sequential, once a step has been completed, developers can’t go back to a previous step. There’s no room for change or error, so a project outcome and an extensive plan must be set in the beginning and then followed carefully.”*

These responses reveal that the respondents also have negative perceptions of the SDMs.

xii. Neutral SDM perceptions

Some neutral perceptions were provided by the respondents with regards to SDMs. These neutral perceptions included comments such as choosing the most appropriate methodology to suit the required product. Whereas some respondents indicated that the quality of the software was dependent on the budget, other respondents mentioned the fact that an SDM does not affect the quality and traditional approaches used planning as their control mechanism. For example, OP6 said:

> *“We are using Ad-hoc SDM not traditional.”*

In addition, OP9 said:

> *"In terms of Agile SDM, we have no comment as our organisation hasn't done anything on Agile SDM."*

These responses point out the fact that in some instances the respondents do not openly express support or preference for any specific SDM.

xiii. Positive SDM perceptions

Most of the respondents had some positive perceptions about both the Agile and the traditional SDMs. For example, OP1 was very positive about Agile SDMs and indicated that the Agile SDM is more people-oriented than traditional SDM since it places more emphasis on people factors on the project amicability, skill and talent. Furthermore, the respondent signalled that Agile is more adaptive and the progress of the project is experienced by the customers at all stages, unlike the Traditional SDMs where the final product is seen by the client at the acceptance testing stage. Also, it was mentioned that Agile is more flexible for clients because the clients can suggest improvements, review the application, change or add requirements at any time during the project. Unlike the Agile SDM, changes in the Traditional SDM can only be made at the information-gathering stage. For the Agile SDM, the user is constantly kept in the loop and any changes can be easily accommodated at any stage of the development phase.

In contrast, OP2 had positive comments to say about both Agile and traditional SDMs. OP2 said:

> *"Agile software is best in terms of communication, adapting and developing. Communication increases the likelihood of developing the software with a high success rate. If the customer is aware of the methodology, it becomes easy. Traditional SDM is a fixed methodology, project managers have control, and the project can be completed on time*

> *because of crystal clear procedures. With Traditional the cost is controllable."*

In addition, although OP6 said they are leaders of Agile and Ad-hoc SDMs, added that Agile provides more opportunities for the team to truly understand the client's vision. OP9 was unapologetic about its allegiance to traditional SDMs. The respondent confidently mentioned that:

> *"We have developed software using Traditional SDM only, as far as Traditional SDM is concerned, utilisation and adoption is very good. It is strongly adopted and utilised for more than 5 years. We managed to implement it well with minor challenges. Our organisation views the Traditional SDM with respect. As far as my understanding of Agile SDM is concerned, it involves users very closely and delivers quality products and avoids any last-minute challenges."*

xiv. Perceptions on Organisational and Technical Infrastructural Support Rendered by the Organisation

Companies have varying perspectives on Agile and Traditional SDMs, and this influences the kind of support they render to their clients. OP1 said:

> *"We have a research and development (R&D) team unit whereby our team researches the latest SDMs. They share their knowledge and ideas with other employees about choosing appropriate SDMs for projects. Then a project is chosen where the SDM will be implemented on a pilot approach where the implementation team including the department head and the R&D team will be working closely on the latest SDM. The project will be implemented and SDM will be reviewed periodically to see if the objective is attained or if there is a need to fine-tune the approach. If it is successful, then the R&D team will share their findings with everyone concerned with knowledge management. They will train the interns, conduct orientation to the new employees and train all the other employees to enhance their knowledge."*

This suggests that this company provides the requisite support for implemented SDMs. Furthermore, the respondent alluded to the fact that they have the requisite technical infrastructure for developing software using Agile SDMs. Research is conducted and relevant technical training is provided to support staff members involved in software development. It is noteworthy that research, training and sharing of knowledge and ideas are at the core of the company's knowledge management strategy. Several respondents including OP4 and OP6 mentioned that they have some research and development teams that share new information with the relevant technical departments in the company.

In addition, OP2 stated the following about their organisational support for SDMs:

> *"Through induction process, we explain to the staff what customer is expecting and what type of prototype we are supposed to follow. Based on the project size, if the project is small, we will choose standardised software development methodology (Waterfall methodology) and for larger projects, we will use Agile SDMs. We will also encourage software developer experts (Teams) to do a thorough investigation and constantly learn about the latest SDMs, which are available in the current markets. Based on the project they will select the particular methodology according to the customer needs. Agile SDMs are not fixed in the style, it is something, which can be adopted for a particular situation. Based on the experts' ideas we will choose the methodology. Teamwork is encouraged; knowledge and different ideas can be shared with other employees and implemented based on the project size and customer needs. In terms of agile, teamwork is encouraged. In terms of standardised SDMs (Traditional) the business database is endless, and each software developer acts on different specifications. In terms of Agile, more verbal communication is needed and it is translated to specific needs."*

Additionally, the company has standardised policies and procedures to support SDMs in software development. Policies and procedures in combination with expert involvement allow for some knowledge management in the organisation. C-sharp and Visual Basic are used to give technical infrastructural support. This suggests that Agile SDMs and Traditional SDMs are supported when developing quality software both from the organisational point of view and from the technical infrastructural support rendered by the company.

From OP3 responses, it shows that the company is well supported with regards to the use of Agile and Traditional SDMs in the development of quality software applications. This is reflected in the following response from OP 3:

> *"We make sure to communicate the policies and procedures to staff and ensure that they are being used and followed. We also develop a training program that all staff and volunteers have to undertake. In addition, employers and volunteers are given information as to where they can access guiding policies and procedures. We create posters or process charts to post around the workplace to keep key messages firmly in mind. Also, we regularly promote a policy or procedure in internal newsletters or at team meetings."*

In addition, this type of response shows that knowledge management is practised at notable levels because different platforms are availed to staff to access important information for future use. Communication and monitoring of the implementation of policies and procedures indicate that there is a flow of important information in the company about software development.

OP7 stated that, in terms of knowledge management, they conduct regular training and workshops for staff members. This is over and above the many online learning platforms that are made available to staff to equip themselves. In addition, the company has the requisite resources and infrastructure to support both Agile and traditional SDMs. This shows that Agile and Traditional SDMs are supported by the company, through the provision of the relevant technical infrastructural support, when developing quality software.

OP8 went on to say,

> *“Our organisation predominantly implements ISO certification, which is the first step of a process of continual improvement that will provide an organisation with the necessary management tools to improve working practices throughout the entire organisation. ISO 9001 certified organisation has implemented Quality Management System requirements for all areas of the business including facilities, people, services, training, and equipment.”*

The respondent proceeded to say,

> *“Our organisation has different knowledge repository systems including different processes and methodologies, which are mapped through the system and shared through internal intranet which has procedures, write-ups, and videos which are meant for the employees. New interns are oriented and trained via the repository system through the intranet. Operational procedures are mapped in. SDM will be communicated and trained to the development team during the beginning of every project. Every 3 months our procedures are reviewed and internally audited. We have a complaint database which is created by the operation manager at the end of every project. Through the database, we store the complaints which are launched by the customers and stakeholders. This feedback can be used in future project development to enhance the quality.”*

The above-mentioned comments show that the company in question has a well-supported SDM system that has allowed the organisation to be International Organization for Standardization (ISO) certified and a different knowledge repository system that allows for knowledge management and implementation.

The company has put in place relevant policies and procedures that create an enabling environment for software development processes to be carried out.

Furthermore, OP9 drew attention to the fact that, unlike most companies, Agile and traditional SDMs are not well supported in their company since they use methodologies that are suggested by Oracle.

All the responses given by the different company respondents seem to suggest that software development is largely supported by the respective companies. There are some technologies and mechanisms that are being followed to allow software development to successfully take place. Information is being generated, developed, and presented on different platforms for the benefit of both the companies and interested company stakeholders. Technological infrastructure is also provided to support software development in the companies. However, OP3 gave a general comment and indicated that their support is geared towards Oracle methodologies. The OP3 respondent commented as follows:

> *"Knowledge is one of your key assets. Like your staff, your money, your customers, your brand, process, and policies. It is one of your more valuable assets too - just imagine how your organisation would perform if you did not know, and your staff did not know what the goal and drive of the organisation is! It is good practice to manage your valuable assets. You almost certainly have implemented financial, people, customer relationship, brand management, project management and implementation. So, it makes sound business sense to implement knowledge management too; to derive maximum business benefit from the invisible asset which is the operational knowledge held in the heads of your employees. We are an Oracle partner; we are more biased toward Oracle's implementation methodology."*

4.4.2.2. Adoption or Non-Adoption of Modern Agile SDMs by the BSDCs

Since the study was aimed at exploring and determining the factors that affect the adoption of modern SDMs by the BSDI, it was important to establish the drive behind the adopted and implemented SDMs in the software development organisations in Botswana. In this section, the SDMs that are being adopted and implemented as well as the driving force behind these adoptions and implementations are explored. This is important to fulfil the quest of this research to devise a framework for the adoption of modern SDMs in Botswana.

4.4.2.2.1. Adopted and Implemented SDMS and Levels of Adoption

Looking at the SDMs being adopted and implemented by the companies under investigation, Organisation Participant1 (OP1) and OP8 indicated that, unlike Traditional SDMs, Agile SDMs adoption and implementation requires the project management team to have control over every stage of project development. They elaborate further by saying that Agile SDMs need more monitoring, continuous follow-up, experience, and experts to successfully develop the project. Furthermore, OP2 explained that their company has put in place standardised policies and procedures for Agile adoption and implementation; this suggests that they are using Agile SDMs. OP3 and OP4 also added that the success of using Agile SDMs was dependent on how well the developer involves the customer in the project delivery phase.

OP5 said: "*We are using the traditional SDMs*", whilst OP6 opined, "*We are a leader for Agile and Ad-hoc SDMs*." Whereas OP7 indicated on one hand that: "*We are market leaders in software development in either of the SDMs, that is, Agile and Traditional*", OP8 alluded to the fact that they use a conventional methodology, for which they have heavy documentation. On the other hand, OP9 said: "*We have developed software only using Traditional SDM*."

These results show that out of the nine companies that were interviewed, six were able to clearly articulate the SDMs that they have adopted and implemented. The balance of the companies was just responding in general without giving specifics. From the six companies that listed the SDMs they have adopted, it was clear that the companies that use the Traditional SDMs were more direct and confident, unlike the companies that mentioned that they use the Agile SDMs.

4.4.2.2.2. Agile adoption and implementation

Although most of the respondents commented about the adoption and implementation of Agile SDMs, they did not make clear whether as a company they adopted and implemented it. Such comments seem to indicate that although some of the software development organisations in Botswana have some level of knowledge about Agile SDMs, they might not have adopted them for software development. For example, OP8 only talked about having heavy documentation on conventional methodology but failed to elaborate on whether they were using Agile SDMs or not. Nonetheless, a few of the organisation respondents were able to openly divulge their usage of Agile SDMs during software development. In fact, some of these respondents went on to proclaim their market leadership status in Agile adoption and implementation. These findings clearly show that Agile SDMs are to some extent being adopted and implemented by BSDCs. Most employer respondents agreed that the Agile SDM is more transparent, involves planning for the needs of the user, helps the developers to stay focused, and is customer-oriented.

4.4.2.2.3. Traditional adoption and implementation

Whereas most of the organisation respondents confirmed that their organisations use traditional SDMs, it was also evident that several other organisations were using a combination of traditional and Agile SDMs. Very few organisations indicated that their use of SMDs was exclusively limited to traditional SDMs. Similarly, with regards to the acceptance and application of Agile SDMs, some of the organisation respondents pointed out their market leadership status in the adoption and implementation of traditional SDMs whilst others indicated that they were mere followers. Most of the respondents shared similar sentiments about traditional SDMs (e.g., traditional SDM is modular, it is difficult to implement changes after completion, it has extensive and complex documentation, and the development process is observable only after completion of the entire product or at least a module). Nevertheless, not only did most of the BSDCs interviewed mention that they were comfortable with using the Traditional SDMs, but they also indicated that they work well with them. Thus, the findings of this study seem to suggest that most of BSDCs use Traditional SDMs.

4.4.2.2.4. Comparisons of Agile and Traditional SDMs

Most of the respondents indicated that, unlike Traditional SDMs, Agile SDMs require a project management team to attain some level of control over every stage of the project development. Also, the respondents indicated that, in project development, Agile SDMs always have an edge over traditional SDMs. Furthermore, the respondents indicated that Traditional SDMs involve a lot more documentation compared to Agile SDMs. While changes are easily accepted during the development process of Agile SDMs, Traditional SDMs are slightly complicated concerning introducing any changes during the development process since changes are only allowed at the end of the traditional SDM cycle. Other than Agile SDMs taking a shorter period than Traditional SDMs to develop the software, most of the respondents find Agile SDMs easier to use and apply to small projects. To this end, Traditional SDMs are usually earmarked for bigger projects because of ease of use in such projects.

4.4.2.2.5. Planning for software development

Participants were also asked to explain the usefulness of policies and procedures when adopting and utilising the latest ITs such as Agile SDMs for software development in their organisations. OP1, OP4, OP5 and OP6 did not specifically discuss the policies and procedures of their respective companies. These OPs discussed in general terms the usefulness of the policies and procedures of their companies since they guide staff on the latest IT technologies such as Agile SDMs. The OPs went on to suggest that policies and procedures should specify what can be achieved, on which project the Agile SDMs should be adopted, what are the benefits and challenges that the staff should expect, and finally, which process should be followed when adopting the latest technologies. The policies and procedures are designed to enhance and improve productivity.

However, OP2 stated that:

> *"They have standardised policies and procedures which may be changed based on requirements. These policies and procedures are helpful during the induction process. We encourage the software developers to constantly learn processes and conduct thorough investigations about the latest software development methodologies. We do not review our policies and procedures regularly. We may change them based on the requirements."*

The sentiments expressed by OP2 indicate that the policies and procedures are useful because they help companies to equip and direct their staff when they search for and adopt new methods. In addition, the policies and regulations are not fixed but are subject to change depending on the prevailing situation.

Similarly, OP3 said they have policies and procedures that they use. Specifically, OP3 said the following:

> "*As part of the process, we make sure to communicate the policies and procedures to staff, and ensure they are being used and followed, they are used to develop a training program that all staff and volunteers have to undertake. At the end of any training or information session, participants are asked to complete a short quiz to show that they have understood the information. We incorporate a segment about the policies and procedures. Induction and orientation programs are also in our volunteer handbook. We create posters or process charts to post around the workplace to keep*

> *key messages firmly in mind and we regularly promote a policy or procedure in internal newsletters or at team meetings."*

In this case, the policies and procedures are used for developing training programs for staff and volunteers.

In addition, OP7 said:

> *"There are elaborative guidelines for SDM selection."*

Furthermore, OP8 said:

> *"Yes, we have policies and procedures which are very significant which are made to improve productivity. For conventional methodologies, we have heavy documentation. We used to mismatch both the methods for developing the small, medium and large projects. The risk is involved in every project in terms of time, security and cost. We can rectify the errors as software developers and secure our projects using standardised procedures and policies."*

This means that in addition to training, policies and procedures are used to enhance productivity, correct errors and secure projects. However, OP9 did not indicate if they have policies and procedures. In this regard, OP9said:

> *"Our organisation is very flexible and open to adopting any latest IT technologies as well as any SDM for software development."*

Failure to confirm the existence of policies and procedures in companies serves as evidence for lack of planning in the companies because policies and procedures are designed and established at the planning stage of software development. Abdalhamid & Mishra (2017, p.823) agreed with Hajjdiab & Taleb (2011, p.3) that successful Agile adoption and practices are supported by systematic frameworks and guidelines, which are put in place during the preparation or planning stages of software development. The findings of this research seem to suggest that very few BSDCs are guided by set policies and procedures.

4.4.2.2.6. Resources gathering

Resource gathering relates to the available human resources and the type of support given to the employees by the BSDCs. It was important to establish if the BSDCs have qualified software developers who are knowledgeable and experienced in software development. In addition, it was of significance to the researcher to establish how knowledge management and any other type of support were being given to the developers by their companies. This was deemed important so as to establish the company's capability in developing software.

i. Qualification, Knowledge and Experience Required

The employees gave a variety of responses concerning the qualifications, knowledge and experience needed for software development such as good business analytical skills, ability to ask users the right questions, ability to interpret and implement what is written in the methodology documentation. Other skills such as being artistic, a team player, flexible, innovative and team oriented, a qualified project manager as well as having good communication skills, industry experience, customer background knowledge, knowledge on software development were also deemed important by the developers. Specific technical skills such as designing skills, software development and testing skills, technical and project specific skills, and training on Agile SDM are also required. For example, EP8 indicated that a software developer should have:

> *"Technical and project specific skills, experience in similar projects, project management qualification, domain knowledge, customer background knowledge and should have industry experience."*

EP9 highlighted that a software developer should have:

> *"Analytical skills, designing skills, software testing skills, software development skills and knowledge on Software development."*

However, it is of critical importance to note that, for the knowledge, qualifications and skills needed for an Agile or traditional software developer, the respondents gave general answers without specifically talking about themselves. This may imply that the respondents are not very sure of the skills, knowledge and qualifications required for software development. Also, this may be an indication that the developers lack the necessary knowledge, qualifications and skills needed for their respective jobs, hence their reluctance to disclose their personal qualifications. As much as software development is a universal concept, software developers that use different methodologies should however be able to distinguish different methodologies from each other. However, this may not apply in this case because most of the software development companies in Botswana mainly use Traditional SDMs and other methodologies other than the Agile SDM.

ii. Marketability of Software Developers

Most of the employee respondents gave general comments without specifically talking about their own personal marketability. The respondents intimated that software developers are much more marketable if they are able to use the Agile SDM. This is because software developers tend to be generally more skilled because knowledge and use of Agile SDM enables them gain to gain key knowledge on collecting and implementing key information about the clients' specification (requires business analysis skills). With respect to traditional SDM, the SDM is marketable because it is risk sensitive and requires understanding of each stage before proceeding to the next stage. For an Agile SDM, developers become much more marketable if they can easily adapt and implement the changes according to the highly rated user's specification. A traditional SDM software developer must be very good at following instructions and documenting guidelines while implementing the application. In addition, EP2 said:

> *"Agile SDM is becoming more and more popular across the world, universities and education institutions, thus it is more marketable across the globe. Traditional SDM will not die completely; it will be used based on*

> *the situation. They can mix match both methodologies in future. It is more corporate."*

This is one example where the respondent had positive perceptions about the marketability of Agile SDMs. However, the respondent was cautious enough not to disregard the relevance of Traditional SDMs.

And EP3 said:

> *"Agile SDM software developers are more marketable and flexible whilst for Traditional software developers they must be very good at following instructions and guidelines documented when implementing the application. Traditional developers are more marketable with respect to cost and fixed deadline for delivery."*

In the case of EP3, the respondent indicated that both Agile and Traditional software developers are marketable based on specific conditions. However, EP8 and EP9 said opportunities abound for traditional SDMs with respect to meeting user expectations. The employees' responses on their marketability reveal that they know the market requirements needed for marketability in the software development industry and they expressed a healthy level of confidence in themselves. However, these employees conveyed a lack of important skills in some areas that will make them more marketable in the industry.

iii. Knowledge Management and Support

Software development requires some level of knowledge management and support for it to be carried out effectively and properly. Responding to knowledge management skills and support that needs to be given to software developers, the software development companies in Botswana offer some knowledge management and support to their companies depending on their SDM preferences and organisational resources. To this end, OP1 said:

> “*Our organisation is open and flexible to adopt any technology and methodology, so it is highly supported. From the technical infrastructural, we do have human resources who can develop software using Agile SDM. However, they have been newly trained and do not possess the capability to develop quality software for large projects.”*

This shows that the respondent is sure about the support being given to the software developers by their company. In addition, OP2 mentioned that they solve complex projects by buying rapid application for using Agile and they use it for developing small projects. Also, the respondent indicated that, for Traditional SDM, they use C-Sharp and Visual Basic. Also, the fact that OP3 said they are an Oracle partner shows that they are biased towards Oracle’s implementation methodology. In a way, this means that Oracle use is supported by the company.

In agreement with OP5 and OP6, OP4 said:

> *“As a company, Agile SDM is not much supported as it requires more effort and time than traditional SDM to develop software. Normally, it is used only for small and medium projects. Quality software can be achieved by using traditional SDM as long as the human resources have the required experience and capability in understanding business process and requirement gathering.”*

Such a type of response shows that the respective companies support the Traditional SDMs more than they do Agile. More so, OP7 openly stated that they have resources and infrastructure to support both Agile and Traditional SDMs. However, OP8 said:

> *“Quality products can be developed using traditional rather than Agile SDMs. We have a technical infrastructural support team which works with the ISO framework.”*

This shows that the company is designed to provide support for Traditional SDMs, and they believe that they can produce quality software products using Traditional SDMs. In general, the respondents indicated that they are well supported in terms of knowledge management and any other support required for their work.

4.4.2.2.7. Software development

Regarding transparency and observability of the software development process when using Agile and Traditional SDMs, a few of the employee respondents were able to easily explain their positions. In terms of Agile SDMs, some of the respondents indicated that Agile SDMs are much more transparent than traditional SDMs since Agile SDMs allow errors to be detected early. Another reason provided for the superior transparency associated with Agile SDMs is that every phase of the development process is revised regularly during the lifecycle of the project. For example, EP3 said:

> *"Agile SDM is straightforward, simple to develop and easy to manage the change and complex to complete the product as the iteration keeps continuing till the user is satisfied."*

In addition, EP6 said:

> *"When a team stops and re-evaluates the direction of a project every two weeks, there's always time to steer it in another direction."*

EP9 echoed similar sentiments by saying:

> *"With Agile SDM the progress of development is easily noticed as the user is involved in a phased manner hence it is transparent and observable at close interval of times."*

However, regarding the transparency and observability of the software development process when using the Traditional SDMs, some of the employee respondents stated that traditional SDMs are less transparent and customer oriented than Agile SDMs. The development process in traditional SDMs is observable only after completing the entire product or at least a module. Most of the employee respondents including EP1, EP2, EP4, EP8 and a few others highlighted the complexity of traditional SDMs, which emanate from extensive documentation required and the fact that this SDM is only observable at the last phase of the software development process. These findings indicate that the respondents were able to relate to the software development processes they have experienced, used or come to know.

i. Organisational Market Technological Leadership Status

In terms of market technological leadership status of the companies as measured against utilisation, adoption and implementation of Agile and Traditional SDMs and the developed software marketability, respondents had varying responses including being leaders or followers whilst others failed to articulate their respective positions on the subject matter. The respondents who failed to identify their leadership positions gave descriptions of how they use the SDMs. For instance, OP1 noted that:

> *"Agile SDM is usually used for developing small and medium custom developed application. It is used mainly when the client is not sure on the detailed requirement for the application to be developed and the timeline constraint. In other projects, we use SDMs that are recommended by technology vendors or on the standard required by the client."*

However, this was just a mere description of what they do. Similarly, OP3 could not state their technological position in the market. In addition, like several other companies, OP8 could not clearly articulate their market technological leadership position. Instead, OP3's remark was as follows:

> "*In terms of technology, we use open-source technology, which is customized according to customer needs, and it is affordable to the customers and easy to use for development."*

Similarly, although no comment was offered on their market technological leadership status, OP9 opined:

> *"We have only developed software using Traditional SDMs and as far as Traditional SDMs are concerned, utilisation and adoption is very good. It is strongly adopted and utilised for more than 5 years. Implementation has also gone well with minor challenges."*

These comments were meant to justify what the companies do without necessarily specifying their market technological positions. In addition, OP2 indicated that they are followers when it comes to the Agile and traditional SDMs since they operate based on the education and experience, they have adopted in the market. Furthermore, OP4 commented that:

> *"We are more of a product-based company. Based on the nature of the project we will choose the methodology. If the project is small, we will go for Agile otherwise we will use Traditional, which is a conventional process. We also use ad-hoc SDM. We are not leaders."*

In contrast, OP5 boasted about their leadership position and gave a detailed account of how they have been implementing and developing software for the past 25 years. OP5 mentioned that they have successfully developed and implemented 250 projects African. Similarly, OP6 said they are leaders in Agile and Ad-hoc SDMs whilst OP7 indicated that they are leaders in Agile and Traditional SDMs.

Among the explanations given by the respondents, it was clear that an SDM is chosen according to the needs of the market and how well it will fit the technical requirements of a project. In addition, the size of the project and timeline for implementation determine the SDMs to be adopted. Agile SDM is usually used for developing small and medium projects whilst Traditional SDMs are used to develop big projects. The respondents indicated that they are of the understanding that Agile is used mainly when the client is not sure on the detailed requirements of the application to be developed and the timeline constraint. Also, the SDMs to be used can be recommended by technology vendors or the clients. The requirements and solutions evolve around collaborations between the customers and the development teams. The Agile method was said to create deliverables early in a project and allows these deliverables to be refined through an iterative approach involving the customer. This means customers can see the work in progress and make inputs during the development phase.

4.4.2.2.8. Software testing

Employee respondents reacted and responded differently to software performance and quality relative to software developed by competitors. Some of the employee respondents indicated that the Agile SDMs lifecycle is more successful in the achieving the desired goal than the traditional software development lifecycle. The employee respondents also alluded to the fact that Agile developers tend to attach more importance to compliance and compatibility testing. They went on to say that software development is a matter of decision making during the design stage of the software. As a further remark, EP 4 said:

> *"All Traditional approaches are systematic methods of software development where the building phase follows the design stage. Most of the important decisions are made during the design stage and the product building follows which is very predictable. Thus, the building stage only follows the "perfect" design of the system."*

Interestingly, EP7 maintained that:

> *"SDM does not affect quality."*

Although EP8 proclaimed their use of both Agile and traditional SDMs, the respondent mentioned that:

> *"If a competitor will be using Agile SDM which is an evolving procedure, quality will differ but if a competitor will be using traditional SDM, we will be better because we are good in traditional."*

Moreover, EP 9 said they have no comments since they had not come across such as a situation.

It is important to note that the employers and employees gave similar comments with regards to the SDMs adopted and implemented by their companies. However, the employers seemed to possess more knowledge on the choice of SDMs adopted and implemented by their companies. To some extent, software adoption and development differ across different companies depending on the organisational SDM preferences. However, employees from the different companies had almost similar experiences and comments.

Companies that are only using Traditional SDMs or Traditional SDMs in combination with other SDMs responded almost in the same manner to software development companies that adopted and utilised Agile SDMs in relation to job relatedness, results and progress demonstration like software product quality, success rate and completion in time. In addition to the responses given earlier about Agile, the employers highlighted that Traditional SDMs have proper boundaries and control. Other than receiving more input from software developers, other organisation respondents indicated that Agile SDMs require constant interaction and more knowledge sharing with clients to gain a better understanding of the required software functionalities and related processes. This means that clients and developers need expertise for better discussion of processes. However, in terms of Traditional SDMs, the employers indicated that everyone know their roles thus less interaction between clients and developers is required.

For example, OP3 said:

> *"Using Agile SDMs in project development is always having an edge over traditional SDMs. The success of the project always depends on customer satisfaction and how well you have involved the client in your project delivery phase."*

However, OP9 said:

> *"We haven't come across any such organisation which is using Agile SDM hence unable to comment on this."*

Such a type of response indicates that some of the BSDCs only know about and have only used Traditional SDMs. After conducting a comparative analysis on SDMs, Despa (2014, p.37) established that there are several classes of SDMs. Also, Despa (2014, p.37) established that traditional SDMs rank amongst the oldest SDMs available, with Waterfall being regarded as a version of the traditional SDMs.

4.4.2.3. Challenges in Adopting Agile

Different companies and employees face different challenges when adopting or thinking of adopting Agile SDMs. Regarding the challenges being faced by companies, OP1 said:

> *"IT companies and developers already using Traditional SDM might find it difficult to switch to Agile SDM as they are already familiar to Traditional SDM. Furthermore, there is a lack of skills and training on Agile SDM. Also, most universities in Botswana are still teaching their students how to use traditional SDMs with no focus on Agile SDM. More efforts will be needed to use Agile SDM. IT Companies might not be willing to invest more time and money on Agile projects. Adopting Agile SDM also means that the client will need to put more effort on requirement gathering sessions and acceptance testing. There might be resistance from client."*

In this response, the respondent is highlighting the fears that accompany changing the usual or the familiar SDMs, which ultimately discourage companies from investing their time and money on Agile projects. In addition, the respondent states that the developers lack the requisite skills and training required for adopting the Agile SDM. In the same degree, the issues of tertiary institutions in

Botswana not delivering courses on Agile SDMs and lack of skills and knowledge were also confirmed by OP2. Furthermore, OP2 went on to say,

> *"In Botswana, the client rarely knows about what they want using a particular methodology. In terms of Agile, it is easy to adopt as it is structured, but it needs more constant interaction and understanding between the customer and clients."*

This response shows that the respondent believes that the customers do not understand of the methodologies that are used in software development. As such, the software developers need to spend additional time with the client to understand the client's requirement. OP3 and OP2 are of the view that Botswana still lags behind in terms of the adoption of Agile SDMs.

In fact, OP3 said:

> *"Botswana is not yet an Agile friendly community, IT organisations are not using Agile as a core software development methodology, a few organisations might be following this methodology and it's the organisation's choice which methodology to adopt. If we need to improve the organisations to adopt Agile SDMs, then it must be criteria in the tender document, so that suppliers will adopt Agile as a preferred IT project development methodology."*

OP3 alluded to the fact that software development organisations in Botswana need motivation such as tendering requirements for them to adopt Agile SDMs. However, OP3 explained that following Traditional SDMs is a challenge when it comes to applying for ISO/CMML certification because companies need to follow an internationally proven methodology that helps to meet the modern software-centric adaptability requirement.

