A Model for integrating User Experience into Agile software development

by

Mpile Manakaza

Submitted in fulfilment of the requirements for the degree

Master of Information Technology

in the

Faculty of Engineering, the Built Environment and Technology

of the

Nelson Mandela University

Supervisor: Prof. Darelle van Greunen

2023

DECLARATION

I, Mpile Manakaza (student number s208108752), declare that the dissertation titled: A Model for Integrating UX into Agile Software Development, which I have submitted to the Nelson Mandela University, has not been submitted before by anyone at any University and, therefore, I am the originator of this dissertation.

Mpile Manakaza

2023

Official use:
In accordance with Rule G5.11.4, I hereby declare that the above-mentioned treatise/ dissertation/ thesis is my own work and that it has not previously been submitted for assessment to another University or for another qualification. However, material from publications by the student may be embodied in a treatise/dissertation/ thesis

ACKNOWLEDGMENTS

First and foremost, praises and thanks to God, the Almighty, for His showers of blessings throughout my research study right to its successful completion.

Completing this dissertation was challenging but a great learning experience that has portrayed my capabilities. Therefore, I am grateful for the opportunity provided to me by Nelson Mandela University.

I would like to express my deepest and most sincere gratitude to my research supervisor, Prof. Darelle van Greunen, for her invaluable guidance, patience, and supervision throughout this research. Her constant motivation, vision, sincerity, and support kept me going. It has been a great honour and privilege to study under her guidance.

I would also like to thank my family for their love, prayers, support and caring for the success of my education.

Special thanks go out to my colleagues and friends for their unwavering, support, motivation, love, and contribution to this research study.

ABSTRACT

The Agile software development methodology has progressed over time and has become the most popularly used methodology for many projects owing to its adaptable and advanced nature. Agile is an iterative and incremental method that supports project teams to maintain the growing demands of present-day organisations. In software development, it is imperative to build software that ensures a holistic user experience and meets requirements. Software Developers and User Experience Designers focus on users' preferences and requirements while navigating an application through an approach called user-centred design (UCD). The purpose of Agile user-experience (UX) methods is to build user-centric software solutions using the Agile methodology. Literature indicates a scarcity of Agile-UX models to integrate user experience design processes effectively within Agile software development. Hence, the motivation for this study was to create a practical model for integrating user experience design processes into Agile software development for implementation by software development teams.

Initially, the study defines the problem that formed the basis for the initiation of this research, followed by the research questions and objectives that are the fundamental portion of the development of this study.

This study highlights the benefits and challenges of Agile-UX methods through a case study of Agile software development teams in South Africa. Furthermore, additional best practices are recommended.

A survey questionnaire was distributed to software development teams to identify the various factors that lead to implementing Agile-UX methods. The distribution of the survey questionnaire supported the theoretical perspective of this study. Furthermore, Agile development (AD) and UCD are not easy to combine as UCD aims to understand users before software product development, and no activities are performed before development starts in AD. Lastly, a significant benefit is that user-centred software systems create and promote team collaboration and communication and, ultimately, successful software projects.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

TABLE OF ACRONYMS

ACRONYM	**TERM IN FULL**
ABAP	Advanced Business Application Programming
AD	Agile Development
ASD	Agile Software Development
Agile-UX	Agile User Experience
AM	Agile Modelling
BA	Business Analyst
DSDM	Dynamic Systems Development Methodology
XP	Extreme Programming Methodology
FAQs	Frequently Asked Questions
HCD	Human-Centred Design
IT	Information Technology
MVP	Minimum Viable Product
SD	Software Development
SDLC	Software Development Life Cycle
SDM	Software Development Methodology
UCD	User-Centred Design
UX	User Experience

ACRONYM	TERM IN FULL
UI	User Interface
UXD	User Experience Design
WC	Western Cape

GLOSSARY OF TERMS

Terminology	Description
Business Analyst	Business Analysts (BA) help guide businesses in improving processes, products, services, and software through data analysis. They straddle the line between IT and the business to help bridge the gap and improve efficiency (Pratt & White, 2019).
Cross-Functional Teams	A cross-functional team is a group of people with different functional expertise across various sectors working towards a common goal (Indeed,2020).
Developer	A Developer is a computer science specialist who creates computer software applications. They are proficient in the use of computer programming languages, which they use to write code to perform software functions (Techopedia, 2017).
Development Sprint	In Agile product development, a Sprint is a fixed period which detailed tasks must be completed and made ready for release. The product owner and the development team agree upon exactly what work will be accomplished during the Sprint (Brunskill, 2020).

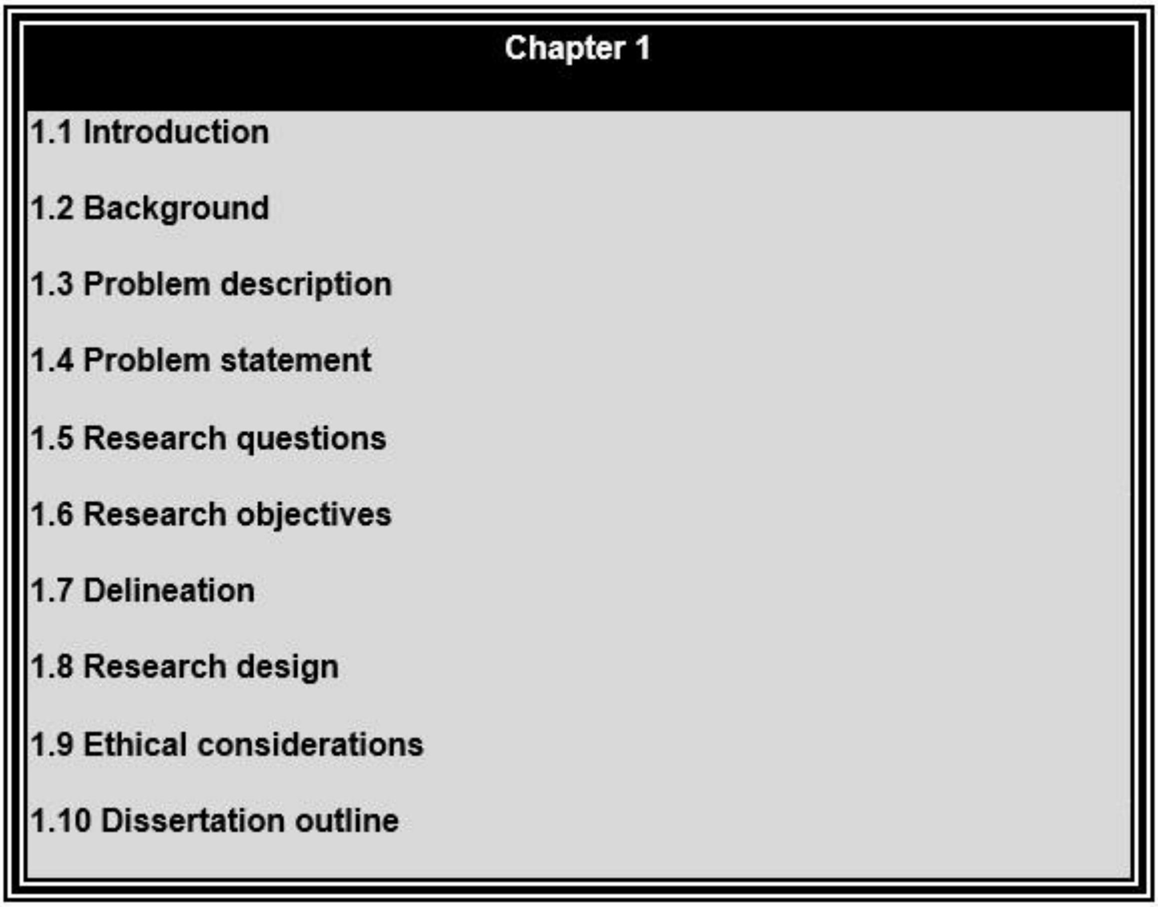
Terminology	Description
Domain Owner	Domain Owners lead a team of software engineers. They often have a similar role to product owners. Domain owners are viewed as subject matter experts in a specific field and deal with defining and capturing technical requirements (Swords, 2017).
End User	In information technology, an end user is referred to as a person who eventually uses or plans to use a product (Hayes, 2022).
Head of Engineering	Head of Engineering is regarded as the 'peoples champion' that ensures that software engineers thrive by inspiring, mentoring and resolving any issues that arise and ultimately ensuring software product delivery aligns according to Chief Technology Officer's objectives (Kuizinas, 2021).
Head of Product	Head of Product development is responsible for supervising team members as well as product development within an organisation while ensuring the products in question meet the organisation's vision (Johnson, 2021).
Integrating	To integrate is to form, coordinate or blend into a functioning or unified whole. To incorporate into a larger unit (Merriam-Webster, 2022).
Kanban	Kanban is a workflow management method for describing, running, and refining services and maximising delivery. It aims to help visualise your work and increase efficiency (Riaz, 2019).
Manager	A Manager is a leader in an organisation who oversees, developing and encouraging employees. They are responsible for the functioning of the team they manage within the organization. Ensuring that

Terminology	**Description**
	operations happen efficiently and effectively (Gartenstein, 2018).
Prototype	A prototype is an early example, framework or release of a developed product to assess a concept or process (Ramírez, 2018).
Project Manager	A Project Manager is a member of a software development team responsible for planning, executing, monitoring, and closing projects (Alexander, 2021).
Product Owner	A Product Owner is a member of the Agile team who is on behalf of the customer, responsible for working with product management and other stakeholders to prioritise stories in the backlog (Scaled Agile,2022).
Scrum Master	Scrum Masters are responsible for the Scrum team's effectiveness. They enable the team to improve its practices within the Scrum framework (Keatinge, 2021).
Software Development Process	A software development process is a step-by-step method of building software (Indeed, 2021a).
Software Development Project	A software development project is a multifaceted undertaking by two or more persons within the boundaries of time, budget and staff resources that produce new or enhanced computer code that increases business value (O'Reilly, 2022).
Software Life Cycle Models	A software life cycle model is a visual and graphic representation of the software life cycle representing all the methods required to make a software product transit through its life cycle phases (Techopedia, 2022).

Terminology	**Description**
Software Process	Software processes are activities for designing, implementing, and testing a software system (Sulemani, 2021).
Systems Architect	Systems Architects are responsible for planning, designing, operating, and sustaining computer and networking systems. They evaluate desired processes and results and give direction on the correct combination of IT systems and mechanisms to achieve specific organisational objectives (Shiff, 2020).
Tech stack	A set of technologies that are used by an organisation to build software solutions (Heap, 2022).
Technical Lead	Technical Leads are experts who manage a team of technical persons in a software development organisation (Indeed, 2021b).
Test Analyst	Test Analysts are the persons responsible for testing computer hardware or software before it is packaged and shipped to the market (The Career Coach Company, 2022).
User-Centric Agile Development Model	A user-centric development model integrates UX Designers into the Agile development team into the UX process to build user-centred software products (Ventera, 2021).
UX Methods	These are techniques used by UX Designers to create great experiences for users (Babich, 2018).
UX Designer	UX Designers focus on all aspects of product development, design, usability, branding and marketing. They’re responsible for a user’s entire journey while

Terminology	Description
	interacting with a product or service and identifying new opportunities for it and the organisation (White, 2020).
UX Experts	UX experts specialise in the design and development of user interaction and the use of techniques for developing human computer interaction (Holishevska, 2022).
UX Sprint	A UX design sprint is a short, time-framed process for validating concepts and resolving challenges through co-creation, research, prototyping and testing ideas (Lo, 2017).

Chapter 1: Introduction



Chapter 1

1.1 Introduction

1.2 Background

1.3 Problem description

1.4 Problem statement

1.5 Research questions

1.6 Research objectives

1.7 Delineation

1.8 Research design

1.9 Ethical considerations

1.10 Dissertation outline

1.1 Introduction

The purpose of this chapter is to present the outline of this research study. Section 1.2 presents the background to the study. The problem description is outlined in section 1.3, followed by the problem statement in section 1.4. Research questions and objectives are presented in section 1.5 and 1.6. The delineation is highlighted in section 1.7. Finally, the research paradigm is defined in section 1.8 followed by ethical considerations in section 1.9 and the dissertation outline in section 1.10.

1.2 Background

User experience (UX) refers to everything that has to do with a user's interaction with an organisation, its products, or services (Norman & Nielsen, 2022, Sauer et al., 2020). The purpose of UX in software development is to develop usable software products that meet users’ needs. Hassan and Galal-Edeen (2017) state that UX is an essential part of the success and failure of user-centred software development projects and ultimately, the organisations that produce them. Pillay & Wing (2019) highlight that UX methods place the needs of the user higher than those of the system. These authors also mention that an analysis of existing literature often shows that Agile software development and user-centred design are equally two important processes for warranting effective UX.

The role of user experience (UX) design is closely linked to user interface (UI) design, which enhances simplicity of use (The Interaction design Foundation, 2022; Sauer et al., 2020). UX design focuses on user interaction-related research, which evaluates data to determine the best UX approach. UI design focuses on the user’s visual interaction and navigation with the interface (Silva-Rodríguez & Nava-Muñoz, 2018).

Laubheimer (2017) states that there are challenges in integrating UX design and Agile. According to Jones (2019), although UX and Agile might compensate for the challenges introduced by traditional software development, it may be a difficult task to successfully combine the two. Agile is implemented in organisations through the Scrum methodology where the UX Designer focuses on a specific feature (per user story). Focus is, therefore, put on the feature instead of the entire software product, which can be difficult for UX Designers to adapt to (Laubheimer, 2017). Although it may not be an easy task to integrate Agile and UX, it is not impossible to implement the integration successfully (Justinmind, 2018). The decision to integrate UX design into the Agile software development process depends on the type of project and the organisation that produces it (Follett, 2017). The entire organisation needs to comprehend the role UX experts play and how they will benefit the organisation (SilvaRodríguez & Nava-Muñoz, 2018).

Levitt (2022) emphasises that UX design should not take place in isolation of software development. The UX design process usually starts before code development in the Scrum process. However, this does not imply that UX Designers should work in isolation from the rest of the software development team, which would result in developers not having any visibility or input into the UX design process. Likewise, UX Designers would not be able to determine the technical restrictions from a developer’s perspective. Silva-Rodríguez & Nava-Muñoz (2018) also state that as user expectations continue to increase, UX Designers need to emphasise thoroughness when conducting user experience (UX) tasks and, likewise, developers need to develop exceptionally operational software products.

Levitt (2022) highlights a research report known as “Business Value of Design” issued by an organistion by the name of McKinsy & Company. This report established that UX is what sets an organisation apart from the rest, giving it a competitive advantage. Furthermore, the report suggests that organisations that regard attention to UX as less of a priority or that it should be isolated or reduced, are adopting a flawed approach. Instead, teams should strive to understand the importance of UX design holistically. The report further states that proclaiming that your organisation is “user-centred” is simply not enough. Organisations should adopt user-centred practices instead (Levitt,2022).

MJV Team (2021) states that resolving issues is the leading concern from UX Designers and Developers. These experts spend far too much time resolving issues that could have been avoided if the gap between design and development had been bridged by ensuring clearly defined requirements, user knowledge, prioritization, and communication by organisations stakeholders. TechTarget (2022) highlights that investing in UX will ensure that Developers work with accurate requirements that have been analysed by users.

Although the Agile manifesto emphasises customer relationships over contract agreements, Agile solely does not support the usability of the software product, states Pillay & Wing (2019). The same authors argue that the need for effective UX has become more important in current day software development organisations. Building software systems that the user will find beneficial and meet their needs affect the success or failure of that software system. This can ultimately affect the organisation negatively if not managed properly. Organisation executives must realise the value of user-centred Agile software development by following an approach that will work well for the company and the software development team. This approach should aim to implement agility in UX and from UX to Agile and provide some flexibility to the whole process and a competitive edge for the business (Justinmind, 2018, Laubheimer, 2017).

1.3 Problem description

Agile software development remains the most popularly used approach for software development projects according to Hoda at el. (2018). Similarly, UX design is significant for the development of a software product as it ensures that the product is created to customer satisfaction. However, UX design is isolated from the Agile software development process in practice (Ilievska,2019). Bruun et al. (2018) state that UX professionals are not considered as part of the Agile development team, the Agile development process, or given sufficient time to create user-centred designs. Therefore, integrating UX design into Agile software development will ensure that users are at the centre of Agile software development. The software product will be incrementally evaluated for user experience functionality and allow for successful Agile software development projects through the combined benefits of both processes.

1.4 Problem statement

Current research shows that insufficient models exist to integrate user experience design processes within Agile software development effectively.

1.5 Research questions

The following are the research questions that direct the focus of this research study:

Primary research question:

What elements should be included in a model for integrating user experience design process into Agile software development?

Secondary research questions:

- What best practices currently exist to integrate user experience into Agile software development?
- What challenges exist for integrating user experience into Agile software development?
- What is the impact of integrated UX practices in Agile software projects?

1.6 Research objectives

The primary objective of this research study is aligned with the primary research question. Similarly, the secondary objectives align with the secondary research questions.

Primary research objective:

To develop a model for integrating user experience design processes into Agile software development.

Secondary research objectives:

To identify existing best practices to integrate user experience design processes into Agile software development.
To identify the challenges that exist for integrating user experience design processes into Agile software development.
To investigate the impact of integrated UX practices in Agile software projects.

1.7 Delineation

This research study is restricted to software development teams. The scope of the research is focused on companies that apply Agile software development methodologies to their software development projects. The geographical area of the study is limited to the Western Cape Province of South Africa where there is a large concentration of software development companies.

1.8 Research design

Research design is the plan that guides the sequence of events for data collection and analysis to meet the research objective. It plays a crucial role in bridging the gap between the research question and the implementation of the research (Boru, 2018).

The Saunders Research Onion *(*Figure 1.1*)* provides a detailed description of the phases involved in the development of the research process (Saunders et al., 2019).

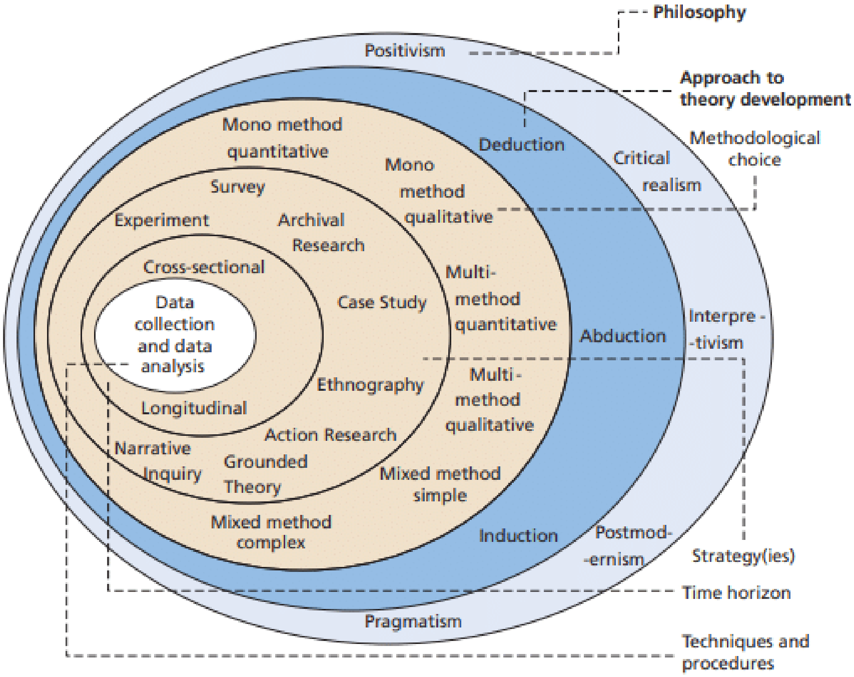


Figure 1.1: Research onion (Saunders et al., 2019)

It begins with the research philosophy, which defines the principles for developing the research. Following is the research approach, which is the detailed method used to conduct it. The research strategy is the following layer that introduces the research approach. Several strategies exist, and selection is based on the purpose of the study. The following layer consists of research choices. This layer determines how many data types will be used for the research and, again, selection is based on the purpose of the research. Time horizon is the following layer, and it explains when and how often data is to be collected. Lastly, decisions must be made regarding the techniques and procedures that will be utilised for the research study. These include decisions on what data to collect, data collection methods, sampling, data analysis and setting up materials to utilise during data collection (Phair & Warren, 2021).

Phair & Warren (2021) state that it is important to understand that the technique selected should align with all the other selections made at each layer of the research onion. Similarly, all the selections made at each layer should align with the purpose of the research. Chapter 4 explains the details of the research design process for this study.

1.9 Ethical considerations

Participants completed an informed consent that outlined the purpose of the data collection and explained that participation was voluntarily. The participants did not record any information that could identify individuals or their organisations.

1.10 Dissertation outline

Chapter 2 – Software design methodologies

This chapter focuses on the current state of knowledge of software development methodologies used in software development projects in the South African context. It explores the concepts that affect the interaction between these methodologies and user experience.

Chapter 3 – User-Centred Agile development

This chapter discusses the Agile-UX concept, explores the contributions and challenges affecting software development projects.

Chapter 4 – Research Methodology and design

This chapter highlights the approach followed for the research. It also presents the philosophical expectations, the research process, and the methods used in the study.

Chapter 5 – Case study results and findings

This chapter presents the research results collected from the case study conducted for this research.

Chapter 6 – Discussion of findings

This chapter defines an in depth and thorough discussion on the findings of the case study and presents recommendations for Agile SD teams based on these findings.

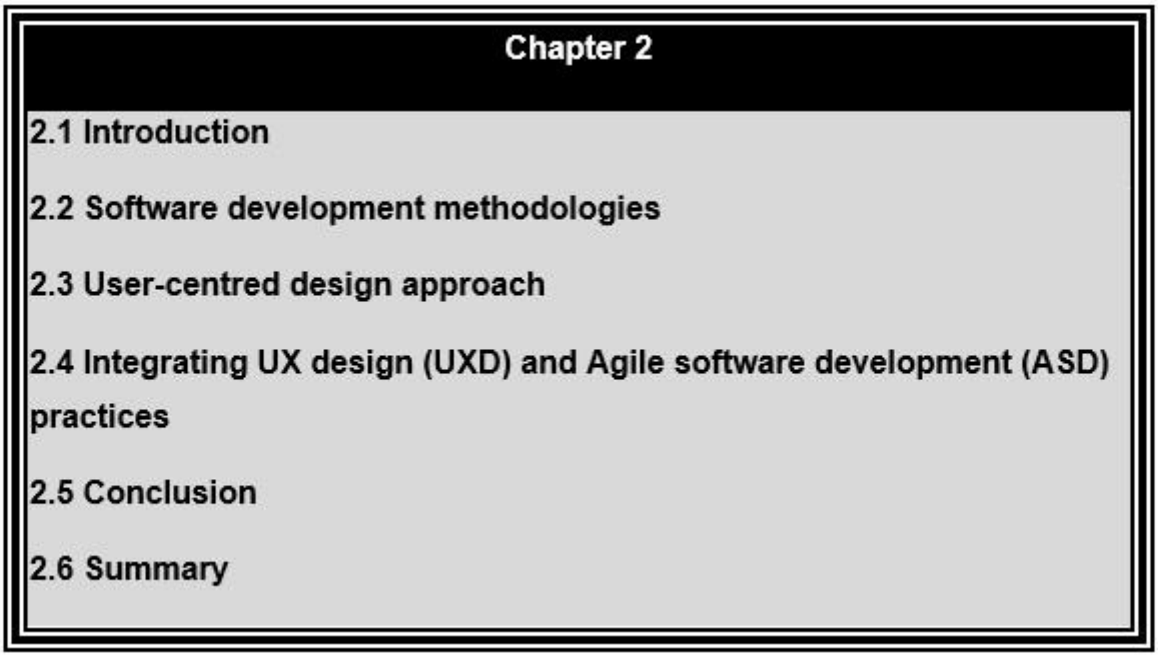
Chapter 7 – Formulation of model

This chapter identifies existing knowledge on the model development process and the approach used to construct the model for this research. Lastly the proposed model is presented.

Chapter 8 – Conclusion

This chapter is an evaluation of the research findings and validation of research objective.

Chapter 2: Software Design Methodologies



Chapter 2

2.1 Introduction

This chapter outlines literature relevant to this study. It focuses on a few of the software development methodologies used within the South African software development context. This is followed by outlining the need for the user centred design process when aiming to achieve a good user experience. Then lastly the chapter covers the related pragmatic work on the collaboration between user-centred design (UCD) and Agile software development (ASD) and further findings into the relationship and characteristics related to the successful integration of UX design (UXD) in software development teams.

The following sections in this chapter discuss software design methodologies beginning with commonly used software development methodologies (SDMs) in the South African context (section 2.2), followed by an analysis of the UCD approach (section 2.3), then the integration of UXD and Agile software development is explored (section 2.4). Lastly the conclusion and summary of the chapter is discussed (section 2.5 - 2.6). This literature review answers the research question:

What best practices currently exist to integrate user experience into agile software development?

2.2 Software development methodologies

The world of technology is one with no boundaries and innovation is inevitable. Innovation in collaboration with conceptualisation develops the best platform for undertaking software development projects. Furthermore, software development methodologies are defined as the structures required to design and monitor the process of developing expert information systems. These innovative processes emphasise what is called the software development process (Jamsheer, 2018).

Saeed et al. (2019) state that striving for effective software methods has existed since 1960. There are several existing software development methodologies. Often, companies adopt the software methodology best suited to develop their software, but generally, the two most popular methodologies are Waterfall and Agile (Joseph et al., 2016). The same authors define waterfall as the traditional method of developing software. This method uses intensive preparation, comprehensive documentation, and extensive design. Agile, however, has gained much more popularity in the software development scene states Saeed et al. (2019). The Agile methodology develops software in small iterative phases, unlike the traditional Waterfall. Agile relies on the team’s skills and knowledge rather than documentation like Waterfall (Pillay & Wing, 2019).

This section discusses these software development methodologies in more detail. It will also take an in-depth look at other existing software development methodologies. The section will discuss their differing characteristics, strengths and weaknesses and lastly propose which one is best suitable for software development projects and why. The purpose of this section is to give an in-depth understanding of why this research study focuses on the Agile software methodology and, ultimately, why it makes sense to integrate it with user experience. A general summary of SDMs is discussed in subsection 2.2.1.

2.2.1 Overview and characteristics of SDMs

Software Development Methodology (SDM) outlines the different tasks performed on a software product from inception to final completion. There are different types of methodologies that can be used to develop software products categorised under Waterfall and Agile (Mahapatra & Goswami, 2015). Since there are different methodologies to choose from, the challenge is to decide which to select. However, the selection is influenced by the organisation and the project features. Mahapatra and Goswami (2015) state that there is no ideal methodology or one that is always best suited. Sub-section 2.2.3 discusses these SDMs in detail.

Saeed et al. (2019) suggest that unsuccessful software development projects are often due to the improper use of SDMs, or not following development theory. The factors to be considered when selecting an appropriate SDM are addressed in sub-section 2.2.2.

2.2.2 Selecting the correct SDM

The initial attempts to identify criteria for selecting SDMs dates back to the 1980s. Boehm and Belz (1990) offered the first theory. Their approach was based on five stages and provided various criteria. As much as this theory was quite ahead of time, it was limited to the Waterfall and spiral models. Furthermore, this section addresses the characteristics of SDMs in a broader context.

Mahapatra and Goswami (2015) state that it is essential to use a cost-effective SDM that has a high success rate. Research by The Standish Group (2015) reports that 32% of all projects were more successful due to accurate, cost-effective delivery with required features and tasks. However, 44% encountered challenges where they were late for delivery, over budget or under-delivered on the required project and tasks. Therefore, the selection of a suitable SDM plays a vital role in ensuring that the software created is delivered on time, cost-effectively and meets the user's requirements. Often, the selection of an SDM can be based on organisational characteristics, the software development team, requirement analysis, user interaction, the nature of the project, and potential risk. The following sub-section (2.2.3) identifies and discusses the SDMs mainly utilised in South Africa.

2.2.3 Main types of SDMs used in South African industries

Joseph et al. (2016) state that unsuccessful software development projects have been an issue for quite a long time. The South African software development sector is no exclusion. The size of South African software development organisations is between small to medium states Joseph et al. (2016). Commonly, most software development teams do not follow a well-processed approach to developing software, and often do not follow any project management approach. This behaviour leaves the South African sector vulnerable to ineffective software development projects. Therefore, it is imperative to find a software development methodology approach that ensures successful software development projects.

Authors such as Sharma (2020) and Saeed et al. (2019) state that SDMs support the software development process by ensuring that software development projects are more operative, risk-free, quality cleared and, ultimately, successful. Even though SDMs run software development projects effectively, there have also been unsuccessful software projects, which query their usefulness. Arguably, some challenges exist in using SDMs emphasises Saeed et al. (2019).

Moyo (2020) states that two types of SDMs exists namely: traditional (which includes the Waterfall method) and Agile. The author describes the traditional SDM's origin as a resolution to a software crisis. It was created to bring stability into the software development process. This process, however, tends to be rigid, continuous and accompanies a great deal of documentation. This documentation serves as a guide for the software development team within the software development cycle (SDLC). Agile on the other hand, is more flexible and has shorter periods between completing tasks in the SDLC. It has subsequently become popular due to its capability to adapt to the changing software development environment, reduced time, and costs to build software Moyo (2020) mentions.

Agile is regarded as the most dominant software development methodology in South Africa, with Scrum being the method used in Agile

Sub-subsections 2.2.3.1, 2.2.3.2, 2.2.3.3 and 2.2.3.4 review the primary used SDMs in South African organisations.

2.2.3.1 Agile software development methodology

Pillay & Wing (2019) state that Agile has been around close to two decades. The Agile manifesto guides the Agile methodology process and comprises the following values: individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan. As mentioned in sub-section 2.2.3, prior to the existence of Agile, traditional methodologies such as Waterfall were used. These methods followed a more strict and progressive approach to developing software. According to Pillay & Wing (2019), Agile methods allow for the incremental development of software. They are flexible, rapid, effective and have reduced cost of developing software. All these factors compensate for the challenges introduced by the traditional methodologies as mentioned in sub-section 2.2.3.

Saeed et al. (2019) and Khoza & Marnewick (2020) highlight Agile as the most popularly used software development methodology in South Africa. Khoza & Marnewick (2020) state that organisations are reaping the results of integrating Agile methods to their organisations. The same authors identified findings from the 2015 Chaos Chronicles that suggest that software projects using Agile as a development methodology are more successful than those that still adopt the traditional ways of creating software. Agile projects were identified as 28% more successful than traditional software development projects. The Agile process starts with defining requirements, development, testing code and lastly deploying the final system to its intended users. See the Agile process in Figure 2.1 below.

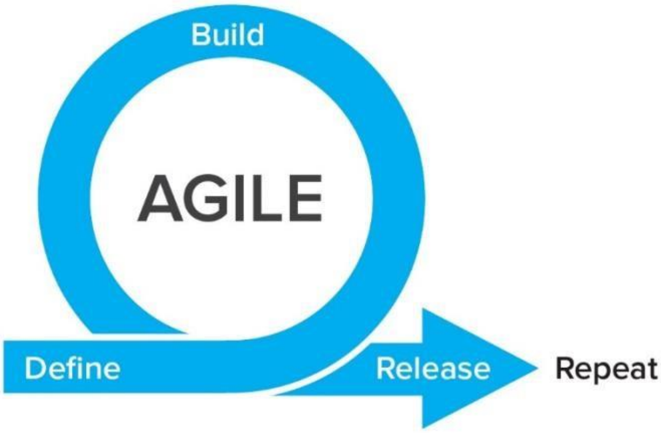


Figure 2.1: Agile Development: quick overview (Ibanez, 2017)

There are various types of existing Agile methodologies. Two of which are discussed in sub-subsections 2.2.3.3 and 2.2.3.4. The traditional Waterfall methodology is discussed in detail in sub-subsection 2.2.3.2.

2.2.3.2 Waterfall model

Moyo (2020) (as highlighted in sub-section 2.2.3) defines Waterfall as a traditional way for developing software. According to Khoza & Marnewick (2020), these methodologies follow a sequence of phases when developing software. This sequence starts with accurately defining all requirements, then the product is developed and tested, and it then goes through the release management and product support process. Figure 2.2 depicts the Waterfall process. The success of software developed using the traditional approach highly depends on clear requirements prior to development commencing.

Khoza & Marnewick (2020) further argue that the adoption of the Waterfall method has introduced challenges such as limited budget costs, project delivery overruns and frustrated users.

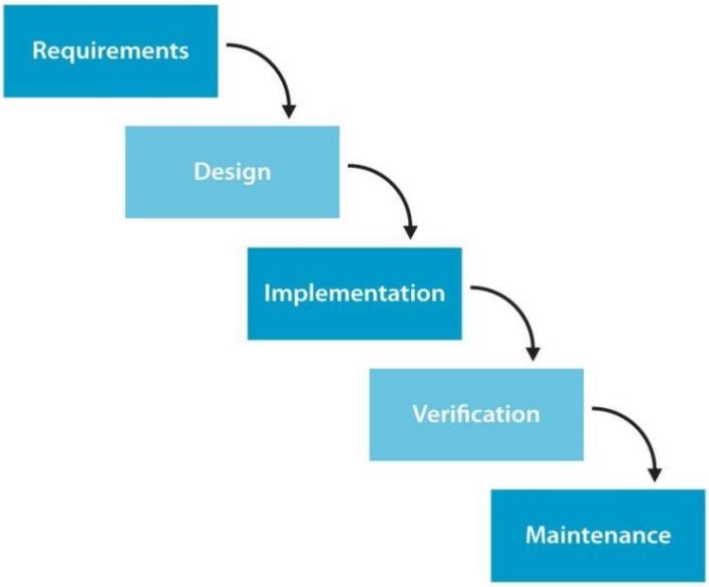


Figure 2.2: Software Engineering Fundamentals (Wordpress, 2016)

2.2.3.3 Rapid application development (RAD)

Rapid Application Development (RAD) is defined by Kissflow (2022) as a development model that emphasises quick prototyping and quicker feedback over lengthy development and test cycles. RAD allows Developers to make several iterations and modifications to software quickly without the need to start the development phase from scratch. RAD was introduced due to the disadvantages of traditional methodologies that hindered developers from modifying software functionality once it has reached the testing cycle. According to Fustik (2017) and Pillay & Wing (2019), RAD has been replaced by Dynamic Systems Development Method (DSDM) which was created as a project delivery framework for RAD. DSDM has progressed since its inception in 1994 (Fustik, 2017) and provides basis for planning, organisation, implementing and scaling the agile process as well as the software development process. The RAD process is outlined in Figure 2.3.

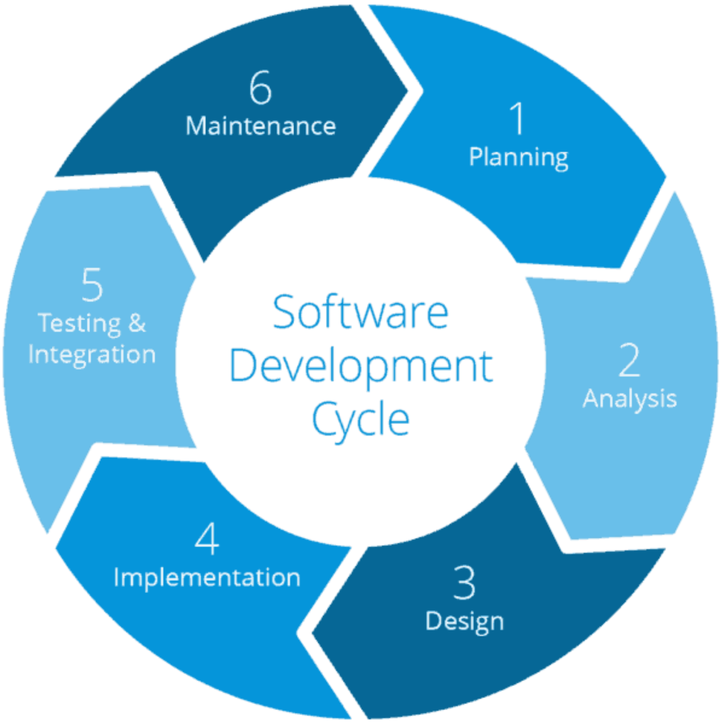


Figure 2.3: Rapid Application Development (RAD) (Mendix, 2022)

2.2.3.4 Extreme programming methodology (XP)

Extreme programming (XP) has been defined by Fustik (2017) as a meticulous technique for delivering high-quality software rapidly and continuously. It supports high customer participation, rapid feedback cycles, continuous testing, continuous planning, and team collaboration to deliver functioning software at regular intervals (e.g., every 1-3 weeks). Ease of use, communication, feedback, and courage are the four values that form the basis of XP. These expand to support the twelve practices depicted in Figure 2.4.

End users leading participation in the software development cycle is key in XP. This methodology is used for creating software in unstable environments. It also creates an opportunity for transparency within the modelling process. The purpose of XP is to decrease software development costs. Communication within the XP methodology takes a transparent, robust, and simple approach (Jamsheer, 2018).

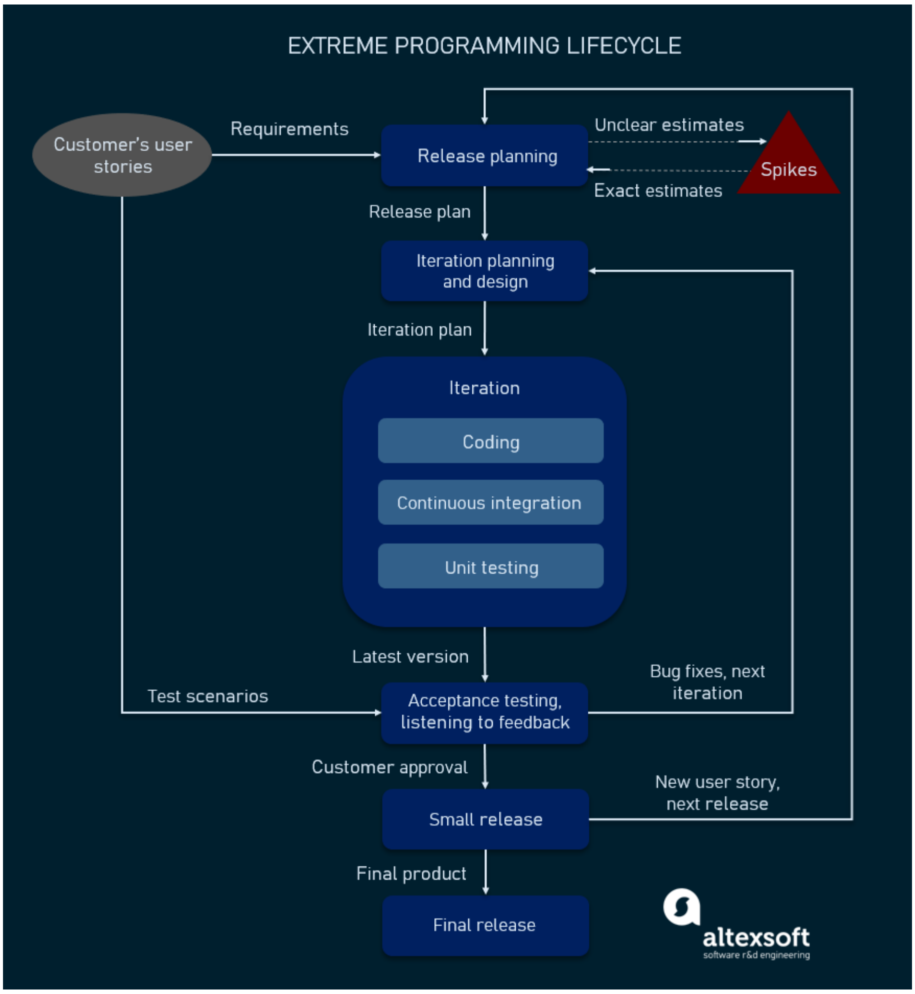


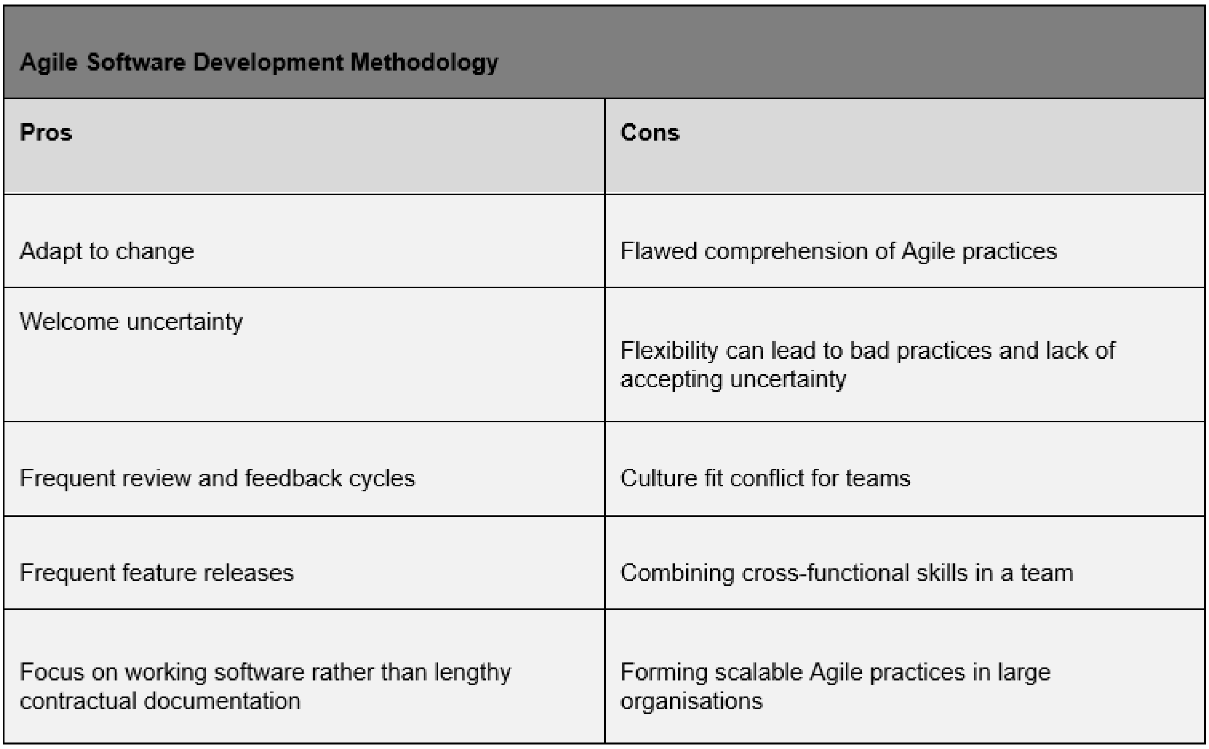
Figure 2.4: Extreme Programming: Values, Principles, and Practices (Altexsoft, 2021)

The following sub-section 2.2.4 identifies some of the advantages and disadvantages of the SDMs discussed in section 2.2.

2.2.4 Software development methodology (SDM) pros and cons

2.2.4.1 Agile

As mentioned in the sub-sections 2.2.3 and 2.2.3.1 above, Agile has shown its advantages against traditional methodologies by adapting to change regarding requirements, technologies, and the market. This is the main advantage of Agile. It can reduce costs of information exchange between parties involved in the software project and the time between making decisions versus the outcomes of those decisions (Moyo, 2020). Determining the project success or failure is influenced by the selection of suitable type of software development methodology states Fustik (2017). Some of the pros and cons of Agile methodologies are depicted in Table 2.1.

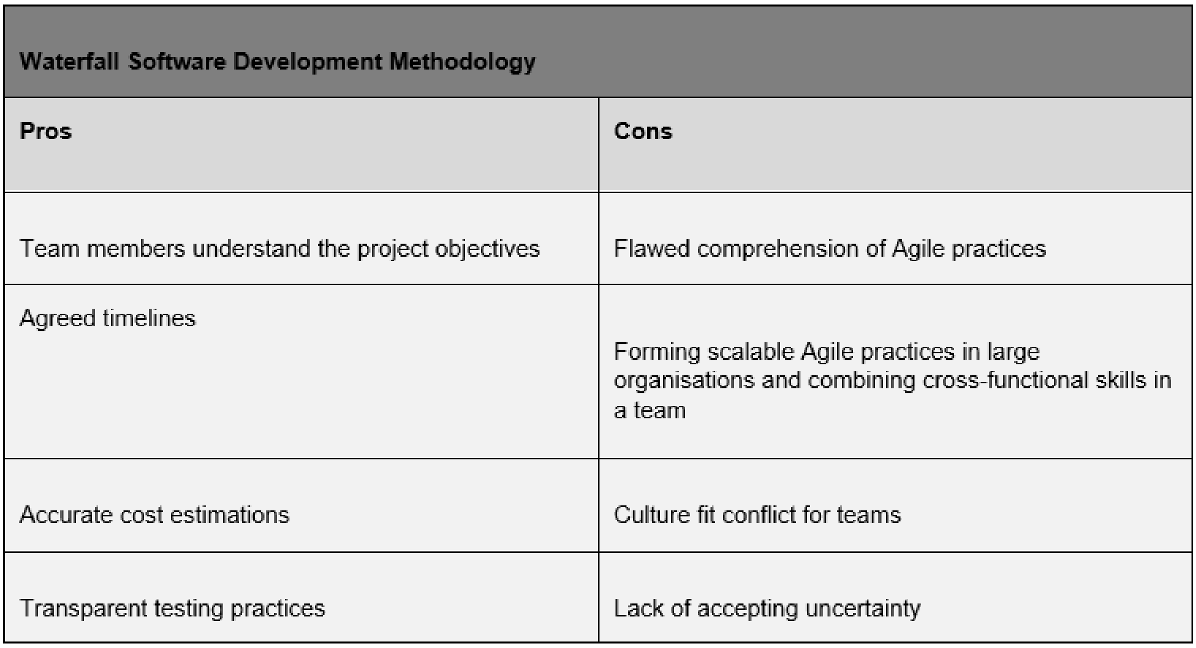


Agile Software Development Methodology	
Pros	**Cons**
Adapt to change	Flawed comprehension of Agile practices
Welcome uncertainty	Flexibility can lead to bad practices and lack of accepting uncertainty
Frequent review and feedback cycles	Culture fit conflict for teams
Frequent feature releases	Combining cross-functional skills in a team
Focus on working software rather than lengthy contractual documentation	Forming scalable Agile practices in large organisations

Table 2.1: The Pros and Cons of Agile Product Development (Gilley, 2021)

2.2.4.2 Waterfall

Kienitz (2017) states that the Waterfall software development model has been around for over 40 years. Waterfall follows a sequential process that starts at the system level and ends with testing (like cascading waterfalls) as highlighted in sub-subsection 2.2.3.2. Some of the challenges affecting Waterfall were mentioned in sub-subsection 2.2.3.2 by Khoza & Marnewick (2020) as limited time and cost as well its lack of flexibility. Table 2.2 lists more of the Waterfall pros and cons.

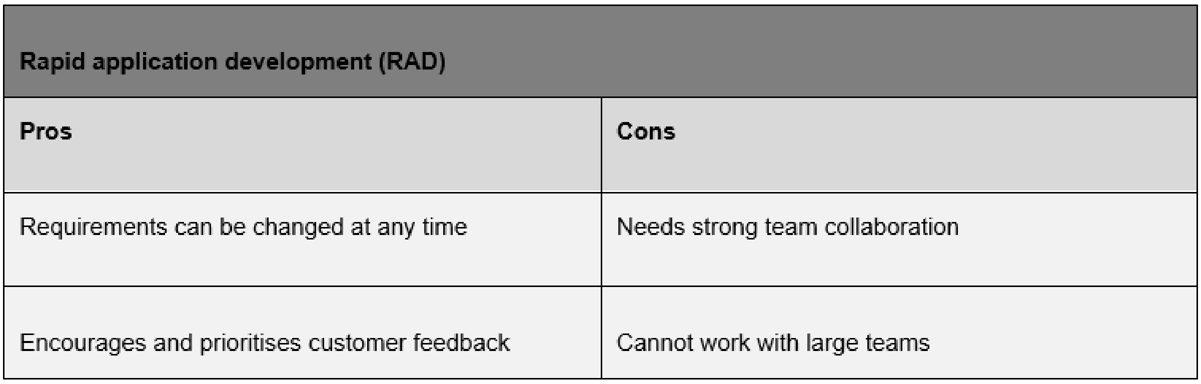


Waterfall Software Development Methodology	
Pros	**Cons**
Team members understand the project objectives	Flawed comprehension of Agile practices
Agreed timelines	Forming scalable Agile practices in large organisations and combining cross-functional skills in a team
Accurate cost estimations	Culture fit conflict for teams
Transparent testing practices	Lack of accepting uncertainty

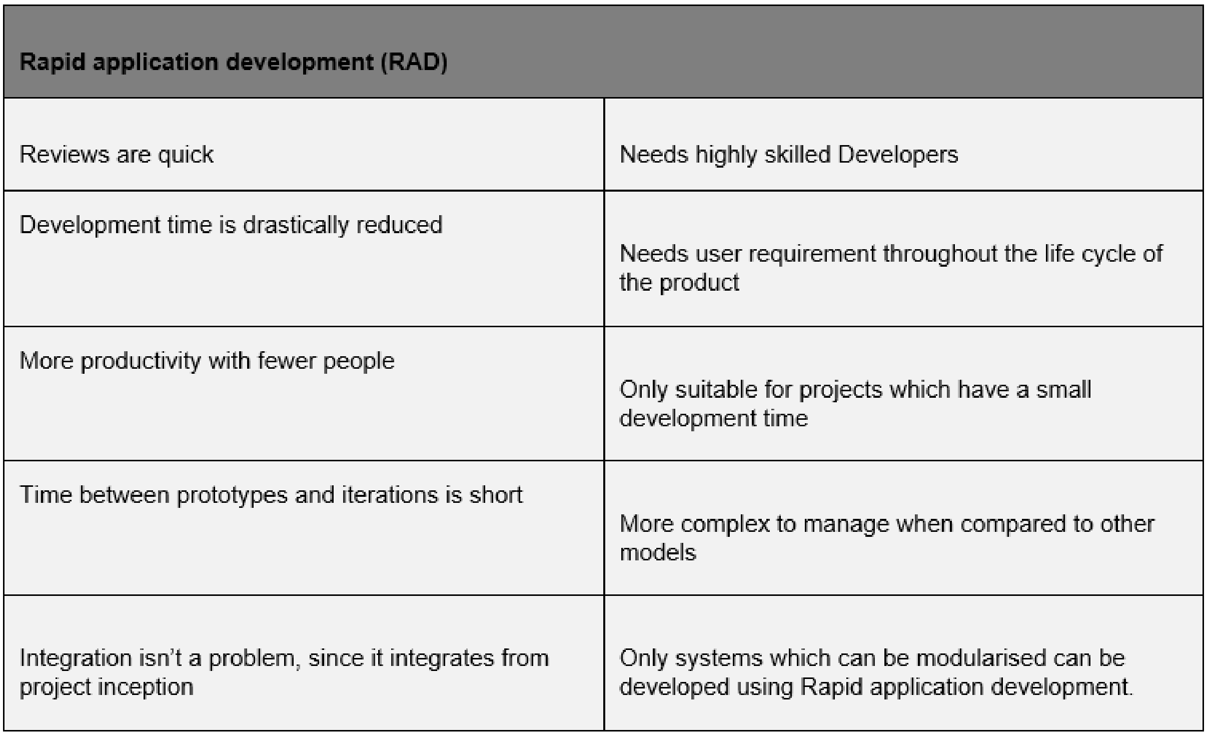
Table 2.2: The pros and cons of Waterfall Software Development (Kienitz, 2017)

2.2.4.3 Rapid application development (RAD)

According to Kissflow (2022), RAD was implemented to speed up the software development process. RAD is meant to reduce the time between the creation of the requirements and when they are implemented, which has caused debate and tension. Kissflow (2022) further suggests that RAD is suitable for small teams and quick projects and is not a solution for all software development issues. Table 2.3 outlines the pros and cons of RAD*.*



Rapid application development (RAD)	
Pros	**Cons**
Requirements can be changed at any time	Needs strong team collaboration
Encourages and prioritises customer feedback	Cannot work with large teams

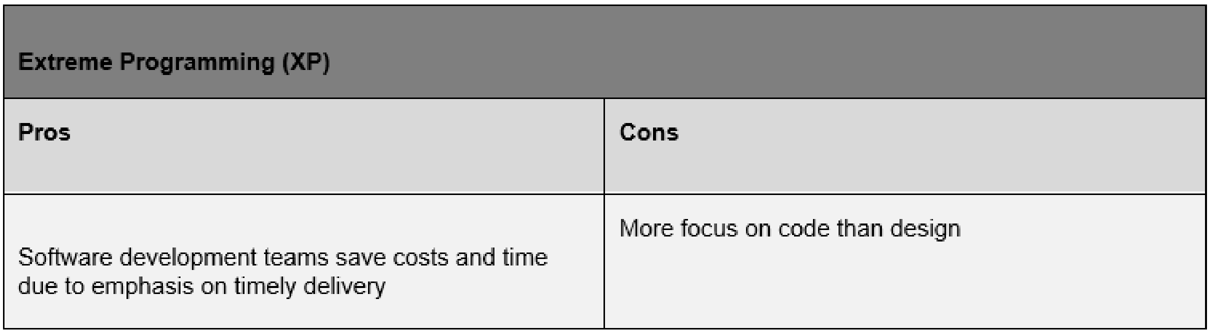


Rapid application development (RAD)	
Reviews are quick	Needs highly skilled Developers
Development time is drastically reduced	Needs user requirement throughout the life cycle of the product
More productivity with fewer people	Only suitable for projects which have a small development time
Time between prototypes and iterations is short	More complex to manage when compared to other models
Integration isn't a problem, since it integrates from project inception	Only systems which can be modularised can be developed using Rapid application development.