Similarly, OP4 and OP5 confirmed the stance of OP1 and OP2 about software developers and IT companies in Botswana. OP4 and OP5 indicated that the developers are more familiar with Traditional SDMs and may therefore face challenges when changing from Traditional to Agile SDMs. In addition, OP4 and OP5 cited lack of skills and training on Agile SDMs in Botswana since most universities in the country still use Traditional SDMs with no focus being placed on Agile SDMs. Furthermore, OP4 went on to state that other challenges associated with adopting Agile SDMs include the need for more effort, time, and money.

Furthermore, OP5 said:

> *"There is a need for staff that already knows Agile SDM which might be difficult to find in the market. It is sometimes not easy to switch to Agile SDM due to high costs involved for a company. High cost will mean that the price for the products will increase and that means the company is not competitive. As a company, there is a need to educate the clients first on the advantages of using Agile as compared to traditional SDMs. If they see the value, then it will be easier for IT Companies to include Agile SDMs in their proposals."*

In this case, the respondent emphasised the point of view that there are few people who are knowledgeable about Agile SDMs in BSDI, and that the lack of knowledge about Agile amongst clients necessitates educating the client about Agile SDMs. OP6 reinforced the challenges faced by the BSDCs by also stating that:

> *"Botswana is not ready for accepting agile. It is an expensive process to switch from traditional to Agile. More investment will be needed. Skills and expertise are not available in Botswana. Clients are not aware of Agile SDMs. Most of the software companies in Botswana are using Traditional SDMs."*

OP7 agreed with the other respondents on the challenges encountered in the sector by stating that there is lack of training and exposure to various systems and as well as lack of planning in the companies. OP7 concluded by saying that it is difficult to educate the clients.

Despite all the challenges mentioned by the respondents, OP8 differentiated the expectations of the two types of customers, namely government and private. The OP8 said:

> *"Government is a matured customer compared to private and failure to engage them properly may lead the organisation to lose market creditability. In addition, it would be difficult to migrate to Agile because resources, tools, techniques and everything else are changing in the development and project management field."*

OP8 pointed out that most of the software developers in Botswana depend on the jobs they get from the government, and failure to satisfy the government's needs or expectations can cost a software development company their reputation and marketability. However, the following comment by OP9 on challenges that accompany the adoption of Agile SDMs threw a spanner in the works by giving a different view:

> *"There are no challenges that I can think of, but most of the organisations in Botswana prefer to implement a product rather than developing any software. However, if there are going to be challenges it is going to be training the entire development team and make them understand Agile SDMs."*

4.4.2.4. Chapter Conclusion

Although most of the respondents indicated that they know about Agile SDMs, they lack experience and know-how to adopt and implement Agile SDMs. Botswana software development companies have positive attitudes and views towards adopted and implemented SDMs in relation to Agile. However, some negative and mixed perceptions were also highlighted specifically against traditional and other SDMs. Other respondents have even indicated that they have only used traditional SDMs and could not provide any input on usage of Agile SDMs.

Despite the popularity of traditional SDMs in Botswana organisations, software development in the country is still at its infancy as most organisations were found to have developed one hundred or less softwares in ten years or more. Most of the respondents have positive perceptions towards adopted and implemented SDMs in relation to Agile SDM. Although most of the respondents found Agile SDMs to possess many advantages over traditional and other SDMs, BSD companies still prefer the traditional SDMs and other non-Agile SDMs for delivering their desired goals.

Despite the well-established benefits associated with Agile SDMs, most of the software development companies in Botswana adopt and implement traditional SDMs in the development of software. There are, however, still a few organisations in the country that adopt and implement Agile SDMs in addition to other SDMs they are using. The number of software development companies using traditional SDMs is the highest followed by some that work with a combination of both Agile and traditional SDMs. Very few organisations have adopted the Agile and Ad-hoc SDMs combination.

Few organisations involved in software development in Botswana were able to confirm the existence of policies and procedures in their organisations. This gave evidence of lack of planning in the organisations as policies and procedures are designed and established at the planning stage of software development to guide processes. In addition, looking at resource gathering, most respondents failed to talk about their personal qualifications, knowledge and skills showing their lack of confidence in the work they are doing. However, the respondents indicated that the organisations support software development in terms of knowledge management and any other support to which promote software development.

Though several organisations confirmed their use of Traditional SDMs, they could not avoid talking about the complexities of using traditional SDMs. The respondents were able to clearly recite the software development processes they have experienced and used including what they know. In addition, the respondents were able to state their leadership status with respect to utilisation, adoption, and implementation of Agile and Traditional SDMs.

Furthermore, it was established that there are some marked levels of knowledge management and support that are provided to software developers in BSDCs. Despite organisations in Botswana having the requisite technological infrastructure and human resources that support software development, most of the personnel lack the necessary experience, knowledge, and skills to effectively develop software especially for Agile SDMs.

The labour marketability of software developers in BSD companies is very limited since most respondents opted not to disclose their personal qualifications and skills but instead chose to mention marketability in general terms. It is well known that software adoption and implementation require some form of qualifications, knowledge, and experience. For instance, a project manager with vast knowledge on specific or related SDMs and industrial experience on software development is difficult to find in Botswana.

Other than challenges relating to lack of knowledge, skills, and experience on Agile SDMs, other challenges that were cited involve the unaffordable costs arising from migrating Traditional SDMs to the more modern Agile SDMs as well as resistance to change on the part of the company and client.

In summary, the pace and extent to which BSDCs are adopting and implementing Agile SDMs to develop software for business processes is not satisfactory, and most of the software developers and BSDCs need to be equipped and empowered for the successful implementation and adoption of Agile SDMs. In the next chapter, an interpretation of the results is presented.

CHAPTER 5: INTERPRETATION OF THE RESULTS

5.1. Introduction

In Chapter 4, the results of the study were presented explaining the generated themes and patterns according to the implied phases of software development life cycle. This chapter presents detailed discussions on the interpretation of the research results. It shows how each of the research objectives were met. Other aspects of this research study such as limitations of the research, gaps not covered by the research, possible future research opportunities, validation of results, practical implications, impact of the research on the relevant society, and clarification of the contribution made by the work to the relevant body of knowledge are also covered.

5.2. Discussion

The goal of this research is to explore and determine the factors that affect the adoption of modern software development methodologies (SDMs) such as Agile, Scrum, Extreme Programming, Crystal Clear and others by the Botswana Software Development Industry (BSDI). This goal was achieved by answering the following fundamental research study question, which was generated from the problem statement:

> “*What are the factors affecting Botswana’s software development companies and software development professionals in adopting and utilising modern software development methods?*”

The research objectives and the associated research sub-questions were discussed in this section by considering a global view of all the emerging themes and implications in terms of software development practices.

5.2.1 Perceptions of respondents towards adoption or non-adoption of modern Agile SDMs

The results presented in this research study indicate that Botswana software development companies have positive perceptions about adopted and implemented SDMs (i.e., the traditional SDMs in comparison to Agile). However, like many companies and software practitioners in the world, most of the respondents in this research found Agile to possess more advantages compared to traditional and other SDMs. This view is also shared by Habib (2013), Penn (2016, p.17) and Al-Zewairi, M. et al. (2017, p.74), who have expressed a generally positive outlook with respect to the adoption and implementation of Agile SDMs worldwide. Similarly, Asnawi, Gravell and Wills (2012b, p.30) have established that software developers in Malaysia that use Agile SDMs prefer Agile SDM instead of traditional methodologies when it comes to problem solving capabilities. However, a South African study by Penn (2016, p.17) indicated that some of the companies are still doubtful about whether agile development is a good course of action to follow. The latter view seems to be aligned with the findings of this research; some of the respondents had mixed feelings about Agile and others were so comfortable with the traditional SDMs to the point where some were so negative and did not want to make any comment about Agile SDMs. For example, organisation participant 9 (OP9) said:

“Concerning Agile SDM, no comment as our organisation hasn’t done anything on Agile SDM; and concerning traditional SDM our organisation views it with respect and above all it is manageable and quite straight forward”.

Other respondents in this research study have gone to an extent of saying that they only knew traditional SDMs and could, as a result, not provide any views regarding how Agile SDMs compare with traditional SDMs. By the same token, a study conducted by Asnawi, Gravell and Wills (2012b, p.30) has established what could be construed as a negative perception of Agile SDMs by Malaysian companies. The negative perception was anchored on Agile team members’ failure to be accountable for execution of software development operations because they lacked confidence in implementing Agile SDM. Interestingly, completely opposite and positive views were expressed by Asnawi, Gravell and Wills (2012b, p.30), who suggested that users of Agile SDMs found the SDM to be very helpful in minimising and managing software development of some volatile functional requirements of certain business processes.

Although many companies have a generally positive perceptions about Agile SDMs to the extent that they have now adopted Agile approaches, these positive perceptions are often preceded by complications, hindrances and other challenges (Penn 2016, p.17). Results emanating from this research study seem to reflect mixed perceptions about the adoption and implementation of Agile SDMs by the Botswana software industry. The views and experiences of some of the companies that have adopted and implemented Agile SDMs in Botswana have generally been positive. On the other hand, companies that have not adopted and implemented Agile SDMs have displayed both mixed and negative attitudes towards these methodologies.

For example, organisation participant (OP) 6 stated that:

> *"We use Agile SDMs and we are leaders for Agile and Ad-hoc SDMs*"

In addition, an employee participant (EP) 6 said:

> *"When a team stops and re-evaluates the direction of a project every two weeks, there's always time to steer it in another direction".*

In addition, OP7 commented that:

> *"We are market leaders in software development in Agile and traditional SDMs. The Agile method creates deliverables early in a project and refines them through an iterative approach involving the customer. The customers seemed to be happy with the Agile approach as it enables them to see the work in progress and make comments during development".*

The above-mentioned comments show that the organisations that are adopting Agile SDMs are generally positive about these methodologies.

However, the factors working against Agile adoption include doubt, lack of confidence, insufficient knowledge about Agile and resistance to change. To tackle such challenges, motivational training sessions, seminars and conferences should be conducted on a regular basis. It is also recommended that motivational speakers and software engineering experts should be invited to present positive stories on successful adoption and implementation of Agile SDMs. Poster presentations as a mode of organisational communication can also be used to reinforce the adoption of Agile SDM with the view to empower software developers. As articulated by (Hajjdiab & Taleb 2011, p.3), constant and continuous communication contributes towards strengthening and facilitating team decision-making processes. In addition, novices can be allowed to learn directly form experts through job shadowing. Furthermore, there is a need to establish a working knowledge management strategy and sharing of information within the organisation as proposed (Gandomani et al. 2013, p.622). Moreover, management roles and styles should change, and managers should find enough time and attempt to choose appropriate people whom they should train, coach and mentor with the intention to develop a set of work practices. In addition, organisations should use equipment that can allow for incremental evolution, on-going integration, re-working, version administration and other Agile technologies as suggested by (Gandomani et al. 2013, p.623).

5.2.2 Adopted and Implemented SDMs and Levels of Adoption

Results of an investigation of SDMs that have been adopted and implemented for the development of software for business processes by the Botswana software development companies (BSDCs) show that the BSDCs that were surveyed have opted to adopt and implement traditional SDMs more than Agile SDMs. Although the software development industry in Botswana is dominated by companies that use traditional SDMs, it is evident from the statement of organisation participant 2 (OP2) that the few that have opted to adopt and implement Agile and/or other SDMs do so at a small scale. For example, OP2 opined:

> "*We have developed 11 – 50 and more than 100 software products in 5 – 10 years and for larger projects, we use waterfall method. For example, to develop software for financial software niche market. Alternatively, we use agile and we operate based on the education and experience we have adopted in the market".*

OP2's sentiments concur with the May 2009 final report of the Botswana Ministry of Communications, Science and Technology, which highlighted limited innovation in Botswana (Final Report May 2009). It is quite clear that software development in Botswana is way below the international software development standards.

Despite gaining widespread popularity in other parts of the world, the use of Agile SDMs in Botswana is similar to that of other similar developing countries such as Malaysia where there is very limited use of Agile SDMs (Asnawi, Gravell and Wills (2012b, p.30). Agile SDM is apparently still new in Malaysia, and the awareness thereof remains limited notably in the public sector (Asnawi, Gravell and Wills (2012b, p.30). Like Botswana, Malaysia was at the time still going through low adoption and implementation rates of Agile SDMs, and the traditional SDMs and other approaches were still dominating due to lack of awareness and knowledge of Agile SDMs and skills as well as resistance to change. So, whilst Agile SDMs have become popular in the software development industry, for various reasons, not all countries are adopting and implementing them at the same rate. Penn (2016, p.17) have highlighted that some of the companies adopt Agile SDMs as part of a competitive strategy.

Duka (2013, p.426) has, however, proclaimed that the adoption and implementation of Agile SDMs has become more popular as compared to traditional SDMs. Duka's Croatian company has decided to adopt the Agile culture to enhance their operations and share the Agile culture within and outside their organisation. Moreover, Kumar & Bhatia (2012, p.49) has noted that: "*adopting Agile development methodologies has a positive impact on both the productivity and the quality of a company*". When discussing the adoption of Agile SDMs in Vietnam, Tran and Duong (2014, p.1) indicated Agile SDMs were being widely adopted and steadily used by the software industry in many companies across the world. This view is also supported by (Al-Zewairi et al. 2017, p.93) . Tran and Duong (2014, p.1) went on to explain that Agile SDMs are viewed as solutions to the challenges resulting from the adoption and implementation of traditional approaches.

Similarly, Bhatia and Kumar (2012, p.49) indicated that many research studies have confirmed that Agile SDM has been welcomed in the software development industry because it facilitates resolving precise software requirements through constant interaction with the client. The software application is delivered early to the client and in incremental stages. Lastly, software development occurs in controlled stages and within the budget limit. The software industry in Vietnam has apparently been developing rapidly and, in order for the industry to keep up with the global trends, many companies have adopted Agile SDMs to enable them to compete for contracts with many international companies (Tran & Duong 2014, p.1). Mbelli and Hira (2016, p.219) agreed with Duka (2013, p.426) and Tran and Duong (2014, p.1) that Agile methodologies have been welcomed globally and South Africa is also following a similar trend. However, Mbelli and Hira (2016, p.219) have reported that most companies in South Africa are adopting the Agile methodologies without the full knowledge and understanding of the processes.

5.2.3 Planning for software development

Very few organisations have guiding policies and procedures for developing software in Botswana. du Plessis (2016, p.92) has explained that a policy provides guidance and thus assist with decision making processes whilst procedures assist with the course of action to be taken in each situation. Policies and procedures should therefore be designed as part of a suggestion system of an organisation that encourages employees to be active participants in the ideation processes of the organisation (du Plessis 2016, p.92). Therefore, policies and procedures provide structures that can be used to guide the software development process.

Squires, Moralejo and LeFort (2007, p.1) have professed those policies and procedures are guidelines that are prepared by organisations to direct and lead employees as they execute their duties. This means policies and procedures are designed to influence and determine all major decisions and actions, and all activities take place within the boundaries set by them (IAS 2019). Procedures are the specific methods employed to express policies in action in day-to-day operations of the organisation (IAS 2019). Thus, lack of structures may have implications on the way things are done on the end-product. In addition, lack of structures may have a negative impact on adoption and implementation of modern SDMs.

According to Mbelli and Hira (2016, p.219), most South African organisations do not have structures to guide them as they shift from the traditional to the Agile approach. Similarly, this research has established that most software development companies in Botswana do not have guiding structures and they also lack knowledge on how to adopt the Agile SDMs. In terms of the provision of guiding frameworks and structures, the Botswana software development industry situation seems to be at par with the African continent and the world at large.

It has already been mentioned that Botswana software development companies generally tend to lean more towards the adoption and implementation of traditional SDMs compared to Agile-related SDMs. Several factors such as limited awareness on Agile methodologies including lack of knowledge on how to adopt these methodologies, lack of skills, expertise and organisational infrastructure that supports Agile development as well as resistance to change were cited by Abdalhamid, S. and Mishra, A (2017, p.420) as challenges that scupper the adoption and implementation of Agile SDMs. In addition, lack of guiding policies and procedures was also found to be one of the factors at the forefront of this impediment.

These challenges can be addressed in many ways. However, a great need exists to put in place a well-organised plan or framework to guide and direct the acceptance and utilisation of Agile SDM within BSDCs. The most critical aspect when adopting Agile methodologies is to have a plan that will assist to streamline the relevant processes. A need also exists for benchmarking and learning how previous Agile SDMs have successfully been implemented locally and internationally. Based on the findings of this research, which concur with studies undertaken by (Tiky 2016, p.8) and Swersky (2018), it is important to consider during the planning stage resource gathering that includes team building, training and recruiting of human resources to handle different phases of Agile software development methodologies, and gathering of physical equipment and technology upgrading needed to maintain, motivate and support Agile adoption. The authors also suggested that documentation such as policies and procedures be adopted at the planning stage. This means an organisation needs to establish its guiding principles and procedures from an early stage.

Kakar (2012, p.137) and Hungund and Kiran (2017, p.355) have discussed the Job Characteristic Model (JCM) for organisation job profile design with the intention to enhance organisation performance. This research has taken a leaf from these studies by adopting some of the key recommendations for designing the framework for Agile SDM adoption. When piloting the adoption and implementation of the software, it important to ensure that any major issues arising during the trials are attended to and thus allow the feasibility of the design, function and other key components of the software to validated (Silva and Goldman (2014, p.64). At every stage of the software development process, it should be made possible to assign value to the adoption of the Agile SDMs. Piloting will enable informed assessment and evaluation of the software behaviour at different stages to be made and to identify areas of improvement. After assessment and evaluation, an acceptable master plan of implementing the Agile SDMs can be designed and adopted for the organisation. Once an acceptable design is adopted, software development will take off and the end-product will be tried and tested to assess if it is working properly. The software will be assessed for errors or malfunctioning, and they will be noted for correction or improvement. As the software is perfected for implementation, it becomes important to have a maintenance plan to keep it functioning at its best and give room for upgrading to cope with new changes in the field. It is also important to provide awareness of the product to the public so that choices can be made, and the product is publicised.

5.2.4 Resources gathering

Failure to make personal reference about knowledge, qualifications and skills needed for an Agile or traditional software developer by the respondents left a lot to be desired with respect to their suitability or adequacy to do the work they are doing as software developers. This provides evidence that there is lack of knowledgeable developers in BSDCs, and by extension limited knowledge and skills. This may be the reason why most of the companies seem to cling on the adoption and implementation of traditional SDMs as they are more familiar with them and thus fail to see the need to adopt modern SDMs such as Agile variants. As much as the employees' responses on their marketability reveal that they know the market requirements and expressed a healthy level of confidence in themselves, they nevertheless demonstrated a lack of important skills in some areas that will make them more marketable in the industry. These findings concurred with a study by Abdalhamid and Mishra (2017b, p.420), which pointed out the lack of skills and expertise in Botswana for supporting Agile development. As discussed earlier in section 5.2.3, poor planning results in poor or wrong recruitment. In addition, Flora and Chande (2014, p.3636) highlighted that the adoption and implementation of Agile methodologies in software development requires developers with good personal skills and experiences. Furthermore, the authors elaborated that software development using Agile SDMs appears to be impossible if developers lack personal knowledge, qualifications and skills.

This research study established that there are some marked levels of knowledge management and support given to software developers in BSDCs. This was supported by some of the respondents who indicated that their organisations are supported in terms of resources and infrastructure needed for Agile and Traditional SDMs. They have the requisite human resources that can develop software using traditional and Agile SDMs. However, as indicated earlier, most of the human resources lack experience, knowledge, and skills to effectively develop software especially with respect to Agile SDMs. In addition, this research established that some of the organisations have research and development (R&D) team units where teams conduct research on latest SDMs. Knowledge management and support are also offered in terms of training, workshops, and online learning platforms to equip the staff. It is important to note that these support systems were not common across the study organisations and others had very little to nothing to show in this regard.

These findings clearly show that BSDCs have some marked levels of knowledge management and support given to software developers. This means that although Agile SDM adoption is to some extent supported. However, there are limitations in terms of experience and expertise for effective adoption and implementation. Nevertheless, it needs to be emphasised that this type of situation is not unique to Botswana. Other countries such as Malaysia and South Africa are also struggling with the same issues relating to the skills and know-how for the adoption and implementation of Agile methodologies (Asnawi, Gravell & Wills 2012a, p.219; Mbelli & Hira 2016). For example, Mbelli and Hira (2016, p.219) have stated that most companies in South Africa are adopting the Agile methodologies with little knowledge and understanding of the processes.

As alluded by Jain (2011, p.1557), limitations in experience and know-how for the effective adoption and implementation of Agile pose serious problems for software development. Development of software inherently requires several people with different set of knowledge and skills to work as a team and use new ideas to obtain efficient results, thus giving quality and reliability as properties of the end-product. In this regard, knowledge management becomes significant in an organisation as it allows knowledge to be collected, improved and shared with employees with the view to enhance organisational performance (Jain 2011, p.1557). This means that an organisation needs an efficient knowledge management system for knowledge processing and transferring from one individual to another. However, although some marked levels of knowledge management and support are being provided to software developers in BSDCs, some organisations indicated that they lack efficient knowledge management systems. In many instances, the employees or the software developers lacked the requisite knowledge, and the available technological infrastructure tended to lean towards supporting the traditional SDMs more than it did the Agile SDMs. This means that the level of knowledge management and support given to software developers in Botswana is somewhat limited.

In addition, it was established through this research that most software development companies in Botswana are small-scale developing software companies. Small-scale software developing companies are defined by Ayalew and Motlhala (2014, p.121) as companies developing 50-100 software applications per year. This observed phenomenon contributed towards the limited skills, knowledge, technological and human resources available in such companies. Challenges relating to skill and knowledge gaps and limited technological and human resources are also commonly experienced in South Africa (Mbelli & Hira 2016, p.219), Malaysia (Asnawi, Gravell & Wills 2012b, p.30) and Jamaica (Chevers Whyte & Chevers 2015, p.1). As mentioned by Chevers Whyte and Chevers (2015, p.1), executive management support for Agile methodologies is also an important prerequisite for the effective implementation of information systems projects.

Knowledge management is vital in software development because knowledge is viewed as a critical asset that allows human resources to operate. Furthermore, knowledge management facilitates collaboration of skilled and knowledgeable personnel to share ideas and learn from each other. It allows movement of information from one person to another within an organisation even long after that person has left the organisation. Efficient knowledge management systems allow the organisation to keep on moving forward without losing information even after a team member has left. According to Jain (2011, p.1561), knowledge management is a transformation procedure where knowledge is changed into value for the company to minimise cost, increase workers' production, and enhance quality of the end-product and services.

The factors working against Agile SDM adoption in the BSDI is are driven largely by many small-scale software developing organisations that have limited skills, knowledge, technological and human resources. This is also made worse by lack of efficient knowledge management systems and technological infrastructure that supports the implementation of Agile SDMs. To solve such a problem, better planning, preparation and upgrading and fostering for a large-scale software development capacity or status is very important and use of frameworks is recommended (Abdalhamid, S & Mishra, A 2017, p.822). In addition, building a dedicated multi-disciplinary team with a wide range of skills for the different stages of Agile SDM adoption can make a huge difference.

5.2.5 Software development

This research established that the respondents were able to relate to the software development processes they have experienced and used. The respondents were also able to provide information about the knowledge they have about software development using traditional and Agile SDMs. It is important to note that, for Agile SDMs, most of the respondents appears to have the general knowledge without necessarily possessing the experiential or practical knowledge. These findings align well with earlier findings about the adopted software, levels of adoption and developer knowledge, qualifications, and skills. The findings of this research concur with those of Betta and Boronina (2018, p.446), which supported the notion that Agile SDMs are more transparent and observable in terms of process of developing software when compared with Traditional SDMs. Also, this research confirmed what the authors revealed Traditional SDMs as being less customer oriented than Agile SDMs. In contrast, a study by Flora and Chande (2014, p.3636) revealed that limitations of Traditional SDMs such as slow response to fast transformations in business requirements differentiate Traditional SDMs from Agile SDMs.

However, Nkone (2013, p.20) has argued that software development does not have to follow strict rules given in the manuals. In addition, the author indicated that more than one SDM can be used to deliver a software product. Nkone (2013, p.20) argues that different SDMs can be used in different stages of software development to produce a software product that can meet set objectives. This is contrary to what the BSDCs are practicing because they are mostly focused on a specific SDM per project. Such an approach is often ascribed to the developers' limited knowledge and expertise. As discussed earlier, limited expertise and knowledge can be addressed through training, workshops, collaborations with successful organisations, expert coaching, and the introduction and enhancement of knowledge management systems.

5.2.6 Software testing

Employee respondents responded differently to software performance and quality relative to software developed by competitors. Some of the employee respondents indicated that the Agile SDM lifecycle is more successful in the achievement of the desired goal than the traditional software development lifecycle. The employee respondents also alluded to the fact that Agile developers tend to scribe more importance to compliance and compatibility testing. They went on to say that software development is a matter of decision making during the design stage of the software. These findings support the stance taken by Haiderzai and Khattab (2019, p.5), that software testing is key to software development because it assesses every stage of software development to identify and deal with bugs. With this mind, the authors argued that software testing is not a simple process, but rather a major phase in software development that needs experts to produce quality products. This means software developers in BSDCs were able to link their expertise to software testing since they talked about a matter of decision making. Decision making is directly related to one’s expertise, which influences organizational practices and effectiveness (Salas, Rosen & DiazGranados 2010, p.941). This means that decision making is not a casual process that can be done as one pleases; instead, it requires systematic reasoning to come up with logical outcomes (Salas, Rosen & DiazGranados 2010, p.941). In addition, Stoica, Mircea and Ghilic-Micu (2013, p.72) have argued that Traditional SDMs are different from Agile SDMs in that the Traditional SDMs allow for software testing once coding is completed whilst Agile tests at every iteration. In a way, this implies that the quality of software produced using Agile SDMs is better than the one produced by Traditional SDMs. It is not necessarily obvious that the software qualities for Traditional and Agile are the same. The findings of this study concur with that of Gandomani and Zulzalil (2013, p.5089), which confirmed the notion that Agile SDMs demand compliance and compatibility testing. This reflects that Agile SDMs place more value on software testing since software development progresses.

5.2.7 The challenges experienced by BSDCs when adopting Agile SDMs in Botswana

The case of software development in Botswana is similar to that in Zambia, where challenges of Agile development and implementation were classified as organisational, people, process and technical challenges (Kunda et al. 2018, p.585). Furthermore, Kunda et al. (2018, p.585) has mentioned that the challenges of Agile development and implementation that are unique to developing countries. In the African continent, Botswana seems to be experiencing Agile adoption challenges such as lack of knowledge, skills, experience and documentation, which are also being experienced in South Africa (Hanslo & Mnkandla 2018, p.951). In the same vein, Vanker (2015, p.1) has reported that South Africa adopted Agile SDMs late and listed several barriers to adoption such as organisational culture, lack of skilled manpower and missing management support.

Asnawi, Gravell and Wills (2012b, p.39) have also identified the following people-factor-based challenges, which require mind-set change to overcome them: organisational culture, involvement, knowledge management, resistance to change. According to Jain (2011, p.1561), geographical location and multicultural organisations apparently make the development of software much more complicated. As alluded to by Hajjdiab and Taleb (2011, p.1), Agile adoption frequently brings unique problems that call for shifts in company thinking and policies that are necessary for successful outcomes. A comparative analysis of two different companies that adopted Agile SDMs has revealed that Agile adoption challenges are not always uniform, but are rather unique depending on the environment in which the Agile SDM is adopted and implemented (Hajjdiab & Taleb 2011, p.1).

Challenges due to Agile SDM adoption and implementation are unique to a given situation, are contextual and would require different approaches to overcome them. These challenges can be solved by adopting the remedies suggested in the preceding sections. Therefore, findings emanating from this research study indicate that organisations trying to adopt and implement Agile SDMs will be confronted with challenges that are specific to that organisation. This also implies that, like many across the world, Botswana is also experiencing difficulties in trying to adopt and implement Agile SDMs. without prescribing what organisation should do in order to effectively adopt Agile; some of these challenges could be resolved using some of the approaches recommended in sections 5.2.1, 5.2.2 and 5.2.3 for those organisations that want Agile development as the way forward.

In addition, organisations interested in Agile adoption could use equipment that support Agile SDMs' characteristics of incremental evolution, ongoing integration, re-working, version administration and other Agile technologies as suggested by (Gandomani et al. 2013, p.623). Facilitating a smooth acceptance and utilisation of Agile SDM, organisations that plan on adopting and implementing Agile methodologies for the first time will need collaboration with organisations that are already using Agile SDM. The government, tertiary institutions and private companies can form partnerships that supports Agile adoption through financial assistance, human resources, training and physical infrastructure.

Since there is evidence of a desire or willingness to adopt Agile SDMs, interested stakeholders can also come together and form partnerships that are aimed at compiling guidelines that can be used by institutions to adopt and implement Agile SDMs in Botswana. Poor Agile adoption in Botswana will require interested or concerned stakeholders to work as a coherent unit so as to achieve a common goal of successful Agile SDMs' adoption. This will require interested BSD companies to initiate and lead the initial process of advertising and marketing ideas about Agile SDMs.

5.2.8 An evaluation of the adoption and implementation of Agile and related SDMs to develop software for business processes by BSDCs

This research has established that the BSDCs are not adopting and implementing relevant Agile SDMs for the development of software for business processes at a satisfactory rate. In addition, most of the software developers and organisations need to be equipped and empowered for the adoption and implementation of relevant Agile SDMs.

Furthermore, this research has established that the marketability of software developers from BSDCs is very limited internationally. Software adoption and implementation require some form of qualifications, knowledge and vast experience in software development and project management (Alqudah & Razali 2017, p.530). In sharp contrast to the rest of the world, all these skills are lacking in Botswana, and this has resulted in low adoption and implementation rates of agile methodologies in the country. The annual report of the Botswana Ministry of Communications, Science and Technology (May 2009) indicates that the country does not have enough information and technology (ICT) personnel including software developers and project managers.