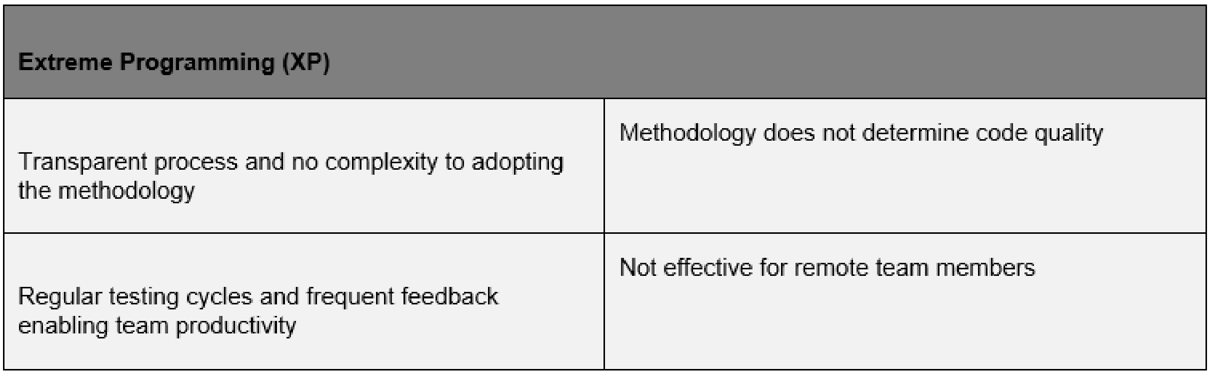
Table 2.3: Advantages and Disadvantages of Rapid Application Development (RAD) (Kissflow,2022)

2.2.4.4 Extreme programming (XP)

Kukhnavets (2018) states that one of the most important XP principles is high customer collaboration with the software development process as well as communication with the team itself. Table 2.4 highlights the pros and cons of adopting the XP software development methodology.



Extreme Programming (XP)	
Pros	**Cons**
Software development teams save costs and time due to emphasis on timely delivery	More focus on code than design



Extreme Programming (XP)	
Transparent process and no complexity to adopting the methodology	Methodology does not determine code quality
Regular testing cycles and frequent feedback enabling team productivity	Not effective for remote team members

Table 2.4: Advantages and Disadvantages of Extreme programming (XP) (Kukhnavets, 2018)

Sub-section 2.2.5 outlines which SDM is regarded as the most suitable.

2.2.5 Which methodology is better?

In the sub-section 2.2.3 the different types of SDMs used in South Africa were discussed to find the most appropriate one for implementation in software development projects now and in the future (Moyo,2020). Based on the findings of sub-sections 2.2.3 and 2.2.4, Agile methodologies have been found to be the most widely used with a higher project success rate. According to Khoza & Marnewick (2020), project success is the most important aspect of project management. Project success can be influenced by software quality, project timelines, user satisfaction and budget compliance. These factors are accounted for in Agile methodology. These findings, provide an indication that Agile software projects are and should be more successful than software projects adopting traditional methodologies. The following section (2.3) discusses the User-centred Design (UCD) approach.

2.3. User-centred design approach

2.3.1 Background

Zorzetti et al. (2020) define user-centred design (UCD) as a set of events, processes and methods that focus on placing the user at the centre of the design process and development process allowing developers to realise user needs and build value-added software with enhanced usability. Furthermore, Signoretti et al. (2020) validates these findings by highlighting UCD as a process that aims to provide tools and techniques for developers to enable them to explore and understand user expectations while anticipating for greater project outcomes.

Each iteration is part of the design process. In the first phase, UX Designers work in groups to try and get the users' perspectives of how they may want to interact with the system or product. The user requirements are derived from the previous phase. This is followed by designing the solution and then the assessment phase commences. The assessment phase verifies if the design complies with the users' needs. In this phase, the design should mimic and satisfy the users' specific requirements and perspectives. From this point, the design team will make additional iterations of these four phases and continue assessing the design results until they are satisfied (The Interaction Design Foundation, 2022). See UCD process in Figure 2.5 below.

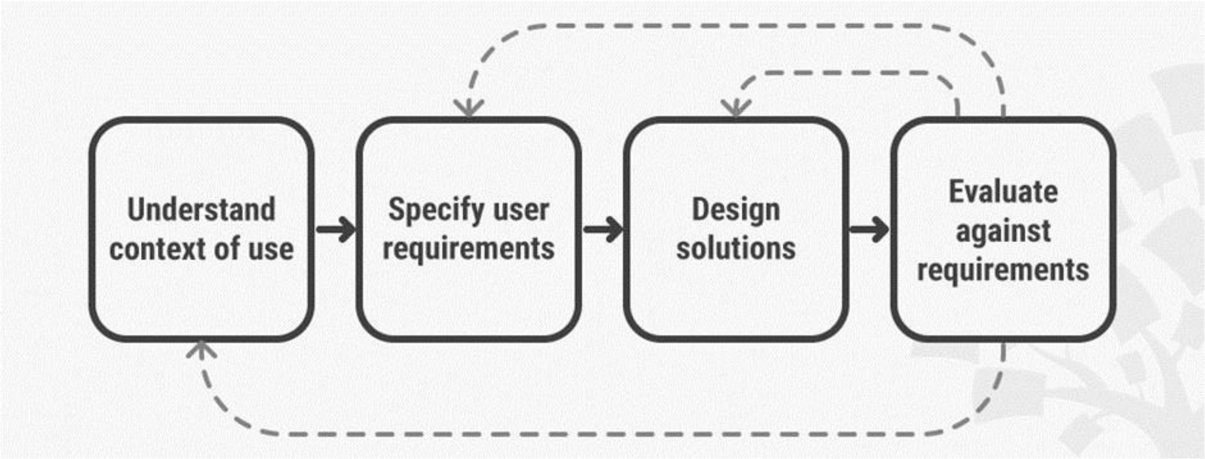

Understand context of use
Specify user requirements
Design solutions
Evaluate against requirements

Figure 2.5: User-Centred Design (The Interaction Design Foundation, 2022)

The Interaction Design Foundation (2022) argues that the benefit of UCD is involving the user at every stage of the UCD process as a great way of discovering aspects that work effectively and those that do not. Furthermore, this enables the team to understand the reasons for any of these outcomes. Sub-section 2.3.2 discusses UCD principles in detail.

2.3.2 Characteristics

UCD principles allow UCD to focus on the users throughout the UCD and software development process, and principles that govern UCD. This section presents a list of identified characteristics that are required for the implementation of UCD to meet collaboration needs. Babich (2019) identifies the following UCD principles:

User-centred designs which can be mapped to user tasks

It is imperative to involve users when designing and developing software to ensure that what is developed is what is required. The system or product should be user-centred and task-oriented. This must be taken into consideration during the software development process. UX Designers should consider the features or characteristics of the user population, environment and the tasks involved for the system or product being created.

Preserve Reliability

The user requires a simple system that is easy to understand. The performance of the system components should be consistent. This needs to be defined in the design phase through to implementation and usage by the user. The design approach is important as it defines and influences how users will view and access the system. It is advised that the approach be assessed as well as the period taken to learn how to use it.

Utilise plain and unambiguous information flow

A system should be designed in a way that is easy for users to interact with and to enable dialogue with the users. Users should only see significant information related to a tasks' successful completion. Where irrelevant information is displayed, users are left in a confusing and frustrating state. The terminology is to be defined in an unambiguous manner.

Decrease redundant psychological user effort

Users like to focus on the task at hand and not have to concern themselves with system design, use and interaction. In other words, they should not be concerned about learning how to interact with the system as that can cause issues and be inefficient. Therefore, instructions on how to use the system should be clearly defined for the user.

Provide suitable feedback

Users like to know when the action they have performed has been done successfully. This can be achieved by an alteration in the appearance once the action has been successfully completed. If the action takes longer than expected to complete, then that should also be indicated to the user. This gives users assurance and a feeling of reliance to use the system. However, the integral issues and status of the system should not be revealed to the user.

Provide user-friendly navigation

Ensure easy navigation through the system by implementing efficient and consistent tools that allocate titles to screens, and display indicators such as page numbers and scrolling bars.

Allow user to take control

As users already know what they want and need for a system or product, it is up to the development team to build the solution. The solution should, therefore, mainly cover the user requirements.

Information displayed clearly

All information displayed in the system should be plainly presented. The information should be displayed in a way that users are able to differentiate between all the components on the screen or product. This can be done by utilising boxes, spaces, and visual coding expertise. Providing more information than required should be avoided.

Offer user support

The system should be easy to understand. Furthermore, users should be able to acquire assistance by interacting with the system with minimal referral to documents or frequently asked questions (FAQs) provided, e.g., labelled buttons or tooltips.

Quality assurance

Minimise issues in the system by navigating users through the right path to attain their objectives. Keep user feedback at a minimum, where required, for user tasks. Furthermore, this should not prohibit users' choices to attain their objectives. Emphasise system data validation of input parts. Validation messages should be as clear as possible.

The following sub-section (2.3.3) explores the user experience (UX) and user-centred design space and the necessity to place users at the centre of the design process.

2.3.3 User experience (UX) and user-centred design (UCD)

Chaione (2017) states that user experience (UX) and user-centred design (UCD) are often confused and considered to have the same meaning. Indicating that content created by integrating UX and UCD gives a false outlook on the UX and UCD sector. This situation makes it difficult for organisations to sell products or services as they must clarify how the two concepts work before selling. However, it would be beneficial for these concepts to be clearly described in the sector so that everyone is clear.

UX is a concept concerning users prior, during and after interacting with a certain product or service. UX is normally a combination of usability, effectiveness, emotional influence, and relevance states Hartson & Pyla (2019). The Interaction Design Foundation (2022) identifies UX as a process used by design teams to build products that offer significant experience to users. The popularity of this concept is derived from an organisation's concern to understand users' experiences with their products or system with the goal to develop improved ones.

UCD has already been described in sub-section 2.3.1 as a collection of processes and approaches regarding psychology and human factors to create a strategy for developing the user experience that organisations desire their users to have.

According to Chaione (2017): “The point of UCD is to 1) understand the current UX, and 2) produce a better one that benefits the stakeholders.”

Additionally, Noman & Draper (1986) present UX (as cited in Justinmind, 2021) as a philosophy that changed the way humans and computers think. This design intended to vigorously include users at every step of the design process for more operative, proficient, and innocuous systems and products.

The UX design process is used to create products that provide meaningful and relevant experiences to users. The process involves creating and integrating the product and focuses on elements of UX such as branding, design, usability, and function. Creating overall user experience includes elements relating to the product itself such as marketing, product packaging and customer services/ support. It is critical to design products that are useful, therefore UX should address the needs and difficulties of intended users. There is no single definition of a good user experience. A good user experience is one that meets the needs of its intended user in the specific context that they use the product (The Interaction Design Foundation, 2022).

Additionally, The Interaction Design Foundation (2022) states that UX design should be user centred. Designing for human users means working with an intensive scope regarding accessibility and considering potential user’s physical restrictions. The tasks of UX designers can differ from organization to organization. These may include research, creating personas, designing wireframes and prototypes and testing designs. However, UX designers are encouraged to be the champions for users and keep user needs at the centre of the design process. In such a process UX designers can make informed decisions that address all relevant issues and user needs. UCD is a repetitive process that allows you to encompass the perception of the intended users and their circumstances as an entry point for all design and development.

Allabarton (2022) states that the UX design process consists of 4 steps which are user research, design, testing and implementation. These steps are defined further below and highlight how the user is involved throughout the process:

User research: This is the starting point of the UX design process as it is critical for UX tasks not to be based on assumptions that are not the goal of the intended users. User research provides data for the entry point to build the product. The UX design process involves the user through user research using the following methods:

One-on-one interviews: In-depth one-on-one discussion between interviewer and intended user. The discussion is intended to identify the underlying needs and requirements of the user when using the product.

Online surveys: Questionnaire form consisting of set questions sent to sample intended users over the internet. The length of the form may differ from project to project. Data collected will be reviewed at a later stage.

User testing: Users interact with the product. This is a valuable way to identify intended user's difficulties with the product.

Design: This is where wireframes and prototypes are created which will provide a tangible item to use when testing with intended users. It is crucial in ensuring that the designs are functional and usable.

User testing: A fundamental part of the UX designers' profession and a crucial part of the entire UX design process. Testing is done to eliminate issues or difficulties that were missed in the design step before moving into the implementation phase. Testing must be done with the intended users.

Implementation: The UX designer is responsible for implementing (by handing over designs to Developers) and championing the UX design process. This may differ from organization to organization. The UX steps mentioned above should be revisited and repeated in the UX design process as you continuously improve the designs.

The following sub-subsection (2.3.3.1) argues why users are prioritised during the design process.

2.3.3.1 Why should the design process be user-centred?

User-centred design allows everyone to be part of the design process. From the users' viewpoint, it is the variation between concluding a task or not. Developers view it as the success or failure of the software system or project entirely. Lastly, it saves businesses time and money and increases satisfaction (Justinmind, 2018).

In conclusion, there is a difference between the UX and UCD concepts. UCD strategies can differ depending on the project objectives.

Sub-section 2.3.4 below explores SDMs and the factors necessary to support UCD practices.

2.3.4 SDM requirements to support UCD

SDMs play an important role in developing software. They form a basis for planning and managing software development processes (as mentioned in sub-section 2.2.3). For this research study, it is vital to identify SDMs that support UCD. These will be discussed in this sub-section.

Agile methods have been established as the most popularly used methodologies in software development. As Agile follows a continuous, transparent, and test-driven strategy, organisations are gaining confidence in using it to build systems that are effective, efficient and make a good impact on its users. As one of its principles is collaborative customer focus, this methodology is expected to support and successfully collaborate with UCD (Novoseltseva, 2019).

According to Kumana (2009), many Agile methodologies recognize the benefit of the involvement of users as experts in the Agile software development process. These are some of the requirements for Agile methods to support UCD described in the subsection below:

2.3.4.1 Extreme programming

XP expects the user to be part of the team continuously. Users should interact with the created software at least twice in the process to provide feedback before the software is deployed.

2.3.4.2 Scrum

This Agile methodology holds regular feature demonstrations for stakeholders at least once each sprint iteration is complete.

However, users cannot be available to participate in the software development process daily. The users that do participate do not make up the whole population of users who will be using the software product. Therefore, it is rather impractical to want users to be part of the team. Agile teams' need to find ways to extract information from users and apply it to the rest of the process in order to ensure quality software is built. One way is UCD, by meeting the following requirements:

Build the right software product by understanding user needs

Involve the users early and throughout the software development process

Deliver software products on time

Maintain the software product after deployment

The following section (2.4) presents requirements needed to combine UX and Agile practices successfully.

2.4 Integrating UX design (UXD) and Agile software development (ASD) practices

This section discusses the acknowledged characteristics for integrating UXD and ASD. Sub-section 2.4.2 identifies current day UX methods and lastly factors required to select an appropriate Agile-UX method are discussed (sub-section 2.4.3).

Laubheimer's (2017) research findings indicate that Agile methodologies have advantages that allow the team to be more effective. These are: follow a continuous and iterative approach, transparency, allow for adjustments to features and can be built continuously and deployed in short timeframes. Laubheimer (2017) also emphasises that Agile focuses more on the Developers and not the rest of the software development team, as they are building the software product. Furthermore, the Agile manifesto (that governs Agile methodology) does not embrace UX, nor does it consider the time and tasks required to complete designs for software development.

Pillay & Wing (2019) elaborate that some of the essential characteristics of similarity between UX and Agile is that they equally depend on changing requirements continuously spanning a sequence of iterations, continuous development for quick delivery of software, and close relationship with the customer.

Reduced user testing time for user testing and less prompt planning were similarly identified by Pillay & Wing (2019) identified as additional vital characteristics.

Laubheimer's (2017) findings also reveal that UX Designers who work in Agile teams have prospered through the challenges experienced through the collaboration of the two practices by embracing Agile principles. In sub-section 2.4.1 below are additional characteristics identified by Laubheimer's (2017) to integrate UX and Agile practices.

2.4.1 Characteristics to integrate UX and Agile practices

Superiors and organisation leaders should understand the benefit of UX

Various successful Agile-UX experts have agreed that the success of UX is based on the organisational superiors acknowledging the importance of embracing UX into their Agile software development projects. Fundamentally, they need to understand what it is and why it is needed. It should also be understood that it is important for UX Designers to be included in the initial step of the software development process. Superiors should also understand that software products are created with users at the centre of it all. Therefore, user research is essential to understand user needs and requirements. They should also determine what the UX Designers will need to be able to do their tasks successfully.

UX Designers should possess leadership qualities

UX Designers need to be able to lead in Agile. They must have the confidence to point out expectations about user actions. Additionally, they must actively collaborate and communicate with team members to understand how the processes function for successful projects. They must invite team members or colleagues at large to user testing tasks so that they can understand UX and user behaviour. UX advantages that succeed in Agile probe the team to understand the overall objective. Similarly, allowing users to test the products during the development process ensures that they will use them once they have been completed.

Flexible Agile development process

Agile is a method based on transparency and adapting to change. Laubheimer (2017) mentions that Scrum is commonly the approach used to implement Agile in most organisations. This is because Scrum enables Agile conversion through procedural standards and recurring team meetings, and clear terminology. Unfortunately, Scrum cannot work well with UX as the latter was not originally part of the Scrum process. UX works successfully in Agile when the Agile process is not too stringent and regulated. It works well when the Agile environment can adapt to change rather than the length of a progress meeting. Teams need to figure out how to manage UX tasks successfully to start working on user stories or tasks before the Sprint commences. **UX Designers and Developers work collaboratively in the same team**

The most effective Agile teams are those that work collaboratively while understanding the project goals. UX Designers should be included in the software development team. Commonly they are treated as consultants who work in isolation and are only included when required. This takes away the collaboration and communication elements of working in a team that ensures that everyone is aware of the progress of the development of the software product. Furthermore, this decreasingly chips away at the trust element of the team. Both UX Designers and Developers should earn each other’s trust by consciously validating design concepts and improvements and communicating their consistency to the rest of the team.

Despite the limitations, UX and Agile can successfully integrate provided that the organisation understands and supports UX efforts. UX Designers should show leadership qualities and work collaboratively with Developers and the rest of the software development team. Lastly, both practices should account for adapting to change and ensure successful user-centred Agile software development.

The following sub-section (2.4.2) looks at the existing methodologies used to integrate UX and SDMs.

2.4.2 Existing UX software development methods

2.4.2.1 Waterfall UX

Arasu (2019) identifies Waterfall UX as a method of combining software development and UX. The Waterfall methodology is consecutive and has no adaptability. In this process, UX Designers follow a set timeline and scope. Typically, there is no room for trial and error. The project cost is normally limited. This approach (displayed in Figure 2.6 below), therefore, is more prone to restrictions, specifically if there is a shift from the set tasks within the process. Enlisting all the required stakeholders at the initiation of the project is encouraged.

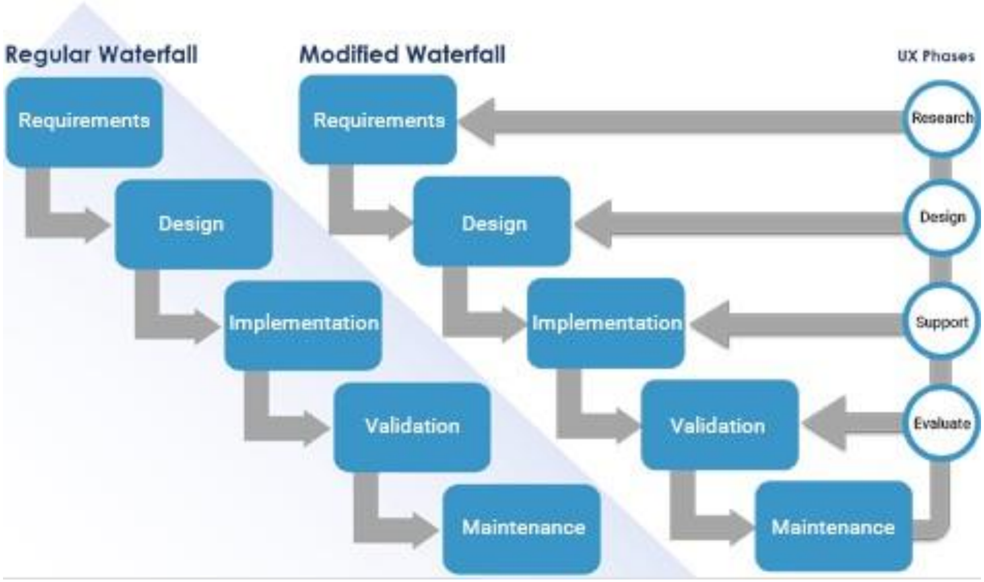


Figure 2.6: Modified Waterfall model based on Semantic Scholars Waterfall model (Arasu, 2019)

It is encouraged for this method to be used when working on short-term projects with a fixed cost and timelines. The software development team should also be quite skilled with the latest technology. However, the project scope should be broad and set with constant requirements.

2.4.2.2 Lean UX

Gothelf and Seiden (2016) define Lean UX as the most used method for interaction design suitable for Agile software development teams. It emphasises how Agile teams can successfully integrate design, research, continuous development, and training through a set of defined principles. Lean UX focuses on ensuring that users have the most appropriate and satisfactory experiences when interacting with software products rather than focusing on deliverables. Through Lean UX, teams learn how to embrace design over continuous short development iterations and work for themselves and the organisation. The team is encouraged to identify the problem and form concepts of how to resolve it, for example, objectives and how to meet them.

Furthermore, the team must incorporate the UX Designer's thoughts early in the process. The team is also encouraged to create a Minimum Viable Product (MVP) to decide which ideas or concepts are more efficient and effective for users. This will ensure team and project productivity. With Lean UX, UX is integrated into the Scrum process, ensuring that the organisation adapts to accommodate the integration of the Lean UX method.

Lean UX has been created specifically for UX Designers who realise they can add value within their software development teams. It should also encourage Product Managers to develop their products with their users at the centre. Developers need to understand the importance of integration between team members to achieve project objectives. Lastly, the organisation's superiors need to understand the value UX brings to the organisation. Ultimately, it is safe to say that Lean UX is for the entire Agile software development team (See Figure 2.7 below).

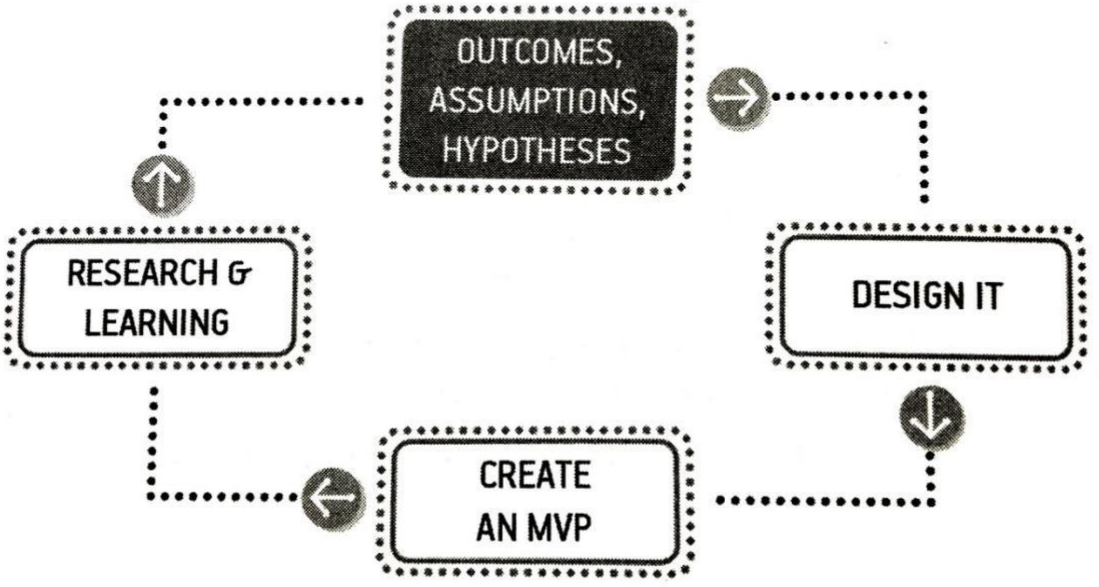


Figure 2.7: The principles and process of Lean UX (Pollard, 2018)

2.4.2.3 Agile process of human-centred design (HCD) and software development

This method looks at the advantages of the Scrum Agile methodology and the HumanCentred Design (HCD) process and creates an approach based on the collaboration of the two processes.

Scrum has already been defined in the sub-section 2.4.1 above. However, Forbrig and Herczeg (2015) illustrate the Scrum process in Figure 2.8 below which consists of a 30-day sprint starting from backlog refinement, sprint planning and software development iterations:

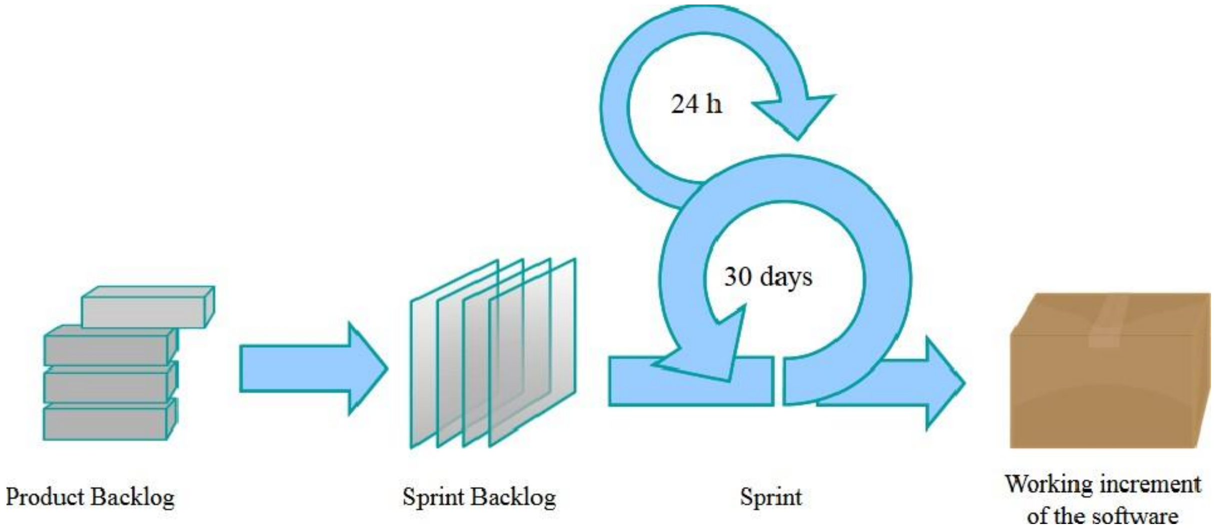


Figure 2.8: The Scrum process (Forbrig & Herczeg, 2015)

Human-Centred Design (HCD) is commonly known for usability and user experience. The fundamental reasons for HCD success are perspective of practice, user requirements and design assessment. User requirements take preference over technical requirements, which Developers might find more important. Users get exactly what they want through HCD. The HCD process is governed by ISO 924-210. See Figure 2.9 for process steps below. The process starts with planning, clear requirements, finalise user requirements, present design solution and evaluate design against requirements.

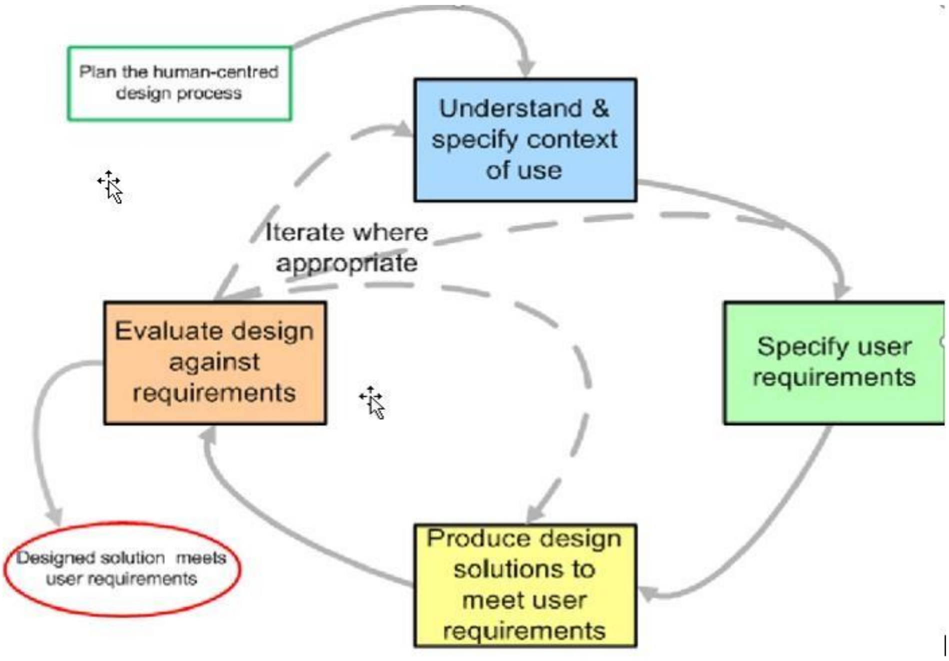

Plan the human-centred design process
Understand & specify context of use
Iterate where appropriate
Specify user requirements
Evaluate design against requirements
Produce design solutions to meet user requirements
Designed solution meets user requirements

Figure 2.9: The design process from ISO 924-210 Human-centred design process (Forbrig & Herczeg, 2015)

Sub-subsection 2.4.2.4 explores the concept of integrating HCD and Scrum approach.

2.4.2.4 Combining HCD and Scrum

The development process commences with project requirements gathering and analysis. According to the project objectives, similar needs are highlighted and happens prior to the Sprint commencing. Business Analysts, or whichever team members have been tasked with the analysis role, do this task. However, Developers can also perform this task and are encouraged to be part of this step.

According to the identified needs, requirements should be defined and added to the project backlog. These requirements will be used in the following Sprint. The Sprint should be managed in parallel to the following cycle of identified project requirements and the backlog (Scrum process) should increase by including these. This method follows the HCD process. Executing this process ensures that the software built has the required benefits, is usable and provides the required user experience. See Figure 2.10 for the detailed process:

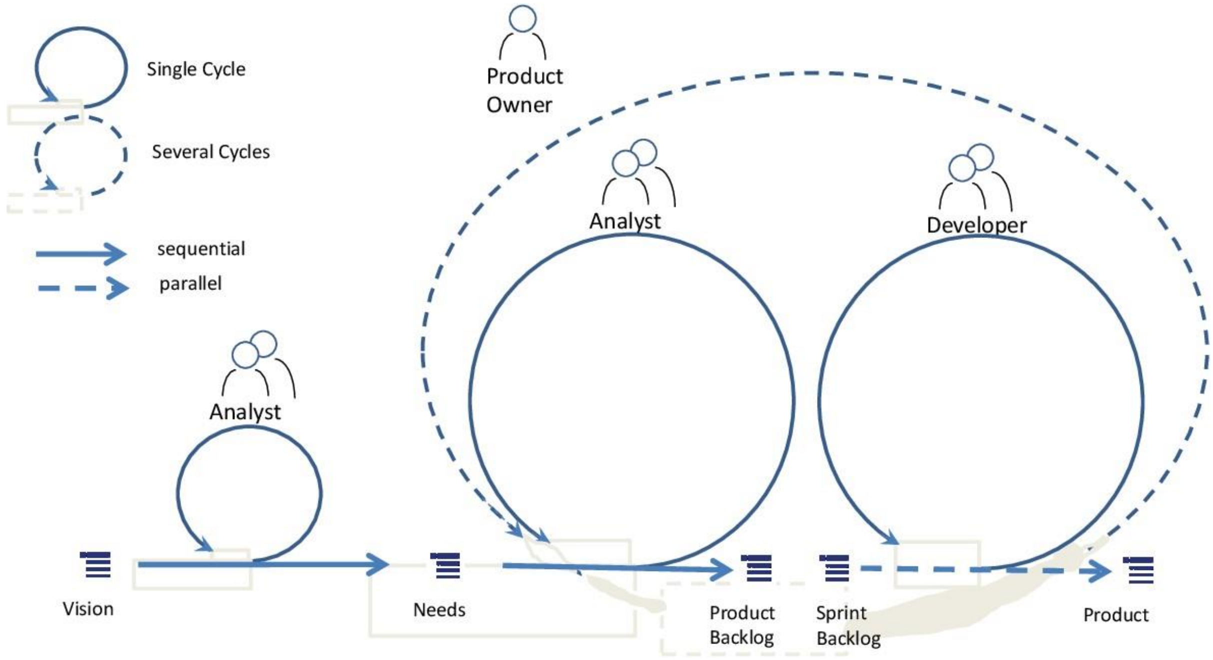


Figure 2.10: Suggested HCD process for Scrum (Forbrig & Herczeg, 2015)

After identifying the requirements, there must be an approach on how the tasks to be executed in the upcoming HCD process embrace the requirements gathering and analysis. The current position and perspective of use are assessed and signified by various models. Then the requirements are specified and later weighed. This is done before the solution is created. After identifying the requirements, they are added to the backlog later. There is less focus on the Scrum process because the process approach is isolated from the details of these models. Each Agile cycle can be sustained. See Figure 2.11.

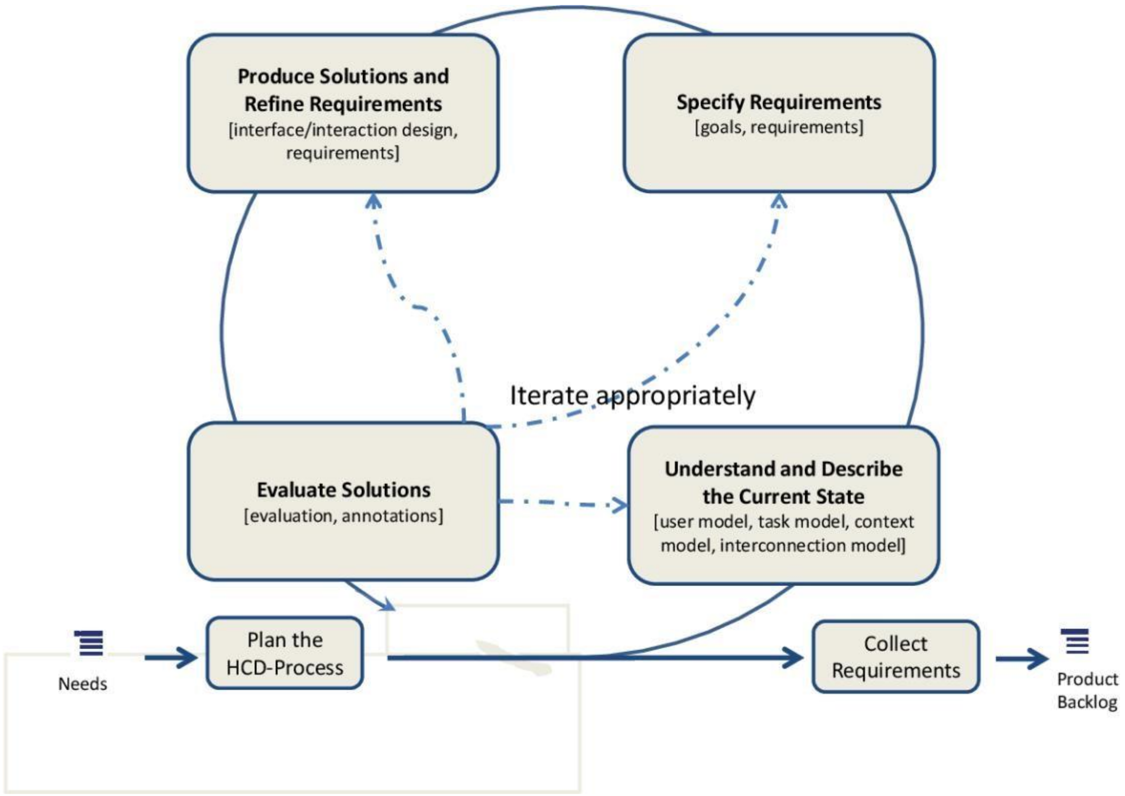

Produce Solutions and Refine Requirements
[interface/interaction design, requirements]
Specify Requirements
[goals, requirements]
Iterate appropriately
Evaluate Solutions
[evaluation, annotations]
Understand and Describe the Current State
[user model, task model, context model, interconnection model]
Needs
Plan the HCD-Process
Collect Requirements
Product Backlog

Figure 2.11: Detailed HCD process (Forbrig & Herczeg, 2015)

The following sub-subsection 2.4.3 presents a discussion on the requirements to be considered when selecting the most relevant Agile-UX approach.

2.4.3 Selecting an appropriate Agile-UX method

Arasu (2019) states that determining the most suitable Agile-UX approach to adopt may be quite a challenge due to conflicts between product requirements and software products built for organisations generally. This is particularly relevant if there is no framework or process to govern the influence of the software development teams on the product release. Conflict occurs when there is a discrepancy between the created product and the user requirements, and this is detrimental to the success of the team.

The Agile-UX team should ensure that the framework adopted guarantees continuous development and deployment and is simple to learn. Often, UX Designers are also uncertain about their UX processes. To decide on the most suitable UX processes, UX

Designers should ensure that they are sure about any insecurities that they may have about their processes and understand their Agile teams' approach to the SDLC.

Teams should also understand that software development is not only about building feature-based software and assessing its usability. It is also about useful and appealing software products that meet user needs. The most suitable UX process for a team should ensure continuous delivery, research and design, user involvement, team management, the right skills and knowledge, tractability, influence, and innovation.

Arasu (2019) furthermore identifies important questions to ask when selecting an approach:

How consistent are the requirements or objectives?
Who is the software product being built for?
What is the project scope?
Is the project cost constant or operational?
Where is the location site of the project team members?

Lastly the team needs to modify the identified approach to meet their needs. They can start by using a more accustomed approach and then modify it to fit the team objectives.

2.5 Conclusion

Various software development methodologies exist from Waterfall to Agile. A software development team will select the most suitable methodology based on the organisation type, project requirements and type of software development team. It has been established in this chapter that UX works much better when combined with Agile methodologies, as Agile is considerably more flexible and allows for continuous development. UX experts often work in isolation and there are, thus, difficulties when integrated into Agile software development teams. Ultimately, the superiors in software development organisations need to understand the value of UX in the organisation.

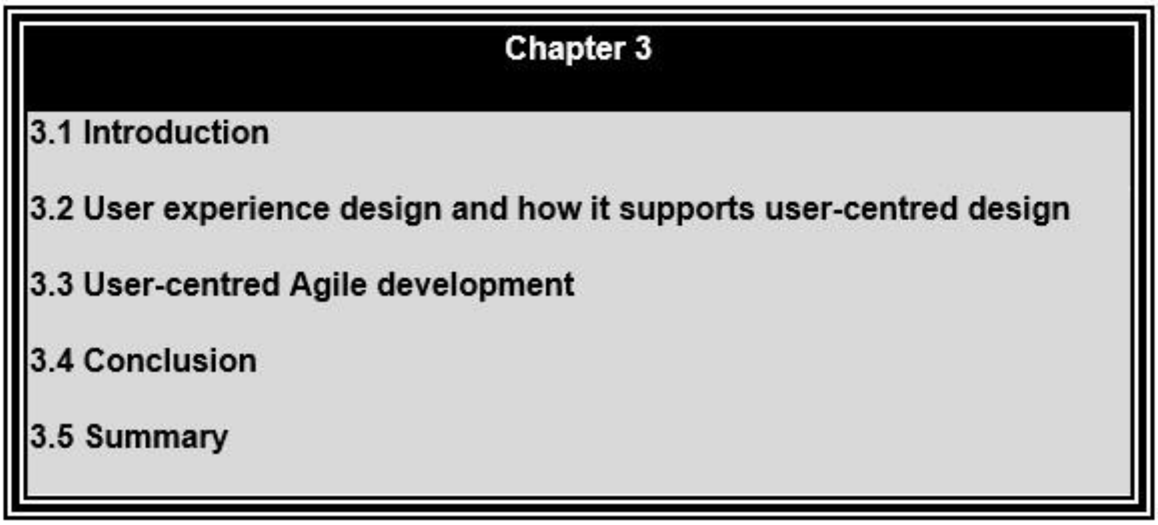
The team therefore needs to decide on the best approach for the two practices to be able to function effectively and cohesively.

2.6 Summary

This chapter explained existing software development methodologies in section 2.2, user experience design techniques in section 2.3 and the relevance of integrating UX design and software development in section 2.4. This research study takes the stance that it is imperative to integrate UX processes into the software development process by allowing users to get involved in the process through a sequence of tasks including research and design to develop high quality and adaptable software products as mentioned in sub-section 2.3.3. Sub-section 2.4.1 explored the benefit of combining UX and Agile practices is a software product or service created that will be reasonably useful for users and meet organisational objectives. To achieve this, the team must understand user needs and project scope. It must also ensure that it is well equipped with skills and knowledge and is willing to embrace change. It must also ensure product or service quality, adopt a model-driven approach, and have stakeholder buyin.

Furthermore, existing UX software development methods are highlighted in subsection 2.4.2 and lastly factors for selecting the appropriate Agile-UX method are explored in sub-section 2.4.3 to guide software development teams. The following chapter explores the user-centred Agile development concept.

Chapter 3: User-Centred Agile Development



Chapter 3

3.1 Introduction

This chapter presents a comprehensive study of current literature that identifies ways Agile and UX can be integrated. The chapter explores each process cycle (sub-section 3.2.1, 3.3.1 and 3.3.2), benefits (sub-section 3.2.2, 3.3.4 and 3.3.6), potential challenges (sub-section 3.2.2 and 3.3.6) that have been identified in the literature with the aim of addressing the following research questions:

- ***What challenges exist for integrating user experience into Agile software development?***
- ***What is the impact of integrated UX practices in Agile software projects?***

In the following section, the literature has been categorised initially to discuss the context of user experience and ways in which it relates to user centred design. This entails describing the UX process, benefits and limitations and the key factors of user experience design.

3.2 User experience design and how it supports user-centred design

Pillay & Wing (2019) state that the term user experience dates back to the 1990's. Lodato (2015) defines UX using the two words that comprise it: *user* and *experience*.

The same author argues that UX unites two vital concepts; the *user* who indicates the suggested person that will interact with the product or service. The second is *experience* which is meant to indicate the suggested or hypothetical character of interacting with the product or service. Stevens (2022) agrees and states that user experience is based purely on the users' experience. UX Designers need to understand the meaning of what kind of 'experience' a user desires. Additionally, Usertesting (2019) also defines user experience as the practice used to define the experience between users and the product or service.

Tran (2019) states that UXD describes what is designed, the experience or interactions users have with the product or service, while UCD describes the process itself. In practice, most UX Designers attempt to work in a user-centred manner, but that is not always easy to achieve under cost-effective constraints, especially when the users and customers are not actually the same people.

3.2.1 User experience design process

Gladkiy (2018) states that UCD is an approach used by UX Designers to ensure that the end product or service provides the ultimate most desired experience for its users. Essentially, UXD is the concept and UCD is the process. The UCD process is broken down into the different stages below and stakeholders are encouraged to be involved throughout (Minhas, 2018):

Understand requirements: Design intends to resolve a problem, and UX Designers need to first comprehend the problem or requirement proposed. Interviews and observations of users in their natural environments are some research methods that can be used to solicit information from users. UX Designers can also facilitate brainstorming meetings with clients and demonstrate existing products to receive feedback. An organisational stakeholder interacts closely with clients and receives requirements from them. A UX Designer's outcomes should be personas, user stories and use cases. The knowledge solicited in this stage steers the UX Designer in a very clear design direction.

Market research: This is the most basic stage of UXD. In this stage the UX design team conducts research to study how the outside world utilises proposed features. The aim is to identify current user interface (UI) trends, design values and UXD. UX Design Institute (2019) adds that UX Designers in this stage are better able to recognise prospects to advance existing product features or discover new features in the case of a new product. Outputs in this stage are ideas and measurable artefacts that UX Designers can use to create designs.

Sketch design: This stage comprises UI classification of the required features. The UX design team drives this stage of the process needed for the final two stages. It is an iterative process where the team brainstorms and shares ideas with stakeholders and illustrates drawings, mock-ups, and user flows. These outputs are then tested and shared with the stakeholders.

Design: In this stage, the layout and flow have been defined and decided. The last step is to create final illustrations. UX design translates the initial wireframes, mockups, and user flows into captivating graphics with themes and styles applied to them. Thereafter, these design specifications (styles, colours, principles) should be shared with the development team. Ultimately, the development team should be part of this stage of the process.

Implement: In this stage, the development team can build functionality based on the created design specifications. The UX Designers should also be involved in this step to assist with the development stage and changes raised during development.

Evaluate: Once the product features have been applied, usability testing is conducted with potential users to assess the product for the following characteristics: usability, flexibility, resolution of user problems, and user experience or interaction. Based on the user feedback, there could be areas for improvement and the process could start again until it achieves user approval.

Great user experience can be achieved by following an iterative process, with stakeholders contributing at each stage to ensure that the organisation retains its existing users and draws in new ones. It is important for the UX design team to ensure that the objective of every stage is to build usable designs to meet user needs. See Figure 3.1.

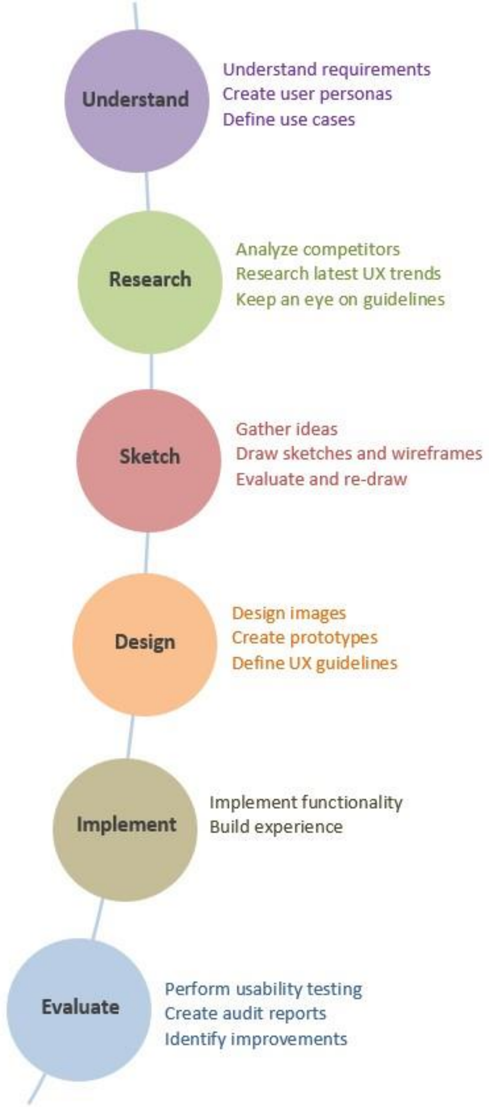


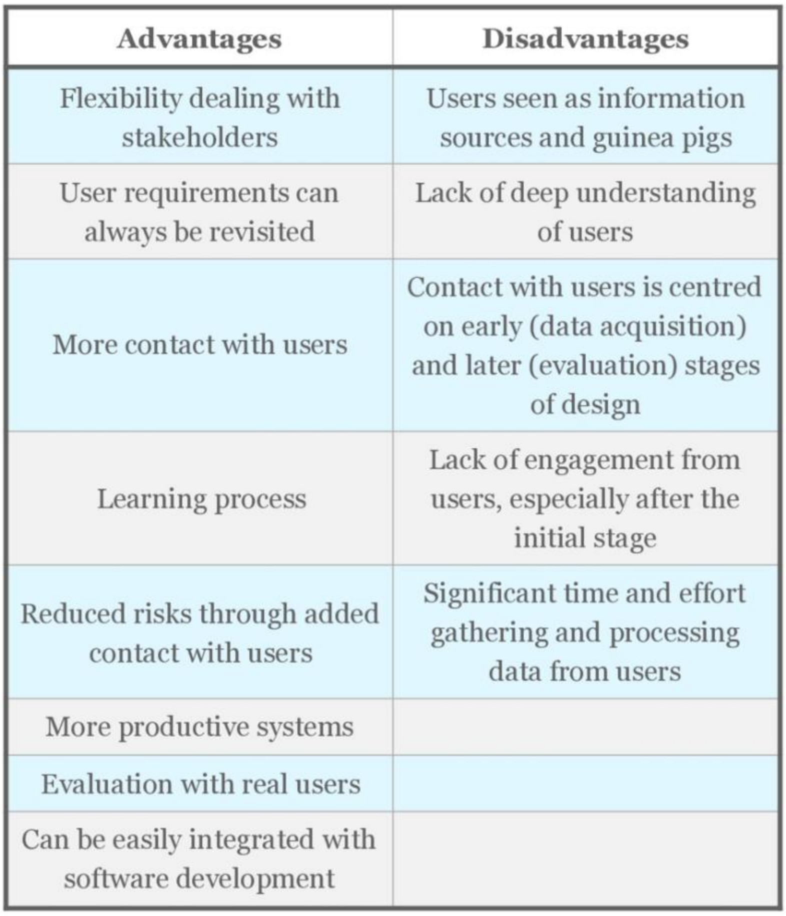
Figure 3.1: User Experience Design Process (Minhas, 2018)

3.2.2 What are the advantages and disadvantages of UCD?

Gladkiy (2018) identifies creating a unique chance to design together with communities as one of the advantages of user-centred design. The author further adds that UX Designers deeply understand the users they are building software systems for. They generate a lot of concepts and ideas to build innovative new software products rooted in user’s real needs. UCD values experimenting, iteration, learning from failure and research. UCD provides a mutual language for users, development team and stakeholders stated Gladkiy (2018).

Experience UX (2022) states that UCD may be viewed as a concept and as a set of instruments which can be challenging to integrate into current processes. Hence, acquiring buy-in and contribution from stakeholders to implement a user-centred process can be a challenge. Stakeholder and project team involvement is required for UCD processes to be properly implemented. These are some of the disadvantages identified by Experience UX (2022).

Antunes (2019) shares additional the advantages and disadvantages of user-centred design in Table 3.1 below:



Advantages	Disadvantages
Flexibility dealing with stakeholders	Users seen as information sources and guinea pigs
User requirements can always be revisited	Lack of deep understanding of users
More contact with users	Contact with users is centred on early (data acquisition) and later (evaluation) stages of design
Learning process	Lack of engagement from users, especially after the initial stage
Reduced risks through added contact with users	Significant time and effort gathering and processing data from users
More productive systems	
Evaluation with real users	
Can be easily integrated with software development	

Table 3.1: Advantages and disadvantages of user-centred design (Antunes, 2019)

3.2.3 The essentials of user experience design

Benyon (2019) states that the most important part of user experience design is putting users first. User experience Designers need to understand the users who will be using the products they are designing. They need to understand the events that need to occur between the users and the product or service. Seth (2020) states that user experience design should be about how the product or service functions rather than how it feels. Therefore, it is imperative to implement usability testing that can paint a precise image of the pros and cons. Below are the five essential elements of user experience design:

- **Properly arranged information structure**: Information should be structured to make it easy for users to locate certain product or service functionality and information and to be able to complete certain actions. The information structure ensures that the structure is designed in a manner that links functionality with the type of product or service being offered to the users.
- **Collaborative design**: This is the collaboration between the users and the product comprehensiveness, impression, and design visuals. In simple terms, this focuses on user activities and how user experience design can personalise the user’s experience. It is vital for UX Designers to ensure they create designs that are simple and clear for the users.
- **Usability centred design**: Usability is one of the most important characteristics of user experience as it allows users to obtain their goals for the designed product or service in a transparent manner.
- **Graphically engaging design**: Visual design deals with first impressions. Hence, it is a vital characteristic which directly affects the user experience process. Visual design refers to the aesthetics, e.g., outline, pictures, colours, and positioning. Incorporating modern trends when creating visually appealing designs is recommended.
- **Intentional user research**: Users are essentially the most important participants of the user experience process. It is, therefore, essential to

study user activities, perceptions, outlook, and objectives so that the final product meets user needs.

The following section is a discussion of the user-centred Agile development concept. It starts by presenting the user-centred and Agile cycles (sub-section 3.3.1 and 3.3.2), fundamental origin follows in sub-section 3.3.3, combining user-centred and Agile processes is presented in sub-section 3.3.4 followed by the stages of user-centred Agile development in sub-section 3.3.5. Lastly, the pros and cons of combining user-centred and Agile processes are identified in sub-section 3.3.6.