According to Moalosi et al. (2016, p.42), software is classified under Botswana's creative industries. Contrary to the findings of this research, Moalosi et al. (2016, p.42) listed tertiary institutions that offer programs to capacitate the industry, however, the authors did not give further information as to how these institutions are doing it. According to this research's findings, most of the institutions that provide training use Traditional SDMs. Botswana's creative industries are new, and they are by world standards not very competitive. In addition, there are vast limitations in the industry in terms of knowledge and experience (Moalosi et al. 2016, p.48). Similar sentiments about the Botswana creative industry have expressed by (Nyamaka et al. 2018, p.1). For example, Botswana's mobile application development industry (which falls under creative industries) is new and the industry is mostly privately owned by young men (Nyamaka et al. 2018, p.1).

Furthermore, many challenges faced by the industry limit the competitiveness of the industry and the country in the global market (Moalosi et al. 2016, p.47). Some of these challenges include lack of qualified personnel, lack of technology information, onerous Botswana regulations and international standards (Moalosi et al. 2016, p.48). This shows that Botswana is lacking in many areas that support and are required for the smooth adoption and implementation of Agile SDMs. Our research findings have since confirmed issues raised by other authors (Moalosi et al. 2016, p.38; Nyamaka et al. 2018, p.1), which relate to the Botswana ICT and software development industry. This means Botswana still has a long way to go with regards to ICT and software development and it will take serious efforts for the country to be on par with the rest of the world. However, it is important to note that the government of Botswana is putting in place measures to address these challenges such as the establishment of the Botswana Innovation Hub, a company tasked with driving innovation in the country (Botswana Innovation Hub 2016).

To emphasise poor adoption of Agile methodologies in Botswana, Organisation Participants 1 and 8 (OP1 & OP8) agreed that the adoption and implementation of Agile SDMs requires the project management team to have control over each and every stage of project development as compared to Traditional SDMs. In addition, OP1 and OP8 went on to say that Agile SDMs need more monitoring, continuous follow up, experience and experts to develop the project successfully. This means that Agile SDM needs experts who are qualified and experienced. In fact, OP9 was confident to say that they have not come across any organisation uses Agile SDMs in Botswana.

This means that some Botswana software developers are using some of the oldest SDMs, and Traditional SDMs being used by these developers ranks amongst the oldest SDMs in the world (Despa (2014, p.37). Furthermore, most of the software development organisations interviewed highlighted that they were comfortable with the Traditional SDMs and were working well with them.

The poor adoption of Agile SDMs in Botswana was further confirmed by negative perceptions about Agile SDMs amongst participants. For example, OP1 revealed that their organisation took Agile SDMs as one of the SDMs (but not necessarily the major) they depended on. At the same time, OP2 indicated that the project manager finds Agile SDM difficult to monitor because of time constraints, lack of flexibility, cost constraints and a need for regular interaction between the teams and client because it does not have a start and an end process. In this regard, OP8 commented that an Agile SDM is expensive for customers compared to a Traditional SDM.

Similarly, the employees also shared the same negative sentiments about Agile. For example, Employee Participant 2 (EP2) believes that Agile SDM is more complex, a lot of discussions the arise during the development phase cause confusion, and it is difficult to apply and control. This research also established that the poor adoption and implementation Agile SDMs are due to lack or poor levels of knowledge management and support for software developers. OP6 mentioned that in their company, Agile SDMs are not well supported because they require more effort and time to develop software than Traditional SDMs. In addition, OP8 opined that quality products can be developed using traditional rather than Agile SDMs. All these views suggest that Agile SDMs are not viewed as relevant and necessary in Botswana as they should. Poor adoption and implementation of Agile methodologies in Botswana has been accompanied by a myriad of challenges and these sentiments are articulated in a list of challenges given by OP1. This list of challenges range from difficulties being experienced when switching to Agile SDMs, which emanating from a familiarity with Traditional SDMs, and lack of skills and training on Agile SDMs. In this research, it was established that more efforts are required to adopt Agile SDMs since IT companies might not be willing to invest more time and money on Agile projects. As evidenced by the statement from OP3, Botswana still lags behind the adoption of Agile as an SDM. The respondent said:

> *"Botswana is not yet an Agile friendly community, IT organisations are not using Agile as a core software development methodology, a few organisations might be following this methodology and it's the organisation's choice which methodology to adopt".*

This was supported by OP6 who also added that Botswana is not ready to accepting Agile because skills and expertise are not available in Botswana and most of the software companies use Traditional SDMs.

In summary, poor Agile SDM adoption in Botswana is ascribed to an intimate relationship with and dominance of traditional SDMs. The adoption of Agile SDM is also faced with resistance to change, insufficient levels of knowledge management, lack of support and skilled labour and other challenges associated with the adoption of Agile SDMs (Boehm & Turner 2005, p.34; Hajjdiab & Taleb 2011, p.2; Kunda et al. 2018, p.590). Lack of knowledge and skilled manpower in Agile SDMs are at the core of Botswana's poor adoption of Agile methodologies. Similarly, GESCI (2017, p.26) has pointed out a scarcity of basic competencies needed to operate ICT equipment in Botswana, which are used during the development and implementation of software.

Botswana's situation is unique in that, despite the poor adoption of Agile by organisations, the government and some private organisations are trying to equip the sector in terms of technological infrastructure needed for Agile adoption. Similar sentiments were expressed by GESCI (2017, p.26), which highlighted a number of ICT initiatives in Botswana that are geared towards assisting the country to establish a knowledge-based economy and community. In addition, as already established by this research, there are marked levels of knowledge management and support in some organisations, and the government has established a number of organisations such as the Botswana Innovation Hub to encourage innovation by software developers and software development companies (GESCI 2017, p.42).

The findings of this study show that while there is limited adoption of Agile SDMs, a desire exist within Botswana to adopt these methodologies. For this reason, this study is proposing a framework for the adoption of Agile methodologies.

5.2.9 Framework for Agile adoption

This section presents the proposed framework for the adoption of Agile methodologies in Botswana.

5.2.9.1 Proposing a framework for the adoption and implementation of Agile SDM in Botswana

The aim of this research study is to explore and determine what factors affect the adoption of modern SDMs such as Agile, Scrum, Extreme Programming, Crystal Clear and others by the BSDI. Specifically, the research study is aimed to achieving a summarised and comprehensive description of the phenomenon through reviewing of literature, conducting of interviews and the analysis of collected data. Literature review, interviews and data analysis served to provide the researcher with important insight into the state of Agile SDM adoption and implementation in Botswana. Thereafter, it was envisaged that a model, conceptual system, or a conceptual map would be developed that support and encourage BSDCs to adopt and implement Agile methodologies or increase and grow their businesses as they adopt and implement Agile SDMs.

The researcher suggested a specific course of action that needs to be taken by BSDCs as they adopt and implement Agile SDMs. The course of action was based on the findings of the research including challenges being faced by developers and BSDCs and the concomitant remedies. With this background in mind, the researcher developed a theoretical framework that was informed by the reviewed literature, interviews conducted, and analysed data before suggesting solutions cited in sections 5.2.1 to 5.2.8 and software development life cycle (SDLC) punted by (Swersky 2018), (Tiky 2016, p.1), and (Ruparelia 2010, p.8). This framework can be used as a standard for Agile adoption and implementation by BSDCs. The framework provides software developers with possible options for the successful acceptance of Agile SDMs. The developed software can be used as a guide and a tool by interested organisations or stakeholders in software development in Botswana. The framework presents the developers with a basis for creativity, exploration, and innovation as they expand their Agile adoption and implementation efforts. In addition, the framework can be adopted as a set of standards around which software developers in Botswana can work as they migrate from traditional to Agile SDMs on a large scale. This study was intended to develop knowledge and expand currently available knowledge, and the framework proposed in this research study is a modification of the SDLC presented below.

The SDLC that guided the proposed framework was designed using the findings of this research and is presented in Figure 32. The interrelationships and order of different stages, which are included in the SDLC, are clearly illustrated. These links and order can be used to augment the inception, acceptance, and implementation of Agile SDMs in Botswana software development.

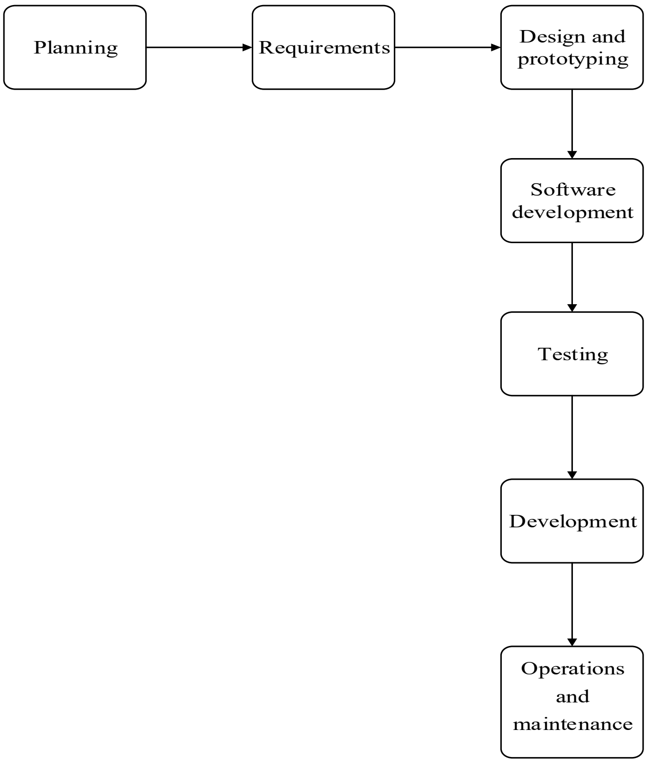


Figure 32: An illustration of the software development life cycle (SDLC) (Ruparelia 2010, p.8; Swersky 2018; Tiky 2016, p.7)

A discussion on the themes emanating from the research findings as well as other factors that guided the development of the proposed framework are presented in the subsection that follows.

5.2.9.2 Factors that informed the proposed framework for the adoption and implementation of Agile SDMs in Botswana

Several themes emerged from research study results, which informed the development of the proposed framework for the adoption and implementation of Agile SDMs in Botswana. These themes included planning for software development, resources gathering, software development, software testing and the challenges experienced by BSDCs when adopting Agile SDMs in Botswana. Findings that informed the framework included the preponderance of traditional SDMs in Botswana. In addition, this research study has established that development of software in BSDCs is being done at a small scale in terms of a small number of software applications developed over a given period of time (Ayalew & Motlhala 2014, p.121). Furthermore, BSD companies and their employees have mixed and negative perceptions towards the adoption and implementation of Agile methodologies. However, although marked levels of knowledge management and support were observed for traditional methodologies, some of the organisations demonstrated insufficient knowledge management and support for the adoption and implementation of Agile SDMs. Moreover, adoption and implementation of Agile SDMs in Botswana is often dogged by several challenges.

These include lack of skills, lack of Agile knowledge, familiarity with traditional SDMs and resistance to change (by both developers and customers). Furthermore, the organisations and developers indicated that there is fear of the high costs associated with adoption and implementation of Agile SMDs. More so, the lack of knowledge about Agile SDMs by the customers and the misalignment resulting from infrastructure that supports Agile adoption poses serious challenges for organisations that are well equipped with technological infrastructure to supports traditional SDMs. Other challenges included lack of guiding policies and procedures in some organisations, lack of knowledge on how to adopt the Agile SDMs and limited awareness of Agile methodologies. All these challenges contribute towards the low rates of acceptance and application of Agile SDM by BSDCs. In brief, software development in Botswana suffers from poor or lack of planning for software development and limited or lack of resources for software development and testing.

There are also many other challenges experienced by BSDCs when adopting Agile SDMs in Botswana.

Furthermore, the Job Characteristic Model (JCM) for job designs was also guided by the proposed framework. The JCM was adopted for this research study because, according to Kakar (2012, p.137), job design aspects of SDMs need to be taken into account when adopting and implementing SDMs. The job designs of Agile SDMs have been described by Kakar (2012, p.137) as being non-Taylorist simply because they are designed in such a way that they quickly react to dynamic situations (i.e. they are agile). The JCM was designed to provide a systematic framework to software developers for the adoption and implementation of SDMs (Kakar 2012, p.137). This assists software developers to analyse the characteristics of the SDMs they work with to enable them to make all the necessary adjustments for successful achievements of the results. Similarly, Batchelor et al. (2014, p.3) has submitted that the JCM was developed to assess and make adjustments on the work carried out by employees. JCM promises to work well for the BSDCs and evaluate how Agile SDMs are beneficial to them, to decide on any necessary enhancements and make recommendations for upgrading and meeting world standards. This helps to improve on the results, drive productivity and promote the satisfaction of employees (Batchelor et al. 2014, p.3).

Finally, the proposed framework was guided by the Technology Acceptance Model (TAM) theory developed by (Davis, Bagozzi & Warshaw 1989, p.982). Figure 33 is an illustration of the proposed framework for Agile adoption and implementation in Botswana. The framework was developed by considering the stages and order of steps presented in Figure 33 while at the same time supporting the objectives, findings and suggested courses of action presented in sections 5.2.1 to 5.2.8 of this research study.

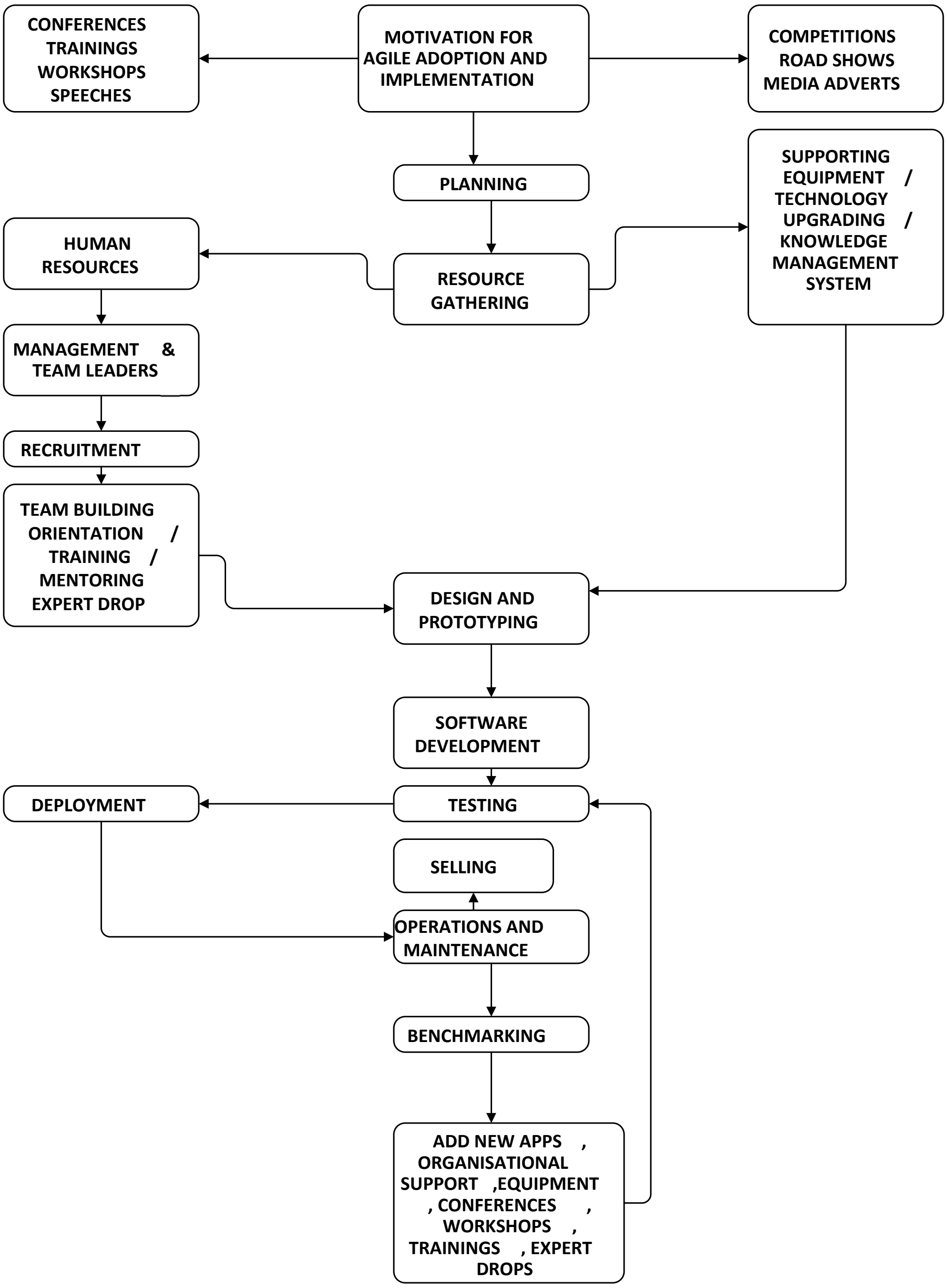

CONFERENCES
TRAININGS
WORKSHOPS
SPEECHES
MOTIVATION FOR AGILE ADOPTION AND IMPLEMENTATION
COMPETITIONS
ROAD SHOWS
MEDIA ADVERTS
PLANNING
SUPPORTING EQUIPMENT / TECHNOLOGY UPGRADING / KNOWLEDGE MANAGEMENT SYSTEM
HUMAN RESOURCES
RESOURCE GATHERING
MANAGEMENT & TEAM LEADERS
RECRUITMENT
TEAM BUILDING ORIENTATION / TRAINING / MENTORING EXPERT DROP
DESIGN AND PROTOTYPING
SOFTWARE DEVELOPMENT
DEPLOYMENT
TESTING
SELLING
OPERATIONS AND MAINTENANCE
BENCHMARKING
ADD NEW APPS , ORGANISATIONAL SUPPORT ,EQUIPMENT , CONFERENCES , WORKSHOPS , TRAININGS , EXPERT DROPS

Figure 33:Framework for Agile adoption and implementation in Botswana

5.2.10 Recommended practices for the successful adoption and implementation Agile SDMs in Botswana

Having discussed in detail how the results relate to the objectives of this research study, the researcher created a set of recommended Agile adoption practices that could assist Botswana software developers to successfully implement Agile. The framework is composed of several elements, which include:

a) **Stakeholder motivation for Agile adoption** – This will be conducted through collaboration amongst BSDCs that are still lagging in the adoption and implementation of Agile SDMs, government departments, pre-identified academic institutions, and customers and other stakeholders with an interest in software development. Depending on the targeted groups, workshops, lectures, motivational speeches, media advertisements, road shows, training sessions, competitions, posters, billboards, and many other strategies will be employed to disseminate information about Agile SDMs in Botswana. Such an approach is also expected to create awareness within the community so that interested stakeholders may make informed decisions when planning and gathering resources.

b) **Planning** – According to Swersky (2018) and Tiky (2016, p.8), planning looks at the project requirements and the management of the product. This planning process will also include the planning and allocation of required human resources and technical infrastructure. For human resources, there is a need to plan for required knowledge, skills, teams, orientation, training, mentoring or expert drop to empower and equip the employees with the needed skills. Expert drop is a process where an expert is assigned to join a team facing difficulties or implementing a specific task with a view to enhance the process delivery mentioned by Chen (2017, p79). Depending on the state of the organisations, relevant supporting machinery to upgrade or new machinery that supports Agile adoption and implementation need to be identified and planned for procurement prior to deployment. An efficient knowledge management system needs to be planned for by identifying the requirements, which include human resources and possible platforms.

Planning should be carried out to determine the production capacity needed by the organisation and prepare for changing product demand. Also, the organisation can plan for an amount of work they can achieve in a given space of time and estimation of time to be taken to complete specific stages. Planning also involves deciding on the level of software development that will be adopted, the version or type of Agile to adopt and size of projects to work on. Planning for policies and procedures should also be conducted. Once all the necessary resources and infrastructure are in place, the implementation phase can begin.

c) **Resource gathering** – Resource gathering involves gathering human and non-human resources. The management or the company will now put in place the human resources planned for, recruiting where a need exists, identifying team leaders and other personnel needed to beef up the management. Different skilled people for different tasks should be positioned and information on their needs should be gathered. Once recruitment has been completed, the empowerment can begin by conducting orientation, training, and mentoring in preparation for the execution of the project. Teams should be established, and their requirements should be presented and put in place for project execution. Supporting equipment, technology upgrading, and knowledge management systems should be put in place during resource gathering. Actual procurement of new or additional infrastructure identified during planning should be carried out. Policies and procedure guide documentation should be prepared and instituted. A need also exists to engage different business stakeholders and experts to assist with essential resource gathering. During resource gathering, management issues at organisational level will also need to be addressed.

d) **Design and prototyping** – When all the necessary and essential resources have been put together and mastered, software developers and architects can start to plan and design the software. For beginners, a visual Agile pipeline skeleton that provides a full view of Agile adoption can also be used as a guide. Software developers and architects should work with architecture frameworks and design patterns to assist them with the software designs. Software requirement specifications should be clearly documented and targeted customers should agree to the specifications (Tiky 2016, p.8). Prototyping should be put in place to determine possible solutions and for clients to make choices.

e) **Software development** - After designs have been made and identified, software development should begin, depending on the identified methodology. Programmers should work with programming tools provided by the organisation, and they should provide a document giving all the functional specifications of their product (Tiky 2016, p.8). In addition, interested stakeholders should be consulted as frequently as possible to assist so that expectations are met and a functional product is produced.

f) **Testing** – Once a functional software has been developed, it can be tested; this should be done by programmers, quality assurance experts and customers (Tiky 2016, p.9). The testing should also be done by different stakeholders to ensure that the software functional requirement meet the software client needs. Testing can be computerised so that it is never missed and/or for easy tracking. Lastly, several tests can be conducted to measure the quality of the software (Swersky 2018).

g) **Deployment** – Once testing has been completed, the developed and working software is deployed to production environments. At this phase, a deployment plan, contingency plan, user manuals, installation guides and administration manuals should be prepared and availed to the relevant stakeholders (Tiky 2016, p.9). The software should be released with adequate information to accommodate any future adjustments or changes.

h) **Operations and maintenance** – Once software has been deployed to production environments, it must be checked and reviewed for continued maximum functionality. Errors and bugs that manifest during production should be addressed, and the product should be restored to its intended operations. In addition, there is need to maintain the human systems and equipment. The system needs to be maintained for efficiency and a smooth flow of operations for the organisation to build reputation and relationships with its customers. When an organisation is satisfied with its operations and monitoring of the software products developed, products can then be developed for sale to clients who might not know about Agile SDMs.

i) **Benchmarking** – Once an organisation is comfortable with their products, they can compare their products and processes with those of other companies that are leading in the industry locally and internationally. This will allow the organisation to grow and adapt to international and acceptable standards. Benchmarking will allow the organisation to do an assessment and evaluation of their processes and products and to adjust accordingly to size up with their competitors or the industry.

j) **New additions** – These will be implemented after a specific period of time as the organisation grows and gets comfortable with handling specific projects. Additions can be made to the Agile-adopted SDMs by introducing new applications, buying new equipment or upgrading available machinery, hosting motivational and educative conferences to keep the developers up to date with the world standards, workshops to enhance skills and motivate teams, training sessions, expert drops to encourage and boost employee confidence, to demonstrate and to help struggling team members, to bring new and up-to-date information and skills, and to lead the teams. Maintenance will be carried out on an on-going basis, and it is important that a routine assessment of the system be conducted to avoid surprises and serious breakdowns. Since this research has established that executive support is important and lacking in Botswana, new additions will need organisational or executive support. Chan and Thong (2009, p.810) elaborated that innovation occurs in companies where the executive management is more supportive by upgrading technological infrastructure. Furthermore, executive support motivates developers to use Agile SDMs and to partake in knowledge administration tasks, which may result in management's recognition of developers (Chan & Thong 2009, p.810). This can change the developers' perceptions of Agile SDMs. Again, developers are more likely to grow in their interest in Agile SDMs and adopt positive perceptions toward the methodologies that may improve their acceptance of Agile SDMs. This means organisational management support plays a pivotal role in the adoption and success of Agile SDMs implementation (Chan & Thong 2009, p.810).

Therefore, considering the research study results and the JCM models, the Agile SDM adoption and acceptance framework was developed according to the goal of this research. This Agile SDM adoption and acceptance framework can guide BSDCs in their adoption and implementation of Agile SDMs. The Agile SDM adoption and acceptance framework presents ideas for optional and alternative actions that can be adopted to successfully implement Agile SDMs in the BSDI. The theoretical framework presents ideas of possible courses of action that can be taken to successfully adopt Agile SDMs in the BSDI. This theoretical framework can be taken as a tool and a guiding framework for software development organisations in Botswana and other countries facing similar situations as Botswana. In this research study, the theoretical framework added to the body of existing knowledge as new knowledge was generated. The theoretical framework highlights relationships that can facilitate use of Agile SDMs in Botswana. Since the adoption of Agile SDMs is still very limited in Botswana, the theoretical framework should be regarded as a tool for use by first-time adopters and should be viewed as an option to those who have long adopted and are using Agile SDMs. The framework provides an organised platform for old, new and future adopters to assess and evaluate factors that influence Agile adoption. This framework will have an impact in decision making for Agile SDMs adopters and in the software development industry at large.

5.3 Validation and Practical Implications and Impact on the Relevant Society

Validation of the research study findings is an important component of any research. The validation facilitates the establishment of results consistency with other previous studies conducted within the software development industry worldwide. For example, the factors affecting the adoption of Agile SDMs in Botswana related very well with what Mbelli and Hira (2016, p.219) and Asnawi, Gravell and Wills (2012a, p.60) established about South Africa and Malaysia, respectively. Botswana is a developing country that is still grappling with issues concerning education, technology infrastructure, technology skills, human resources, world-competitive standards and position. Thus, results of this research study have several applied values on the software development industry and Agile SDM acceptance and utilisation in Botswana, in the Southern African region, in the African continent, and in many other developing countries.

The findings of this research study can be utilised to determine influential factors on the adoption and implementation of Agile SDMs in Botswana and African countries. The research findings will shed light and give direction for further software development activities in the BSDI. The research study has also established an awareness of Agile SDMs to potential consumers who might have not been aware of these methodologies. All the concerned stakeholders in Botswana will now consider their operations with respect to the industry and not as individuals and only for the sake of getting government projects. The government of Botswana and the policy makers may want to go back to the drawing board and reconsider what can be done to boost and grow the industry. The BSDCs will be guided as they make decisions about adopting and implementing Agile SDMs and the requisite knowledge will be availed through this research thesis.

In addition, the research has implications on how the information systems theory is applied in real life in relation to software development practices. An information system theory is needed to solve real life problems and is applied in many forms including in SDLC methodologies. Grounded on the results of this research study, the findings and outcomes have potential to impact on the practice of information systems analysts and developers in the BSDI and neighbouring countries. Adopted methodologies have an impact on the quality of products and services in an organisation (Bhatia & Kumar 2012, p.46; Singh & Gautam 2016, p.666). Software development practices bring about a link between systems, information and information technology infrastructure to process information by following or adopting information systems theories and models (Al-Mamary, Shamsuddin & Aziati 2014b, pp.1279,84; Alter 2008, p.448).

In addition, validation of the proposed framework was conducted through interviewing some of the BSDCs that participated in the data collection exercise of this research study. The identified companies were presented with the discussions of the research findings and the proposed framework. The experts in these companies were asked to try and implement the proposed framework to confirm its viability, practical implications and impact in the industry. Organisations were given four weeks to try and use the framework in their everyday work. The researcher did interview the experts on their views concerning the framework after receiving their responses. Expert validation of qualitative research findings is supported by (Simon & Goes 2011, p.1). The testing of the framework was undertaken to understand the perceptions of the experts. The semi-structured interview approach was used to establish the usefulness and relevance of the framework in adoption of Agile SDMs in Botswana.

5.4 Limitations of the research

The research was limited in terms of sample size and sampling frame. Only companies that are active in tendering were considered for this research; other companies that are not tendering but may be doing better than the tendering ones were not considered in this research. The reason for targeting the tendering companies was to get the detailed results of specific cases. In addition, the research was also limited in terms of sample size. Since only a few companies were interviewed, the sample size may not necessarily be fully representative of entire software development industry. To this end, the limitations induced by the method and sample size means that any generalisation of the findings or results cannot necessarily be extended to all software development companies in Botswana and/or other countries.

5.5 Implications for policy, practice, and research

The findings of this research have implications on policy, practice, theory, and research. In addition, the findings of this research have implications on policy development by the government of Botswana, the BSDI and the software developers in the industry. This research established that Botswana is lacking key skills that are needed for superior adoption and implementation rates of Agile methodologies because the developers lack the requisite qualifications, knowledge, and experience in software development. This implies that the government needs to devise policies that promote higher levels and standards of software development in the country. Therefore, the government may need to form collaborations with policy makers, tertiary education institutions, The Human Resource Development Council (HRDC), BSDI and software developers to devise working policies that promote software development and the empowerment of software developers in Botswana. It will be important to include Give the important role the HRDC plays in planning and funding of the education and training as well as advising the Government of Botswana on all matters related to human capital development, it is important to include this state entity.

In practice, the findings of this research have implications on software developers and the BSDI. These findings imply that the software developers in Botswana and the software development industries should institute accrediting bodies that monitor, accredit, and empower software developers so that they become relevant and qualified in terms of international standards and best practice. Most of the software developers in Botswana need to upgrade their skills and qualifications to be on par with international standards and to practice at higher levels using modern Agile SDMs. There is a need to motivate and empower the BSDI and its practitioners for the adoption of modern Agile SDMs. The findings of this research also imply, that although the companies and the developers are aware of modern Agile SDMs, they lack proper knowledge, support, and capacity to adopt and implement the SDMs successfully. Therefore, additional experts may need to be brought into Botswana to train the industry. Alternatively, software developers may need to be taken to other countries for training where software development has been practiced and adopted successfully. Furthermore, the findings of this research imply that personal development is lacking within developers, and they may need the support of all concerned stakeholders to obtain further education and qualifications.