3.3 User-centred Agile development

The need to maintain a successful software development industry increases through its evolution, states Zorzetti et al. (2022). Combining user-centred design and Agile is one way to introduce innovation while ensuring that the end products meet user's needs. This combination ensures that user needs are met, presents experiment-driven development technique, reduces risk and while ensuring great value to the business. Zorzetti at al. (2022) further mentions that there is existing research relating to the success and progression of user-centred design. However, there is still a lack of existing research indicating comprehensive investigations. Both Agile software development and user-centred design processes are discussed thoroughly below.

3.3.1 Agile development life cycle

The objective of the Agile methodology is to adapt to change and provide functioning software. Windsor (2020) depicts five stages of Agile software development (Also see Figure 3.2 below). The life cycle includes:

Project initiation: This is the inception stage of the cycle which entails discussing the project objectives. The project vision is discussed on a high-level, as not many details are available yet. Project team members, time estimates and all the necessary resources to complete the project are identified.

Planning: In this stage the project team gathers to discuss the project details. The team identifies the requirements and converts them into backlog stories. These stories are basically how the user would define a feature or product. Project risks, deadlines, prioritisation, and estimations should be identified further and discussed. This step is complete once all the backlog items have been identified, discussed, and prioritised based on business importance and dependency.

Development: Once the requirements have been identified and defined, the development work begins. The expectation is to deliver high quality working software in iterations. The Developers then start building the first iteration with the objective of delivering an entire working project at the end of the Sprint (iteration). Testing (quality assurance) is done after each iteration for the Developer to ensure that working software is built. Final testing is carried out by the testers to find any bugs in the software during the final leg of the cycle or iteration. Usability testing can be done by including users in the final testing tasks before the product is deployed to the users (production). Any bugs detected in the testing stage are fixed until the product features are working as expected. All these tasks should already be accounted for in the planning schedule that was conducted during the planning stage.

Production: During this stage the product is handed over and is used by the intended users. It is very important to look out for any bugs that may come up during this stage. Training should take place to educate the users about the product.

Retirement: This stage is not definite and occurs when the product has reached the end of life and is being discontinued. This could be due to several reasons, namely, a newer release is available, or an older version is no longer available.

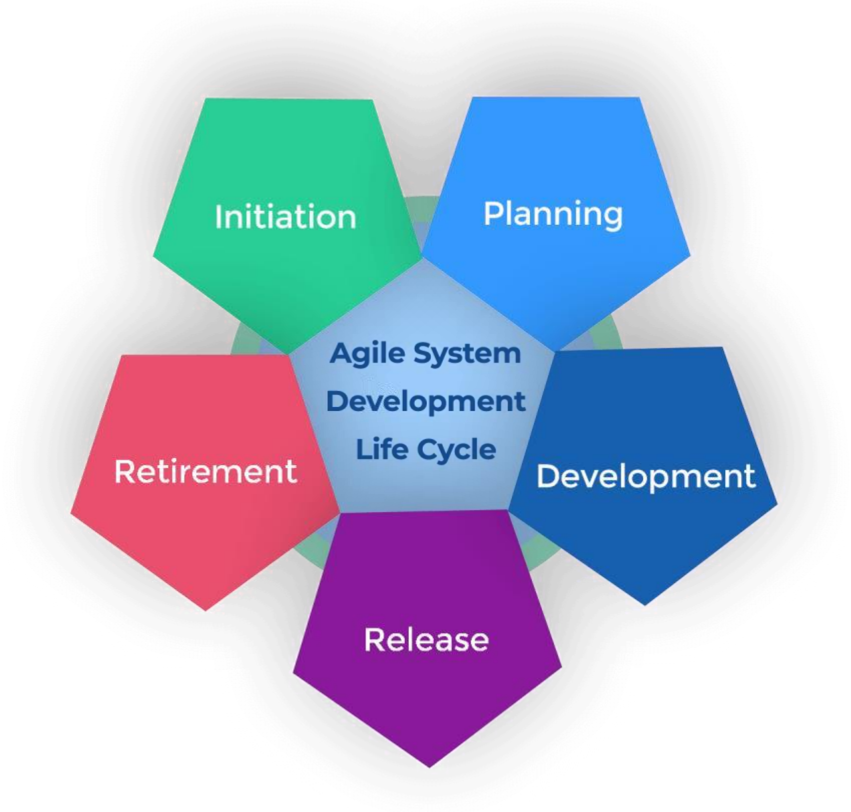


Figure 3.2: Stages of the Agile System Development Life Cycle (Windsor, 2020)

3.3.2 User-centred design life cycle

Gladkiy (2018) mentions that to design in a user-centred manner, the people that will utilise the product need to be identified, together with what they'll use it for and the environment in which it will be used. User observations need to be conducted to understand their needs and challenges. User-centred design affords a mutual language for users, stakeholders, and engineers for end users. Six phases are identified below as part of the user-centred design process. The phases are:

1. Define context and user needs.
2. Define business requirements.
3. Construct design solutions from irregular concept to complete design.
4. Evaluate designs with usability testing.
5. Develop and deliver the product.
6. Deployment – the complete product is analysed as user needs change.

The following sub-section (3.3.3) discusses the origins of user-centred Agile development methods.

3.3.3 Foundation of user-centred Agile development

Pillay & Wing (2019) define the objective of user-centred Agile development to integrate UX Designers and Agile Developers during the software development cycle. Cockton et al. (2016) state that combining UCD into software development has been a challenge for quite some time. These challenges did not exist during the first two decades of UCD because methods enabled UCD to be clearly structured and simply integrated in principle. However, this changed after the introduction of Agile software as it varies from premeditated development. Cockton et al. (2016) further state that there has been a lot of progression over the past decade. When efforts were initially made to present UCD into software development in the 1980s, the main challenge was gaining buy-in from stakeholders for UCD practices across all development phases.

UCD thrives in an iterative environment which makes it compatible with Agile. Progressive requirements, set of iterations and incremental software development are some of the other similarities of UCD and Agile. Pillay & Wing (2019) further argue that both Agile and UX have differing perceptions of how to build software. Agile focuses on how software is developed while UX highlights user interaction with the software product. Despite these contradicting perceptions, both these methodologies have a vital role in building quality software.

3.3.4 Combining user-centred design and Agile development

The integration of UCD and software development has been found to work better in principle (theory) than in practice. The absence of fitting UCD tasks in the software development stages was a challenge in the 1980s despite multiple efforts reports Cockton et al. (2016). UCD proved to be more compatible with the traditional method of software development. However, UCD requires development activities to be repeatedly redone until the software product developed fulfils user functionality and usability needs. Unfortunately, the traditional software development method (e.g., Waterfall) does not support iterative development, hence the introduction of Agile. Agile was created to account for the shortcomings of Waterfall. Changing iterative activities and progressive development, to allow for functioning software products and user interaction, are some of the essential features that make UCD well-suited to Agile.

However, there are different types of Agile software development methodologies (as discussed in Chapter 2). Therefore, it is imperative not to generalise when referring to Agile development. The activities that occur during software development are far more important than what Agile software development methodologies may seem to dictate should happen in theory. Therefore, it is important to understand how to achieve Agile software development activities and integrate them into UCD in practice rather than just in theory. Keep in mind, however, that the theoretical methods form the basis for practice (Cockton et al., 2016).

"You've got to start with the customer experience and work backwards to the technology" (Jobs, 1997, as cited in Cockton et al., 2016)

As quoted by Steve Jobs above, the objective of user-centred design is to place customers at the centre of the design process. However, Agile emphasis is on the software (coding) development process. Usability and assessment of the product under design is the focus of user-centred design. Equally, UCD and Agile have iterative processes and incremental improvement as mutual principles (Farebo, 2015). As mentioned above by Cockton et al. (2016), the contrast may be challenging to adopt in practice due to short and continuous delivery being one of the main Agile focuses, while UCD focuses on user experience. An effective integrated model of the two practices requires prior research to determine user needs and the usage environment. This process requires thinking outside of the two practices to get a full understanding of the user needs. Based on the findings indicated in sub-sections 3.3.3 and 3.3.4, we can conclude that the compatibility between Agile and UCD makes it less difficult to integrate the two methods.

Furthermore, as mentioned in sub-section 3.3.3 it requires buy-in from stakeholders as well as the rest of the software development team. In the same context, Cockton et al. (2016) emphasise training Developers or any Agile team members as vital as Agile would need to integrate UX into daily software development activities.

The following sub-section looks at the user-centred Agile development process based on existing methods.

3.3.5 Stages of user-centred Agile development

Method 1: Gothelf and Seiden (2016)

Another approach introduced by Gothelf and Seiden (2016) is Lean User Experience (Lean UX). Lean UX uses underlying values in three ways. Firstly, it removes unnecessary tasks from the UX design process. Secondly, there is minimal focus on documentation. Alternatively, documentation that is created during the design process can assist the entire development team moving forward. Thirdly, continuous testing and evaluation is used to assess whether the product will meet user needs. Lean UX has design thinking and Agile values at its core.

Method 2: Kieffer et al. (2017)

The Agile-UX development cycle is depicted in Figures 3.3 and 3.4 below. The Agile team track is represented in yellow and the UX team track in purple. This cycle includes three stages: product vision, product co-creation and product enhancement. Features are iterative versions of the product prototype deployed after each Agile-UX development cycle. Feature 1 represents the MVP (Minimum Vital Product), and Feature N is the completed product. The UX team gathers user requirement tasks during the product vision stage, while the Agile team can start high-level development (front and backend modules).

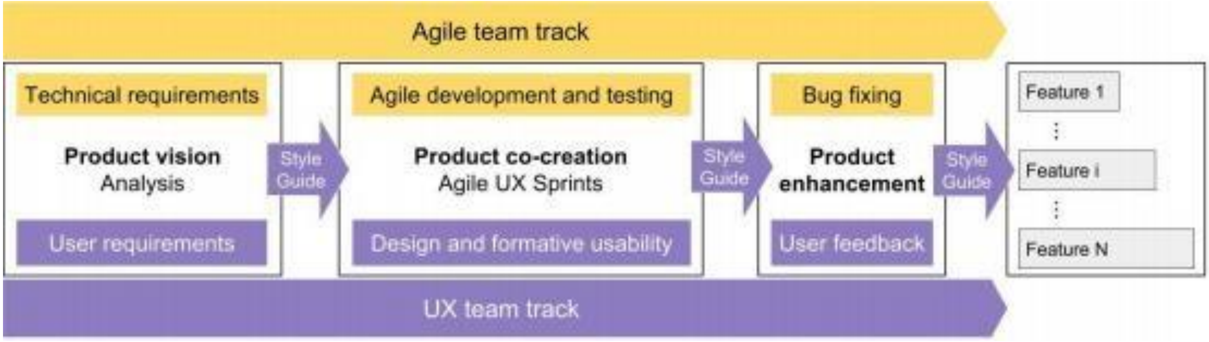


Figure 3.3: Agile-UX development cycle (Kieffer et al., 2017)

Both teams cooperate efforts in the product co-creation stage and integrate these into the Agile-UX Sprints. The UX team report their usability results in the style guide before determining which design changes are essential for the following Sprint. Agile-UX Sprints are quickly reworked until each feature is released.

In the product enhancement stage, the Agile team focusses entirely on fixing outstanding bugs, while the UX team analyses user responses which are directly incorporated into the user requirements of the following feature.

The style guide is a fundamental interface between both teams as it documents the intended user interface principles for Designers and Developers to follow during design and development. It indicates a user-driven viewpoint, investigation of user profiles, and realistic instances of tasks and usability outcomes. The aim is to meet user requirements and ensure that the product supports realistic tasks in a real-world environment.

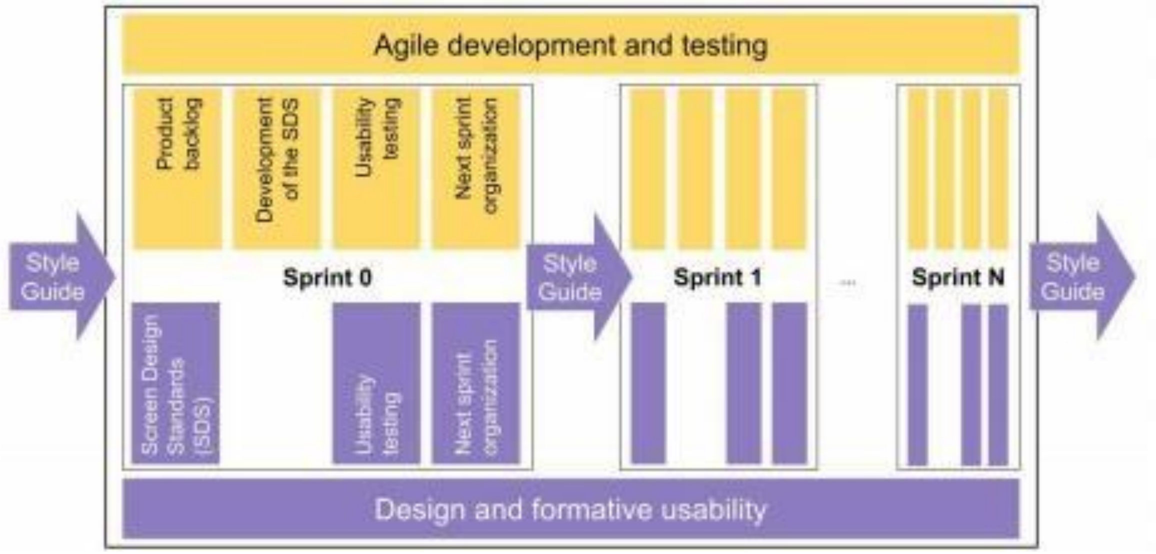


Figure 3.4: Agile-UX Sprint (product co-creation phase) (Kieffer et al., 2017)

3.3.6 Advantages and disadvantages of combining UCD and Agile development

A user-centred software development project may work successfully due to the similarities between the two practices, but it may also encounter challenges due to the differences that exist. In Table 3.2 Pillay & Wing (2019) present existing challenges including solutions which serve as advantages as well.

Known difficulties In the Literature	**Good practices for resolving difficulties Authors Research Question**
Sprints are too short	Upfront design
	UX work being desynchronised – UX work is one of two sprints ahead of development
Difficulties in prioritizing UX activities	Assigning the responsibility to the UX Designer/UX practitioner
	Separate UX product backlog
	Scrum
	Separate UX team to prioritize UX activities
Difficulty in performing UX testing	Method of UX testing: • Using discount usability engineering techniques • Using the RITE method – identify and fix in the shortest possible time • Low fidelity prototyping

Known difficulties In the Literature	Good practices for resolving difficulties Authors Research Question
	Scheduling of testing: UX testing done during Agile development tests
	Access to users: • Planning in advance for user inclusion. • Use an existing user pool to act as development and design partner and conduct testing
	Shorter time to iterate design: Dedicate cycles for user feedback and incorporate into development life cycles
Customer versus User	• Separate the "chooser" from the "user" • Involve real users to avoid biases • Use interaction designers to bridge gap between customer and development team
Lack of documentation	User stories were reported as being the most prevalent. However, there are a wide range of artifacts.

Table 3.2: Good practices for integrating UX in Agile (Pillay & Wing, 2019)

The Interaction Design foundation (2016) states that in 2003 Nielsen suggested that development teams assign 10% of their project budget to usability. However, with the current situation in terms of finance, organisations cannot afford to do so. Below are some additional disadvantages of combining UCD and AD in some organisations:

Less time to design than needed: Often, UX Designers and Developers are pushed to meet deadlines due to the nature of both the UCD and AD processes, which can shift focus from quality to time or deadlines.

Short iterations: Short process iterations may lead to less time for testing or having it aborted altogether. Possibly there could be less time to finish creating an intended prototype.

Insufficient user feedback: Based on the limitation of time, it may be easier to abandon user feedback more than any other part of the process, thus leading to insufficient user feedback. Vital feedback that can assist in meeting user requirements ends up excluding the UX aspect in the process.

Lack of communication and collaboration: Team members need to communicate effectively so that information does not fall through the cracks.

Designated team members: Team members working in remote areas can potentially cause communication issues.

Reliability on external parties to complete tasks: Involvement of other stakeholders could potentially hold the process back if the team is awaiting stakeholder feedback or involvement for too long.

Evidently, combining UCD and AD can be a cumbersome task as there are a lot of factors to take into consideration when undertaking such an approach.

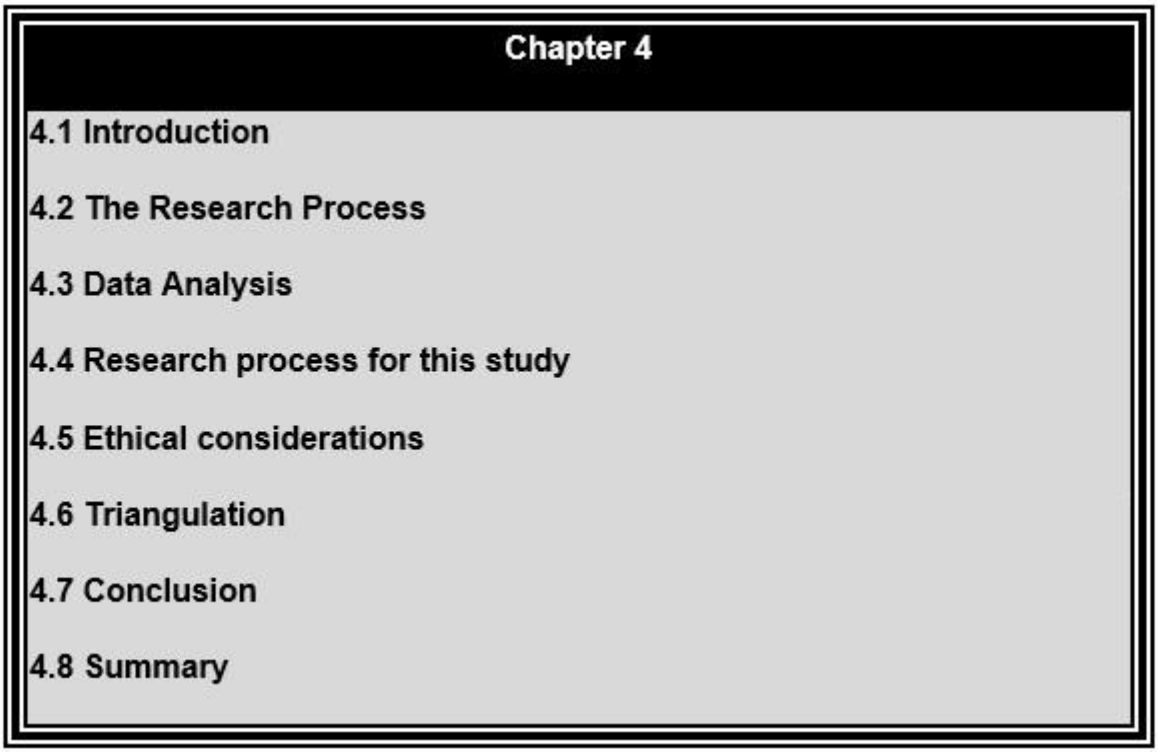
3.4 Conclusion

A positive user experience can be accomplished by following an iterative UCD process. The UCD process leads to developing products that are efficient, effective, and secure. In this process, stakeholders must be involved at inception to ensure that the organisation maintains existing users and attains new ones. Research shows that to build software products that meet user needs, the users would have to be involved in the development process. Therefore, the UCD process would need to be integrated with Agile as they have similarities of being iterative and involving user. Hence, the introduction of user-centred design. However, there are certain factors that work against combining these two practices that need to be overcome. The primary focus of UCD is user experience and continuous delivery for Agile, and it may be challenging to combine the two based on those factors. Organisations need to find a way to curb these issues to ensure user-centred software products are created. It has been highlighted that an effective Agile-UX model is one that has stakeholder buy-in, involves the customers by determining their needs and environment and then work backwards to technology.

3.5 Summary

This chapter explained the concept of user-centred Agile development. An overview of UCD and UXD is provided as the two tend to get confused. Section 3.2 and 3.3 detail the processes entailed, relevance, and limitations of UCD and UXD. The following chapter reveals the research methodology and design of this research study.

Chapter 4: Research Methodology and Design



Chapter 4

4.1 Introduction

The impact of integrating Agile and UX methods has been visibly identified in the literature review presented in Chapter 3. The findings of this literature provide clarity on the Agile-UX concept but indicate a shortage in Agile-UX methods from a South African perspective that provide a clear and simple Agile-UX process. This drives the basis for this research study. This chapter sets out the research methodology and design process for this research study. It identifies the most appropriate research design, data gathering, and data analysis process required to address the primary research question. Saunders et al. (2019) developed the research onion that depicts the fundamental aspects to be followed when conducting an effective research study. The sub-sections below will peel off the research onion starting from the outer layer; philosophies (sub-section 4.2.1), approach (sub-section 4.2.2), techniques (subsection 4.2.3), methodological choices (sub-section 4.2.4), time horizon (sub-section

4.2.5) and data gathering methods (sub-section 4.2.6).

4.2 The Research Process

Research design is defined by Leavy (2017) as a process of developing a structure or strategy for a research project. Research methods are the strategies used for research design. Quantitative, qualitative, and mixed methods are some of the most used. Pillay & Wing (2019) further state that the concept of research design methods can be deemed as strategies of addressing research questions and cover comprehensive objectives including data collection methods and challenges that the researcher might encounter. The different layers for this research are presented in 'the research onion: research process' in Figure 4.1 derived from Saunders et al. (2019). Each of these layers are identified and discussed in the sections that follow within the context of this research.

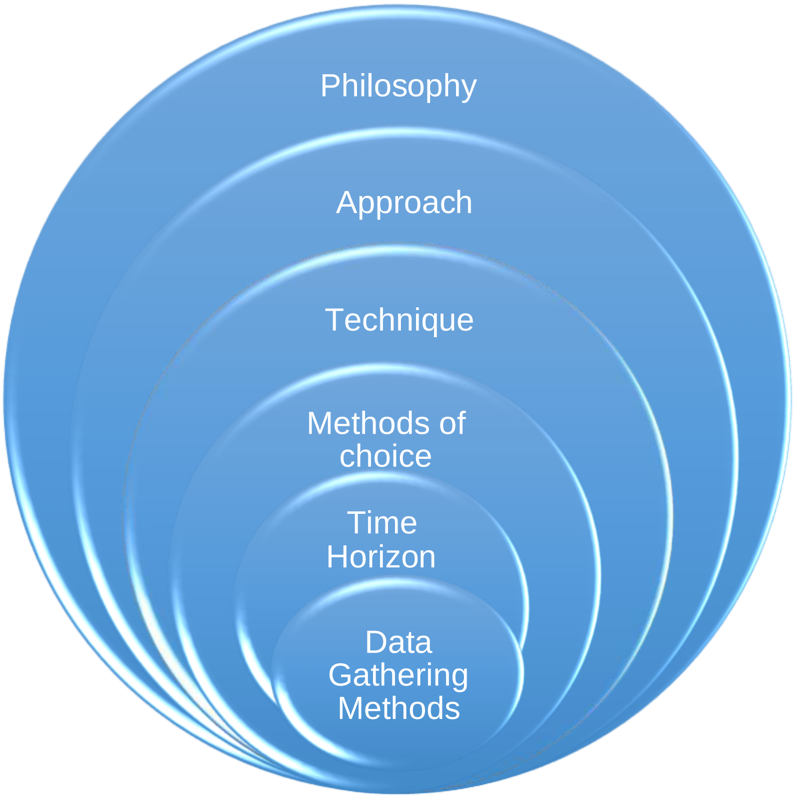


Figure 4.1: The Research onion: research process (As adapted from Saunders et al., 2019)

4.2.1 Philosophy

Research philosophy is defined as “the development of knowledge and the nature of knowledge” (Saunders et al., 2019). In basic terms, it is a belief about how the data for a certain occurrence should be collected, analysed, and utilised (Dudovskiy, 2022c). It is vital for a researcher to comprehend philosophical issues when conducting research in a specific field. Integrating research philosophy into your research involves being conscious and creating your own assumptions and beliefs. Figure 4.1 above, outlined by Saunders et al. (2019), illustrates the positioning of the research philosophy in the outer layer of the research onion. This is the first topic to be addressed for this research study. The research onion also illustrates different types of philosophies. The choice of which research philosophy to use is impacted by real-world implications. It is important for researchers to be aware of which research philosophies exist before deciding on which philosophy to adopt. Dudovskiy (2022b) states that there are four types of research philosophies in the space of business studies, namely, pragmatism, positivism, realism and interpretivism, which are also represented in the research onion. The research philosophy reflects researchers’ view of the world, and these views are, therefore, the foundation for the research strategy.

Based on the understanding of the different philosophies as outlined in literature, this research study will not focus on determining whether a theory is true based on social science research, as defined by the positivism philosophy. Furthermore, realism also does not apply to this study as it highlights the independent existence of reality and the human mind. Both aspects are needed to address the problem and research questions in this study. This research study focuses on issues of truth and reality based on the research questions. Its focus rules out the use of pragmatic philosophy, which focuses on what works as the truth regarding the research questions.

Interpretivism is the most suitable philosophy for this research study as it focuses on finding the most appropriate method to integrate UX and Agile. It also addresses the research questions based on the expertise and perceptions of software development teams, using different research methods to validate findings.

4.2.2 Approach

The next layer of the research onion in Figure 4.1 considers the research approach. This sub-section focuses on presenting knowledge on the research approach concept and describing the research approach that will be followed for this research study. Creswell (2014), as cited by Ishtiaq (2015) described research approach as a set of strategies and processes that contain the phases of predicted comprehensive methods of data gathering, assessment and interpretation. Dudovskiy (2022b) and Malhotra (2017) identify three research approach types namely deductive, inductive, and abductive. Deductive is defined as an approach used when a set of assumptions requiring validation or dismissal is set for a research study. On the other hand, an inductive approach does not include formulation of assumptions but rather focuses on research questions and objectives that must be accomplished throughout the research process. Lastly, in the research process abductive approach focuses on clarifying the incomplete interpretations, shocking evidence identified at the start of the research study states Dudovskiy (2022b).

Based on the literature, a deductive approach will be followed for this research study as a new theory will be formulated based on pre-existing research. A new model for user-centred Agile software development will be created based on existing research.

4.2.3 Technique

Once the research approach is peeled off, the following layer of the research onion is the strategy or technique. Sahay (2016) describes the research technique as the general framework of the research providing details for several components of design. Saunders et al. (2019) supports this description by defining research techniques as the way researchers aim to approach their research study. Research techniques include experiment research, case study, grounded theory, action research and ethnography. These research techniques are briefly described below:

An **experimental** research method is a scientific technique for examining one or more individual instances that are modified and used in one or more distinct variables to determine how they affect the latter. The effect of the two instances is regularly observed and verified over time to assist the researcher to come to a probable outcome. This technique uses quantitative data and numerical analysis during data collection (Pubrica, 2022).

A **case study** research method evaluates a specific instance to find its significant structures and to establish views. It can also establish perceptions about the nature of any instance or case and draw the significance of philosophy and the differences between them.

Grounded theory is a qualitative research methodology that establishes an inductive approach where trends are identified from the data as a pre-requisite for the research study. The research findings are derived essentially from existing research rather than data that has been established suitably for preceding frameworks.

Action research is perceived as a real-world method for a research problem within the public environment. It includes analysis to establish relations to the best method, insightful practice, and a methodical procedure to analyse the professionalism and the experience of the experts.

Ethnography is a close examination of people, analysing their social or ethnic interactions and their connotations. In research, the spectator researches the perceptions of the individuals under observation to understand the varying connotations and significance of their actions. According to Kozinets (2010) netnography comes from ethnography which is an older method used in anthropology and refers to the observation of participants and learning about their culture. Netnography is used for online observation of online community and their behaviours. Hayes & Sakoda (2013) agree that netnography is about making online observation concerning the understanding of cultural behaviours online. Also, netnography is adaptable, immersive, naturalistic, and descriptive. A study by Kozinets et al. (2022), states that the means of communication for netnographic is through online community members and it can be via Facebook postings, Twitter posts, blogs, forums and LinkedIn.

A case study was identified as a suitable technique for this research study as perceptions of the participants about the Agile-UX concept would allow the researcher the ability to draw meaningful theories. Ethnography through a netnography analysis was also used to gain insight on the software development community forums within the broader South African context.

4.2.4 Methodological choices

Selecting the most suitable research methodology is imperative for the success of a research study. Opoku et al. (2016) advise that the type of research undertaken determines the type of research methods and underpinning data collection methods to use. However, irrespective of the research methods used for a study, the data collection methods undertaken should be able to meet the research study's objectives (Melnikovas, 2018). Research methodologies can be characterised into qualitative and quantitative methods. There is, however, a new classification referred to as the mixed-method research methodology. Quantitative methods are used to test a distinct theory and typically involve experimental data testing, which cannot be addressed using qualitative research methodology. The researcher gathers evidence through statistical data to evaluate later using statistical tools. The objective is to determine whether the theory is accepted or rejected. A qualitative research methodology typically answers the 'what' and 'why' research questions. The purpose of a researcher is to gain in-depth facts to support the research questions. However, there is a constant belief that researchers are biased as the qualitative research methodology is generally perceived as subjective. Most evocative, experimental, and historical studies are generally qualitative.

The most popularly used qualitative data gathering methods are case studies, interviews, and seminars. The mixed-method is one that uses both quantitative and qualitative research methodologies. In this type of research method, the researcher uses multiple data collection instruments as well as data analysis procedures to support findings. Opoku et al. (2016) also state that the mixed-method research method reflects on different beliefs, attitudes, perspectives, and standpoints from quantitative and qualitative research methodologies.

The findings by Pillay & Wing (2019) based on the formulated research questions for this study (see section 1.5 in Chapter 1) validate that the research methods used for this research study to be mixed methods (both qualitative and quantitative).

4.2.5 Time Horizons

According to Saunders et al. (2019), time horizon refers to the time outline to complete within the research study. Two types of time horizons are defined within the research onion, namely, cross-sectional, and longitudinal. Cross-sectional is already wellknown where data must be composed. This data is composed in a certain period. Longitudinal on the other hand involves the collection of data continuously over an additional period. It is usually utilised where a significant aspect of research involves analysing variation over a certain period. It also allows for the regulation of the variables under examination. The type of research methodology chosen does not affect the time horizon of choice.

A cross-sectional time horizon will be used for this research study so that data can be composed at a certain period and does not involve collecting data over an extended period.

4.2.6 Data Gathering Methods

Once the research philosophy, approach, technique, and time horizon have been peeled off, the last layer in the research onion is the data gathering methods. Data gathering methods are means with which research information is managed and evaluated using specific research tools. Literature reviews, surveys and interviews are data gathering methods that have been identified by Paradis et al. (2016) and the University of Minnesota (2022) as the primary data collected from a variety of sources. These various sources can be interviews, questionnaires, surveys, observations, focus groups, ethnographies, case studies, oral studies, documents, and records.

These sources are used to answer the research questions, test the theory, and analyse the results (Dudovskiy,2022b. According to Dudovskiy (2022b), data gathering methods can be divided into primary and secondary data methods. Data gathering methods are divided into two groups, namely, quantitative, and qualitative. Quantitative methods are based on accurate numerical values in a variety of representations. It is easier to make assessments using such data. A survey would be an accurate example of a quantitative data collection method. However, qualitative methods are related to words, numbers, feelings, and emotions. They involve all the non-quantifiable elements and aim to better understand the topic at hand. Questionnaires, interviews, and focus groups are just a few examples of qualitative methods. Secondary data gathering methods is the data that already exist in books, newspapers, journals, and online portals, etc. This type of data collection method plays a very important role in the legitimacy and trustworthiness of data collected.

In this study, a literature review was used to evaluate existing research for better strategies to integrate user experience practices and Agile software development. The literature review was supported by using questionnaires to gather data to understand the impact of integrating UX practices in Agile software development projects. It allowed for different viewpoints or perceptions of integrating the two practices and identified what challenges exist. Once the questionnaires were completed, netnography was used to collect data from non-participant who participate in various forums related to software development and the various methods. The information collected using netnography assisted in understanding the software development community within the broader South African context. It provided insights into understanding the impact of integrating UX practices with Agile software development. Table 4.1 below highlights the research questions and data collection techniques used for this research study.

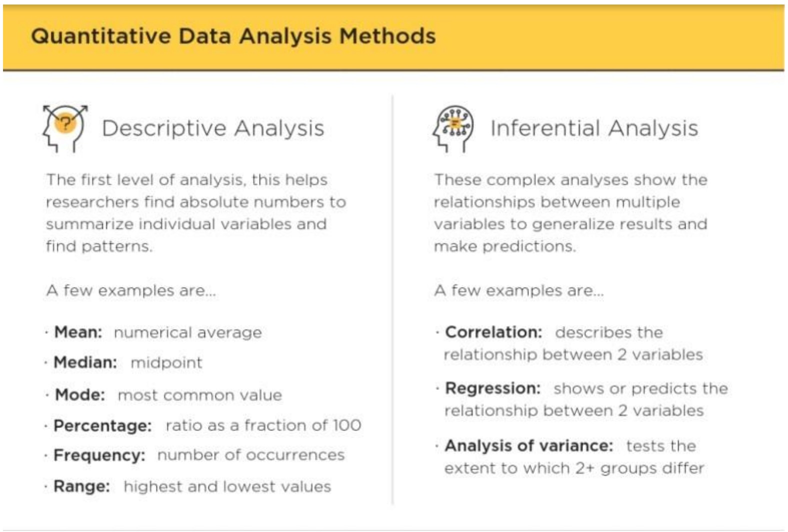
Research question	Data collection technique
What best practices currently exist to integrate user experience into Agile software development?	Literature Review Questionnaire Netnography
What challenges exist for integrating user experience into Agile software development?	Literature Review Questionnaire
What is the impact of integrated UX practices in Agile software projects?	Literature Review Questionnaire

Table 4.1: Research questions and data collection techniques (Source Researcher)

4.3 Data Analysis

Data analysis is the process where researchers go from the bulk of data after data gathering to significant comprehension. There are various data analysis methods, depending on the research study (Debever, 2019). Additionally, Dudovskiy (2022b) states that researchers should include in the summary how they intend to analyse the primary data gathered in the data analysis section of the research study. Respectively, there are two data analysis methods, namely, quantitative, and qualitative as mentioned in above sections. In qualitative research, utilising interviews, experiments, and focus groups, requires identifying patterns and trends within the participants responses. These responses require critical analysis to accomplish the research objectives. Quantitative research data analysis, however, involves critical analysis of numerical figures and searches for validation behind the importance of the results collected. Given the absence of primary data, data analysis methods can assist in analysing patterns and trends within secondary data associated to the research study.

There are two widely used data analysis types, descriptive and inferential, as described in Table 4. 2.



Quantitative Data Analysis Methods

Descriptive Analysis

The first level of analysis, this helps researchers find absolute numbers to summarize individual variables and find patterns.

A few examples are...

- **Mean:** numerical average
- **Median:** midpoint
- **Mode:** most common value
- **Percentage:** ratio as a fraction of 100
- **Frequency:** number of occurrences
- **Range:** highest and lowest values

Inferential Analysis

These complex analyses show the relationships between multiple variables to generalize results and make predictions.

A few examples are...

- **Correlation:** describes the relationship between 2 variables
- **Regression:** shows or predicts the relationship between 2 variables
- **Analysis of variance:** tests the extent to which 2+ groups differ

Table 4.2: Quantitative Data Analysis Methods (Debever, 2019)

This research study adopted descriptive analysis. According to Debever (2019), descriptive statistics are regarded as the first level of analysis as they assist researchers to review data collected and to find common trends. It is essential for researchers to ensure that the selected data analysis method is appropriate for the research study, as descriptive statistics do not explain the perspective behind the number or statistics found. Descriptive statistics are convenient when the research is restricted to its sample and does not need to be generalised to a larger population.

In this research study, descriptive analysis is used to summarise the findings collected from the Agile software development teams. This data analysis method was chosen for its efficiency in simplifying the primary data, the research topic and the research questions selected for this study. The summarised data from the findings will be displayed numerically according to the different results from the participants.

4.4 Research process for this study

This section presents a summary of the approach followed in the research process for this research study. A case study approach was used as it is best suited to address the research questions and purpose of this study. The case study, however, used both qualitative and quantitative methods for data analysis. A literature review, semi-structured interviews and surveys were the data gathering methods for this research study. Mohajan (2017) mentioned that multiple data gathering methods enhance the validity and reliability of this research. Therefore, the various data gathering methods in this research enhanced the accuracy and truthfulness of this research study.

The research questions and objectives are presented in Section 1.5 in Chapter 1.

4.5 Ethical considerations

Research integrity symbolises a variety of good research practices and, which entails trustworthiness, precision, equality and safeguarding human and animal counterparts involved in research. The researcher and the educational institute involved should be accountable for ensuring research integrity (Zayed University, 2022). This research study did not collect information from vulnerable participants or groups. It provided participants with informed consent and the option to discontinue should they feel uncomfortable. No personal information was collected during the study.

4.6 Triangulation

Data triangulation is the method of authenticating the findings of a study using multiple methods to verify its reliability and validity. Triangulation can be used for both qualitative and quantitative research. There are different triangulation methods according to the techniques researchers use. It involves the use of time, space and people or participants. Investigator triangulation involves various researchers for the same study, while theory triangulation requires utilising multiple theoretical schemes to understand the phenomenon under study. Lastly, methodological triangulation requires different data gathering methods (Nightingale, 2022). Furthermore, Denzin as cited by Wilson (2014) discovered two types of methodological triangulation — the 'within method' and the 'between' or 'across' method. The 'within method' is one methodological triangulation method but entails various strategies within it. Between or across methods require two different methods in one study. It is vital to understand which triangulation method to use for a research study and why.

This research study used both qualitative and quantitative data gathering methods (mixed). Findings from different sources and studies were compared, analysed, and assessed where methodological triangulation was utilised.

4.7 Conclusion

The research onion that this study followed illustrates the aspects that must be covered when creating a research study such as Philosophies, Approaches, Strategies, Choices and Techniques and Procedures for data collection and analysis. This study followed the following aspects:

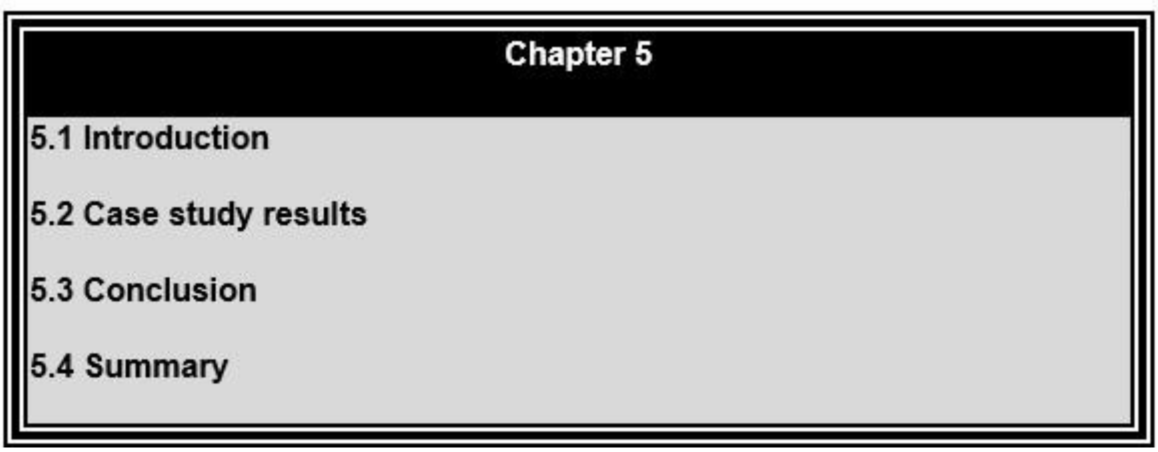
- Philosophy: Interpretivism philosophy was followed to obtain a deep understanding of the facts and factors concerning this study.
- Research Approach: Inductive research approach was followed as the new theory was generated from the data collected from the software development teams within the South African context.
- Research Strategy: Case study research strategy was followed due to this study being a single case of determining the UX designers and software developers' perception towards Agile and UX.
- Research Methodology choice: Mixed method research was followed to allow for the collection of quantitative and qualitative data.
- Data Collection Methods: Literature review was followed to analyse what has been documented on the existing studies by various researchers, the questionnaire was followed to collect data from development teams. Also, netnography was utilised to include comments made in online forums.
- Data Analysis: Descriptive statistics were followed to summarize the data collected from the respondents and the summarized data is in a form of numbers.

Convenience sampling was chosen for this study, and this sample means that it is all up to the participants to volunteer and take part in answering the questionnaire, they are not obligated.

4.8 Summary

This chapter has discussed the research process guided by Saunders' research onion (Saunders et al., 2019) in section 4.2. This study takes a stance that existing research shows that there are insufficient Agile-UX methods that exist hence the need for this research study (See section 1.2). The interpretivism approach has been chosen as the research philosophy for this research study as a new model for user-centred Agile software development will be created based on existing research. The study also uses a deductive approach as a new theory will be formulated based on pre-existing research. Section 4.3 presents knowledge on the problem statement, research questions and objectives. In addition, a new model for user-centred Agile software development was created based on existing research. Furthermore, a case study research strategy was conducted as an investigation focusing on real-life processes. A cross-sectional time horizon was used for this research study as multiple data gathering methods were used, namely, surveys, interviews, focus groups and a literature review. Similarly, the study adopted descriptive statistics to analyse data collected from the survey. Triangulation (See section 4.6) was adopted as this study has chosen more than one methodology. The following chapter (Chapter 5) presents case study findings.

Chapter 5: Case Study Results and Findings



Chapter 5

5.1 Introduction

This chapter presents the case study results collected during this study. Section 5.2 presents the findings from semi-structured interviews in the Western Cape Province of South Africa with the aim to gain an understanding of how SD teams approach their development processes. Two SD teams from the WC were interviewed to gain insights (Appendix A outlines the guiding questions). The findings from the semi-structured interview are outlined in the table followed by some conclusions. Based on the knowledge gained from the semi-structured interviews as well as the literature review, a survey was formulated and distributed to different software development teams in the WC Province. Appendix B depicts the questions that were put to the teams. Sub section 5.2.2 describes the results collected from this survey followed by netnography results in sub section 5.2.3. Lastly, the chapter conclusions (section 5.3) and summary (section 5.4) are presented.

5.2 Case study results

A case study was conducted of SD teams in South African companies in the Western Cape Province that develop software solutions for the local and international market. Some of these software development teams are UX inclusive. The aim of the case study was to determine impact Agile-UX has on the implementation of software projects.

Semi-structured interviews were conducted with members of different software development teams to determine their understanding of existing best practices for integrating Agile and UX methods and the factors that impact the integration of these methods. Convenience sampling was used to identify suitable participants from the place of work of the researcher. Participation was voluntary and no personal data was collected. Interviews were conducted via digital platforms. Members of the software development teams participated and shared their views.

The interview report is described in sub-section 5.2.1 whilst the full set of questions can be seen in Appendix A.

5.2.1 Interview Report

This section highlights a report from the semi – structured interviews that were conducted for this case study. This report reflects responses from the interviewees relating to the roles of the teams they belong to and how they execute the software development process. See responses in Table 5.1 below:

Discussion point	Response SD Team 1	Response SD Team 2
Type of software developed by organisation	Import and export software systems	Mobile software for educators
What is the size of the SD team and what are their roles?	Medium-sized: Business Analyst, Test Analyst, Project Manager, Systems Architect and UX Designers	Small-sized: UX Designers, Head of Engineering, outsourced Developers, Product Owners, and Head of Product
SD approach used	Waterfall	Agile
Advantages of current approach	• Works well with current SD processes relating to analysis and development	• Delivery on Sprints for Developers • Collaboration between team members • Integration real users into approach through user research and demos

Discussion point	Response SD Team 1	Response SD Team 2
Disadvantages of current approach	• Testing happens at the end of the cycle. Sometimes testers must wait very long to complete testing tasks • Intensive documentation	• UX Sprints may be tight • A lot of upfront UX work • UX is a challenge for short sprints
Awareness of UX?	Yes	Yes
Is UX included in SD approach? Provide detail	Yes. However, UX Designers work in isolation to the rest of the SD team. This is the preferred approach as UX Designers are not required throughout the SD process	Yes. Through the UX Design process. UX Designers work on tasks prior to the development sprint. They perform user research and demonstrations to their users to showcase software functionality
Consideration to change existing processes	• Document existing processes for new starters • Adopt Agile methods to improve existing processes and team culture	Streamlining design and developer tasks by: • Not rush through the UX process for fear of delaying the following Sprint • Avoid context switching by working up front and having to integrate back into the current Sprint. Find better ways of working to avoid this

Table 5.1: Semi-Structured interviews responses (Source Researcher)

The findings indicate that the interviewees belong to small to medium sized SD teams that use the Agile and Waterfall approach. SD Team 1 uses the Waterfall approach and is UX inclusive, however UX designers work in isolation from the rest of the team. The current approach works for analysis and development processes for SD team 1; however, testing is delayed as it happens right at the end. Furthermore, SD Team 1 indicates that adopting the Agile approach can improve their SD processes and team culture. SD Team 2 uses the Agile approach and is also UX inclusive. UX designers work one sprint ahead of the development sprint. These UX sprints can be tight as they need to be completed before the development sprint starts. The Agile approach works well for delivering development sprints, supports team collaboration and integrates real users into the SD processes. Existing processes can be improved by streamlining developer and UX tasks.

5.2.2 Survey of SD Sector in Western Cape

An online survey was sent to South African software development teams in the Western Cape province that form part of companies that develop software solutions in the local and international market. This study samples companies that compete in the software development sector, the context of the study. The survey aimed to collect data at team level as the target was software development teams inclusive and exclusive of UX practices within the SDLC. The survey was also aimed to collect data mostly from teams that use Agile as a software development methodology to address the impact of implementing Agile-UX in software development projects.

The participants are members of software developments teams that are inclusive and exclusive of UX practices within the SDLC. The SDLC comprises Developers, Test Analysts, Business Analysts, Technical Leads, Scrum Masters, Project Managers, Managers, Domain Owners, and User Experience Designers, depending on the nature of the business and the team itself. A copy of the survey is contained in Appendix A.

Demographical data

The participants that responded to the survey are representative of a variety of age groups, gender, work roles and experience. A total number of 35 respondents were recorded (n=35).

The survey question responses indicate that most responding participants were between 18–30 and 31–40 years. The least participants were between 41–50 years of age. Overall, 44.4% of the responses were from participants aged 18–30 and 31– 40 and 11.1% aged 41–50 years.

Of the 35 participants that responded to the survey, 58.8% were female, and 41.2% were male. The length of working experience for participants was between 7 to 9 years and 10 and above, each indicating 37.5% of the participants, and the least being between 4 to 6 years, making up 25%.

The participants were asked about the roles they occupy in their various organisations. Of the 33 participants that answered this question, 12 were Test Analysts, 5 Developers, 4 Business Analysts, 4 UX experts, 3 Technical Leads, 2 Scrum Masters, 2 Project Managers and 1 Manager. 2 participants did not answer this question.

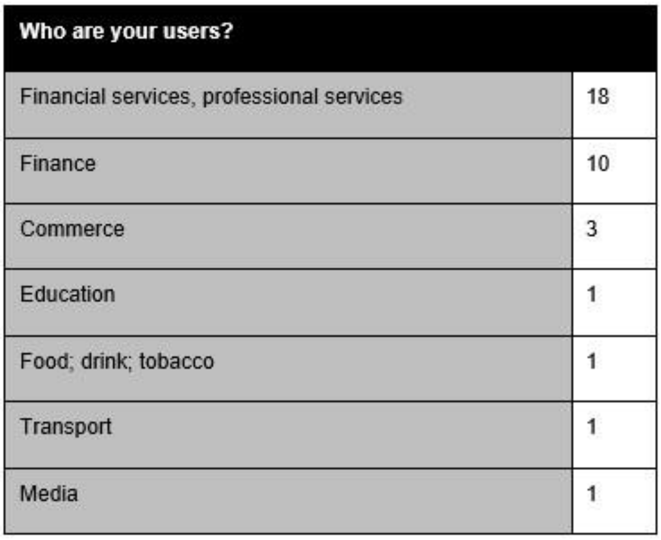
Agile-UX software development

This sub-section discusses the participants' knowledge and experience within a user experience inclusive Agile software development team. It presents information about the participants' involvement with and understanding of Agile-UX software development projects.

In answer to the question:

Who are your users?

As depicted in Table 5.2, most of the users are from companies that provide financial services. 10 are purely from Finance and 3 from Commerce. Lastly, Education, Food, Drink; Tobacco, Transport and Media all had 1 participant each.

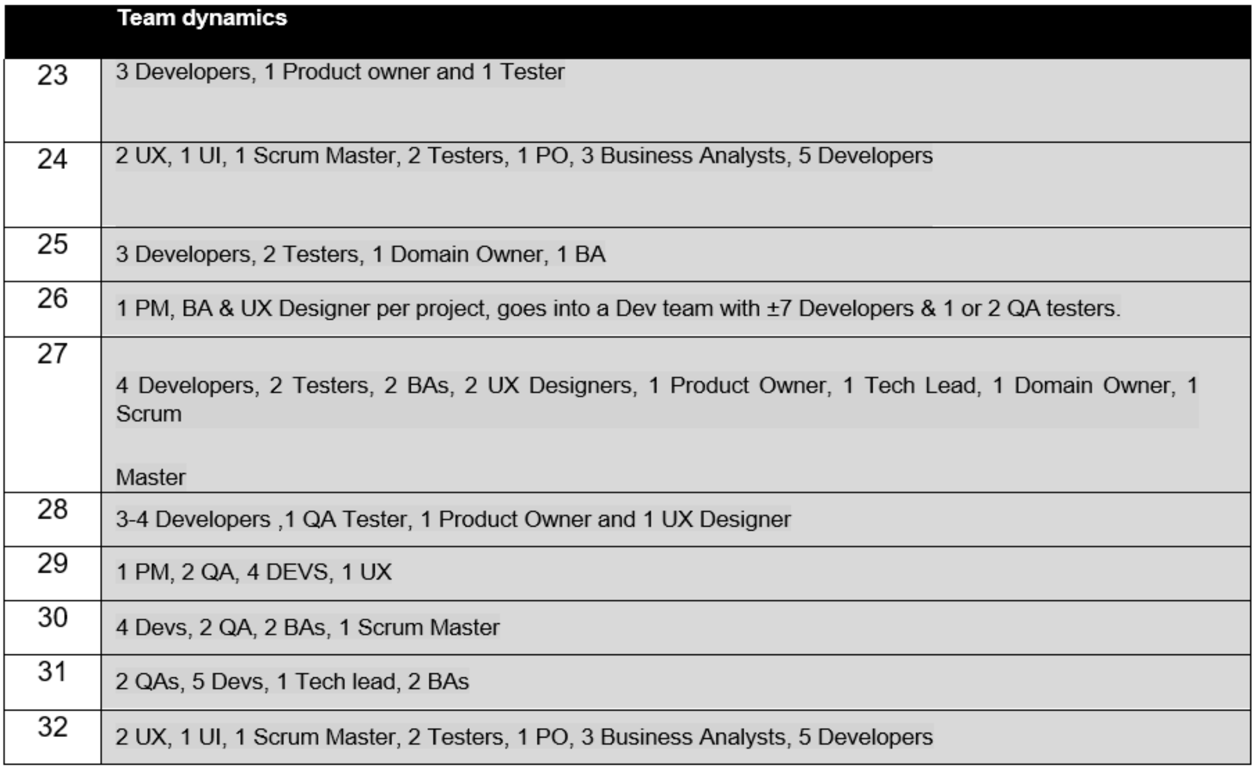


Who are your users?	
Financial services, professional services	18
Finance	10
Commerce	3
Education	1
Food; drink; tobacco	1
Transport	1
Media	1

Table 5.2: Who are the users of your organisation? (Source Researcher)

The results in Table 5.3 indicate the participants' team dynamics. The question posed was added to the survey to gather information on teams who have UX experts and the nature of the company. Out of the total 35 participants, 3 did not answer this question. According to 20 participants, these teams do not have UX experts included in the team. However, the 13 remaining participants who have UX experts as part of the team, come from companies that focus on user interface related systems.