The findings of this research suggest that little research is being conducted in the BSDI thus resulting in limited awareness and knowledge of how other organisations and developers in other countries and the world at large are performing. Additional research and theory development need to be done to inform and to encourage the BSDI and its practitioners to step up their practices.

5.6 Gaps not covered by the research

The research has not revealed the personnel leading the companies and their qualifications, knowledge, and experience in relation to the adoption, implementation, and support of Agile SDMs. The research did not cover the relationship that exists between the government of Botswana and the software development industry. In addition, qualifications, training, and curriculum offered by tertiary institutions as far as Agile SDMs are concerned was not covered. A need still exists to consider the policies and regulations that guide software development in Botswana.

5.7 Future research opportunities

Future research should consider an inquiry involving the government, tertiary education institutions, research bodies, software development companies, consumers, and software development practitioners with a view to establish a working relationship that could drive the industry to world standards. Also, further research should consider establishing partnership programs within the local and international software development sector. Future research may consider the different Agile methods and how they are being or can be implemented by the BSDI to execute projects. It is important to establish if different Agile methods are beneficial to the industry. In addition, further research may include the application of Agile SDMs to cloud technologies and how cloud computing and agile methodologies can be integrated for better project management by business leaders in the Botswana business environment. Additionally, further research may require the integration Agile methodologies and internet of things to establish how the Botswana culture promote or reject Agile adoption in managing business projects in Botswana. Further research may also look for the applications of Agile methodologies in blockchains, supply chains, Industrial Internet of Things (IIoT) and other related fields to create competitive advantages in the business markets and communities in Botswana. It would be of necessity to understand the logistics involved for successful adoption and implementation.

5.8 Contribution of the Study

The research has made contributions to theory and practice of software development in Botswana and the world at large. The researcher can safely assume that there is limited research about software development in Botswana.

5.8.1 Extending knowledge on Agile SDMs Adoption Factors in Botswana

The proposed framework shows how the elements that can influence Agile SDMs adoption can be organised to support the adoption and implementation of Agile SDMs in Botswana. The proposed framework is not a linear process but consists of several interactions of the associated elements. This research and the proposed theoretical framework in Section 5.2.9.2, Figure 33 constitute a significant contribution to the body of knowledge of information systems, ICT and to the BSDI, which is part of software engineering in Botswana.

In addition, this research study has contributed immensely to the body of knowledge in general and in software engineering, information systems, software development and information and communication technologies. Software development is not only about information and communication technology, but it is about different personnel possessing different knowledge and skills and coming together to address specific human needs. From this research study, it is evident that most people in Botswana who are not in the software development industry (and to some degree even some people within the industry) lack information about Agile SDMs even though these methodologies have become so popular worldwide. It is important that people's doubts, misconceptions, and fears about Agile adoption are dealt with so that more and more companies, the government, and the public at large take the plunge to adopt and use Agile to their advantage. Most of the limiting factors are not impossible to resolve; this means that software engineering practitioners, the government, and tertiary education institutions can easily come together to alleviate the situation.

5.8.2 State of Agile Adoption in Botswana

This research found and presented details of the state of Agile adoption in Botswana. It has contributed to the pool of knowledge in this field by providing more suggestions about options that are available within software engineering and software development sectors. The thesis also highlighted knowledge and awareness about Agile methods to countries of Botswana and South Africa. The Agile adoption and implementation influencing factors and perceptions discussed will go a long way in providing a reference of procedures and processes for the acceptance and implementation of Agile SDMs in BSDCs and other software development companies in the region or the world. This research study will serve as a reference point for the early and future adopters of Agile methods. Moreover, has provided an awareness and important information about the factors that influence the adoption and implementation of Agile SDMs by software development practitioners and companies in Botswana and countries in the region. Lastly, the research will serve as a guide to software developers and companies intending to adopt and use Agile SDMs in the future.

5.8.3 Development of an Agile Adoption Framework

This research has proposed a framework for the adoption of Agile SDMs in Botswana. The research was built on existing knowledge about SDLC, SDMs, information systems, software engineering, and ICTs. The research builds and improves on currently available processes and procedures for adopting and implementing the Agile SDM as a software development life cycle through the proposed Agile SDM framework. The Agile SDM framework was aligned to address the unique situation of adopting and utilising Agile SDMs by considering the contextual issues existing in the BSDI. The proposed adoption framework was developed for the first time in this research and thus adds new information to the body of knowledge on the adoption of Agile SDMs in Botswana.

5.9 Chapter Conclusion

In Chapter 5, the findings of the study were discussed. The limitations of the study, its implications for policy, practice and research, gaps not covered, recommendations for future research and contribution of the study were also mentioned. This chapter presented detailed results and findings emanating from the research study. The results and findings were addressed by considering the research study objectives, reviewed literature, and the research findings arising from collected and analysed data. The chapter also presented a proposed Agile adoption framework that can be used as a guide by software developers and current and potential adopters of Agile SDMs in Botswana. The chapter concluded by presenting the limitations of the research, gaps not covered by the research, possible future research opportunities, validation and practical implications and impact on the relevant society, and contributions made by this research study as well their relevance to the pool of knowledge.

CHAPTER 6: THEORETICAL FRAMEWORK VALIDATION

6.1. Introduction

This chapter presents the validation of the theoretical framework developed in this research and based on the title "*A framework for the adoption of modern software development methodologies in Botswana*". In other words, the validation sought to evaluate the findings of this research. Souza and Silva (2011, p.778) assert that validation is one of the strategies used for the evaluation of the findings of qualitative research. In addition, Souza and Silva (2011, p.778) professed that validation of research findings assists in establishing whether the findings are realistic and are able to represent the given situation in practical terms. This means that the authentication of the theoretical framework suggested in this research for the adoption and implementation of Agile SDMs in Botswana was undertaken to show that the researcher's findings and interpretations are practical, and they represent reality instead of mere imagination.

Specifically, the aim of the validation step is to evaluate the relevance and value of each stage that was proposed in the theoretical framework. To establish the relevance and value of each stage, one-on-one in-depth interview sessions were held with five of the organisations that were interviewed in the main research study. The five organisations were selected based on their willingness and availability as well as the convenience of the researcher.

A summary of the research study results, including a diagram and explanation of each of the stages of the proposed Agile Adoption and Implementation framework was collated. The summary was e-mailed to willing and interested organisations 3 – 4 weeks prior to the scheduled interviews. The validation was targeted at software development experts and the senior management who are responsible for decision making and are very knowledgeable about software development. In other words, purposive sampling was employed for reasons of fulfilling the validation objective. The validation participants were given 3 – 4 weeks to scrutinise the summary and attempt to implement the proposed framework to check its practicality, including relevance and value of each stage. Upon completion of the 3 – 4 weeks' face-to-face interviews with identified teams or participants within the selected organisations, a content analysis was undertaken to determine the relevance and value for the proposed Agile software development methodology (SDM) framework model. The content analysis of information generated through the validation process is presented in the section that follows.

6.2. Valuation approach for the validation of the proposed framework

The proposed model was validated through communicative validation since the researcher went back to the research participants to confirm the research findings with respect to their lived software development experiences (Call-Cummings, 2017, p.192). Call-Cummings (2017, p.192) argues that the level and quality of representation of a research report to the reality can only be confirmed by the readers of the report who understand the content and context of the research. In this regard, Lankshear & Knobel (2018, P. 45) explained that communicative validity is interested in the quality of a qualitative research in terms of the contents because the readers of the final report can confirm that the report indeed represents the real-life experiences appropriately. Thus, in this research, the researcher had to go back to the research participants with the finished research report and the identified respondents were asked to make an evaluation of the research findings including considering the proposed framework and its applicability to the Botswana software development industry (BSDI). For the research findings to be credible, the targeted audience must be convinced that the finished research report represents the real situation, and it provides adequate grounds for quality research processes. Thus, based on these conditions, the researcher analysed the validation data and made conclusive remarks from the remarks and perceptions of the respondents based on their practical and lived experiences.

6.3. Analysis of Validation Data

Based on the interactions of the respondents and the research report, the researcher collected written responses and noted down responses and expressions that were made by validation participants during the validation process. The collected data were analysed using qualitative content analysis.

6.3.1. Expertise of the Respondents

All the respondents who participated in the validation of the findings of this research including the development of the theoretical framework were managing directors and technical directors who doubled as programmers and consultants. This means that experts in software development were mainly interviewed for validation. However, for the two organisations that were interviewed, software development teams and their expert leaders were involved in the interviewing process. In cases where teams were involved, a discussion of the posed question took place before a conclusion on the adopted response was reached. This means experts and other team players who are not necessarily experts but are instead engaged in the software development tasks were interviewed and their responses were recorded.

6.3.2. Framework Relevance

Most of the respondents found the research findings, especially the proposed theoretical framework, to be relevant for the BSDI, refer figure 33. Framework for Agile SDM Adoption and Implementation in Botswana Validation Letters for further confirmation. Most of the respondents indicated that the proposed framework was relevant, is needed, and useful to the BSDI. For example, validation participant 5 (VP5) said the following:

> *“Yes, the framework is relevant for Botswana Software Development Companies, but most of the works we do in Botswana are based on Government Departments’ Tenders, the tendering process doesn’t suit for Agile Development Model, they expect a defined requirement, time and cost estimates. We need to adapt a hybrid model for government tender based projects instead of full Agile i.e., while we bid for the tender with waterfall model, the internal execution can be handled with Agile model”.*

Validation participant 3 (VP3) added that:

> *“I have found the framework to be very relevant to the Botswana software development industry and the software development situation in the country. It is practical and simple to follow thereby making agile adoption possible to SMEs and established companies in Botswana”.*

Whereas validation participant 2 (VP2) said:

> *“I can comfortably call it relevant with finer modifications to make it more practical and valuable in the Botswana software development industry”.*

In addition, validation participant 1(VP1) said:

> *“The framework is very relevant to the Botswana software development industry. It is very easy to follow”.*

It is important to note that most of the respondents found the framework to be relevant and practical, especially on the technical aspects, however, a few business concerns relating to the stages of the proposed framework were raised. The relevance of the stages of the proposed framework is deliberated in more detail in the ensuing section.

6.3.3. Relevance of Stages in the Proposed Framework

All the respondents indicated that the stages of the proposed framework are relevant and well defined, thus making the framework easy to use and follow. Besides adding value to the procedures for the acceptance and implementation Agile SDMs, the respondents indicated that the stages of the proposed framework are necessary and helpful for the process leading to the software application end-product. However, a few concerns were raised on the motivation stage. Some of the respondents, especially those who are well established and are practicing Agile SDMs, felt that while the motivation stage is a very significant stage, this stage can be a problem in a business environment since businesses use the adoption Agile SDMs as a competitive advantage in the market.

On one hand, VP 3 mentioned that:

> *“Agile adoption planning may be followed by Management buy-in.”*

On the other hand, VP 4 stated that:

> *“Yes, the stages are relevant, they are required for the framework otherwise people see it as a big chunk, stages make it easy and simple to understand and follow. Yes, the stages can add value to the process of Agile adoption. Stages make it easy and simple to understand, it makes software delivery faster with good efficiency and quality. The stages are necessary and helpful as any bottlenecks that might come during the software development can be identified easily because of the stages, it also helps to go back to particular stage otherwise it becomes cumbersome to manage.”*

VP 1 said:

> *“Each stage needs validation and very vital expect benchmarking which is optional anyways. However, who will motivate for Agile SDMs, as a business we will not do that because that has been our strength and competitive advantage in the market. Also, the market is very limited since the government is the major customer and everyone is trying to please the government. In addition, I think a SWOT analysis should be done before the designing and prototyping stage. For an organisation to assess their opportunities, disadvantages and strengths in the market”*

VP 2 alluded to the following:

> *“I found the framework to be valuable and up to the mark. Though I have few findings on the framework staging. I have little more concern on the motivation, training, and recruitment stages. These stages might be needful to start with but have to be planned for very cleverly and should be made sure that these are not being redundant or repetitive. Otherwise, there is a risk of increased lead time and project funds”.*

However, organisations that were concerned about the motivation stage have proposed that organisations motivate for internal adoption of Agile SDMs by inviting experts from overseas or established organisations to help their companies because corporate motivation will not work except if the government takes the lead. In addition, these organisations have indicated that, as businesses, they need some type of motivation that will spill over and motivate others to do what they are already doing. Interestingly, organisations that are not yet using or have partly adopted Agile SDM welcomed the idea of all stakeholders coming together and to motivate each other to adopt Agile SDMs.

6.3.4. Framework Practical Use and Addressing of Limiting Issues

All the respondents agreed that the framework can be used by any organisation in the software development industry. In addition, the respondents stipulated that the framework is addressing a lot of issues that have been limiting the adoption of Agile SDMs in Botswana. However, the issue of the government being the largest customer featured prominently and it was mentioned that the framework can work very well especially if the government can itself adopt Agile SDMs. In other words, some of the respondents indicated that they are limited to Traditional Waterfall since it is the only SDM that is accepted by the government, a major consumer and stakeholder.

6.3.5. Provision of a starting point for potential Agile adopters and framework complexity

All the respondents concurred that the proposed framework provides a starting point for potential Agile adopters and implementers in Botswana, as outlined by VP2.

VP2 said:

> "*Yes, very much a starting point.*"

Whilst VP 3 said:

> "*Yes, definitely this framework is going to be a starting point for whoever willing to Adopt Agile as the process is clearly articulated*."

In addition, most of the respondents found the framework to be very easy to follow, especially given the clearly laid-out stages. However, VP1 commented:

> "*Agile is not linear in nature and it allows different modules to run independently thereby quickening the cycles.*"

6.3.6. Proposed Framework and Recommendation

It was evident most of the respondents were going to use the framework and would recommend it to other companies that are struggling with the adoption and implementation of Agile SDMs within the software development industry.
To this end, VP1 said:

> *"Yes, we can make use of the framework and we can recommend to other organisations in Botswana. However, we will recommend after seeing and safeguarding our secret first. We won't tell other organisations our business secrets."*

6.3.7. Proposed Framework Recommendations from the Validation Participants

The following recommendations for improving the framework were made by the validation participants:

VP1 said that:

> *"Though the framework is done, it's in conceptual stage, it has to be used by BSD, and continuous monitoring is required so that improvisation can be made till this is matured."*

VP2’s suggestion was as follows:

> *“Addition of validation of sub-stages between each stage would help a great deal. This will also help rectify the mistakes in production in time and allows the addition of the missing elements in time.”*

VP 5 emphasised the point by saying that:

> *“Most of the works we do in Botswana are based on Government Departments Tender, the tendering process doesn’t suit for Agile Development Model, and they expect a defined requirement, time and cost estimates. We need to adapt a hybrid model for Government tender-based projects instead of full Agile while we bid for the tender with Waterfall model, but internal execution can be handled with Agile model.”*

VP 4 specified that:

> *“Sometimes mix of Agile and Traditional project management methodology works very well. So, opportunity of use the combination should not be negated.”*

In addition, VP 3 said:

> *“The proposed stages like Agile awareness building, training, and management buy-in are very much needed for Botswana Software Industry as the Agile adoption is low in the industry currently. I validate the proposed framework and can comfortably call it relevant with finer modifications to make it practical and valuable in the Botswana software development industry.”*

All the respondents consulted in this validation process accepted the framework and indicated that it is very relevant, practical and a valuable tool for the BSDI.

6.4. Chapter Conclusion

The proposed framework was perceived as relevant, practical, and valuable by all the validators. However, a few suggestions such as conducting a SWOT analysis and identifying the stakeholders to take the responsibility to motivate, bearing in mind the business aspect, were made. The validation of the proposed framework was based on the relevance of the entire framework with a view to identify the value of each proposed stage and its relevance in the context of BSDI and the research. In conclusion, it can be said that the proposed framework was validated in terms of relevance, value, and practicality. The framework can be adopted as proposed and modifications or improvements can be made depending on the position of the Botswana software development organisations. Further consultations can be done to adopt suggested adjustments.

CHAPTER 7: RESEARCH CONCLUSIONS AND RECOMMENDATIONS

7.1. Introduction

This chapter concludes the findings of this research study by looking at how the research questions posed at the beginning of the research study were addressed. The conclusion of the research study is followed immediately by a presentation of recommendations that are informed by the findings emanating from the research study.

This research study explored and determined the factors that affect the adoption of modern software development methodologies (SDMs) such as Agile, Scrum, Extreme Programming, Crystal Clear and others by the Botswana Software Development Industry (BSDI). This goal of the study was achieved by answering the following key research question, which was derived from the problem statement:

> "*What are the factors affecting Botswana's software development companies and software development professionals in adopting and utilising modern software development methods?*"

The research objectives and research sub-questions were deliberated on by considering a global view of all the emerging themes and establishing the implications in terms of information systems theory and software development practice.

7.2. Research Conclusions

The thrust of this research study was grounded on the extent to which Botswana software development companies (BSDCs) are adopting and implementing relevant Agile SDMs. The study undertook a thorough examination of elements affecting the acceptance of Agile SDM in the BSDI. The investigation has led to the development of a proposed framework that will provide guidance for Agile adoption in Botswana. The research objectives were addressed through associated research questions and interviews were conducted with the view to answer the research questions. To answer the research questions, two questionnaires were designed for employees and employers from identified BSDCs. For each research question, several questions were asked to solicit information from the employees and employers to answer the research question. The questions in the two questionnaires were structured differently but were linked to the same research question to assist the researcher to compare the employees' and the employers' responses. This was done to validate the responses given by the research respondents from the same organisation. Gathered data from the employees and employers was examined to identify patterns, themes, and commonalities. Thereafter, the data were summarised according to emerging themes and patterns and the following conclusions were made.

Considering objective 1 - To understand the perceptions of twelve BSDCs (and their professional software developers) about the adoption and implementation of software development methodologies in Botswana.

Findings emanating from this research have led to a conclusion that BSDCs have mixed perceptions about the adoption and implementation of SDMs. Whereas respondents generally reacted much more positively towards Agile SDMs, negative sentiments were expressed towards both Traditional and Agile SDMs. Traditional SDMs were negatively perceived because of the extensive documentation requirement and difficulties when wanting to make changes. The Agile SDMs were specifically criticised for their lack of documentation and possible failure to end the iterations. It is important to note that most of the perceptions towards Agile were based on general knowledge and were not backed by first-hand or practical experience. Although positive feedback on Agile SDMs was received from most of the respondents, the same respondents had never used it practically in software development. Also, the inherent iterative approach of the Agile SDMs, which results in the customer being billed on a continuous basis thus potentially making it expensive due to constant interaction with client, did not sit well with the respondents. These mixed perceptions have led to the low adoption and implementation of Agile SDMs in Botswana.

Considering objective 2 - To explore factors influencing the adoption and usage of SDMs in the BSDI with specific reference to twelve software development companies in Botswana. Firstly, this research concluded that software development in Botswana suffers from poor or lack of planning for software development. Very few of the BSDCs studied were able to show that they follow set principles and regulations to guide their practices. Failure to confirm the existence of policies and procedures in the companies reflects lack of planning in the companies as policies and procedures are designed and established at the planning stage of software development. Secondly, it was concluded that software development in Botswana suffers from poor or lack of resources for software development and software testing. The knowledge management and support offered to software developers by the BSDI remains inadequate. The knowledge management and support standards for the BSDI as well as the adoption and implementation of Agile SDMs is low compared to the rest of the world. The knowledge management support levels are low because there is no effort in building knowledge on Agile SDMs. The software developers lack the requisite knowledge and experience needed for efficient knowledge management and some of the companies are even encountering technological challenges. Furthermore, this research concluded that several challenges are being experienced by the software development industry in Botswana when adopting Agile SDMs. These challenges include resistance to migration from Traditional to Agile SDMs and fear of concomitant costs associated with this switch. The limited knowledge about Agile SDMs coupled with unavailability of skilled personnel, insufficient resource support, and inefficient knowledge management systems works against the adoption of Agile SDMs in Botswana.

Considering objective 3 - To develop a framework in collaboration with the software development community for assisting and guiding the BSDI, BSDCs (and their professional software developers) to adopt and implement modern and Agile SDMs in Botswana. This research concluded that the BSDCs are mostly using Traditional SDMs. However, these Traditional SDMs are often used in combination with Ad-hoc or Agile SDMs. Also, very few companies in Botswana are developing software through the exclusive use of Agile SDMs. Of the few companies that are employing Agile SDMs, they are doing it in combination with Traditional or Ad-hoc SDMs. Therefore, it suffices to say that the adoption and implementation of Agile SDMs by the BSDI is still very low compared to international best practice. In addition, software development in Botswana is still being done at a small-scale level whereby an organisation develops one hundred or less successful software applications over a period of 5 – 10 years. As such, this research developed a framework in collaboration with the software development community to assist and guide the BSDI, BSDCs and their professional software developers to adopt and implement modern and Agile SDMs in Botswana.

7.3. Recommendations

Based on the challenges identified in this research study, the following recommendations are made for improved adoption and implementation of Agile SDMs in Botswana. To address lack of knowledge and associated skills for the adoption and implementation of Agile SDMs, it is recommended that:

a) Companies should put in place measures targeted at staff development. It is envisaged that such measures will upskill their staff and, in the process, ensure that the software development industry is well equipped and uplifted in terms of skills and competence. BSDCs can invite experts to teach and facilitate knowledge transfer sessions at their respective premises. Alternatively, BSDCs can outsource software development experts to work with and transfer skills to local employees.

b) BSDCs should establish partnerships with academic institutions whereby BSDCs are able to sponsor their employees to attend modern SDMs courses at the said academic institutions. In addition, BSDCs can conduct workshops where academic practitioners are invited to the BSDCs to make presentations. Such a collaboration effort can also create an opportunity for a staff exchange program.

c) Software development companies in Botswana should form interactive groups and platforms where software development practitioners can meet, discuss and provide guidance to each other on relevant and topical industry issues. BSDCs can partner to host conferences that are beneficial to their employees. Such an approach could potentially address challenges relating to resistance to change by adopters of traditional SDMs, fear of high costs incurred when adopting modern Agile SDMs, insufficient knowledge and experience about Agile SDM, absence of skilled personnel, insufficient resource support, and inefficient knowledge management systems.

d) To address lack of skilled software developers, it is recommended that:

  i) A professional body for accrediting and auditing software development practitioners, developing policies and procedures for monitoring, and evaluating purposes can be formed.

  ii) Expert drops can be used to allow the inexperienced practitioners to learn from and be mentored by experts.

e) To address lack of knowledge about Agile SDMs and associated experience, it is recommended that:

  i) The BSDCs and the government collaborate by establishing fellowships and/or bursaries whereby Botswana can send software development practitioners to embark on formal qualifications in foreign academic institutions with a demonstrable track record in the training of software development.

  ii) Countrywide awareness campaigns (e.g., place adverts in both print and electronic media) and roadshows of Agile SDMs can be conducted targeting software development practitioners and customers.

f) To address mixed and negative perceptions towards Agile adoption and implementation by companies and developers, it is recommended that:

  i) Marketing and motivation of Agile adoption through countrywide awareness campaigns (e.g., place adverts in both print and electronic media) and roadshows of Agile SDMs be conducted that target software development companies, practitioners, and customers. This will contribute towards increasing knowledge on and awareness of Agile SDMs.

ii) The BSDCs and the developers be educated about Agile through government initiatives and collaborations with the industry and tertiary institutions.

REFERENCES

Aaen, I 2002, 'Challenging Software Process Improvement By Design', *ECIS*, pp. 379-90.

Abdalhamid, S & Mishra, A 2017, 'Factors in agile methods adoption', *TEM Journal,* vol. 6, no. 2, pp. 416-21.

Abdalhamid, S & Mishra, A 2017, 'Adopting of Agile methods in Software Development Organizations: Systematic Mapping', *TEM Journal,* vol. 6, no. 4, pp. 817-25.

Abrahamsson, P, Salo, O, Ronkainen, J & Warsta, J 2017, 'Agile software development methods: Review and analysis', *arXiv preprint arXiv:1709.08439*.

Aceto, G, Persico, V & Pescapé, A 2020, 'Industry 4.0 and health: Internet of things, big data, and cloud computing for healthcare 4.0', *Journal of Industrial Information Integration,* vol. 18, p. 100129.

Adanna, AA & Nonyelum, OF 2020, 'Criteria for Choosing the Right Software Development Life Cycle Method for the Success of Software Project', *IUP Journal of Information Technology,* vol. 16, no. 2, pp. 39-65.

Aggarwal, S & Dhir, S 2014, 'Swift Tack: A New Development Approach.', *IEEE:Internationai Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT).* pp. 476 - 81.

Agile Alliance 2014, *Agile Modeling: Effective Practices for Modeling and Documentation*, updated 2014, viewed 12/10, <http://www.agilemodeling.com/>.

Ahimbisibwe, A, Daellenbach, U & Cavana, RY 2017, 'Empirical comparison of traditional plan-based and agile methodologies', *Journal of Enterprise Information Management*.

Ahmad , A, Z., Shahela , S, Muzafar , K, Javed , I, Adnan , A, Abdul , W, Ahmad , A & Atif , A 2018, 'Taxonomy of factors causing integration failure during global software development', *IEEE Access,* vol. 6, pp. 22228-39.

Ajzen, I 1991, 'The Theorry of Planned Behavior', *Organizational Behaviour and Human Decision Processes,* vol. 50, pp. 179 -211.

Akbar, MA, Sang, J, Khan, AA, Shafiq, M, Hussain, S, Hu, H, Elahi, M & Xiang, H 2017, 'Improving the quality of software development process by introducing a new methodology–AZ-model', *IEEE Access,* vol. 6, pp. 4811-23.

Akbar, MA, Sang, J, Khan, AA, Amin, F-E, Hussain, S, Sohail, MK, Xiang, H & Cai, B 2018, 'Statistical analysis of the effects of heavyweight and lightweight methodologies on the six-pointed star model', *IEEE Access,* vol. 6, pp. 8066-79.

Al-Gahtani, SS, Hubona, G & Wang, J 2007, 'Information technology (IT) in Saudi Arabia', *Culture and the acceptance and use of IT, Information & Management,* vol. 44, no. 8, pp. 681-91.

Al-Kautsar, E, Salleh, N, Hoda, R & Asnawi, AL 2013, 'Preliminary analysis on the adoption of agile software development methodology in Indonesia', *development,* vol. 29, p. 25.

Al-Mamary, Y, H., Shamsuddin, A & Aziati, N 2014a, 'Factors Affecting Successful Adoption of Management Information Systems in Organizations towards Enhancing Organizational Performance', *American Journal of Systems and Software,* vol. 2, no. 5, pp. 333-39.

Al-Mamary, Y, H., Shamsuddin, A & Aziati, N 2014b, 'The role of different types of information systems in business organizations: A review', *International Journal of Research,* vol. 1, no. 7, pp. 333-39.

Al-Zewairi, M, Biltawi, M, Etaiwi, W & Shaout, A 2017, 'Agile software development methodologies: survey of surveys', *Journal of Computer and Communications,* vol. 5, no. 05, pp. 74-97.

Al-Zewairi, M, Biltawi, M, Etaiwi, W & Shaout, A 2017, 'Agile software development methodologies: survey of surveys', *Journal of Computer and Communications,* vol. 5, no. 05, p. 74.

Al Kautsar, E, Salleh, N, Hoda, R & Asnawi, AL 2013, 'Challenges in Adopting Agile Practices: Perceptions of Software Practitioners in Indonesia'.

Al Kurdi, B, Alshurideh, M, Salloum, S, Obeidat, Z & Al-dweeri, R 2020, 'An Empirical Investigation into Examination of Factors Influencing University Students’ Behavior towards Elearning Acceptance Using SEM Approach'.

Aldahmash, A, Gravell, AM & Howard, Y 2017, 'A review on the critical success factors of agile software development. ', *In European conference on software process improvement* Springer, Cham, pp. 504-12.

Alfraihi, H, Lano, K, Kolahdouz-Rahimi, S, Sharbaf, M & Haughton, H 2018, 'The impact of integrating agile software development and model-driven development: a comparative case study', *International Conference on System Analysis and Modeling*, Springer, pp. 229-45.

Alhashmi, SF, Salloum, SA & Mhamdi, C 2019, 'Implementing artificial intelligence in the United Arab Emirates healthcare sector: an extended technology acceptance model', *Int. J. Inf. Technol. Lang. Stud,* vol. 3, no. 3, pp. 27-42.

Alqudah, M, K. & Razali, R 2017, 'Key factors for selecting an Agile method: A systematic literature review', *International Journal on Advanced Science, Engineering and Information Technology,* vol. 7, no. 2, pp. 526-37.

Alshamrani, A & Bahattab, A 2015, 'A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model', *International Journal of Computer Science Issues (IJCSI),* vol. 12, no. 1, p. 106.

Alshamrani, A & Bahattab, A 2015 'A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model ', *IJCSI International Journal of Computer Science Issues,,* vol. 12, no. 1, pp. 106 - 11.

Alter, S 2008, 'Defining information systems as work systems: implications for the IS field', *European Journal of Information Systems,* vol. 17, no. 5, pp. 448-69.

Amani, M, Ghorbanian, A, Ahmadi, SA, Kakooei, M, Moghimi, A, Mirmazloumi, SM, Moghaddam, SHA, Mahdavi, S, Ghahremanloo, M & Parsian, S 2020, 'Google earth engine cloud computing platform for remote sensing big data applications: A comprehensive review', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Ambler, S, W. 2012, *Agile Model Driven Development (AMDD):The Key to Scaling Agile Software Development*, Ambysoft Inc., viewed 27 July, <http://www.agilemodeling.com/essays/amdd.htm>.

Ambler, S, W. 2013, *How Agile Are You? 2013 Survey Results*, Scot W Ambler + Associates, 2013, viewed 2nd July 2014, <http://www.ambysoft.com/surveys/howAgileAreYou2013.html>.