	Team dynamics
1	4 backend Devs, 2 Frontend Devs, 1 Product Owner, 1 Scrum Master, 1 Product Designer, 1 UX Designer
2	2 Developers, 1 Tech Lead, 1 Tester ,1 Product Owner, 1 Project Manager
3	4 Devs, 1 QA, 1 DevOps, 1 PM, 1 Solution Architect
4	1 Developer, 1 DBA, 1 BA, 1 SA, 1 PM
5	3 ABAP Developers, 3 Neptune Mobile Developers, 1 Integration Manager, 6 Functional Consultants (SD MM, WM, PP, PM, FI, Retail)
6	8 Developers, 4 QA Analyst, 1 Automation Engineer, 1 Product Manager, 1 BA, 2 UX Designer
7	1 QA, 3 Devs and 1 BA
8	4 Developers, 3 Testers, 1 Product Owner
9	3 BA, 3 Testers, 5 Developers
10	1 Domain Owner, 5 Developers, 2 Tech Leads, 2 Testers
11	3 Developers, 3 Testers, 1 Product Owner, 2 BAs
12	4 Developers, 2 Testers, 1 Project Manager, 1 BA, 1 UX Designer
13	3 Developers, 3 Analysts
14	8 Developers, 2 testers, 1 Domain Owner, 4 BAs, 1 Scrum master
15	9 Test analyst in my team, but most teams have 1 Domain Owner, approximately 4-6 Developers, 3-4 Business Analysts, 1 Test analyst, no UX Designers
16	7 Developers, 2 Testers, 1 Project Manager, 2 BAs
17	1 Developer, 1 BA, 1 DBA
18	3 Developers, 1 Tester, 1 Product Owner
19	Team of Developers and CEO and freelance UI Designer and UX Designer
20	1 BA, 1 UX Designer, 2 Developers, 3 Testers
21	4 Dev, 1 QA, 1 Project Manager, 1 Data Scientist, 2 UX Designers
22	3 Testers with outsourced Developers



	Team dynamics
23	3 Developers, 1 Product owner and 1 Tester
24	2 UX, 1 UI, 1 Scrum Master, 2 Testers, 1 PO, 3 Business Analysts, 5 Developers
25	3 Developers, 2 Testers, 1 Domain Owner, 1 BA
26	1 PM, BA & UX Designer per project, goes into a Dev team with ±7 Developers & 1 or 2 QA testers.
27	4 Developers, 2 Testers, 2 BAs, 2 UX Designers, 1 Product Owner, 1 Tech Lead, 1 Domain Owner, 1 Scrum Master
28	3-4 Developers ,1 QA Tester, 1 Product Owner and 1 UX Designer
29	1 PM, 2 QA, 4 DEVS, 1 UX
30	4 Devs, 2 QA, 2 BAs, 1 Scrum Master
31	2 QAs, 5 Devs, 1 Tech lead, 2 BAs
32	2 UX, 1 UI, 1 Scrum Master, 2 Testers, 1 PO, 3 Business Analysts, 5 Developers

Table 5.3: Team dynamics (Source Researcher)

Results shown in Figure 5.1 indicate that most of the participants come from an Agile methodology background with 64.5%. Coming in second with 16.1% each is Scrum and Hybrid. Waterfall comes in last at 3.2%.

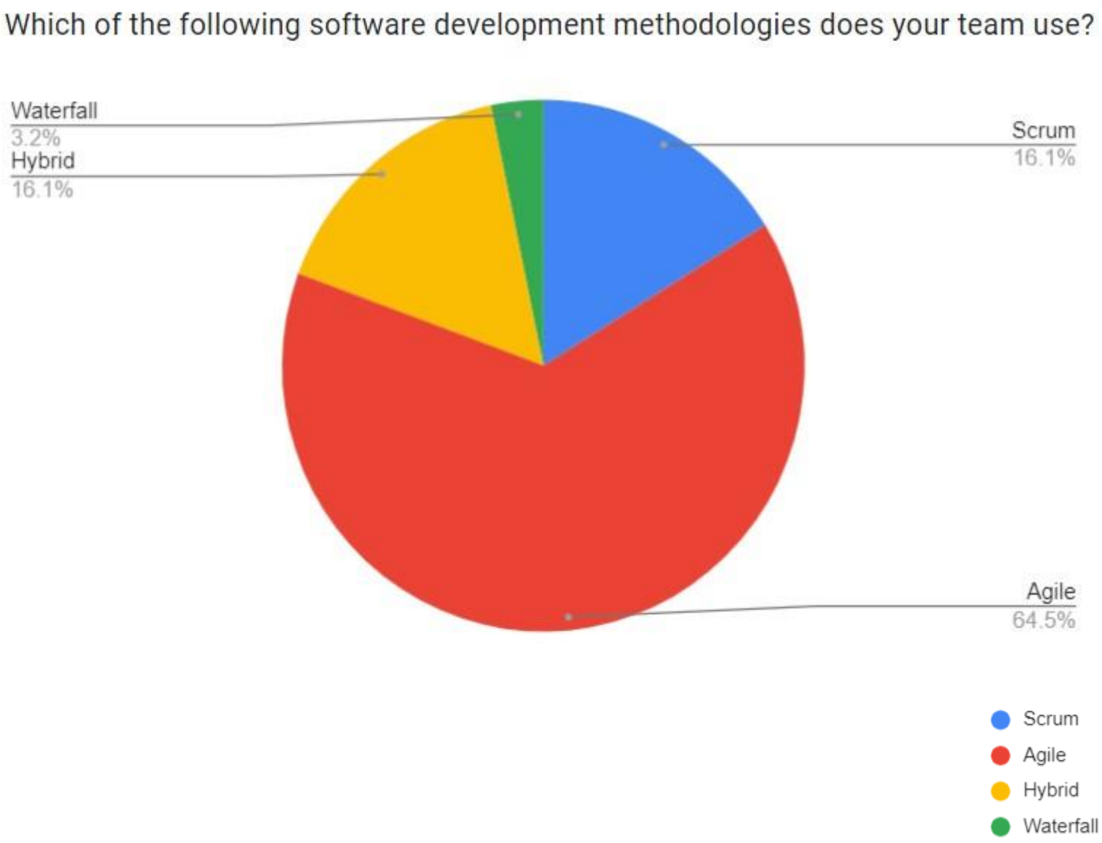
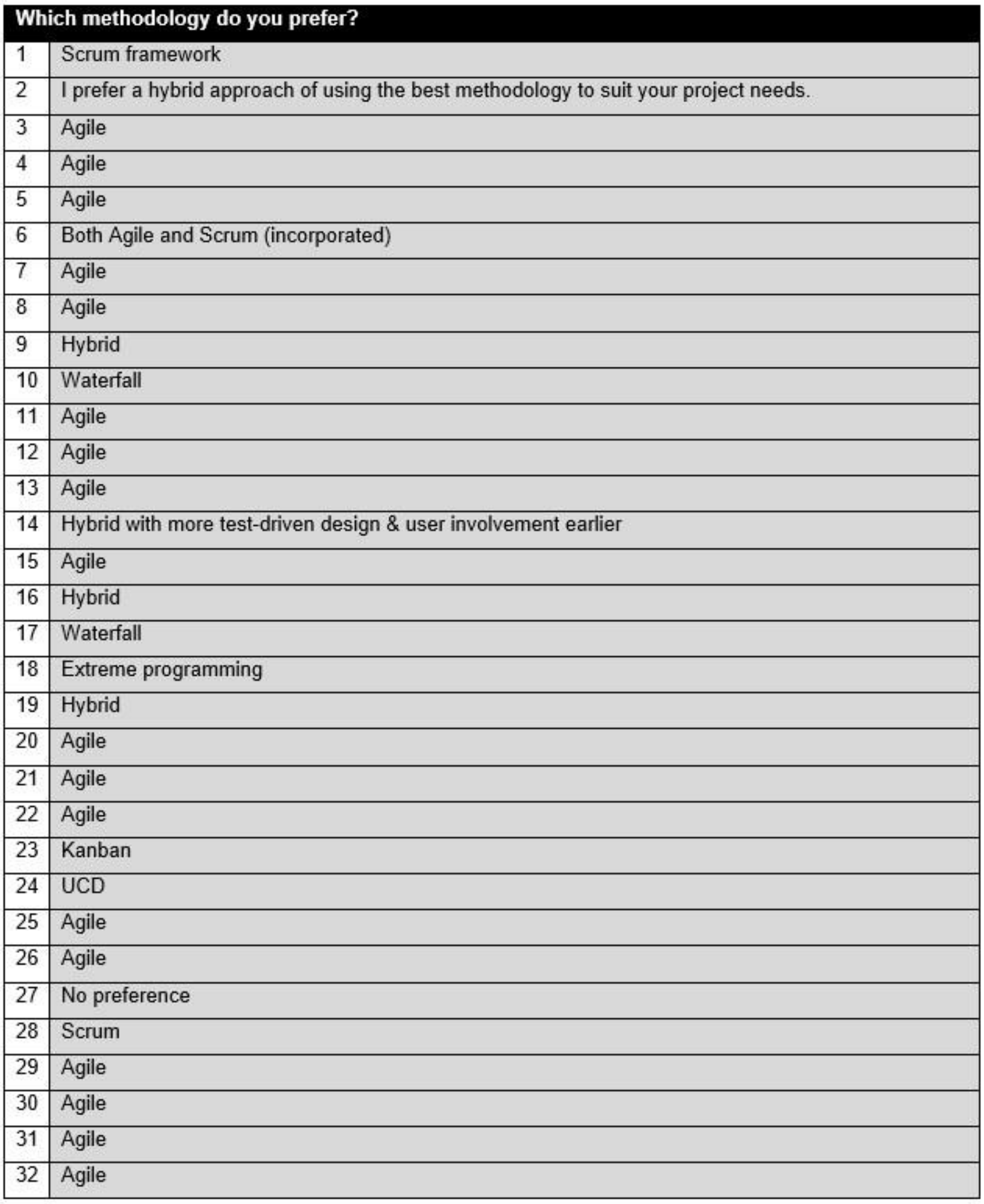


Figure 5.1: Software development methodologies (Source Researcher)

When asked about their software development methodology of preference, depicted in Table 5.4, the majority (19 of 32 that responded) indicated Agile. These findings support the notion that Agile is indeed the most popularly used software development methodology. It also indicates that some participants do not understand the various existing software development methods. 3 participants did not respond to this question.



Which methodology do you prefer?	
1	Scrum framework
2	I prefer a hybrid approach of using the best methodology to suit your project needs.
3	Agile
4	Agile
5	Agile
6	Both Agile and Scrum (incorporated)
7	Agile
8	Agile
9	Hybrid
10	Waterfall
11	Agile
12	Agile
13	Agile
14	Hybrid with more test-driven design & user involvement earlier
15	Agile
16	Hybrid
17	Waterfall
18	Extreme programming
19	Hybrid
20	Agile
21	Agile
22	Agile
23	Kanban
24	UCD
25	Agile
26	Agile
27	No preference
28	Scrum
29	Agile
30	Agile
31	Agile
32	Agile

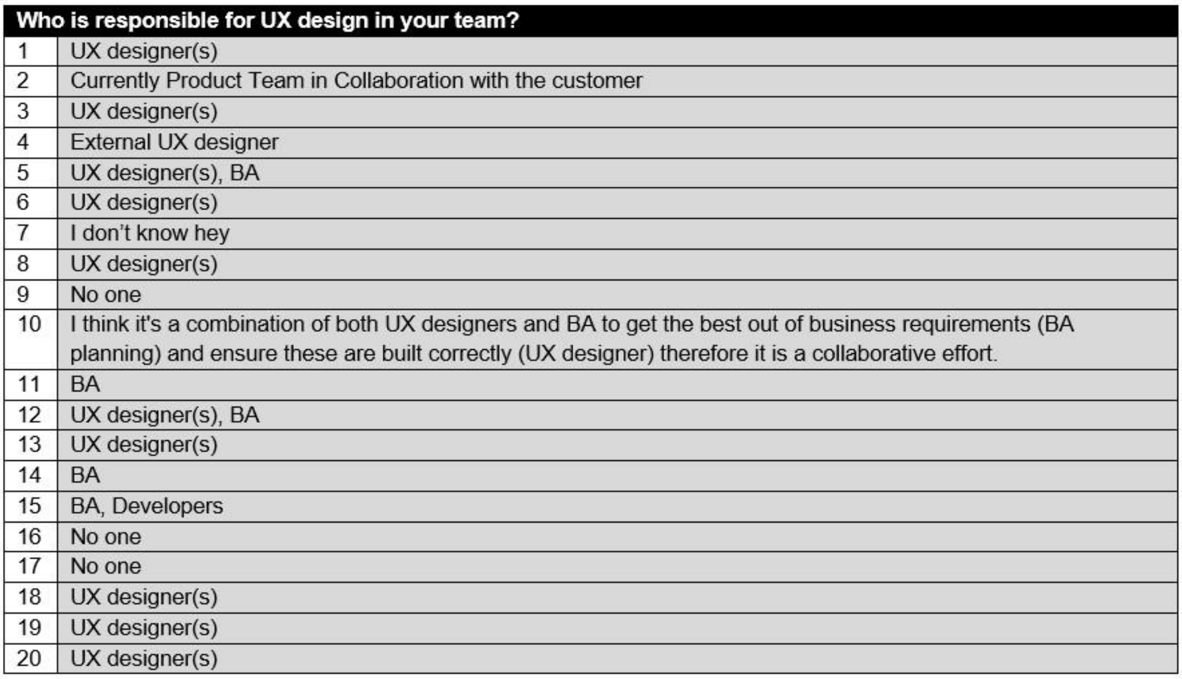
Table 5.4: Preferred Software development methodology (Source Researcher)

Participants were asked about their understanding of Agile. The results indicated that 99.9% of the participants had similar responses indicating that they understood Agile and how it works.

The final question in this section related to their understanding of user experience. Most respondents indicated that user experience refers to designing systems from a user's perspective.

Contribution of UX methods to the success of Agile software projects

Based on the responses, most participants highlighted that user experience is an integral part of Agile, drives iterations and ensures that the software built is easy to use and meets user needs. They are familiar with UX practices as most of them highlight how UX experts start working ahead of the Sprint ensure user research is done to determine if the product will indeed meet user requirements. However, some of the responses identify that UX experts often work in isolation and that they should be included and accounted for in the Scrum process.



Who is responsible for UX design in your team?	
1	UX designer(s)
2	Currently Product Team in Collaboration with the customer
3	UX designer(s)
4	External UX designer
5	UX designer(s), BA
6	UX designer(s)
7	I don't know hey
8	UX designer(s)
9	No one
10	I think it's a combination of both UX designers and BA to get the best out of business requirements (BA planning) and ensure these are built correctly (UX designer) therefore it is a collaborative effort.
11	BA
12	UX designer(s), BA
13	UX designer(s)
14	BA
15	BA, Developers
16	No one
17	No one
18	UX designer(s)
19	UX designer(s)
20	UX designer(s)

Table 5.5: Who is responsible for UX in the team? (Source Researcher)

Participants were asked which member(s) of the software development team are responsible for user experience design, and the responses in Table 5.5 above indicate that UX Designers are mainly responsible. Business Analysts were a close second, as they work very closely with users and user requirements. However, they are not UX experts and are not familiar with UX practices unless the BA has previous UX experience. 10 participants did not respond to this question.

Based on responses, the top 5 factors that contribute to the success of software development projects are team collaboration and communication, client inclusion and usability testing, flexibility, clear requirements, and quality assurance. Furthermore, participants identified the focus on UX, as one of the related factors (client inclusion), as one of the highest contributors.

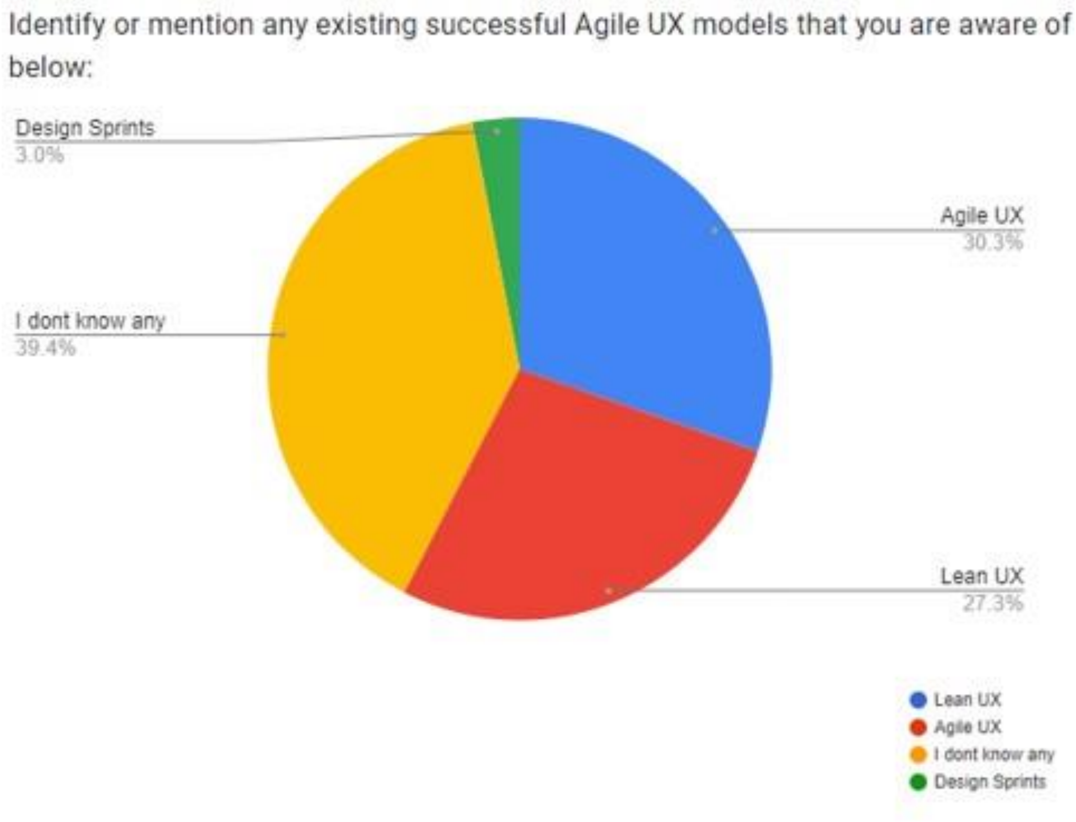


Figure 5.2: Participant's Awareness of Agile-UX methods (Source Researcher)

Figure 5.2 above highlights that 39.4% of the participants were not aware of any existing Agile-UX models. 27.3% were aware of Lean UX based on companies that they had interacted with. Lastly, 30.3% were aware of Agile-UX, which refers to having

UX experts, based on the responses from previously asked questions, and the remaining 3.0% were aware of design Sprints.

The results indicate that 4 participants mentioned Amazon, with 1 participant each identifying Netflix, Google, Visa, Wonga, Clickatell, Maxxor, Apple, IBM, Ipreo, Allan Gray and Woolworths as the organisations adopting Agile-UX models. 1 participant identified start-ups as organisations that adopt Lean UX models. Vodacom, PPS, and Discovery were each identified by 1 participant as adopting Lean UX models. Lastly, 1 participant said they were not aware of any and the remaining 15 participants did not respond to the question.

The UX experts mentioned in the results in the above section are either internal or external. The results represent an even split of 50% global and 50% local. Furthermore, they either work within the software development team or in isolation. Research indicates that the latter is not regarded as Agile-UX. This solidifies the need for awareness and inclusion of Agile-UX models in software development teams.

Software development team members can participate either directly or indirectly in ensuring that UX practices are integrated effectively into the software development team. The results show that some team members are aware of how they contribute successfully to including UX practices. However, a small percentage indicated that they were not aware of any role that they play. Lastly only 23 participants responded to this specific question and 12 did not. See Table 5.6 below.

	Comments:
1	Currently excluded
2	Backend development, ensuring that the data is efficient and accurate
3	Close work with Developers and QA
4	I don’t play a direct role
5	Collaborating with development and being part of the planning sessions with either CTO or if PMO
6	Facilitating valuable conversations with all involved
7	As a tester, I have to verify and ensure that the developed application meets the provided designs. Communicating with the UI/UX team and Developers

	Comments:
8	Always make sure a feature is easily accessible to the end user with clear understanding
9	No role
10	As a tester, I am often required to anticipate how a typical user would interact with the systems/products. When we had a UX team, I would often try to involve or consult UX experts to get their input on a variety of issues that I would come across during the testing phase
11	I am a junior tester (QA) so I need to ensure that the quality of software is upheld always
12	I try my hardest to be involved with the analysis and concept development as I believe that UX design includes deciding on offerings
13	Ensuring that the UX team understands the problem statement and what we are trying to achieve. Having the UX Designer walk the team through the design thinking process
14	Raising any UX issues I encounter while I test
15	Ensure the user is consulted throughout the process especially with anything that might affect the usability of the system
16	None
17	I don't
18	Ensuring System/Application meets criteria set out by UX design model
19	We are a more back-end focused team no need for UX in our space
20	By making sure that we deliver documents that are easy to read in order for the developer to build a system that is pleasing to the user
21	Highlight any areas where I think there could be potential for improvement
22	QA
23	Project Manager/Product Owner/Domain Owner

Table 5.6: What role do you play in successfully integrating UX into your team? (Source Researcher)

Participants highlighted that the biggest advantage was the active and frequent communication between UX experts and development team regarding progress and issues. 34.4 % of the participants jointly selected active and frequent communication between the UX experts and the development team members about project progress and issues as an advantage. 21.9 % of the participants highlighted the inclusion of UX experts. 15.6 % of the participants selected clarity on the roles and responsibilities and the inclusion of the UX experts. 12.6 % jointly selected communication between UX experts and development team members regarding project progress and issues.

Clarity on each other's roles and responsibilities was collectively selected by 9.4% of the participants. 3.1 % selected active and frequent communication between UX experts and development team members regarding project progress and issues. The same percentage (3.1 %) selected clear requirements and clarity about each other's roles and responsibilities, ensuring Business Analysts wouldn't assume the UX role.

Considering the disadvantages of including UX experts in the software development teams, the results show that 43.5% of the participants highlight that UX experts may find it difficult to adapt to Agile practices. Existing research has shown that UX was not intended for Agile and, therefore, it might be a challenge to merge it into Agile and vice versa (Handa & Vashisht, 2016). 39.1 % say that the lack of technical knowledge may pose a challenge to UX experts fulfilling the merger of these two practices successfully. The remaining 17.4% is distributed among those who highlighted that Sprints are developer focused, UX experts work in isolation and those who indicate that Agile teams would not be conducive for UX experts to make intelligent decisions.

Buy-in from stakeholders was selected by 20 participants, company attitude by 18 and understanding roles and responsibilities by 16 participants as factors influencing successful implementation of Agile-UX models. The results show that most participants view buy-in from stakeholders as the most important factor to effectively implement Agile-UX methods.

The participants were asked to indicate whether they had participated in user acceptance testing (UAT) or usability testing in their respective teams. The results below show that 81.8% of them do and the remaining 18.2% do not. These results may differ per participant based on the nature of the company where each one works.

In terms of the effectiveness of performing these testing types, 27 out of 35 participants answered this question, where 13 of them indicated the high effectiveness of the processes. 8 participants did not answer this question.

20 out of 35 participants indicated that the post-implementation issues they encounter are less likely due to not including UX practices in the software development process. 7 had a neutral response, 6 leaned towards quite likely responses, while 1 participant was convinced that UX inclusion was the reason for these issues. 1 participant did not answer this question.

Lastly, the participants were asked for recommendations of how they perceived would guide the successful integration of UX into the software development process. 27 out of the 35 survey participants answered this question. Many of the responses highlighted the inclusion of UX experts in the development process, clear communication and collaboration by all members, stakeholders buy-in and a flexible software development process. 8 participants did not answer this question. See comments in Table 5.7 below:

	Comments:
1	I’d say start with UX - understanding the problem and goals and include a technical lead in these conversations. Reason mainly is because this helps solves technical constraints upfront.
2	Ensure that your team has enough UX work.
3	Do it right
4	Also monitor the users when they test, this way we can pick up so issues they don't even realise they encountered
5	Understanding the importance of UX Work closely with the UX team Have discussion, showcase or talks about UX/UI topics Do demos every 2nd week during the SDLC (in case Developers are deviating from the initial designs)
6	Software team and UX work together
7	Include user experience as part of Sprint planning
8	Choosing a formal process that will be flexible in the accommodation of the changes that may come
9	Communication within the team is important, consult with stakeholders more especially business. So that they are in the know. The continuous performance of UAT testing is very import to source out system limitations and constraints.

	Comments:
10	All systems are geared towards the user, therefore UX should be made a priority otherwise would just waste time developing something that would be rejected
11	That really makes your life beautiful.
12	Invite your users into your refinements of a feature you are doing for them
13	Start with writing proper user stories ... not just functional requirements!
14	Every team and product is different be open to be flexible. Make sure the UX Designer is part of the conversation early on.
15	My recommendation is they should include UX into their software development process, because UX and Agile share the same values which is user satisfaction, collaboration etc.
16	Just open up and communicate. Both sides need to be emphatic and flexible towards each other. Developer is doing a lot of UX work every day, so involving an expert at these points will make the product better.
17	UX/UI experts must be part of Agile rituals
	I would encourage Agile teams to include UX personnel in their teams. It only makes sense to have a 'Champion of the user' involved throughout the product/solution delivery process. After all, teams do not create products/systems for themselves but for the users.
18	Have open conversations with UX Designers to try understand where they fit into the team and which meetings are relevant to them
20	they must first understand User Experience's role
21	Be open to enhancements later even in cases where MVP is the priority for the roadmap currently.
22	Have Agile-UX experts
23	User experience should just keep up with the fast pace of Agile.
24	We do not have user experiences experts in our team
25	Include UX experts as part of the Agile process but also include Developers in the UX process to ensure alignment
26	I have added the UX leads into product refinement so that they are part of the team lead group. All tickets are assigned a related UX ticket or subtask so that UX are aware of all Features and Improvements/ The UX Designers work is pulled into the Sprint so it is visible and they form part of the team ceremonies.
27	Have open conversations with UX Designers to try understand where they fit into the team and which meetings are relevant to them

Table 5.7: User Comments (Source Researcher)

The next section outlines the findings of the netnographic analysis of software development blogs and online forums to observe the occurring discussions and phenomena related to UX and Agile and the integration thereof.

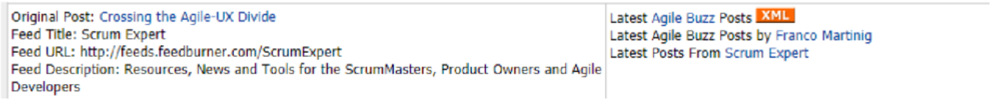
5.2.3 Netnography Results

Eleven unedited blog reviews and comments are displayed in Table 5.8 below. These reviews and comments are taken from online forums such as Scrum.org from https://www.scrum.org/forum/scrum-forum/5508/ux-design-scrum and https://www.toptal.com/ portray the common challenge UX experts face in Agile teams of aligning their tasks with the rest of the team in Scrum. Many have had to work ahead of the Sprint to facilitate user research and prototype creation, which most view as a

'Waterfall' way of working. A comment left on the Artima forum https://www.artima.com/forums/flat.jsp?forum=155&thread=370995 provides an article written by Bulejwiski (2014) on best practices to combat this challenge. These reviews cover the period between 2011 and 2019.