Ameller, D, Franch, X, Gómez, C, Martínez-Fernández, S, Araújo, J, Biffl, S, Cabot, J, Cortellessa, V, Fernández, DM & Moreira, A 2019, 'Dealing with non-functional requirements in model-driven development: A survey', *IEEE Transactions on Software Engineering,* vol. 47, no. 4, pp. 818-35.

Amir, M, Khan, K, Khan, A & Khan, A, M. 2013, 'An Appraisal of Agile Software Development Process', *International Journal of Advanced Science and Technology,* vol. 58, pp. 75-86.

Anureet, K & Kulwan, K 2015, 'Suitability of existing Software development Life Cycle (SDLC) in context of Mobile Application Development Life Cycle (MADLC)', *International Journal of Computer Applications (0975 – 8887),* vol. 116 no. 19, pp. 1-6.

Anwer, F, Aftab, S, Waheed, U & Muhammad, SS 2017, 'Agile Software Development Models TDD, FDD, DSDM, and Crystal Methods: A Survey', *International journal of multidisciplinary sciences and engineering,* vol. 8, no. 2, pp. 1-10.

Arthur, T 2013, *Agile Adoption: Measuring its Worth*, SAS Institute, Inc, San Francisco.

Asnawi, A, L., Gravell, A, M. & Wills, G, B. 2012a, 'Factor analysis: Investigating important aspects for agile adoption in Malaysia', *2012 Agile India*, pp. 60-63.

Asnawi, A, L., Gravell, A, M. & Wills, G, B. 2012b, 'Emergence of agile methods: Perceptions from software practitioners in Malaysia', *2012 Agile India*, pp. 30-39.

Ayalew, Y & Motlhala, K 2014, 'An ISO/IEC 15504 based Software Process Assessment in Small Companies ', *International Journal of Software Engineering and Its Applications,* vol. 18, no. 6, pp. 121 - 38.

Baccelli, E, Eichinger, T, Padilla, F, Javier, Acosta. & Schleiser, K 2016, 'The future of IoT software must be updated', *IAB Workshop on Internet of Things Software Update (IoTSU)*.

Bagnoli, C, Garlatti, A, Massaro, M, Dal Mas, F, Paschetto, M & Paschetto, S 2018, 'Winning business models for the 4th industrial revolution', *Theory and Applications in the Knowledge Economy*, p. 59.

Bandura , A 1977, 'Self-efficacy: toward a unifying theory of behavioral change', *Psychological review,* vol. 84, no. 2, p. 191.

Batchelor, J, H., Abston, K, A., Lawlor, K & Burch, G, F 2014, 'The job characteristics model: An extension to entrepreneurial motivation', *Small Business Institute Journal,* vol. 10, no. 1, pp. 1-10.

Beck, K, Beedle, M, Van Bennekum, A, Cockburn, A, Cunningham, W, Fowler, M, Grenning, J, Highsmith, J, Hunt, A & Jeffries, R 2001, 'Manifesto for agile software development'.

Betta, J & Boronina, L 2018, 'Transparency in Project Management–from Traditional to Agile', *Third International Conference on Economic and Business Management (FEBM 2018)*, pp. 446-49.

Bhadoriya, N, Mishra, N & Malviya, A 2014, 'Agile Software Development Methods, Comparison with Traditional Methods & Implementation in Software Firm', *Int. J. Eng. Res. Technol,* vol. 3, no. 7.

Bhatia, P, K & Kumar, G 2012, 'Impact of agile methodology on software development process', *International Journal of Computer Technology and Electronics Engineering (IJCTEE),* vol. 2, no. 4, pp. 46-50.

Blasi, L, Bordonaro, R, Borsellino, N, Butera, A, Caruso, M, Cordio, S, Liborio, DC, Ferraù, F, Giuffrida, D & Parra, HS 2020, 'Reactions and countermeasures of medical oncologists towards the incoming COVID-19 pandemic: a WhatsApp messenger-based report from the Italian College of Chief Medical Oncologists', *ecancermedicalscience,* vol. 14.

Boehm, B & Turner, R 2005, 'Management Challenges to Implementing Agile Processes in Traditional Development Organizations', *IEEE SOFTWARE ComputerSociety*, pp. 30 -39.

Borge, S & Poonia, N 2020, 'REVIEW ON AMAZON WEB SERVICES, GOOGLE CLOUD PROVIDER AND MICROSOFT WINDOWS AZURE', *Advance and Innovative Research*, pp. 53-58.

Botswana Innovation Hub 2016, *Botswana Innovation Hub*, BOTSWANA, viewed 06/08/2019, <http://www.bih.co.bw/>.

Brummelen, J & Slenders, T 2019, *Modern Software Development*, Compact Magazine, Netherlands. <https://www.compact.nl/en/articles/modern-software-development/>.

Bui, KT, Nguyen, HM, Tran, NVH & Nguyen, BH 2020, 'Lecturers' adoption to use the online Learning Management System (LMS): Empirical evidence from TAM2 model for Vietnam', *JOURNAL OF SCIENCE HCMC OU-ECONOMICS AND BUSINESS ADMINISTRATION,* vol. 10, no. 1.

Burrell, G & Morgan, G 1979, *Sociological paradigms and organisational analysis*, Heineman, London.

Cabot, J 2020, 'Positioning of the low-code movement within the field of model-driven engineering', *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pp. 1-3.

Chan , F, K, Y. & Thong , J, Y, L. 2009, 'Acceptance of agile methodologies: A critical review and conceptual framework', *Decision Support Systems,* vol. 46, no. 4, pp. 803-14.

Chan, FKY & Thong, JYL 2009, 'Acceptance of agile methodologies: A critical review and conceptual framework', *Decision Support Systems,* vol. 46, no. 4, pp. 803-14.

Chau, P, Y, K. 1996, 'An empirical investigation on factors affecting the acceptance of CASE by systems developers', *Information Management,* vol. 30, no. 6, pp. 269-80.

Chen, L 2017, 'Continuous delivery: overcoming adoption challenges', *Journal of Systems and Software,* vol. 128, pp. 72-86.

Chevers Whyte, C & Chevers, D 2015, 'The adoption of agile software development methods in Jamaica', pp. 1-9.

Cho, H-J & Jeong, K 2019, 'A Study on the Connective Validity of Technology Maturity and Industry for Core Technologies based on 4th Industrial Revolution', *Journal of the Korea Convergence Society,* vol. 10, no. 3, pp. 49-57.

Christie, M, Rowe, P, Perry, C & John, C 2000, 'Implementation of Realism in Case Study Research Methodology ', *International Council for Small Business*, 1 -36.

Cobelli, N 2020, 'The Choices of Adoption of eHealth Tools: An Analysis of Research Models', *Innovation in Community-Based Private Practices Through eHealth*, Springer, pp. 21-37.

Cockburn, A & Highsmith, J 2001, 'Agile software development, the people factor', *Computer,* vol. 34, no. 11, pp. 131-33.

Cohen, L, Manion, L & Morrison, K 2007, *Research Methods in Education*, Sixth edition edn, Routledge, London and New York.

Conger, S 2011, 'Software Development Life Cycles and Methodologies: Fixing the old and adopting the new', *Sprouts: Working Papers on Information Systems*, pp. 66-90.

Coolican, H 2017, *Research methods and statistics in psychology*, Psychology Press, London,United Kingdom.

Craddock, A, Roberts, B, Richards, K, Godwin, J & Tudor, D 2012, 'The DSDM agile project framework for scrum', *Dynamic Systems Development Method (DSDM) Consortium,* vol. 1.

Creswell, J, W. 2009, 'Research design: Qualitative, quantitative, and mixed methods approaches', *Los angeles: University of Nebraska–Lincoln*.

Cronje, JC 2012, 'The ABC (aim, belief, concern) instant research question generator', <https://pdfs.semanticscholar.org/e132/2ddf75705095fd33dc573f489861c2aaf11f.pdf>.

Crotty, M 1998, *The Foundations of Social Research: Meaning and Perspective in the Research Process*, Sage,

Cruzes, D, S., Dybå, T, Runeson, P & Höst, M 2015, 'Case studies synthesis: a thematic, cross-case, and narrative synthesis worked example', *Empirical Software Engineering,* vol. 20, no. 6, pp. 1634-65.

Darke, P, Shanks, G & Broadbent, M 1998, 'Succefully Completing Case Study research: Combining Rigour, Relevance and Pragmatism', *Journal of Information System,* vol. 8, no. 4, pp. 273 - 89.

Davis, F, D. 1986, *A technology acceptance model for empirically testing new end-user information systems: Theory and results.*, Sloan School of Management, Massachusetts Institute of Technology., Massachusetts, United States

Davis, F, D., Bagozzi, R, P. & Warshaw, P, R 1989, 'User acceptance of computer technology: a comparison of two theoretical model. ', *Management Science,* vol. 35, no. 8, pp. 982-1003.

Davison, A & Fitzgerald, F 2006, *Information Systems Development, Methodologies, Techniques & Tools*, 4th edn, McGraw-Hill Education; , Berkshire, United Kingdom.

De Souza, JV 2020, 'Moving from traditional to Agile information systems development in a large organisation: two complementary models of organisational Agile implementation', Curtin University.

Dearle, A 2007, 'Software Deployment, Past, Present and Future', *IEEE: Computer Society*, pp. 1 -16.

Denning, S 2015, *Agile: The World's Most Popular Innovation Engine*, Forbes Leadership, viewed 12/10/2015, <http://www.forbes.com/sites/stevedenning/2015/07/23/the-worlds-most-popular-innovation-engine/>.

Denzin, N, K. & Lincoln, Y, S 2008, ' Introduction: The discipline and practice of qualitative research.', *The SAGE handbook of qualitative research* 3rd edn, Sage, Thousand Oaks, CA, pp. 1-43.

Despa, M, L. 2014, 'Comparative study on software development methodologies', *Database Systems Journal,* vol. 5, no. 3, pp. 37 - 56.

Devroe, R 2016, 'How to enhance the external validity of survey experiments? A discussion on the basis of a research design on political gender stereotypes in Flanders', *ECPR General Conference*, pp. 1-24.

Dimitrios, T & Antigoni, F 2019, ' Limitations And Delimitations In The Research Process. ', *Perioperative nursing,* vol. 7, no. 3, pp. 155-62.

du Plessis, A, J 2016, 'The contribution of policies, procedures and rules for successful suggestion systems in organisations: some research findings', *Jurnalul Practicilor Comunitare Pozitive*, no. 1, pp. 92-106.

Duka, D 2013, 'Adoption of agile methodology in software development', *2013 36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 426-30.

Dybå, T, Prikladnicki, R, Rönkkö, K, Seaman, C & Sillito, J 2011, 'Qualitative Research in Software Engineering', *Springer Science: Emprical Software Engineering,* vol. 16, pp. 425 - 29.

Ebert, C, Abrahamsson, P & Oza, N 2012, 'Lean software development', *IEEE Software*, no. 5, pp. 22-25.

Edeki, C 2013, 'Agile Unified Process', *International Journal of Computer Science and Mobile Applications,* vol. 1, no. 3, pp. 13 - 17.

Edeki, C 2015, 'AGILE SOFTWARE DEVELOPMENT METHODOLOGY', *European Journal of Research in Medical Sciences,* vol. 3, no. 1, pp. 22 - 27.

Eisenhardt, K, M 1989, 'Building theories from case study research', *Academy of Management Review,* vol. 14, no. 4, pp. 532–50.

Elo, S & Kyngäs, H 2008, 'The qualitative content analysis process', *Journal of advanced nursing,* vol. 62, no. 1, pp. 107-15.

Emmanuel, O, Onyebinama, I, Chinyere, A, Emeka, E & Chigoze, I 2020, 'INDUSTRIAL REVOLUTION OF AFRICA'S AGRICULTURAL SECTOR A PARADIGM SHIFT TRANSFORMATION', *GPH-International Journal of Agriculture and Research→,* vol. 3, no. 05, pp. 01-10.

Etikan, I, Musa, S, A. & Alkassim, R, S 2016, 'Comparison of convenience sampling and purposive sampling', *American journal of theoretical and applied statistics,* vol. 5, no. 1, pp. 1-4.

Evans, C, Johnson, B, Hackney, R, Rauniar, R, Rawski, G & Yang, J 2014, 'Technology acceptance model (TAM) and social media usage: an empirical study on Facebook', *Journal of Enterprise Information Management,* vol. 27, no. 1, pp. 6-30.

Final Report May 2009, *Consultancy on the Development of the Science and Technology Human Resource Strategy*, Deloitte, Botswana.

Fishbein, M & Ajzen, I 1975, *Belief, attitude, intention, and behavior: An introduction to theory and research. Reading, Mass*, Addison-Wesley Pub. Co., Ontario.

Flora, HK & Chande, SV 2014, 'A systematic study on agile software development methodologies and practices', *International Journal of Computer Science and Information Technologies,* vol. 5, no. 3, pp. 3626-37.

Forman, J & Damschroder, L 2007, 'Qualitative content analysis', *Empirical methods for bioethics: A primer*, Emerald Group Publishing Limited, pp. 39-62.

Forrester Research 2011, 'Agile Software Development And The Factors That Drive Success', *Hewlett-Packard*.

Fowler, M 2018, *Refactoring: improving the design of existing code*, Addison-Wesley Professional,

Gable, G, G 1994, 'Integrating case study and survey research methods: an example in information systems. ', *European Journal of Information Systems,* vol. 3, no. 2, pp. 112-26.

Gagel, G 2017, 'The Intersection of Organizational Agility and Transformational Leadership: A Literature Review. ', *Academy of Management Proceedings,* vol. 1.

Gandomani, T, J. & Zulzalil, H 2013, 'Compatibility of agile software development methods and CMMI', *Indian Journal of Science and Technology,* vol. 6, no. 8, pp. 5089-94.

Gandomani, T, J., Zulzalil, H, Ghani, A, A, A,, Sultan, A, B, M, & Nafchi, M, Z 2013, 'Obstacles in moving to agile software development methods; at a glance', *Journal of Computer Science,* vol. 9, no. 5, pp. 620-25.

Geambaşu, C, V., Jianu, I, Jianu , I & Gavrilă, A 2011, 'Influence Factors for the Choice of a Software Development Methodology', *Accounting and Management Information Systems,* vol. 4, no. 10, pp. 479–94.

Georgiou, S, Rizou, S & Spinellis, D 2019, 'Software Development Lifecycle for Energy Efficiency: Techniques and Tools', *ACM Computing Surveys (CSUR),* vol. 52, no. 4, pp. 1-33.

GESCI 2017, *Assessment of Knowledge Society Development in Botswana*, Global e-Schools and Communities Initaitive(GESCI) Nairobi, Kenya.

Gill, P, Stewart, K, Treasure, E & Chadwick, B 2008, 'Methods of data collection in qualitative research: interviews and focus groups', *British dental journal,* vol. 204, no. 6, pp. 291-95.

Given, M, L 2008, *The SAGE Encyclopedia of QUALITATIVE RESEARCH METHODS*, SAGE Publications, Inc, Thousand Oaks, California.

Glass, R, Ramesh, V & Vessey, I 1994, 'An analysis of research in Computing disciplines', *Communications of the ACM,* vol. 47, no. 6, pp. 89-94.

Golfarelli, M, Rizzi, S & Turricchia, E 2011, *Modern Software Engineering Methodologies Meet Data Warehouse Design: 4WD,*

Guangzhong, L 2020, 'Application of IoT and Countermeasure in Agriculture of Shandong Province, China', *Computing,* vol. 8, no. 1, pp. 8-11.

Guetterman, T, C., Fetters, M, D. & Creswell, J, W. 2015, 'Integrating quantitative and qualitative results in health science mixed methods research through joint displays', *The Annals of Family Medicine,* vol. 13, no. 6, pp. 554-61.

Gupta, D, Sharma, S & Sarkar, D 2012, 'Agile Processes and Methodologies: A Conceptual Study', *International Journal on Computer Science and Engineering,* vol. 4, no. 5, pp. 892 - 98.

Gupta, M, George, JF & Xia, W 2019, 'Relationships between IT department culture and agile software development practices: An empirical investigation', *International Journal of Information Management,* vol. 44, pp. 13-24.

Habib, M 2013, *Agile software development methodologies and how to apply them*, Code project, viewed 01/08/2019, <https://www.codeproject.com/Articles/604417/Agile-software-development-methodologies>.

Haiderzai, MD & Khattab, MI 2019, 'How software testing impact the quality of software systems?', *IJECS,* vol. 1, no. 2, pp. 05-09.

Hajjdiab, H & Taleb, A, S 2011, 'Adopting agile software development: issues and challenges', *International Journal of Managing Value and Supply Chains (IJMVSC),* vol. 2, no. 3, pp. 1-10.

Hallikainen, H, Savimäki, E & Laukkanen, T 2020, 'Fostering B2B sales with customer big data analytics', *Industrial Marketing Management,* vol. 86, pp. 90-98.

Hanslo, R & Mnkandla, E 2018, 'Scrum adoption challenges detection model: SACDM', *2018 Federated Conference on Computer Science and Information Systems (FedCSIS),* vol. 15, pp. 949-57.

Hardgrave, BC & Johnson, RA 2003, 'Toward an information systems development acceptance model: the case of object-oriented systems development', *IEEE Transactions on Engineering Management,* vol. 50, no. 3, pp. 322-36.

Hays, K, T 2007, 'A qualitative cross-case analysis of postsecondary students' performance in asynchronous mechanical system laboratories', Texas A&M University.

Hazzan, O & Dubinsky, Y 2008, *Qualitative Research in Software Engineering*, Technion, Department of Education in Technology and Science, Haifa 32000, Israel. <http://www.researchgate.net/publication/237780988_Orit_Hazzan%27s_Column_Qualitative_Research_in_Software_Engineering>.

Helmo, P 2019, *WEF Africa: Botswana tells the world it wants to reduce dependence on diamonds*, Independent Online and affiliated companies, Botswana, Gaborone, viewed 29th November, <https://www.iol.co.za/business-report/economy/wef-africa-botswana-tells-the-world-it-wants-to-reduce-dependence-on-diamonds-31998283>.

Highsmith, J & Cockburn, A 2001, 'Agile software development: The business of innovation', *Computer,* vol. 34, no. 9, pp. 120-27.

Highsmith, J 2002, 'What is agile software development?', *crosstalk,* vol. 15, no. 10, pp. 4-10.

Hijazi, H, Alqrainy, S, Muaidi, H & Khdour, T 2014, 'Risk factors in software development phases', *European Scientific Journal,* vol. 10, no. 3, pp. 213-32.

Hildayanti, A & Machrizzandi, MSR 2020, 'THE APPLICATION OF IOT (INTERNET OF THINGS) FOR SMART HOUSING ENVIRONMENTS AND INTEGRATED ECOSYSTEMS', *Nature: National Academic Journal of Architecture,* vol. 7, no. 1, pp. 80-88.

Hobbs, B & Petit, Y 2017, 'Agile methods on large projects in large organizations', *Project Management Journal,* vol. 48, no. 3, pp. 3-19.

Hornbæk, K & Hertzum, M 2017, 'Technology acceptance and user experience: A review of the experiential component in HCI', *ACM Transactions on Computer-Human Interaction (TOCHI),* vol. 24, no. 5, pp. 1-30.

Hungund, S & Kiran, K, B 2017, 'Open innovation practices among Indian software product firms: a pilot study', *International Journal of Innovation and Sustainable Development,* vol. 11, no. 4, pp. 355-76.

IAS 2019, *Governance & Policy*, Illawarra Academy of Sport, viewed 13-08-2019, <http://www.ias.org.au/page/ias-policies>.

Iivari, J & Iivari, N 2011, 'The relationship between organizational culture and the deployment of agile methods', *Information and Software Technology,* vol. 53, no. 5, pp. 509-20.

International Telecommunication Union 2018, *Measuring the Information Society Report 2018 - Volume 2*, Intenational Telecommunication Union, Geneva, Switzerland. <https://www.itu.int/en/ITU-D/Statistics/Pages/publications/misr2018.aspx>.

Isaac, O, Abdullah, Z, Ramayah, T & Mutahar, AM 2017, 'Internet usage within government institutions in Yemen: An extended technology acceptance model (TAM) with internet self-efficacy and performance impact', *Science International,* vol. 29, no. 4, pp. 737-47.

Jain, R 2011, 'Improvement in software development process and software product through knowledge management', *International Journal of Computer Technology and Applications,* vol. 2, no. 05, pp. 1557-62.

Jayaswal, KB & Patton, CP 2006, *Software Development Methodology Today*.

Johnston, M, P. 2017, 'Secondary data analysis: A method of which the time has come', *Qualitative and quantitative methods in libraries,* vol. 3, no. 3, pp. 619-26.

Joo, YJ, Park, S & Lim, E 2018, 'Factors influencing preservice teachers’ intention to use technology: TPACK, teacher self-efficacy, and technology acceptance model', *Journal of Educational Technology & Society,* vol. 21, no. 3, pp. 48-59.

Kakar, A, K. 2012, 'A theory of software development methodologies', *Proceedings of the Southern Association for Information Systems Conference*, pp. 137-42.

Kaur, R & Sengupta, J 2011, 'Software Process Models and Analysis on Failure of Software Development Projects', *International Journal of Scientific & Engineering Research,* vol. 2, no. 2, pp. 1-4.

Kerretts-Makua, M 2014, *A Review of the ICT Regulatory Framework In Botswana*, ICT Sector Unit of the World Bank.

Khan, S & VanWynsberghe, R 2008, 'Cultivating the under-mined: Cross-case analysis as knowledge mobilization', *Forum qualitative Sozialforschung/forum: Qualitative social research,* vol. 9, no. 1, pp. 1-26.

Kittlaus, H, B. & Clough, P, N 2009, 'Software Products: Terms and Characteristics', *Software Product Management and Pricing. Key success factors for Software Organizations*, Springer-Verlag, pp. 5 -15.

, 2018, 'Agile manifesto and practices selection for tailoring software development: a systematic literature review', *International Conference on Product-Focused Software Process Improvement*.

Kiv , S, Samed i, H, Manuel , K & Yves , W 2018, 'Agile manifesto and practices selection for tailoring software development: a systematic literature review', *International Conference on Product-Focused Software Process Improvement*, Springer, pp. 12-30.

Kramer, M 2018, 'Best practices in systems development lifecycle: An analyses based on the waterfall model', *Review of Business & Finance Studies,* vol. 9, no. 1, pp. 77-84.

Krishnan, M, S 2015, 'Software Development Risk Aspects and Success Frequency on Spiral and Agile Mode', *International Journal of Innovative Research in Computer and Communication Engineering,* vol. 3, no. 1, pp. 301 - 10.

Kropp, M, Meier, A, Anslow, C & Biddle, R 2018, 'Satisfaction, practices, and influences in agile software development', *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, pp. 112-21.

Kukreja, V, Ahuja, S & Singh, A 2018, 'Measurement and structural model of agile software development critical success factors', *International Journal of Engineering & Technology,* vol. 7, no. 3, pp. 1236-42.

Kumar, G & Bhatia, PK 2014, 'Comparative analysis of software engineering models from traditional to modern methodologies', *2014 Fourth International Conference on Advanced Computing & Communication Technologies*, IEEE, pp. 189-96.

Kunda, D, Mulenga, M, Sinyinda , M & Chama, V 2018, 'Challenges of Agile Development and Implementation in a Developing Country: A Zambia Case Study. ', *Journal of Computer Science,* vol. 14, pp. 585-600.

Lai, PC 2017, 'The literature review of technology adoption models and theories for the novelty technology', *JISTEM-Journal of Information Systems and Technology Management,* vol. 14, pp. 21-38.

Lee , C, S. & Low , S, H , S. 2017, 'Developing interest to share and craft based on the Technology Acceptance Model'.

Lee, Y, Hsieh, Y & Hsu, C 2011, 'Adding Innovation Diffusion Theory to the Technology Acceptance Model: Supporting Employees' Intentions to use E-Learning Systems.', *Educational Technology & Society,* vol. 14, no. 4, pp. 124-37.

Leung, L 2015, 'Validity, reliability, and generalizability in qualitative research', *Journal of family medicine and primary care,* vol. 4, no. 3, pp. 324-27.

Li, Y, Yu, M, Xu, M, Yang, J, Sha, D, Liu, Q & Yang, C 2020, 'Big Data and Cloud Computing', *Manual of Digital Earth*, pp. 325-48.

Licorish, SA, Holvitie, J, Hyrynsalmi, S, Leppänen, V, Spínola, RO, Mendes, TS, MacDonell, SG & Buchan, J 2016, 'Adoption and suitability of software development methods and practices', *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)*, IEEE, pp. 369-72.

Lim, WM 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.

MacCormack, A & Verganti, R 2003, 'Managing the Sources of Uncertainty: Matching Process and Context in Software Development', *Journal of Product Innovation Management,* vol. 20, no. 3, pp. 217-32.

Marnewick, AL & Marnewick, C 2019, 'The Ability of Project Managers to Implement Industry 4.0-Related Projects', *IEEE Access,* vol. 8, pp. 314-24.

Mbelli, M, T. & Hira, J, J. 2016, 'The Perceptions of Agile Methodology in South Africa', *Computer Science & Information Technology,* vol. 6, pp. 219-27.

McGuiggan , R, L., Lee , G, Spanjaard , D, Denize , S, M. & Sharma , N 2008, 'Cross-case analysis: An alternative methodology', *Marketing: Shifting the Focus from Mainstream to Offbeat: Proceedings of the Australian and New Zealand Marketing Academy Conference, held 1-3 December 2008, Olympic Park, Sydney, NSW*.

Mezhuyev, V, Al-Emran, M, Fatehah, M & Hong, NC 2018, 'Factors affecting the metamodelling acceptance: a case study from software development companies in Malaysia', *IEEE Access,* vol. 6, pp. 49476-85.

Mezhuyev, V, Al-Emran, M, Ismail, MA, Benedicenti, L & Chandran, DA 2019, 'The acceptance of search-based software engineering techniques: An empirical evaluation using the technology acceptance model', *IEEE Access,* vol. 7, pp. 101073-85.

Middlemass, JB, Vos, J & Siriwardena, AN 2017, 'Perceptions on use of home telemonitoring in patients with long term conditions–concordance with the Health Information Technology Acceptance Model: a qualitative collective case study', *BMC medical informatics and decision making,* vol. 17, no. 1, pp. 1-13.

Mishra, A & Dubey, D 2013, 'A comparative study of different software development life cycle models in different scenarios', *International Journal of Advance research in computer science and management studies,* vol. 1, no. 5.

Mitsuhashi, A 2013, *Data collection system, data collection method and data collection program to reduce the communication time*, Patent.

MmegiOnline 2021, *Botswana improves in global ICT ranking* updated Friday 08 January 2021, Dikgang Publishing Company Gaborone, Botswana, viewed 10th January, <https://www.mmegi.bw/index.php?sid=4&aid=212&dir=2012/April/Friday13>.

Mnkandla, E 2008, 'A Selection Framework For Agile Methodology Practices: A Family Of Methodologies Approach', Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, South Africa.

Moalosi, R, Sealetsa, O, Molwane, O, Letsholo, P, MOLOKWANE, S, Mwendapole, C, Letsatsi, M, Moalos, R, Sealetsa, O & Molwane, O 2016, 'Assessment of Design-driven Innovation within Botswana's Small Creative Industries', *International jounal of Cultural and Creative Industries,* vol. 3, no. 2, pp. 38-51.

Mogalakwe, M & Nyamnjoh, F 2017, *Botswana at 50: democratic deficit, elite corruption and poverty in the midst of plenty*, Taylor & Francis.

Mohajan , H, K. 2017, 'Two criteria for good measurements in research: Validity and reliability', *Annals of Spiru Haret University. Economic Series,* vol. 17, no. 4, pp. 59-82.

Mokgoabone, K 2004, *Seretse takes over as BTC head*, Mmegi Online, Gaborone, Botswana, viewed 29 December <http://www.mmegi.bw/2004/April/Tuesday20/9437896191056.html>.

Monga, C, Shimeles, A & Woldemichael, A 2019, 'Creating Decent Jobs: Strategies, Policies and Instruments', *Abidjan (Côte d'Ivoire): African Development Bank*.

Moretti, F, van Vliet, L, Bensing, J, Deledda, G, Mazzi, M, Rimondini, M, Zimmermann, C & Fletcher, I 2011, 'A standardized approach to qualitative content analysis of focus group discussions from different countries', *Patient education and counseling,* vol. 82, no. 3, pp. 420-28.

Morteza, P, Motameni, H, Alinejad, H, Pourshaikh, R & Pourshaikh, R 2011, 'Improving Web Engineering and Agile ICONIX Process', *Middle-East Journal of Scientific Research,* vol. 8, no. 1, pp. 274-81.

Mphale , O, Okike , U, E. & Mogotlhwane , T 2016, 'An Assessment of ICT Project Success/Failure in Botswana', *IJCSN International Journal of Computer Science and Network,* vol. 5, no. 6, pp. 966 -77.

Munassar, N, M,A. & Govardhan, A 2010, 'A Comparison Between Five Models Of Software Engineering', *IJCSI International Journal of Computer Science Issues,,* vol. 7, no. 5, pp. 94 - 101.

Myers, M 2009, *Qualitative Research in Business Management*, Sage, London.

Naderuzzaman, M, Rabbi, F & Beg, A 2011, 'An Improved & Adaptive Software Development Methodology', *Computer Engineering and Intelligent Systems,* vol. 2, no. 3, pp. 35 - 41.

Nadri, H, Rahimi, B, Afshar, HL, Samadbeik, M & Garavand, A 2018, 'Factors affecting acceptance of hospital information systems based on extended technology acceptance model: a case study in three paraclinical departments', *Applied clinical informatics,* vol. 9, no. 2, p. 238.

Navimipour, NJ & Charband, Y 2016, 'Knowledge sharing mechanisms and techniques in project teams: Literature review, classification, and current trends', *Computers in Human Behavior,* vol. 62, pp. 730-42.

Ng, E, M , W., Shroff , R & Lim , C 2013, 'Applying a modified technology acceptance model to qualitatively analyse the factors affecting e-portfolio implementation for student teachers’ in field experience placements', *Proceedings of the Informing Science and Information Technology Education Conference*, Informing Science Institute, pp. 355-65.

Njikam, M, Nanna, S, Shahrin, S & Othman, MFI 2019, 'High speed internet development in Africa using 4G-LTE technology-a review', *Bulletin of Electrical Engineering and Informatics,* vol. 8, no. 2, pp. 577-85.

Nkone, MS 2013, 'The relationship between systems development methodologies and Information Technology project success'.

Nyamaka , A, Botha, A, Van Biljon, J & Marais, M 2018, 'Challenges Botswana’s Mobile Application Developers Encounter: Funding, Commercial and Technical Support', *2018 IST-Africa Week Conference (IST-Africa)*, pp. 1-10.

Oke, A & Fernandes, FAP 2020, 'Innovations in Teaching and Learning: Exploring the Perceptions of the Education Sector on the 4th Industrial Revolution (4IR)', *Journal of Open Innovation: Technology, Market, and Complexity,* vol. 6, no. 2, p. 31.

Okesola , O, J., Adebiyi , A, A., Owoade , A, A., Adeaga , O, Adeyemi , O & Odun-Ayo , I 2020, 'Software Requirement in Iterative SDLC Model', *Computer Science On-line Conference*, Springer, pp. 26-34.

Okesola, OJ, Adebiyi, AA, Owoade, AA, Adeaga, O, Adeyemi, O & Odun-Ayo, I 2020, 'Software Requirement in Iterative SDLC Model', *Computer Science On-line Conference*, Springer, pp. 26-34.

Olszewska, JI & Allison, I 2018, 'ODYSSEY: Software development life cycle ontology'.

Park, Y 2009, 'An Analysis of the Technology Acceptance Model in Understanding University Students’ Behavioral Intention to Use e-Learning', *Educational Technology & Society,* vol. 12, no. 3, pp. 150–62.

Parkman, S, Litz, D & Gromik, N 2018, 'Examining pre-service teachers’ acceptance of technology-rich learning environments: A UAE case study', *Education and Information Technologies,* vol. 23, no. 3, pp. 1253-75.

Parveen, A, Khan, A, M. & Sadiq, M 2014, 'A Method for the Selection of Software Development Life Cycle Models using Analytic Hierarchy Process', *IEEE:2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, pp. 534 - 40.

Penn, DM 2016, 'Agile and conventional methodologies: an empirical investigation of their impact on software quality parameters'.

Pressman, R 2010, *Software Ebgineering : A Practitioner's Approach*, Seventh Edtion edn, MacGraw - Hill, NY, America.

Purnama, IWJW & Ginardi, RVH 2019, 'Analysis of Application Based on Cloud Computing in Banking Industries in Indonesia Using Technology Acceptance Model (TAM) 2 Method Case Study The National Private Banks in Surabaya and Bali Region', *IPTEK Journal of Proceedings Series*, no. 5, pp. 519-26.

Putra , B, A., Agussalim , M, S., Lusiarini , Y, Rizky , M, I. & Bestari , P, Y. 2020, 'Design and Development Executive Information System Application with Drilldown and What-If Analysis features', *International Conference on Science and Technology 2019*, IOP Publishing, p. 7.

Putra, IDGRD 2019, 'The evolution of technology acceptance model (TAM) and recent progress on technology acceptance research in ELT: State of the art article', *Yavana Bhasha: Journal of English Language Education,* vol. 1, no. 2.

Ramadubu, D 2021, 'Online Registration Crisis Continues ', *Botswana Guardian*,

Rao, K, N., Naidu, G, K & Chakka, P 2011, 'A Study of the Agile Software Development Methods, Applicability and Implications in Industry', *International Journal of Software Engineering and Its Applications,* vol. 5, no. 2, pp. 35 - 46.

Rapanyane, M & Sethole, F 2020, 'The rise of artificial intelligence and robots in the 4th Industrial Revolution: implications for future South African job creation', *Contemporary Social Science,* vol. 15, no. 4, pp. 489-501.

Riaz, MN, Mahboob, A & Buriro, A 2018, 'Social Success Factors Affecting Implementation of Agile Software Development Methodologies in Software Industry of Pakistan: An Empirical Study', *International Journal of Advanced Computer Science and Applications,* vol. 9, pp. 94-98.

Rigopoulos, G & Askounis, D 2007, 'A TAM Framework to Evaluate Users' Perception towards Online Electronic Payments ', *Journal of Internet Banking and Commerce,* vol. 12, no. 3, pp. 1-6.

Ritchie , J & Lewis , J 2003, *Qualitative research practice: Qualitative Research Practice*, SAGE Publications, London.

Rogers, EM 1995, *Diffusion of Innovations. 4th ed., ,* New York: The Free Press,

Rönnby, S, Lundberg, O, Fagher, K, Jacobsson, J, Tillander, B, Gauffin, H, Hansson, P-O, Dahlström, Ö & Timpka, T 2018, 'mHealth self-report monitoring in competitive middle- and long-distance runners: qualitative study of long-term use intentions using the technology acceptance model', *JMIR mHealth and uHealth,* vol. 6, no. 8, p. e10270.

L Smith (ed), 1993, 'Implications for teaching of a process-based research framework for information systems ', *Proceedings of the International Academy for Information Management Conference*, Orlando, Florida.

Rosenberg, D 2001, *Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example*, Addison Wesley, Canada.

Roses, LK, Windmöller, A & Carmo, EAd 2016, 'Favorability conditions in the adoption of agile method practices for Software development in a public banking', *JISTEM-Journal of Information Systems and Technology Management,* vol. 13, no. 3, pp. 439-58.

Ruparelia, N, B. 2010, 'Software development lifecycle models', *ACM SIGSOFT Software Engineering Notes,* vol. 35, no. 3, pp. 8-13.

SADC 2020, *Selection Of Individual Consultants: For Development Of Southern African Development Community (SADC) Fourth Industrial Revolution Strategy*, SADC, Botswana, Gaborone. <https://www.sadc.int/files/7215/9462/6781/RFP_-_Fourth_Industrial_Revolution_20200713-1.pdf>.

Salas, E, Rosen, MA & DiazGranados, D 2010, 'Expertise-based intuition and decision making in organizations', *Journal of management,* vol. 36, no. 4, pp. 941-73.

Sani, A, Firdaus, A, Jeong, S, R. & Ghani, I 2013, 'A review on software development security engineering using dynamic system method (DSDM)', *International Journal of Computer Applications,* vol. 69, no. 25, pp. 37-44.

Scheller, RM, Sturtevant, BR, Gustafson, EJ, Ward, BC & Mladenoff, DJ 2010, 'Increasing the reliability of ecological models using modern software engineering techniques', *Frontiers in Ecology and the Environment,* vol. 8, no. 5, pp. 253-60.

Schmidt, K, Gummer, T & Roßmann, J 2020, 'Effects of respondent and survey characteristics on the response quality of an open-ended attitude question in web surveys', *methods, data, analyses,* vol. 14, no. 1, p. 32.

Schwab, K 2015, 'The Fourth Industrial Revolution: What it Means and how to Respond', *Council of Foreign Relations*, <https://www.foreignaffairs.com/articles/2015-12-12/fourth-industrial-revolution>.

Schwab, K 2016, 'The Fourth Industrial Revolution (Geneva: World Economic Forum)', *EKONOMIKA PREDUZEĆA EKONOMIKA EKONOMIKA P EKONOMIKA PR EKONOMIKA PRE EKONOMIKA PRED EKONOMIKA PREDUZ EKONOMIKA PREDUZE KONOMIKA PREDUZEĆA NOMIKA PREDUZEĆA OMIKA PREDUZEĆA MIKA PREDUZEĆA IKA PREDUZEĆA A PREDUZEĆA PREDUZEĆA*.

Setimela, M, K. 2018, 'The use of ICT in the greater Gaborone area', Stellenbosch: Stellenbosch University.

Setimela, MK 2018, 'The use of ICT in the greater Gaborone area', Stellenbosch: Stellenbosch University.

Sharma, S, Sarkar, D & Gupta, D 2012, 'Agile processes and methodologies: A conceptual study', *International Journal on Computer Science and Engineering,* vol. 4, no. 5, pp. 892-98.

Shea, P, Pickett, A & Li, CS 2005, 'Increasing access to higher education: A study of the diffusion of online teaching among 913 college faculty', *The International review of research in open and distributed learning,* vol. 6, no. 2.

Siddiqui, F 2019, 'Impact of Agile Model Driven-Development (AMDD) on Test Driven-Development (TDD): An Exploratory Study'.

Silva, C, C. & Goldman, A 2014, 'Agile methods adoption on software development: a pilot review', *2014 Agile Conference*, pp. 64-65.

Simon, M & Goes, J 2011, *Reliability and validity in qualitative studies*, Dissertation and scholarly Research, Seattle, W.A.

Singh, B & Gautam, S 2016, 'The impact of software development process on software quality: A review', *2016 8th International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 666-72.

Siyam, N 2019, 'Factors impacting special education teachers' acceptance and actual use of technology', *Education and Information Technologies,* vol. 24, no. 3, pp. 2035-57.

Song, S 2019, *Afrcian undersea cables*, viewed 15/01/2021, <https://manypossibilities.net/african-undersea-cables/>.

Song , S 2019, *Afrcian undersea cables*, viewed 15/01/2021, <https://manypossibilities.net/african-undersea-cables/>.

Song, Y & Kong, S 2017, 'Investigating students' acceptance of a statistics learning platform using technology acceptance model', *Journal of Educational Computing Research,* vol. 55, no. 6, pp. 865-97.

Sorooshian, S & Panigrahi, S 2020, 'Impacts of the 4th Industrial Revolution on Industries', *Walailak Journal of Science and Technology (WJST),* vol. 17, no. 8, pp. 903-15.

Souza, S, D, S, D. & Silva, D, M, G,V, D 2011, 'Validation of a theoretical model: knowing the interactive processes within the support network for people with tuberculosis', *Acta Paulista de Enfermagem,* vol. 24, no. 6, pp. 778-83.

Squires, J, E., Moralejo, D & LeFort, S, M. 2007, 'Exploring the role of organizational policies and procedures in promoting research utilization in registered nurses', *Implementation Science,* vol. 2, no. 1, p. 17.

Standish Group 2015, *2015 Chaos Report*, <http://www.Infoq.com/articles/standish-chaos-2015>.

Stanek, TB 2017, 'On David Hume', *Wisdom in Education,* vol. 7, no. 1, p. 3.

Stoica, M, Mircea, M & Ghilic-Micu, B 2013, 'Software Development: Agile vs. Traditional', *Informatica Economics,* vol. 17, no. 4, pp. 64 -76.

Subramanian, G, H., Klein, G, Jiang, J, J. & Chan, C, L 2009, 'Balancing Four Factors in System Development Projects', *ACM,* vol. 52, no. 10, pp. 118-21.

Surya, P 2018, *Survey Data Shows That Many Companies Are Still Not Truly Agile* Harward Business Publishing, viewed 13/04/2021, <https://hbr.org/sponsored/2018/03/survey-data-shows-that-many-companies-are-still-not-truly-agile>.

Swersky, D 2018, *The SDLC: 7 phases, popular models, benefits & more [2019]*, Raygun 2018, viewed 04/August/2019, <https://raygun.com/blog/software-development-life-cycle/>.

Taylor, SaT, P. A. 1995, 'Understanding Information Technology Usage: A Test of Competing Models.', *Information Systems Research,* vol. 6, pp. 144-76.

Tegegne, W, E., Seppänen, P & Ahmad, O, M. 2019, 'Software Development Methodologies and Practices in Startups', *IET Software*, pp. 1-16.

Tiky, W, T,Y. 2016, *Software Development Life Cycle*, The Hongkong University of Science and Technology, Hongkong.

Tolfo, C, Wazlawick, RS, Ferreira, MGG & Forcellini, FA 2011, 'Agile methods and organizational culture: Reflections about cultural levels', *Journal of Software Maintenance and Evolution: Research and Practice,* vol. 23, no. 6, pp. 423-41.

Totten, J 2017, 'Critical Success Factors for Agile Project Management in Non-Software Related Product Development Teams'.

Tran, T, M & Duong, N, D 2014, 'Adoption of agile software development in Vietnam', pp. 1-102.

Tsoy, M & Staples, D, Sandy. 2020, 'Exploring Critical Success Factors in Agile Analytics Projects', *Proceedings of the 53rd Hawaii International Conference on System Sciences*.

Ulmer, LW, Watson, LW & Derby, D 2007, 'Perceptions of higher education faculty members on the value of distance education', *Quarterly Review of Distance Education,* vol. 8, no. 1.

Umarji, M & Seaman, C 2005, 'Predicting acceptance of software process improvement', *ACM SIGSOFT Softw. Eng. Notes,* vol. 30, no. 4, pp. 1-6.

Usman, AV & Ogwueleka, FN 2018, 'SDLC Models as Tools in the Development of MIS: A Study', *IUP Journal of Information Technology,* vol. 14, no. 4, pp. 52-59.

Vanker, C 2015, 'The adoption of Agile Software development methodologies by organisations in South Africa', University of Kwazulu-Natal.

Vasantrao, K, V 2011, 'Understanding need of "Uncertainty Analysis" in the System Design Process', *International Journal of Software Engineering & Applications (IJSEA),* vol. 2, no. 3, pp. 95 - 104.

Vaziri, DD, Aal, K, Ogonowski, C, Von Rekowski, T, Kroll, M, Marston, HR, Poveda, R, Gschwind, YJ, Delbaere, K & Wieching, R 2016, 'Exploring user experience and technology acceptance for a fall prevention system: results from a randomized clinical trial and a living lab', *European review of aging and physical activity,* vol. 13, no. 1, p. 6.

Venkatesh, V & Davis, F, D. 2000, 'A theoretical extension of the technology acceptance model: Four longitudinal field studies', *Management Science,* vol. 46, no. 2, pp. 186-204.

Venkatesh , V & Davis, F, D. 2000, 'A theoretical extension of the technology acceptance model: Four longitudinal field studies', *Management Science,* vol. 46, no. 2, pp. 186-204.

Venkatesh, V & Bala, H 2008, 'Technology Acceptance Model 3 and a Research Agenda on Interventions.', *Decision Science.,* vol. 39, no. 2, pp. 273-312.

Venkatesh , V, Thong , J, Y. & Xu , X 2016, 'Unified theory of acceptance and use of technology: A synthesis and the road ahead', *Journal of the Association for Information Systems,* vol. 17, no. 5, pp. 328-76.

Version One 2013, *7th Annual State of Agile VERSION ONE Agile Made Easier Development Survey*, VERSION ONE, viewed 14 October, <https://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf>.

VersionOne 2013, *State of Agile Development Survey Results 2013*, VersionOne Inc, viewed 2nd July 2014, <http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf>.

VersionOne 2015, *10th Annual State of Agile Report*, USA, viewed 12th December, <https://stateofagile.com/#ufh-i-613554198-10th-annual-state-of-agile-report/7027494>.

, 2018, 'A green routing algorithm for IoT-enabled software defined wireless sensor network', *IEEE Sensors Journal*.

Vijayasarathy, LR & Butler, CW 2015, 'Choice of software development methodologies: Do organizational, project, and team characteristics matter?', *IEEE Software,* vol*.* 33, no. 5, pp. 86-94.

Vogelsang, K, Steinhüser, M & Hoppe, U 2013, 'A qualitative approach to examine technology acceptance'.

Vohra, P & Singh, A 2013, 'A Contrast and Comparison of Modern Software Process Models', *International Conference on Advances in Management and Technology*, pp. 23 - 27.

Wallace, LG & Sheetz, SD 2014, 'The adoption of software measures: A technology acceptance model (TAM) perspective', *Information & Management,* vol*.* 51, no. 2, pp. 249-59.

Walliman, N 2017, *Research methods: The basics*, II edn, Routledge, London,United Kingdom.

Wang, Y & Patel, D 2000, 'Editors' introduction: comparative software engineering: review and perspectives', *Annals of Software Engineering,* vol*.* 10, pp. 1–10.

Weigers, E 2005, *Why is Software Process Improvement So Hard: Process Imapct*, https://www.processimpact.com/articles/spi_so_hard.html.

Weng, F, Yang, R-J, Ho, H-J & Su, H-M 2018, 'A TAM-based study of the attitude towards use intention of multimedia among school teachers', *Applied System Innovation,* vol*.* 1, no. 3, p. 36.

Wingo, NP, Ivankova, NV & Moss, JA 2017, 'Faculty perceptions about teaching online: Exploring the literature using the technology acceptance model as an organizing framework', *Online Learning,* vol*.* 21, no. 1, pp. 15-35.

Wixom, B & Todd, P 2005, 'A theoretical integration of user satisfaction and technology acceptance', *Information Systems Research*, pp. 85-102.

Woods, D 2010, *Why Lean And Agile Go Together*, Forbes, viewed 12/10/2015, <http://www.forbes.com/2010/01/11/software-lean-manufacturing-technology-cio-network-agile.html>.

Wu, B & Chen, X 2017, 'Continuance intention to use MOOCs: Integrating the technology acceptance model (TAM) and task technology fit (TTF) model', *Computers in Human Behavior,* vol. 67, pp. 221-32.

Yin, R 2003, *A Review of Case Study Research: Design and Methods*, vol.5, Fourth Edition,p.219.

Yin , R 2003, 'A Review of Case Study Research: Design and Methods', vol. 5, p. 219.

Yin, R, K 2009, *Case Study Research*, SAGE Publications, London.

Yucel, UA & Gulbahar, Y 2013, 'Technology acceptance model: A review of the prior predictors', *Egitim Bilimleri Fakultesi Dergisi,* vol. 46, no. 1, p. 89.

Zafar, I, Nazir, AK & Abbas, M 2017, 'The impact of agile methodology (DSDM) on software project management', *Circulation in Computer Science: International Conference on Engineering, Computing and Information Technology (ICECIT 2017)*, pp. 1-6.

Zainab , B, Awais , B, M. & Alshagawi , M 2017, 'Factors affecting e-training adoption: an examination of perceived cost, computer self-efficacy and the technology acceptance model', *Behaviour & Information Technology,* vol. 36, no. 12, pp. 1261-73.

Zambonelli, F 2017, 'Key abstractions for IoT-oriented software engineering', *IEEE Software,* vol. 34, no. 1, pp. 38-45.

APPENDIX A: NODES\\INTERVIEW QUESTIONS ORGANISATIONS

Table 8: Demographic profile for Organisation Employees

EMPLOYEES	AGE	DURATION	GENDER	HIGHEST EDUCATION	POSITION
Cases\\EMPLOYEES CASES\\Case study employee 1	37	Over 10 Years	Male	Degree	Executive
Cases\\EMPLOYEES CASES\\Case study employee 2	50	Over 10 Years	Male	Degree	Executive
Cases\\EMPLOYEES CASES\\Case study employee 3	35	Over 10 Years	Male	Degree	Project Manager
Cases\\EMPLOYEES CASES\\Case study employee 4	32	5-10 Years	Male	Degree	Analyst
Cases\\EMPLOYEES CASES\\Case study employee 5	39	Over 10 Years	Male	Degree	Other
Cases\\EMPLOYEES CASES\\Case study employee 6	37	Over 10 Years	Male	Degree	Project Manager
Cases\\EMPLOYEES CASES\\Case study employee 7	38	Over 10 Years	Male	Degree	Executive
Cases\\EMPLOYEES CASES\\Case study employee 8	30	Over 10 Years	Male	Degree	Programme r

Cases\\EMPLOYEES CASES\\Case study employee 9	unassigned	0-5 Years	Male	Degree	Analyst

Table 9: Nodes\\INTERVIEW QUESTIONS ORGANISATIONS

Name	Files	References
INTERVIEW QUESTIONS ORGANISATIONS	9	169
CHALLENGES OF ADOPTING AGILE SDM	9	16
During software development when the organization engages software clients and other stakeholders in the software project (developers, managers, sponsors, testers, advisers.) what is the organization's view on using SDM technologies as mentioned below, to	9	9
Elaborate on how useful are your organization policies and procedures when adopting and utilizing latest IT technologies like Agile SDM for software development in your organization.	9	9
How well supported is Agile SDM and Traditional SDM in developing quality software first from the organization point of view and form the technical infrastructural support rendered by your organization.	2	2
How well supported is Agile SDM and Traditional SDM in developing quality software first from the organization point of view and form the technical infrastructural support rendered by your organization~	7	7
LEVELS OF KNOWLEDGE MANAGEMENT AND SUPPORT	9	25
HIGH	9	16

Name	Files	References
LOW	0	0
MODERATE	6	7
NONE	1	2
PERCEPTIONS OF BSD	9	35
MIXED	7	9
NEGATIVE	3	3
NEUTRAL	2	2
POSITIVE	9	21
Question for those using Traditional SDM as well and only~ Considering that you organization is still using Traditional SDM what challenges if any do you see as inhibiting adoption and migration to Agile SDM and how do you deal with modern challenges of c	9	9
Question for those using Traditional SDM as well and only~ What is your reaction and response to software development organizations which have adopted and utilized Agile SDM relative to your organization in terms of job relatedness, results and progress d	9	9
SDM ADOPTED AND IMPLEMENTED	9	12
AD-HOC	1	1
AGILE	4	4

Name	Files	References
COMPARISON OF SDMs	4	4
TRADITIONAL	3	3
What approaches with regards to communication, training, staff development and SDM knowledge management do you do to ensure your organization has well qualified and versatile staff in latest SDMs~	9	9
What do you think could be challenges in adopting Agile SDMs in context of developing software in Botswana~	9	9
What is your opinion and comment regarding market technological leadership status of your organization with regards to utilization, adoption and implementation of Agile SDM and Traditional SDM and the developed software marketability~	9	9
What is your perception about Agile SDM as an SDM which can deliver a service quality to clientele in terms of user involvement, through user interaction, user profound understanding of what is needed~	9	9

Table 10: Case Classification for Employers

ORGANIZATION	DURATION DEVELOPING	MOST PREFERRED SOFTWARE CATEGORY	NUMBER OF SOFTWARE PRODUCTS DEVELOPED	POSITION
Cases\\ORGANISATION CASES\\Case study org 1	5-10 YEARS	TRADITIONAL	11-50	CONSULTANT
Cases\\ORGANISATION CASES\\Case study org 2	5-10 YEARS	AGILE	11-50	CEO
Cases\\ORGANISATION CASES\\Case study org 3	5-10 YEARS	AGILE	100+	CONSULTANT
Cases\\ORGANISATION CASES\\Case study org 4	5-10 YEARS	AGILE	11-50	CONSULTANT
Cases\\ORGANISATION CASES\\Case study org 5	5-10 YEARS	AD-HOC	11-50	CONSULTANT
Cases\\ORGANISATION CASES\\Case study org 6	5-10 YEARS	TRADITIONAL	11-50	CONSULTANT
Cases\\ORGANISATION CASES\\Case study org 7	5-10 YEARS	TRADITIONAL	11-50	CONSULTANT
Cases\\ORGANISATION CASES\\Case study org 8	5-10 YEARS	AGILE	100+	DIRECTOR
Cases\\ORGANISATION CASES\\Case study org 9	5-10 YEARS	TRADITIONAL	11-50	CONSULTANT

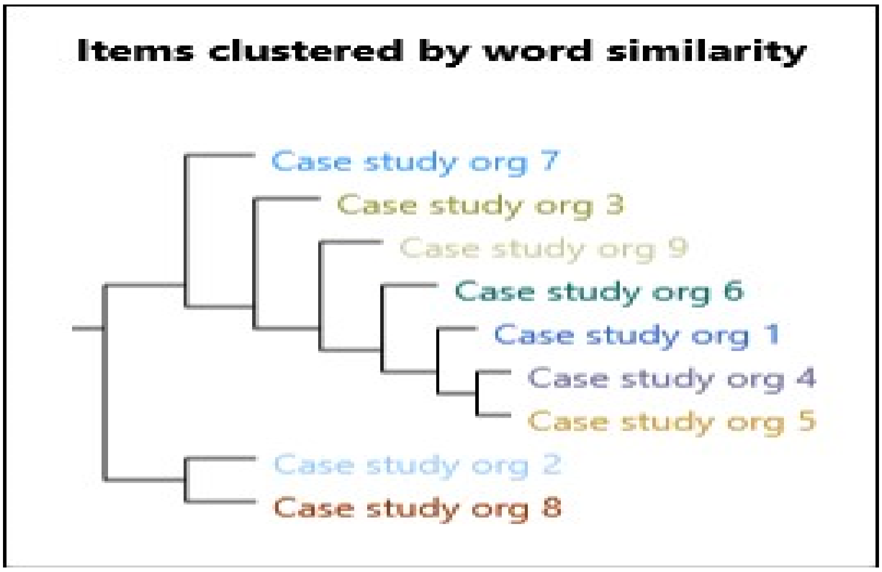


Figure 34:Word Cluster for Employee Interviews

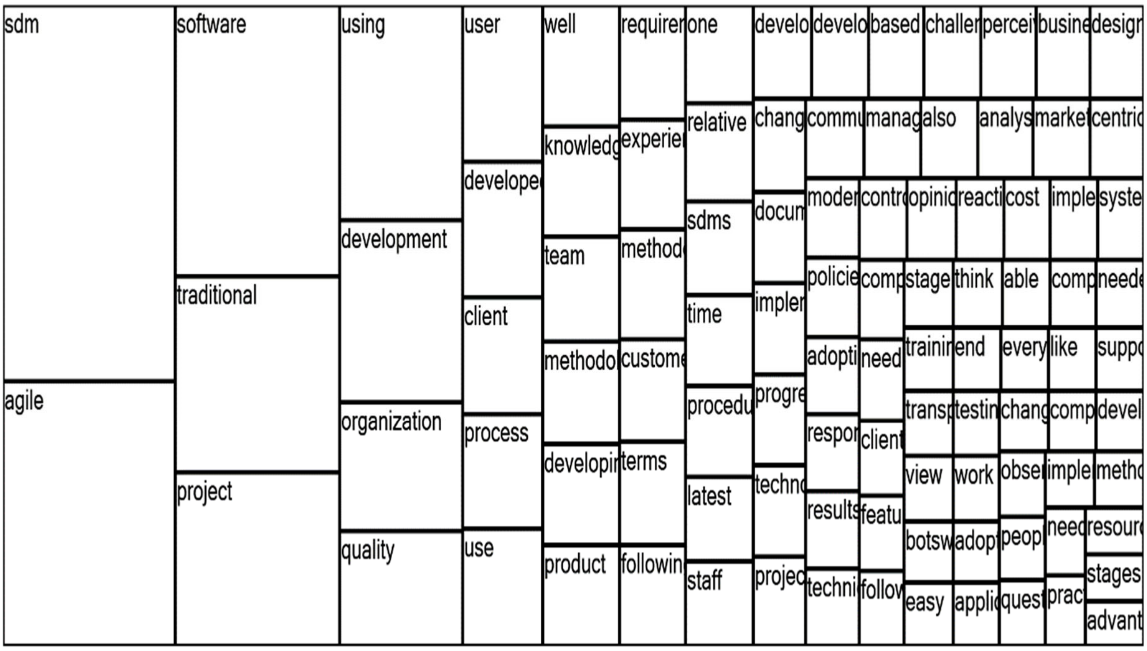


Figure 35 Word Map

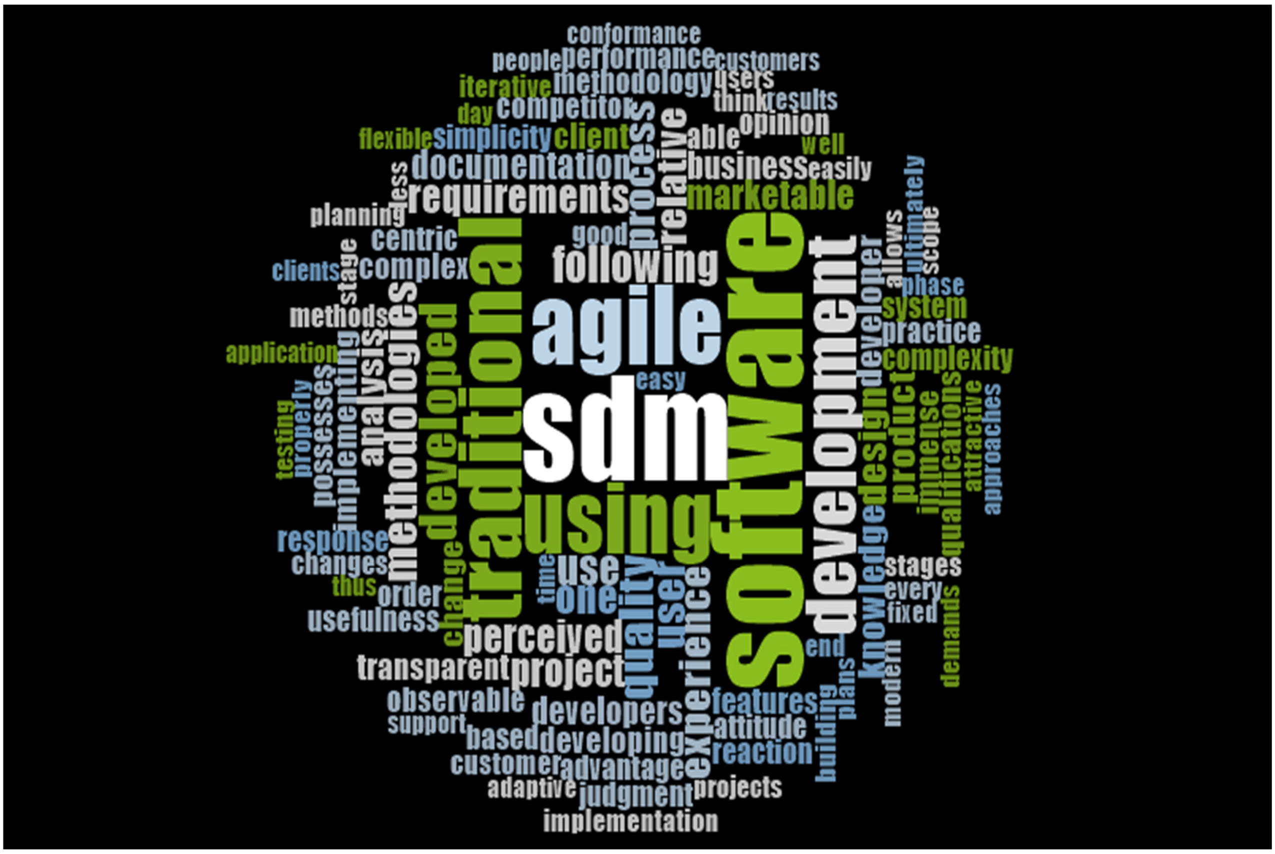


Figure 36 Word Cloud

APPENDIX B: COMPARATIVE DIAGRAMS SAMPLES

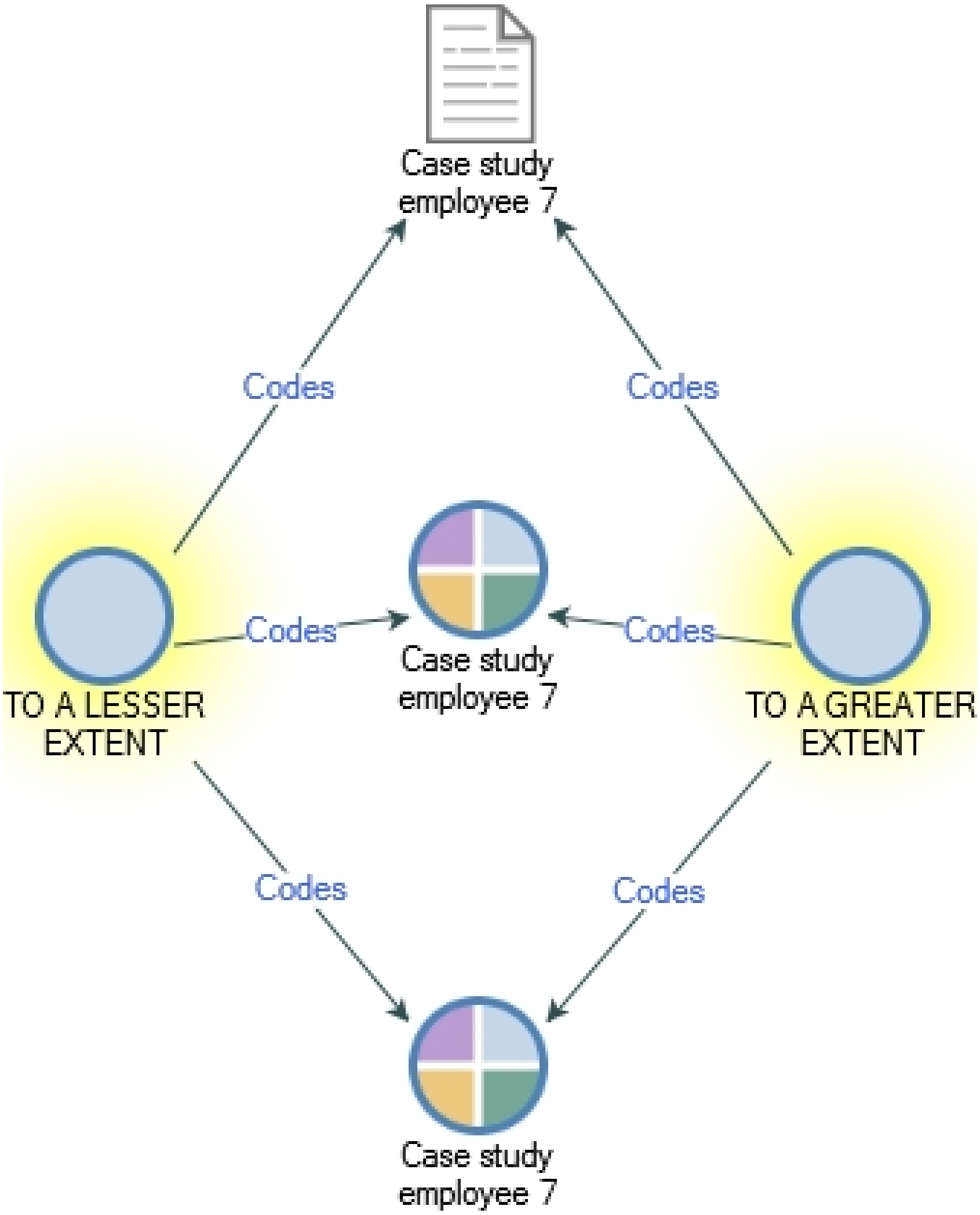


Figure 37Comparative Diagram 1

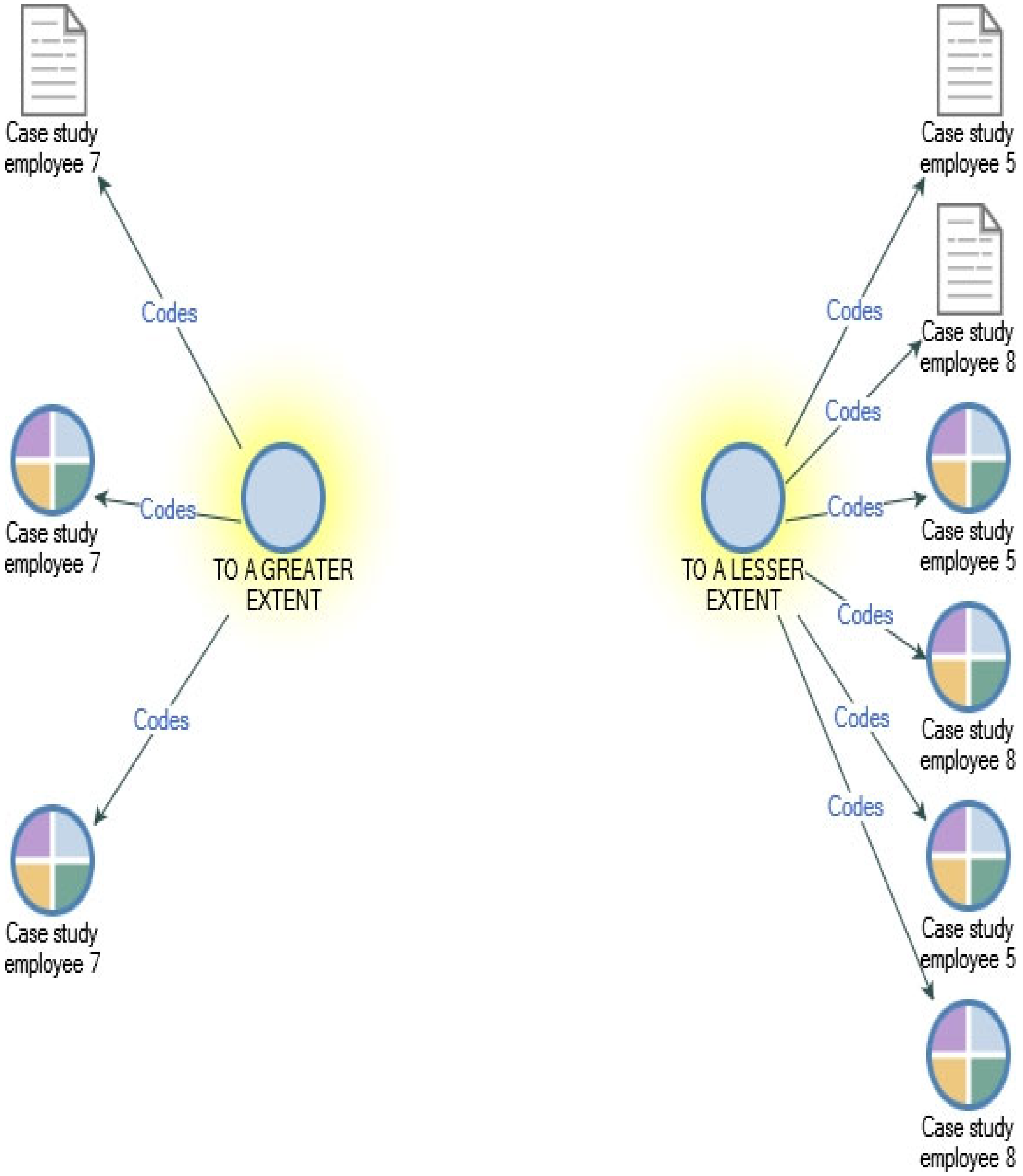


Figure 38 Comparative Diagram 2

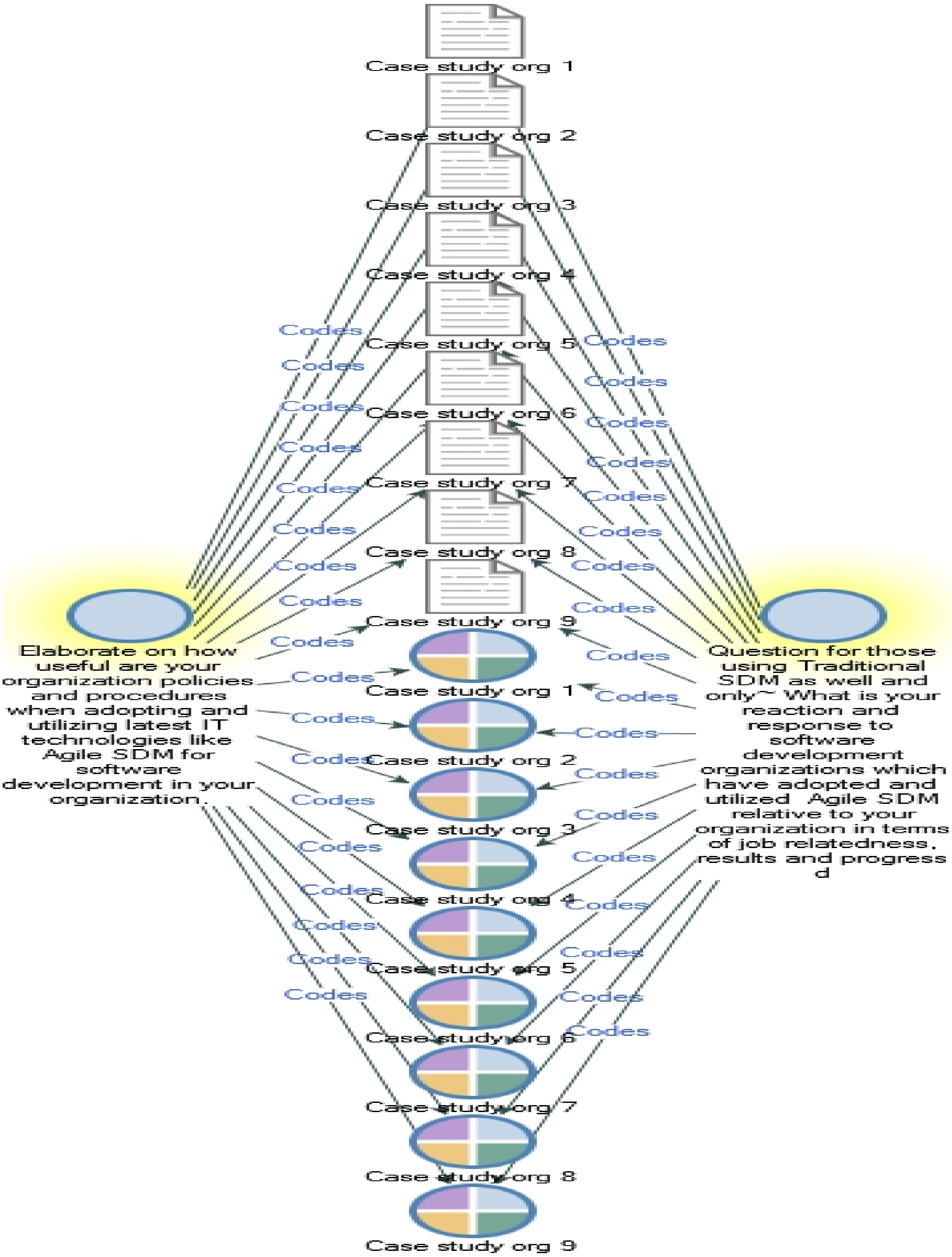

Case study org 1
Case study org 2
Case study org 3
Case study org 4
Case study org 5
Case study org 6
Case study org 7
Case study org 8
Case study org 9
Case study org 1
Case study org 2
Case study org 3
Case study org 4
Case study org 5
Case study org 6
Case study org 7
Case study org 8
Case study org 9
Elaborate on how useful are your organization policies and procedures when adopting and utilizing latest IT technologies like Agile SDM for software development in your organization.
Question for those using Traditional SDM as well and only~ What is your reaction and response to software development organizations which have adopted and utilized Agile SDM relative to your organization in terms of job relatedness, results and progress d
Codes

Figure 39: Comparative Diagram 3

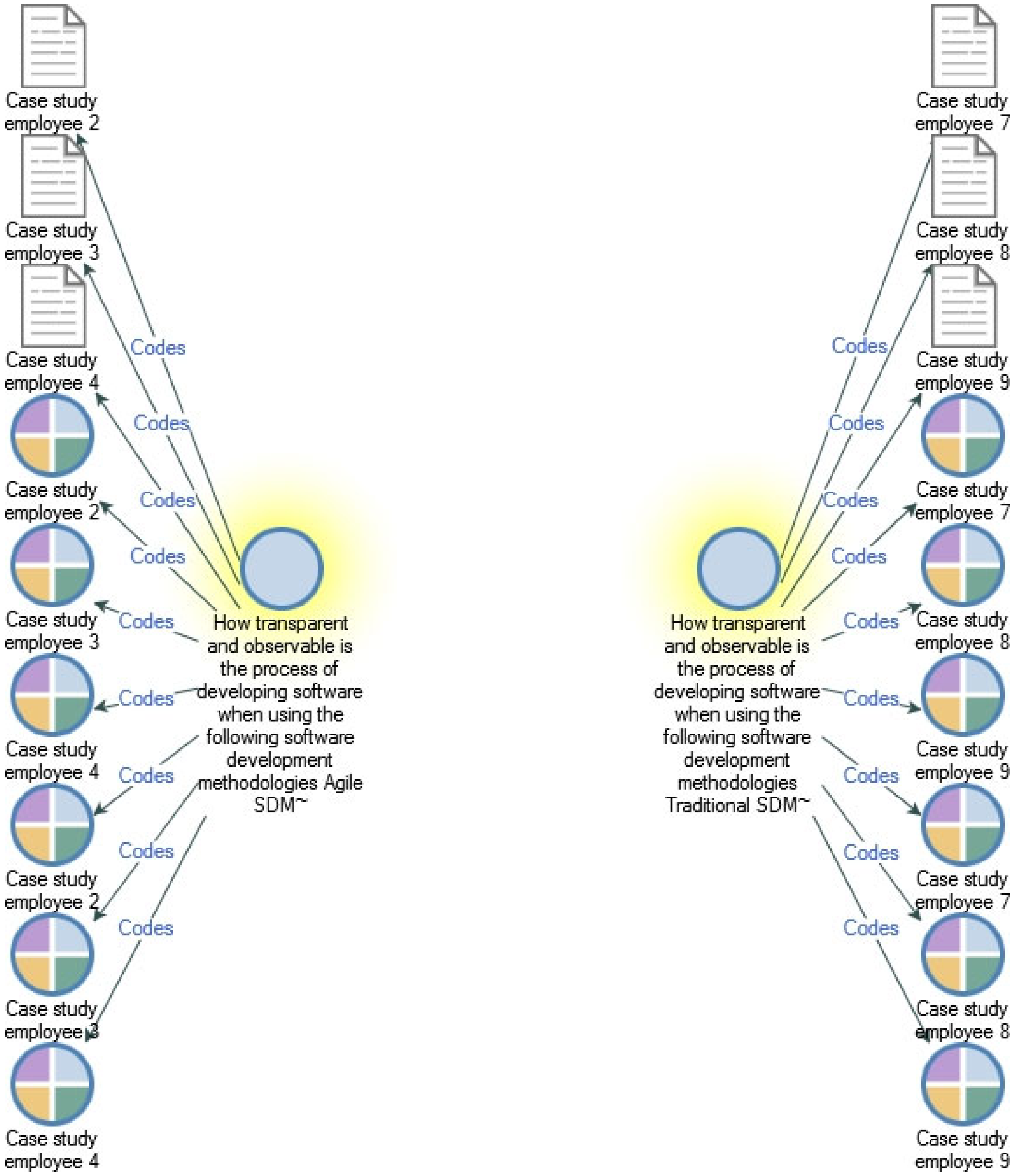


Figure 40: Comparative Diagram 4

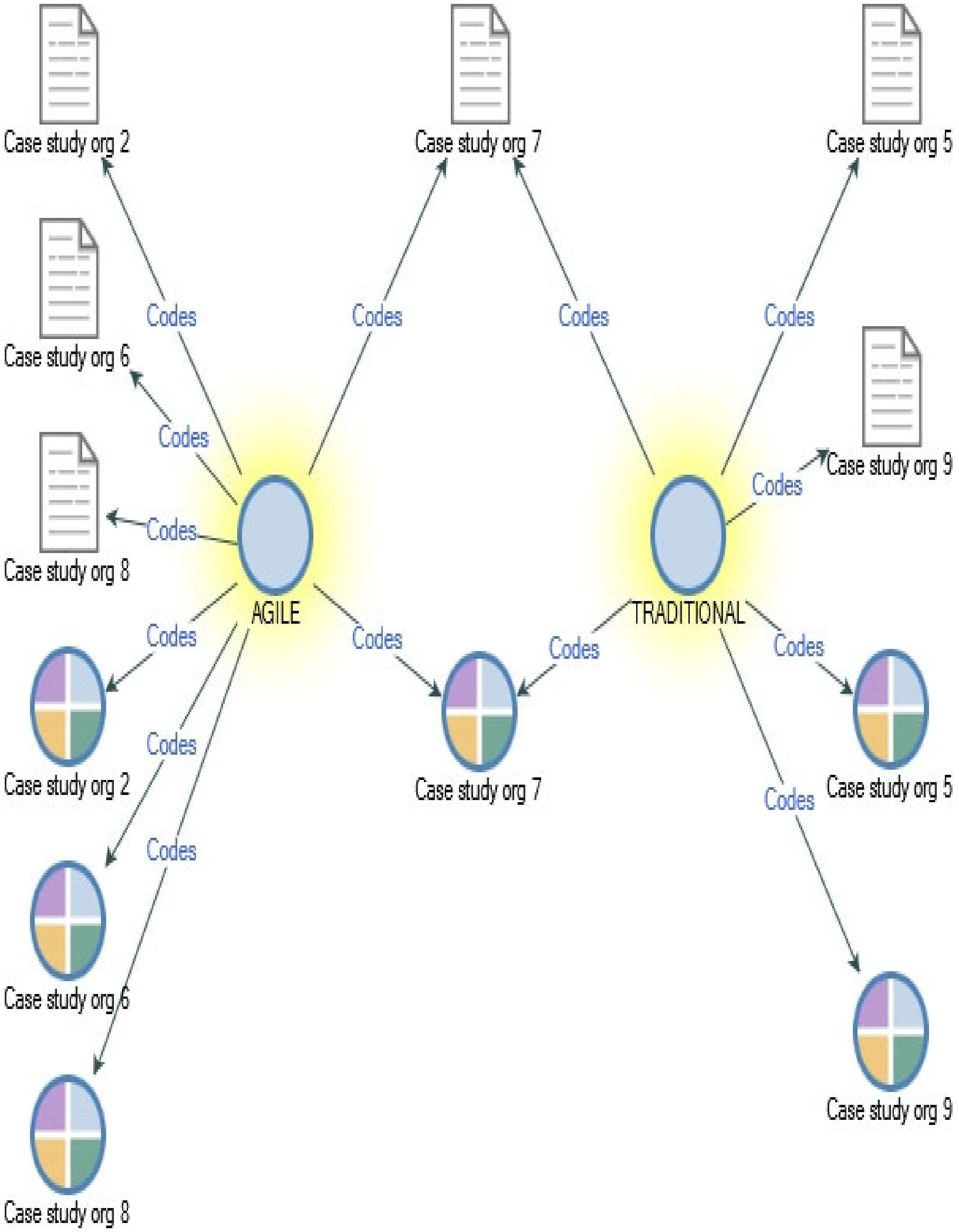


Figure 41: Comparative Diagram 5

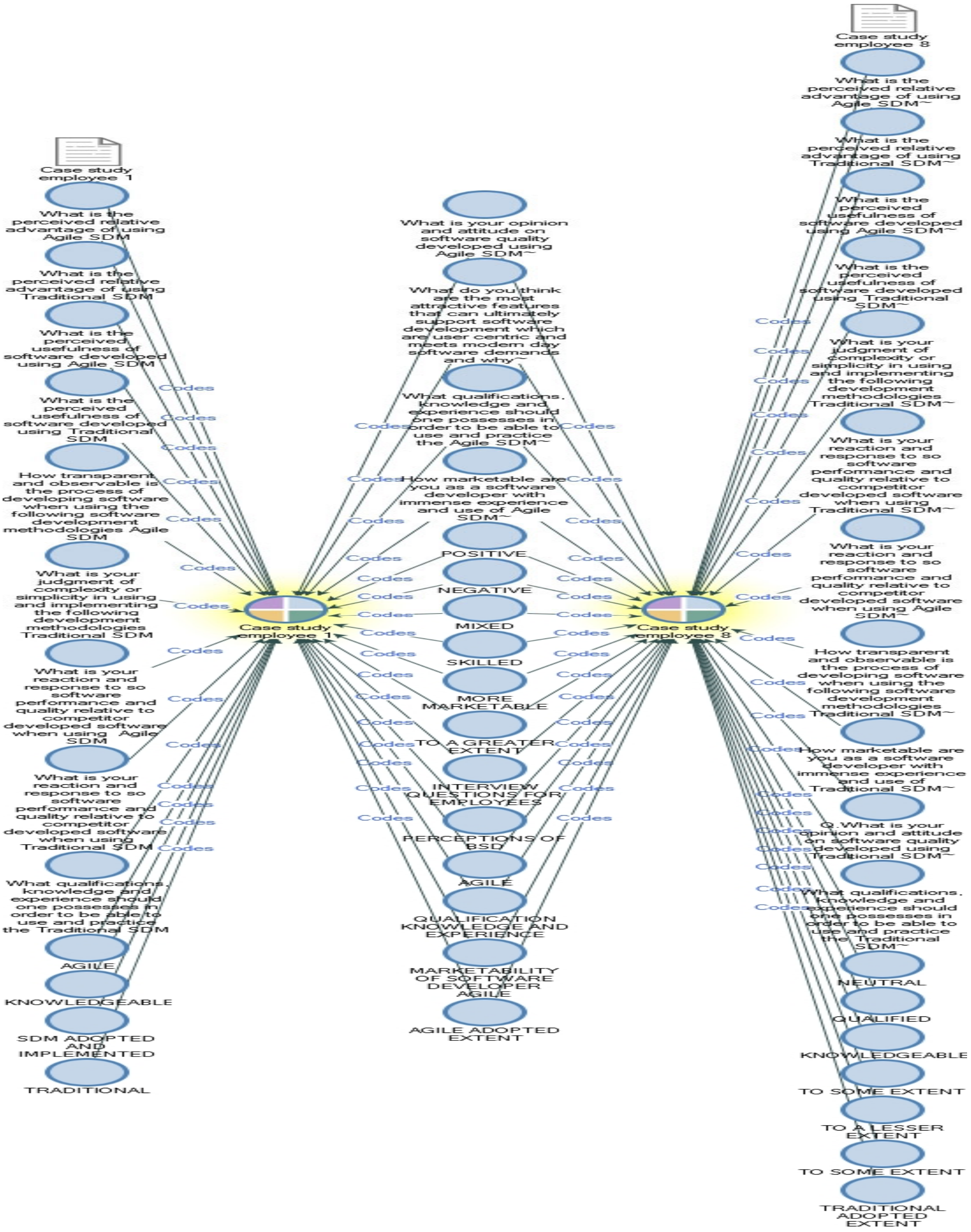

Case study employee 1
What is the perceived relative advantage of using Agile SDM
What is the perceived relative advantage of using Traditional SDM
What is the perceived usefulness of software developed using Agile SDM
What is the perceived usefulness of software developed using Traditional SDM
How transparent and observable is the process of developing software when using the following software development methodologies Agile SDM
What is your judgment of complexity or simplicity in using and implementing the following development methodologies Traditional SDM
What is your reaction and response to so software performance and quality relative to competitor developed software when using Agile SDM
What is your reaction and response to so software performance and quality relative to competitor developed software when using Traditional SDM
What qualifications, knowledge and experience should one possesses in order to be able to use and practice the Traditional SDM
AGILE
KNOWLEDGEABLE
SDM ADOPTED AND IMPLEMENTED
TRADITIONAL
What is your opinion and attitude on software quality developed using Agile SDM~
What do you think are the most attractive features that can ultimately support software development which are user centric and meets modern day software demands and why~
What qualifications, knowledge and experience should one possesses in order to be able to use and practice the Agile SDM~
How marketable are you as a software developer with immense experience and use of Agile SDM~
POSITIVE
NEGATIVE
MIXED
SKILLED
MORE MARKETABLE
TO A GREATER EXTENT
INTERVIEW QUESTIONS FOR EMPLOYEES
PERCEPTIONS OF BSD
AGILE
QUALIFICATION KNOWLEDGE AND EXPERIENCE
MARKETABILITY OF SOFTWARE DEVELOPER AGILE
AGILE ADOPTED EXTENT
Case study employee 8
What is the perceived relative advantage of using Agile SDM~
What is the perceived relative advantage of using Traditional SDM~
What is the perceived usefulness of software developed using Agile SDM~
What is the perceived usefulness of software developed using Traditional SDM~
What is your judgment of complexity or simplicity in using and implementing the following development methodologies Traditional SDM~
What is your reaction and response to so software performance and quality relative to competitor developed software when using Traditional SDM~
What is your reaction and response to so software performance and quality relative to competitor developed software when using Agile SDM~
How transparent and observable is the process of developing software when using the following software development methodologies Traditional SDM~
How marketable are you as a software developer with immense experience and use of Traditional SDM~
Q.What is your opinion and attitude on software quality developed using Traditional SDM~
What qualifications, knowledge and experience should one possesses in order to be able to use and practice the Traditional SDM~
NEUTRAL
QUALIFIED
KNOWLEDGEABLE
TO SOME EXTENT
TO A LESSER EXTENT
TO SOME EXTENT
TRADITIONAL ADOPTED EXTENT
Case study employee 1
Case study employee 8
Codes

Figure 42: Comparative Diagram 6

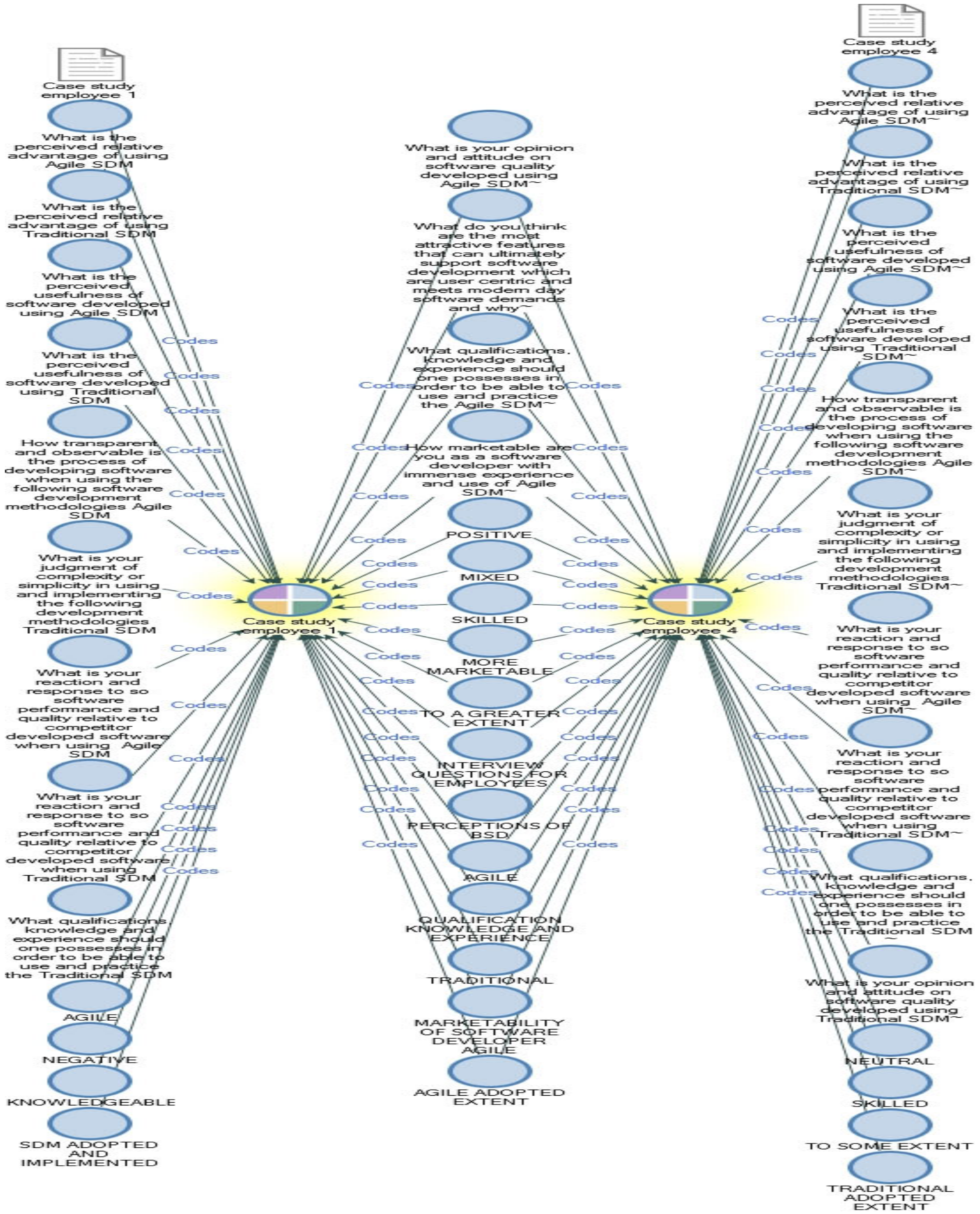

Case study employee 1
What is the perceived relative advantage of using Agile SDM
What is the perceived relative advantage of using Traditional SDM
What is the perceived usefulness of software developed using Agile SDM
What is the perceived usefulness of software developed using Traditional SDM
How transparent and observable is the process of developing software when using the following software development methodologies Agile SDM
What is your judgment of complexity or simplicity in using and implementing the following development methodologies Traditional SDM
What is your reaction and response to so software performance and quality relative to competitor developed software when using Agile SDM
What is your reaction and response to so software performance and quality relative to competitor developed software when using Traditional SDM
What qualifications, knowledge and experience should one possesses in order to be able to use and practice the Traditional SDM
AGILE
NEGATIVE
KNOWLEDGEABLE
SDM ADOPTED AND IMPLEMENTED
What is your opinion and attitude on software quality developed using Agile SDM~
What do you think are the most attractive features that can ultimately support software development which are user centric and meets modern day software demands and why~
What qualifications, knowledge and experience should one possesses in order to be able to use and practice the Agile SDM~
How marketable are you as a software developer with immense experience and use of Agile SDM~
POSITIVE
MIXED
SKILLED
MORE MARKETABLE
TO A GREATER EXTENT
INTERVIEW QUESTIONS FOR EMPLOYEES
PERCEPTIONS OF BSD
AGILE
QUALIFICATION KNOWLEDGE AND EXPERIENCE
TRADITIONAL
MARKETABILITY OF SOFTWARE DEVELOPER AGILE
AGILE ADOPTED EXTENT
Codes
Case study employee 1
Case study employee 4
Case study employee 4
What is the perceived relative advantage of using Agile SDM~
What is the perceived relative advantage of using Traditional SDM~
What is the perceived usefulness of software developed using Agile SDM~
What is the perceived usefulness of software developed using Traditional SDM~
How transparent and observable is the process of developing software when using the following software development methodologies Agile SDM~
What is your judgment of complexity or simplicity in using and implementing the following development methodologies Traditional SDM~
What is your reaction and response to so software performance and quality relative to competitor developed software when using Agile SDM~
What is your reaction and response to so software performance and quality relative to competitor developed software when using Traditional SDM~
What qualifications, knowledge and experience should one possesses in order to be able to use and practice the Traditional SDM ~
What is your opinion and attitude on software quality developed using Traditional SDM~
NEUTRAL
SKILLED
TO SOME EXTENT
TRADITIONAL ADOPTED EXTENT

Figure 43:Comparative Diagram 6

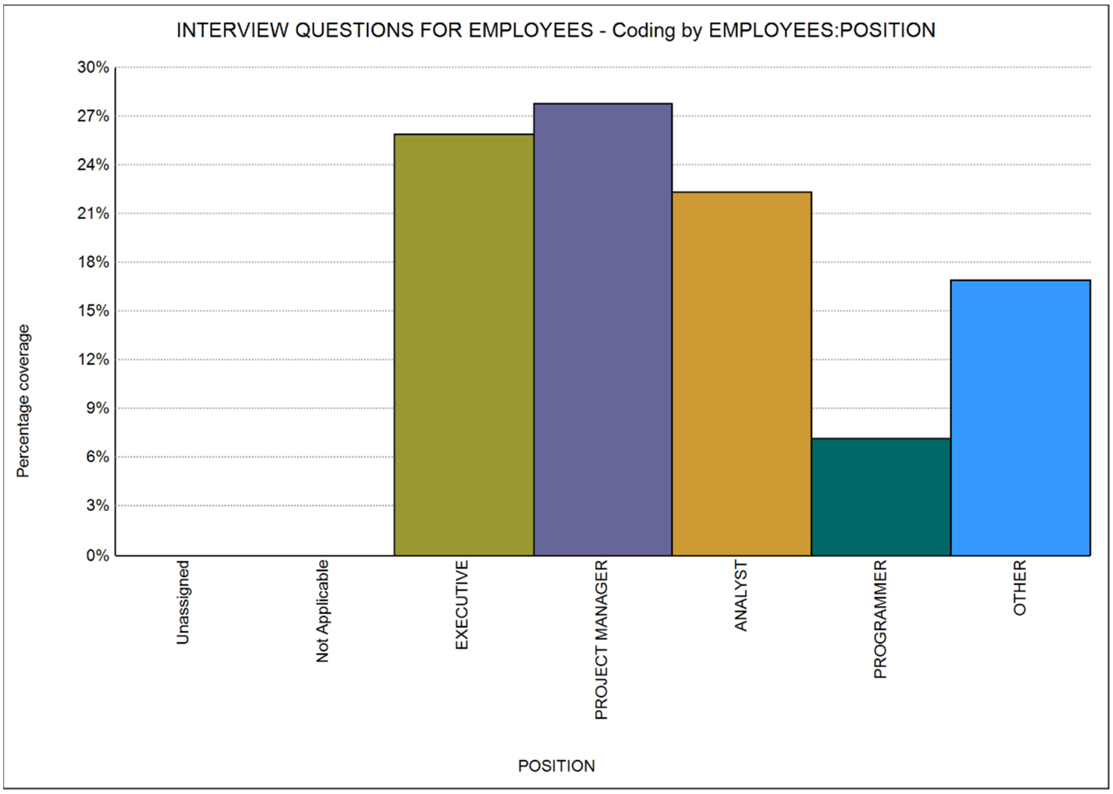
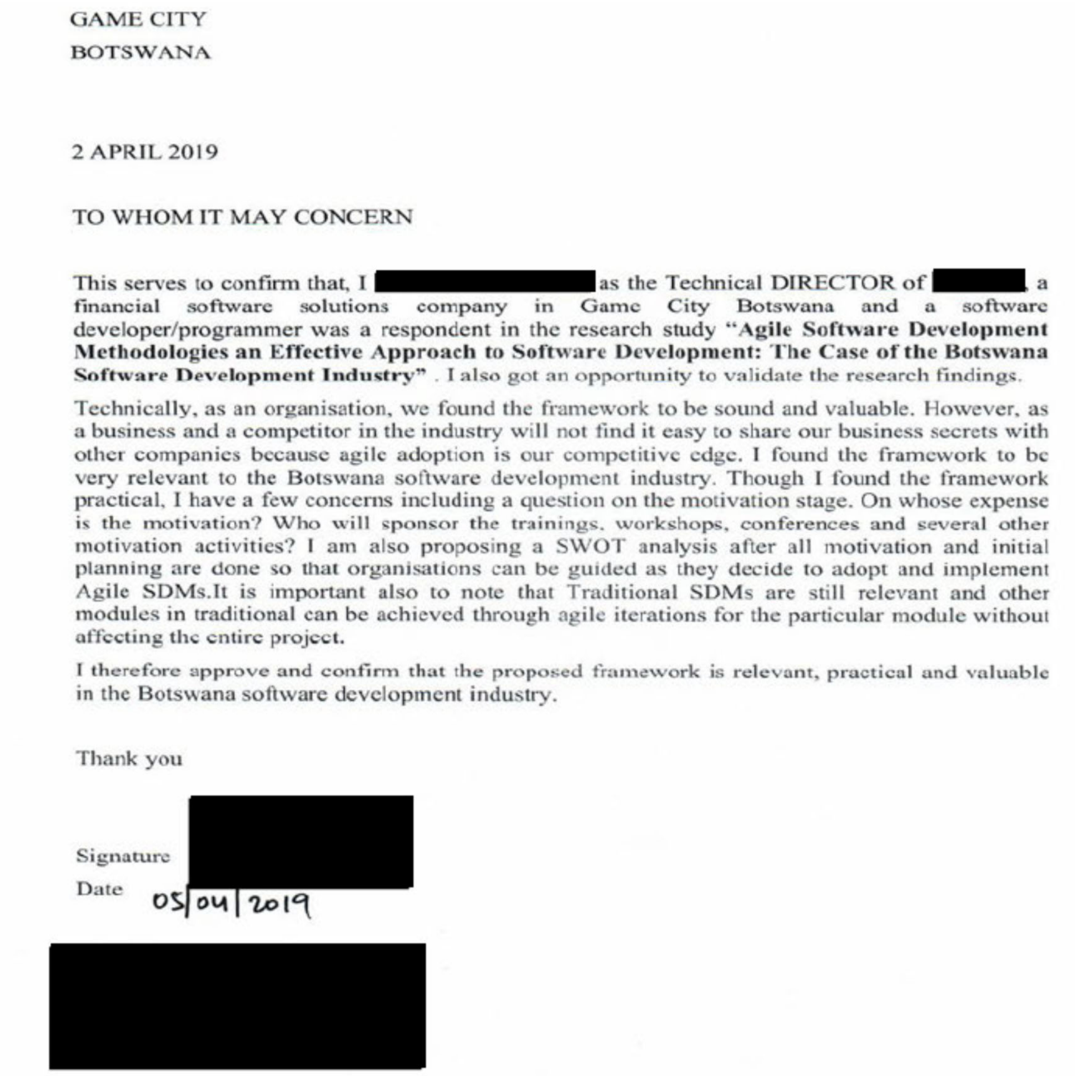


Figure 44: Bar Chart diagram

APPENDIX C: ADOPTION AND IMPLEMENTATION FRAMEWORK VALIDATION LETTERS

Software Development Company A



GAME CITY
BOTSWANA

2 APRIL 2019

TO WHOM IT MAY CONCERN

This serves to confirm that, I [redacted] as the Technical DIRECTOR of [redacted], a financial software solutions company in Game City Botswana and a software developer/programmer was a respondent in the research study **"Agile Software Development Methodologies an Effective Approach to Software Development: The Case of the Botswana Software Development Industry"** . I also got an opportunity to validate the research findings.

Technically, as an organisation, we found the framework to be sound and valuable. However, as a business and a competitor in the industry will not find it easy to share our business secrets with other companies because agile adoption is our competitive edge. I found the framework to be very relevant to the Botswana software development industry. Though I found the framework practical, I have a few concerns including a question on the motivation stage. On whose expense is the motivation? Who will sponsor the trainings, workshops, conferences and several other motivation activities? I am also proposing a SWOT analysis after all motivation and initial planning are done so that organisations can be guided as they decide to adopt and implement Agile SDMs.It is important also to note that Traditional SDMs are still relevant and other modules in traditional can be achieved through agile iterations for the particular module without affecting the entire project.

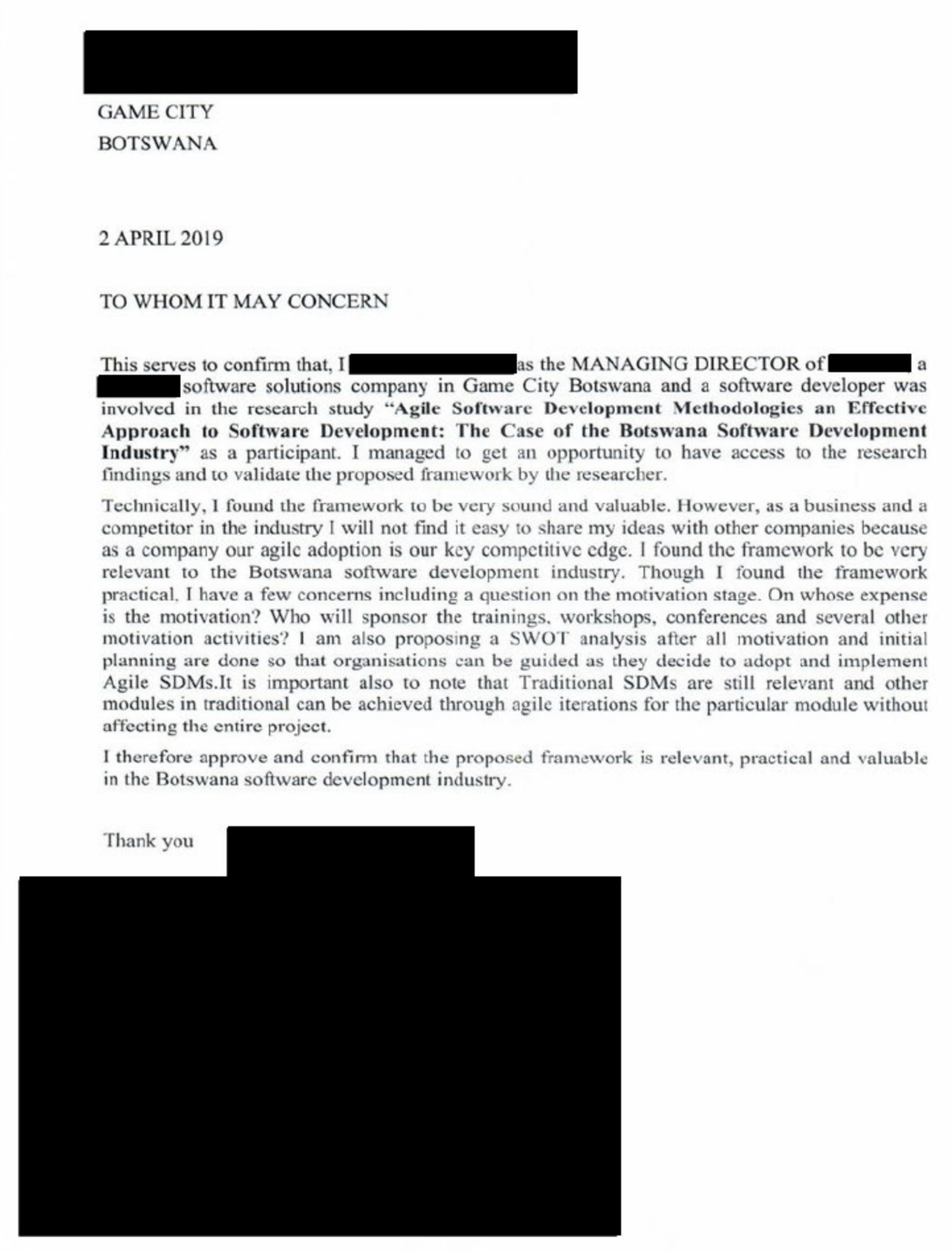
I therefore approve and confirm that the proposed framework is relevant, practical and valuable in the Botswana software development industry.

Thank you

Signature [redacted]
Date 05/04/2019

[redacted]

Software Development Company B



[redacted]

GAME CITY
BOTSWANA

2 APRIL 2019

TO WHOM IT MAY CONCERN

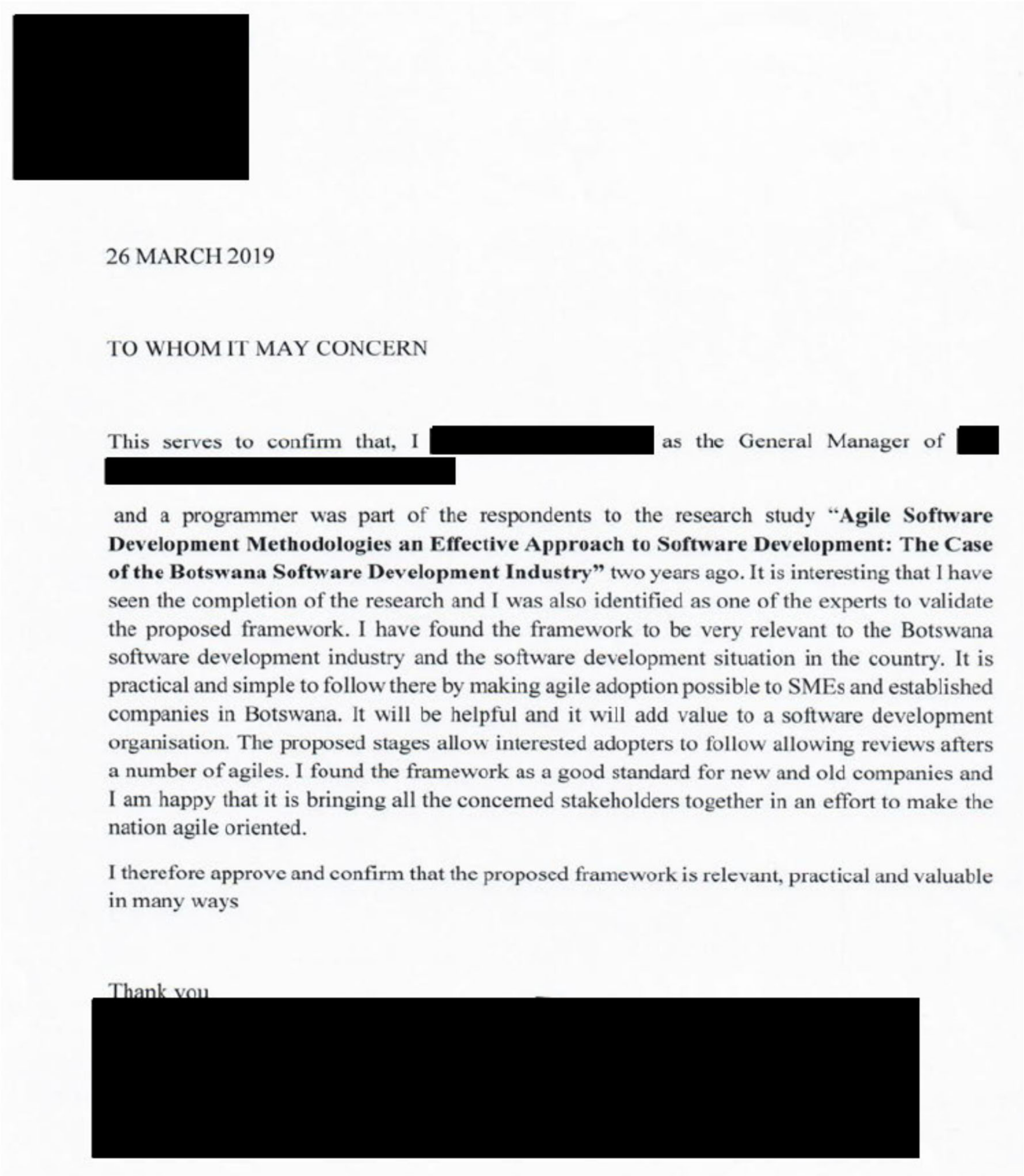
This serves to confirm that, I [redacted] as the MANAGING DIRECTOR of [redacted] a [redacted] software solutions company in Game City Botswana and a software developer was involved in the research study **"Agile Software Development Methodologies an Effective Approach to Software Development: The Case of the Botswana Software Development Industry"** as a participant. I managed to get an opportunity to have access to the research findings and to validate the proposed framework by the researcher.

Technically, I found the framework to be very sound and valuable. However, as a business and a competitor in the industry I will not find it easy to share my ideas with other companies because as a company our agile adoption is our key competitive edge. I found the framework to be very relevant to the Botswana software development industry. Though I found the framework practical. I have a few concerns including a question on the motivation stage. On whose expense is the motivation? Who will sponsor the trainings. workshops, conferences and several other motivation activities? I am also proposing a SWOT analysis after all motivation and initial planning are done so that organisations can be guided as they decide to adopt and implement Agile SDMs.It is important also to note that Traditional SDMs are still relevant and other modules in traditional can be achieved through agile iterations for the particular module without affecting the entire project.

I therefore approve and confirm that the proposed framework is relevant, practical and valuable in the Botswana software development industry.

Thank you [redacted]

Software Development Company C



[redacted]

26 MARCH 2019

TO WHOM IT MAY CONCERN

This serves to confirm that, I [redacted] as the General Manager of [redacted]

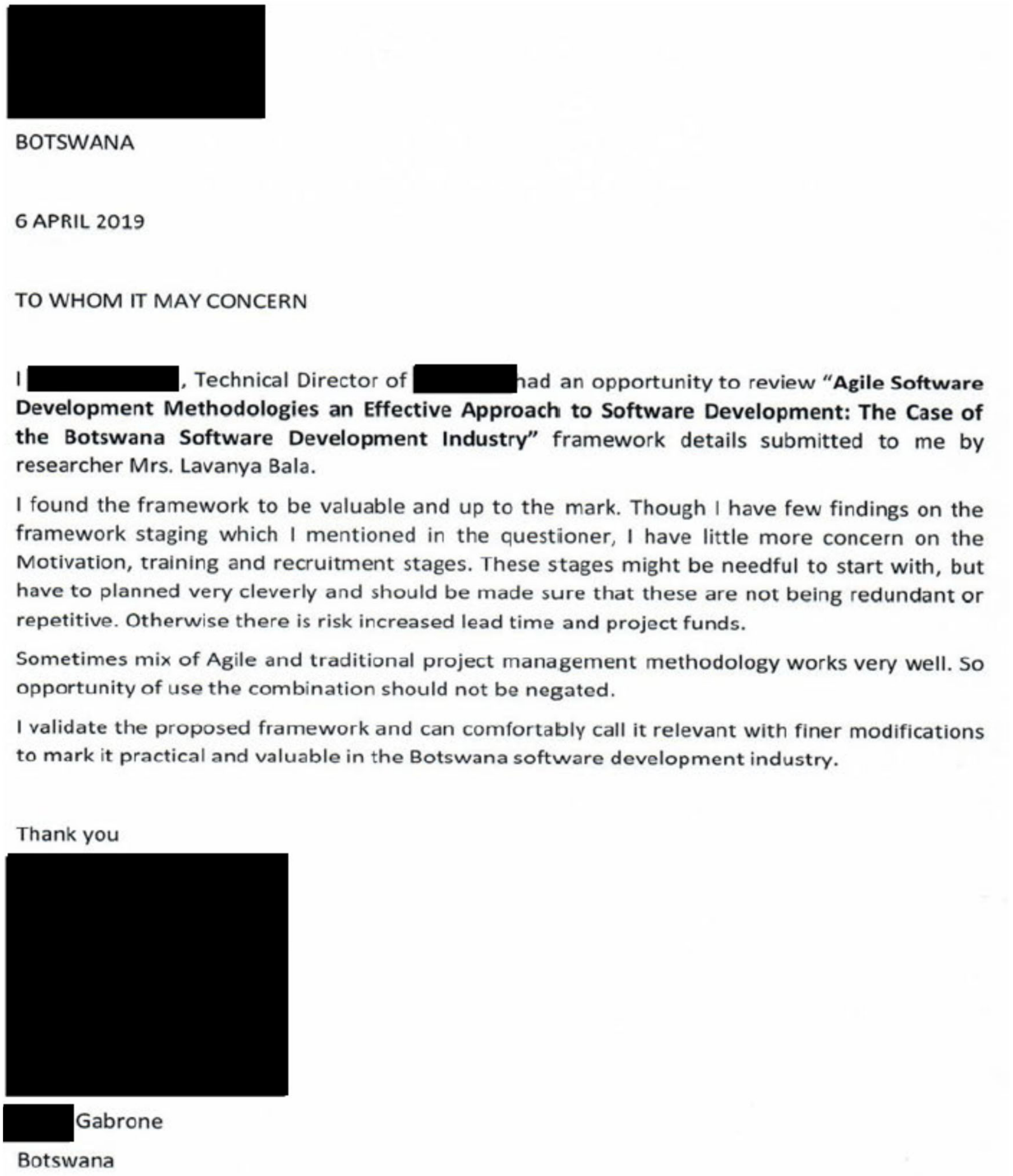
and a programmer was part of the respondents to the research study **"Agile Software Development Methodologies an Effective Approach to Software Development: The Case of the Botswana Software Development Industry"** two years ago. It is interesting that I have seen the completion of the research and I was also identified as one of the experts to validate the proposed framework. I have found the framework to be very relevant to the Botswana software development industry and the software development situation in the country. It is practical and simple to follow there by making agile adoption possible to SMEs and established companies in Botswana. It will be helpful and it will add value to a software development organisation. The proposed stages allow interested adopters to follow allowing reviews afters a number of agiles. I found the framework as a good standard for new and old companies and I am happy that it is bringing all the concerned stakeholders together in an effort to make the nation agile oriented.

I therefore approve and confirm that the proposed framework is relevant, practical and valuable in many ways

Thank you

[redacted]

Software Development Company D



[redacted]

BOTSWANA

6 APRIL 2019

TO WHOM IT MAY CONCERN

I [redacted], Technical Director of [redacted] had an opportunity to review **"Agile Software Development Methodologies an Effective Approach to Software Development: The Case of the Botswana Software Development Industry"** framework details submitted to me by researcher Mrs. Lavanya Bala.

I found the framework to be valuable and up to the mark. Though I have few findings on the framework staging which I mentioned in the questioner, I have little more concern on the Motivation, training and recruitment stages. These stages might be needful to start with, but have to planned very cleverly and should be made sure that these are not being redundant or repetitive. Otherwise there is risk increased lead time and project funds.

Sometimes mix of Agile and traditional project management methodology works very well. So opportunity of use the combination should not be negated.

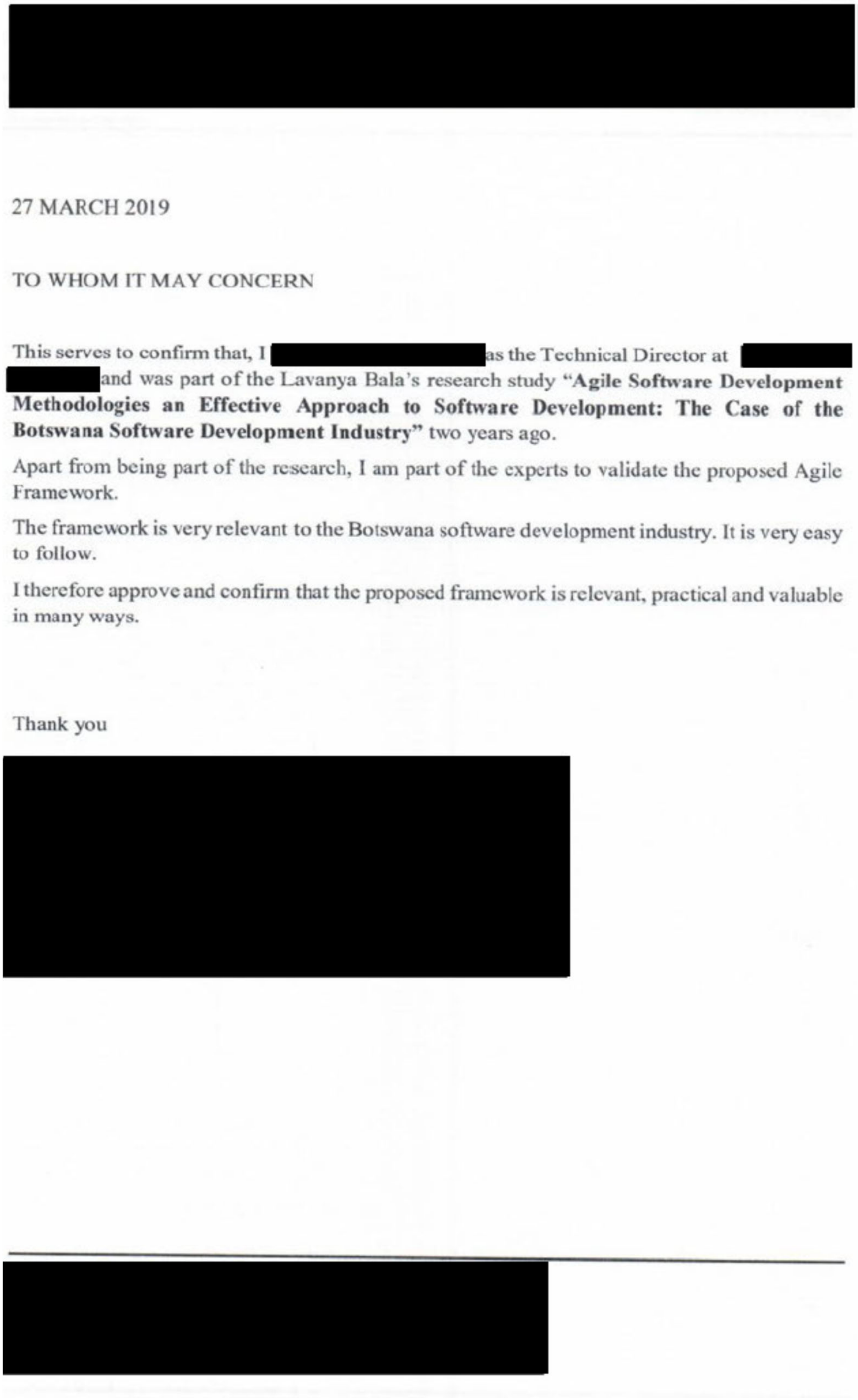
I validate the proposed framework and can comfortably call it relevant with finer modifications to mark it practical and valuable in the Botswana software development industry.

Thank you

[redacted]

[redacted] Gabrone
Botswana

Software Development Company E



[redacted]

27 MARCH 2019

TO WHOM IT MAY CONCERN

This serves to confirm that, I [redacted] as the Technical Director at [redacted] [redacted] and was part of the Lavanya Bala's research study **"Agile Software Development Methodologies an Effective Approach to Software Development: The Case of the Botswana Software Development Industry"** two years ago.

Apart from being part of the research, I am part of the experts to validate the proposed Agile Framework.

The framework is very relevant to the Botswana software development industry. It is very easy to follow.

I therefore approve and confirm that the proposed framework is relevant, practical and valuable in many ways.

Thank you

[redacted]

[redacted]