User	Comment/Review	Date
Castlehills	Agile is a widely embraced software development methodology now. The topic is, can a UI/UX team be agile in the same iteration where the other software developers also are involved in? The problem is, the UI/UX team can't simply work on a prototype which keeps changing in the same iteration. This causes problem to the software developers who integrate the services to the UI who don't have a solid UI to work on. The solution is as I explain below. The UI/UX team has to come up with mock-ups first. Then they develop prototypes. The developers can follow the prototypes to make the final product. This means the UI/UX team has to be ready with their prototypes way before the real iteration starts. The UI team's iteration should be X when the software developer's iterations should be X+3. However, the UI team can be available to help the developers to clarify the doubts in the X+3 iteration. How is this working in your development team? (1) How do you make sure the prototype is up to date with the production version? (2) How are you making sure the software development team is not duplicating the efforts that UI/UX team doing? For example, the software developers can simply merge the changes from the prototypes to the real production version rather than coding the UI again? (3) How do you demo a prototype? Does it run with the real data or the mock data? (4) Who consume the prototypes? Is it just for the demo to the Prod Dev team or made for the software developers (5) Is prototype necessary? Share your experiences.	31 August 2011

User	Comment/Review	Date
Karine Cardona	Hi everyone, I'm currently working on an essay about Agile UX Design for a Master2 at GOBELINS, L'école de l'image in Paris, France. As a UX Designer, I've joined a team working well with Scrum. I do feel that being a UX designer in a Scrum team is not easy, because some times user research takes more than a sprint to be done. On the other hand, developers often "wait for me" so I can deliver new screens to implement them... and so I feel I'm on a waterfalls process fallback. I read blogs and posts about "how to do UX agile design". Some people recommend to "do design spikes" or to "be 2 sprints ahead", some even imagine a "Design owner" role... I'm now interviewing people working with Scrum about this, and I'd be happy to have your feedback... Karine	8 July 2013
Choisel Fogang	Hello Karine, I use to work on this question with the product owner. From my experience (a short one) I'd recommend to be 2 sprints ahead, this way you can have the customers express their feedback so that you have one sprint to adapt your UX before the developers can start implementing the underlying feature. According to my understanding of Scrum, an item should start and end in the same sprint. How does the UX design fits in this affirmation, if this quote is true? Choisel.	8 July 2013
Karine Cardona	Hi Choisel, thanks for your feedback. You're pointing exactly what I'm wondering: I intended to get customers feedback during my first sprint, but I never get a chance to do it: the product I'm working on is a B to B one, has only a few customers I could ask feedback, and I have "the big pictures" only now... 3 sprints later I started working (also I'm alone doing it so I may need more time than a UX team working together). I was thinking that maybe there's a part of their work that could fit well with Scrum (once you have had the time to figure out what your customers need and how things work for them) ; but another that needs to be done before being able to create user stories... (and so am I back to a waterfall conception?)	9 July 2013

User	Comment/Review	Date
	That's that particular point I'd like to have feedback for. I'm interested in how other teams work, regarding their size maybe? their projects? uses to work together? their product owner?.. and whatever parameter could change things. Karine	
Franco Martinig	Integrating UX into an Agile process is not always an easy challenge. In this article, Mike Bulajewski reminds the why the differences between the two vision exist in the first place and proposes some solutions on how they could be solved. This article provides a very interesting point of view of the Agile-UX relationship from the UX side. The article starts by reminding the origin of the Agile movement and how it changes the vision from the traditional software development. He thinks that for many agilists, "he mere existence of designers.  Original Post: Crossing the Agile-UX Divide Feed Title: Scrum Expert Feed URL: http://feeds.feedburner.com/ScrumExpert Feed Description: Resources, News and Tools for the ScrumMasters, Product Owners and Agile Developers Latest Agile Buzz Posts XML Latest Agile Buzz Posts by Franco Martinig Latest Posts From Scrum Expert	21 April 2015
Suzanne Nash	The organisation I work for employs UX/UI contractors, who have worked on so-called 'Agile' teams in the past. Yet within these teams, it has been common practice to spend a significant amount of time **upfront** on mapping out user journeys and flows across the system from an experience and interface perspective. I am not a UX/UI practitioner myself, but this all sounds very 'Waterfall' to me. The designers are keen to spend time working 1-2 Sprints ahead, but say they also need to dedicate time to the upfront period to obtain this bird's-eye-view; it also helps them challenge many of assumptions in advance. I would like to hear from SCRUM Masters and UX/UI experts about how to make these roles as Agile as possible? I appreciate that UX/UI needs to play a part in the definition of 'ready', but what about the bird's-eye-view part? Can this be achieved by only working incrementally, a few Sprints ahead?	6 November 2017
Ian Mitchell	Is this the view of the Development Team? Are the designers actually members of the Development Team, in so far as their involvement is needed to create an increment once a Sprint is underway?	6 November 2017
Ian Mitchell	How are they currently validating the assumptions which do end up being made before a Sprint has started? What quality of empirical evidence are they gathering?	6 November 2017
Suzanne Nash	Hi Ian. Thanks for the reply. Yes, the UX/UI functions were included in a former Development Team, who itself promoted the upfront design work. It became common practice to capture a user journey map, which fulfilled a role not too dissimilar to an entity diagram, before the Sprint began. It was considered as 'empirical evidence' internally as it was signed off by the business and technical staff.	7 November 2017

User	Comment/Review	Date
	I can foresee the response from you already on this! What I'd really appreciate is finding some ways I can engage and inspire the UX/UI function in relation to Agile. The risk here is that we follow inherited ways of working because "it's always been done this way". However, the last thing anyway wants is to hear: "I know you think you are Agile now, but you aren't; please apply the theory and it will be so much better!" So...any examples or reference to articles would be most welcome to help individuals discover for themselves. Perhaps some shared stories about transitioning from Waterfall to Agile, specifically in the UX/UI domain?	
Ian Mitchell	In my experience, one of the key unwritten duties of a Scrum Master is to be the first to tell people the last thing they want to hear. It's his or her responsibility to put transparency over the Scrum process, always,and to highlight any shortcomings or risks in its implementation. In your case it sounds as though UI/UX people were formerly Development Team members, but in practice are not now. The work they do may however be of value to the Product Owner in forming Product Backlog items in advance of each Sprint. My advice would be to discuss the situation with the PO, explaining the meaning and value of empirical evidence proven via actual release, and the risks inherent in taking the present leaps of faith. Where there is delay or wastage in proving value, the PO carries the can	7 November 2017
Neal	I think the point made about resolving bottlenecks in Scrum would be better as a clarification than invalidation of Scrum. A good scrum master will resolve bottlenecks based on the sprint content and available resources. A great way to do that is to get side-lined team members to pair with someone on a task that's a bottleneck, or to split out tasks that are less skilled. For example, a developer, under the leadership of a UX expert, could shorten task time by helping build or modify wireframes as a result of recent customer feedback. Similarly, later down the line, a UX team member could support the developers in testing or reviewing the implemented UI as it is developed. This, to me, is how to squeeze more out of your team instead of side-lining people and seeing velocity drop. Also, I think UX has a massive part to play in "Lean Start-up" approaches, and in rapid testing of ideas such as the approaches suggested in books like"Sprint"	9 May 2019

Table 5.8: Netnography results (Source Researcher)

Themes identified from the netnography include:

Integrating UX into Agile is a challenge

Users from most of the forums highlight the challenge of including UX into Agile software development teams. One user stresses that UX/UI experts cannot function in an iteration that keeps on changing as software Developers will not have any work tasks. Therefore, it is suggested that UX/UI experts need to first work on mock-ups that will be converted into prototypes that Developers can use. Another states that the difficulty for a UX expert comes with the long period spent on user research more than the duration of completing the Sprint; hence the solution to work two Sprints ahead.

- **UX experts working two Sprints ahead, a 'Waterfall' way of work**

One user stated that working ahead of the Sprint is not an Agile practice but rather the traditional way of building software, as all Scrum tasks should start at the same time according to the fundamental principles. Perhaps UX practices cannot be completely Agile to be implemented and successfully fit into Agile software development or is there another 'Agile' solution. This will be analysed further in this research study.

- **Inclusivity of UX experts in a software development team**

Another user states that having UX experts included in the development team aids to increment the Sprint and provides a bird's-eye view for the rest of the team once the Sprint has started. The business approves the results from that phase of the UX process. The same user stresses that the risk is following inherited ways of working as being the norm. However, teams do not want to fall into the trap of saying they are Agile but still adopt somewhat traditional ways of building software.

- **Cross-functional teams**

A user commented and suggested that cross-functional teams can solve Sprint impediments, through the guidance of the Scrum master when UX experts are part of the development team. For instance, team members with sufficient skills can be paired on an impeded task. In the example provided, a Developer under the guidance of a UX expert can assist build wireframes which can be done same time. Equally, UX experts can support Developers with testing or reviewing the user interface during the development process. Overall, the team can gain more from both roles, which can result in a decline in velocity. The same user suggests that UX may have a big role to play in Lean Start-up approaches, and in swift design testing.

- **Agile-UX relationship**

Lastly, on one of the forums, there is a user who suggests an article on understanding the Agile-UX relationship. The article clarifies the variances between the two practices and their fundamental existence. It offers a very motivating perspective of the Agile-UX relationship from the UX side. The article begins by revisiting the foundation of Agile and how it has evolved from the traditional software development. It supports the nature of this research study.

The comments provided by users on these forums prove indeed that Agile-UX implementation is not an easy challenge but that there may be ways to combat these challenges. They also indicate the importance of UX inclusive Agile software development teams.

5.3 Conclusion

Based on the results from the participants, UX inclusive Agile software development teams are advantageous if all involved understand the importance of the integrating these practices. In addition, team members need to know their roles and responsibilities, company stakeholders need to be supportive and company culture and attitudes must comply. However, UX experts may find it challenging to adopt to

Agile practices as UX practices are not Agile by design. The technical competencies might not be on the same bar as the rest of the development team, and lastly, the UX role is not fully recognised in software development teams.

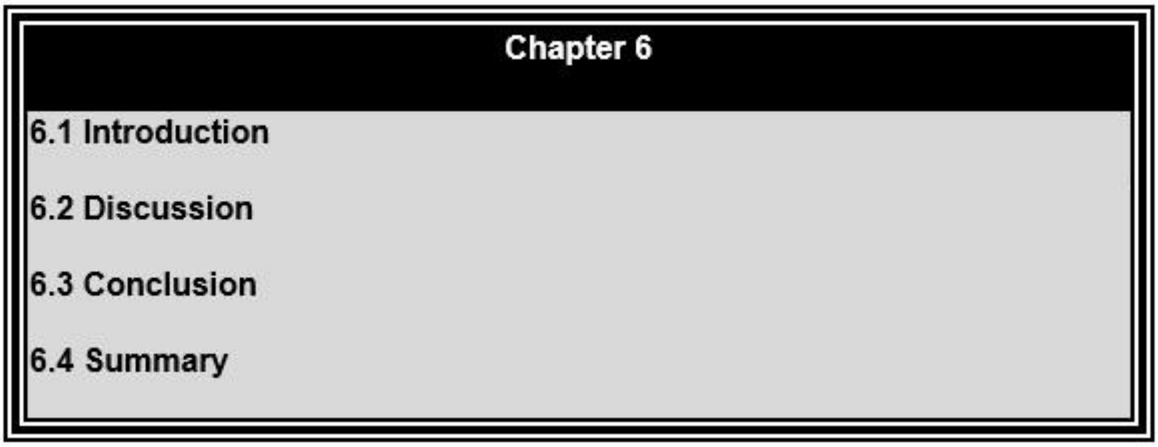
A minority of the participants are aware of Agile-UX. These are inclusive of UX experts during the early stages of the Sprint from which based on the results does not work successfully, especially to the UX expert or the overall team's advantage.

The results also indicate the importance of performing user-related testing such as UAT and Usability testing to ensure that user-related issues are picked up early during the software development process.

5.4 Summary

This chapter presented case study findings from semi-structured interviews and surveys conducted by SD teams in the Western Cape Province of South Africa. Members of the software development teams shared their insights on SD processes and the different ways in which these processes can be approached. Lastly, common themes from the findings of the netnography analysis were presented. The following chapter provides a discussion of the case study findings in detail.

Chapter 6: Discussion of findings



Chapter 6

6.1 Introduction

In chapter 5, case findings derived from interviews and surveys were presented. This chapter provides an in-depth discussion on these findings with the aim to apply them in the formulation of the final model. Sub section 6.2.1 presents the demographic data discussion, Agile-UX software development in sub section 6.2.2 and netnography results in sub section 6.2.3. Lastly the chapter offers a conclusion (6.6) and summary (6.7).

6.2 Discussion

The semi-structured interviews highlighted in sub-section 5.2.1 were conducted with members of SD teams that contribute to UX practices in their respective teams. SD team 1 uses the Waterfall approach to develop software systems. The findings from this case study indicated that teams that still use traditional ways of building software have an interest in transitioning to Agile-UX methods to restructure processes and improve team culture. Furthermore, SD Team 2 uses the Agile approach but identified the need to refine processes to enable better management of the development of software systems. Additionally, the knowledge obtained from these interviews and the literature collected for this study was used to formulate a survey. Both the semi-structured interviews and the survey were aimed at determining how SD teams approach software development.

The survey mentioned in section 5.2.2 comprises two sections, namely, the demographical data and the Agile-UX software development process. The demographical data has contributed to gathering the users' basic information. The information determined the users' roles in the software development team, years of experience in the field, the type of service the organisation they work for offers and who their users are. The second part aims to gather information about the users' awareness of Agile-UX methods to understand approach to software development projects. Each section will be discussed below.

6.2.1 Demographical Data Discussion

Age

As sub-section 5.2.2 states, most of the participants are between the ages of 18– 30 and 31−40, both with 44,4%. The age group between 41–50 years reflected 11.1%. Lastly, the survey results showed no participants from age 51 years and above. The results show that most of the participants are young IT professionals based in the medium age grouping, as indicated by the findings.

Gender

As stated in section 5.2.2, of the 35 participants that responded to the survey, 58.8% were female, and the remaining 41.2% were male. Existing research shows that males dominate the IT profession (Moloko, 2022). However, the results of this research study do not indicate the gender split in the IT profession but the participants of this research survey. Software development teams may comprise both genders, and the split will be different per team and organisation.

Years of working experience

As mentioned in sub-section 5.2.2, participants with the most years of working experience were those that had been working in the IT profession for 7-9 years and 10 years plus, each reflecting 37.5%. The remaining 25% had worked for 4–6 years. These results indicate that the participants of this survey have solid experience in the IT profession.

Roles

In sub-section 5.2.2, participants were asked about the roles they occupy in their various organisations of work. The results indicate that out of the 35 participants that completed the survey, 12 were Test Analysts, 5 Developers, 4 Business Analysts, 4 UX Experts, 3 Technical Leads, 2 Scrum Masters, 2 Project Managers and 1 Manager. The results indicate a good balance among the roles required to gather information for this research study. 3 participants did not answer this question.

6.2.2 Agile-UX software development discussion

This sub-section discusses the participants' results based on their approach to software development by analysing various elements such as experience, awareness, contribution, challenges, and benefits of Agile-UX software development and the role these various elements play in the success of Agile software development projects.

6.2.2.1 Awareness of Agile-UX methods discussion

According to sub-sections 5.2.1 and 5.2.2, the focus of the questions was to assess the participants' knowledge, skills, and experience in both Agile and user experience. These questions are discussed further below.

What type of business or service does your organisation provide?

As mentioned in sub-subsection 5.2.2, 18 of the total 35 participants are from the Finance and Professional Services sector. 10 belong strictly to the Finance sector, 3 were from Commerce, 1 in Education, 1 each in Media, Culture and Graphical, 1 in Transport and the remaining 1 in the Food, Drink and Tobacco sector. Overall, the results indicate that most of the participants in this survey belong to the Finance and Professional Services sector.

Who are your users?

Based on sub-subsection 5.2.2 and the findings discussed in the previous section, the results indicate that the participants in this survey service a wide range of users. These include advisers and investors, internet service providers, import and export, banks and payment providers, editors and news readers, tax institutions, debtors, students, members of the public, gamblers, business members and online shoppers. These results align with the results in the previous section, as expected. Equally, they represent the various sectors that the survey participants are in. They are expected to align with the type of users for whom they intend to build software systems.

What are your team dynamics?

A minority of the teams that the participants of this survey are part of, as indicated in sub-section 5.2.2, are not inclusive of UX experts. These include 12 of the 35 participants. However, the remaining 23 participants have UX experts in their teams and are from companies that build user interface related systems. These results indicate that some existing software development teams are not UX inclusive owing the type of software systems they build. These results are scrutinised further in this case study. However, the findings show gaps within the Agile software development methodology that should be addressed. In this research study, the model produced at the end achieves this.

Which of the following software development methodologies does your team use?

As indicated in sub-subsection 5.2.2, 20 participants indicated that the most widely used software methodology is Agile. Thereafter, 5 participants each, respectively said that Hybrid and Scrum were and, lastly, 2 participants indicated Waterfall. This confirms existing research that indicates that Agile is the most widely used methodology in software development (Section 1,3, 2,2 and 2,2.3.1). 3 participants did not respond to this question.

Which methodology do you prefer?

In sub-section 5.2.2, participants were requested to indicate their methodology of preference. The results show that 57.6% of the participants preferred Agile, as seen in the previous sub-section. This is followed by 9 participants or 1% preferring a Hybrid

methodology, 6 participants or 1% Waterfall, and then 3% of the group of participants indicating Hybrid UX, Extreme Programming and UCD. This confirms that
Agile methodology is the most popular but indicates that the participants were not aware of the other existing software development methodologies. This could be owing to little or no exposure to these methodologies.

What do you understand by Agile?

Participants were requested to indicate their understanding of Agile and this is discussed in sub-subsection 5.2.2. The results show that most participants had similar responses highlighting a good understanding of Agile. 99% of the participants understand Agile to be a modern-day, iterative development methodology that utilises Scrum, which provides the ability to build successfully and respond to change under unpredictable circumstances. These results have proven to be accurate as the previous results indicated that most of the participants had been exposed to the Agile software development methodology. However, the remaining 1% was unsure of the Agile method. This could be due to their not having had any exposure to the methodology. This research study aims to address this.

What do you understand by user experience?

As indicated in sub-subsection 5.2.2, most of the participant's responses indicated that user experience refers to designing systems from a user's perspective. These results show that the participants fundamentally understand user experience, which supports the purpose of this research study.

6.2.2.2 Contribution of UX methods to the success of Agile software projects

It was important to gather information from the participants about their understanding of the UX contribution to the success of Agile software projects. They would have obtained this understanding through their involvement in the process, or their perceptions based on their experience. These results presented in sub-section 5.2.2 determine the impact that UX methods portray in the effective progression and delivery of Agile software development projects. The questions and discussions collected from the participants responses are presented below.

Who is responsible for UX design in your team?

Participants were requested to indicate the person(s) responsible for UX design in their respective teams, as discussed in sub-subsection 5.2.2. The results indicate that out of 35 participants, 13 said UX Designers, 4 said Business Analysts, 4 said Business Analysts and UX Designers and, lastly, 4 indicated that they had no representation of UX experts in their teams. This could be due to the nature of the product or service developed by the respective organisations not requiring any UX involvement. 2 participants indicated that they no longer had a UX team, and that Developers and Business Analysts had assumed that role. 1 participant said they didn’t know, while another said they had a combination of UX Designers, Business Analysts, and clients responsible for UX design in their team. Lastly, another participant indicated that both the product team and the client take up the UX role in their team. The results indicate that UX Designers are mostly responsible for UX design in Agile software development teams. Furthermore, they also indicate that Business Analysts take on UX design tasks and, in some instances, in collaboration with Developers. This is fairly justified as Business Analysts interact with users to solicit requirements and are the best people to undertake UX design tasks. However, they are not UX Designers and do not go through the UX design process. Again, the nature of the product or system under development might not require rigorous UX design involvement. UX Designers would be better suited for carrying out UX design tasks. Lastly, the collaboration between Business Analysts, Developers and UX Designers is vital as the information gathered from the UX design process is used by Developers to design and write code while Business Analysts interpret the requirements to both UX designers and Developers. Therefore, it would be best for UX Designers and Business Analysts to work hand in hand with Developers. This emphasizes the team collaboration UCD, and AD element mentioned in section 3.3.6 in Chapter 3.

What would you say are the factors that contribute to the success of software development projects in your team?

The participants were requested to identify the factors that contribute to the success of software development projects in their teams. As presented in sub-sections 5.2.1, 5.2.2, the results indicate the highest-ranked factors are:

Team collaboration and communication
Client inclusion
Usability testing
Flexibility
Clear requirements
Quality assurance

For this question, participants could select multiple answers. Therefore, 24 out of the 35 participants selected team collaboration and communication, client inclusion and usability testing. 17 chose flexibility, and clear requirements, and 7 quality assurance. 5 participants selected lack of technical knowledge, and 4 said UX experts would find it difficult to adapt to Agile. The results identified an interesting outcome in that the focus on UX was ranked the lowest. Yet, the participants identified one of the related factors (client inclusion) as one of the highest contributors.

Additionally, SD Team 2 in section 5.2.1 indicated client inclusion and collaboration as the advantages of using their Agile-UX inclusive approach to software development.

Identify or mention any existing successful Agile-UX models that you are aware of below:

As indicated in sub-section 5.2.2, 39.4% of the participants confirmed that they were unaware of any Agile-UX models. 33.3% said they were aware of Agile-UX models, while 27.3% said they were aware of Lean UX and the remaining 3% said they were aware of design Sprints. Lean UX is viewed as an Agile-UX model that focuses on team collaboration and customer feedback. Therefore, the results show that 67% of the participants are aware of existing Agile-UX models while the remaining 30.3% are not.

Furthermore, in sub-section 5.2.1 both SD teams indicated UX awareness. The representation of the participant's awareness could be due to their participation, exposure to these models or lack thereof.

Mention any organisations that adopt any of the models mentioned above in the text field below:

According to sub-section 5.2.2, the results show that 4 participants mentioned Amazon, with 1 each identifying Netflix, Google, Visa, Wonga, Clickatell, Maxxor, Apple, IBM, Ipreo, Allan Gray and Woolworths as the organisations adopting Agile-UX models. 1 participant identified start-ups as organisations that adopt Lean UX models, another mentioned Discovery, and yet another mentioned Vodacom and PPS as other organisations. Lastly, 1 participant said they were unaware of any, and the remaining 14 participants didn't respond to the question. This is possibly due to the participants not being aware of any such models as well.

Where are those organisations located?

The results shown in sub-section 5.2.2, indicate an even split of 50% of these organisations being global and 50% local. The UX experts are either internal to the team or organisation or outsourced externally. Existing research states that these UX experts usually work in isolation from the rest of the team (sub-section 2.4.1). The results presented in this section highlight the necessity for awareness of Agile-UX models in software development organisations.

What role do you play in ensuring that UX is successfully integrated in the Agile software development process in your team?

Based on the information in sub-section 5.2.2, the entire software development team has a role to play in guaranteeing that UX tasks are successfully integrated into Agile software development. The results indicate that some participants were not aware of how they contribute to UX inclusion in Agile software development. Only 23 out of 35 participants responded to this question. 1 participant said they were excluded from contributing, 4 said they played no role at all. The remaining 15 out of the 23 who responded mentioned that their contribution is through the respective roles they fulfil in the software development team. 5 of these participants were Test Analysts, 3 Developers, 3 UX Designers, 2 Business Analysts, and the last 2 participants were Project Managers and 1 Product Owner. The remaining 12 participants didn't answer this question, possibly due to them not being sure or aware of how they play a role. Again, this solidifies the need to provide awareness.

What would you say are the pros of having UX experts working as part of the Agile software development team?

Based on sub-section 5.2.2, the participants were allowed to select more than one response. The results indicate that 34.4% of the participants collectively selected active and frequent communication between UX experts and the development team members regarding project progress and issues as a major advantage. 21.9 % of the participants mutually agreed that the inclusion of UX experts is one of the pros. 15.6% of the participants selected clarity on roles and responsibilities and the inclusion of UX experts. 12.6% collectively selected communication between UX experts and the development team members regarding project progress and issues. 9.4% of the participants collectively selected clarity on each other's roles and responsibilities.

Active and frequent communication between UX experts and development team members regarding project progress and issues, clear requirements and clarity about each other's roles and responsibilities are all vital characteristics as they are a running theme within the case study findings. Therefore, subject experts and Business Analysts won't need to play this role.

The 6 advantages highlighted by the participants above define the impact that Agile-UX methods contribute to Agile software development.

Which factors play a role in the successful implementation of Agile-UX methods?

The results presented in sub-section 5.2.2 were from multiple-choice questions where participants could select more than one answer. Therefore, these results highlight the importance of integrating UX and Agile practices as selected by 28 out of the 35 participants. Furthermore, buy-in from stakeholders was selected by 20 participants, company attitude mentioned by 18 and, lastly, understanding roles and responsibilities selected by 16. Stakeholders buy-in is important as they make the decisions in the best interests of the organisation (section 1.2 and sub sub-section 3.3.3). These decisions flow from the top-down to the respective teams that make up the organisation.

Does your team perform UAT or usability testing?

In sub-section 5.2.2, the results indicate that 27 participants perform User Acceptance Testing (UAT) or usability testing in their teams, while the remaining 8 do not. These results may be due to the nature of the organisations for whom these participants work. Each organisation runs their processes differently based on the type of product or service they provide.

If YES, how effective is it?

Based on sub sub-section 5.2.2, the results indicate that 13 participants find UAT extremely effective, 7 just effective, and 7 fairly effective, while the remaining 8 participants didn't respond to this question. In conclusion, UAT can be beneficial for software development teams as the intended users interact with the product or service to confirm that it has been built according to requirements before the final version is deployed to users. This gives the team time to resolve any issues that may come back from the UAT test run before the final version is deployed, ultimately ensuring that high-quality software products or systems are built.

How often does your team encounter production issues that would've been identified in the software development process had users or UX experts been involved?

The results in sub-section 5.2.2 indicate that 20 participants agree that the postimplementation issues they encounter are less likely due to not including UX practices in the software development process. 7 had an unbiased response, 6 leant towards quite likely, and 1 participant was in total agreement. 1 participant did not answer this question.

What recommendations do you have for Agile teams that want to integrate user experience into their software development process?

In sub sub-section 5.2.2, the results presented show the recommendations to effectively integrate UX into Agile software development teams that participants have shared. 27 of the 35 participants responded to this question. Most of the responses indicated that UX inclusive Agile teams, communication and collaboration between team members, stakeholder buy-in and adaptability within the software development process. All these factors have been mentioned in previous responses where participants were asked to share the advantages of UX inclusive Agile software development teams. 8 participants did not answer this question. Therefore, the following guidelines have been identified for Agile-UX software development teams.

Understand UX importance: Understand the importance of UX by working closely with the UX team.

Demo/showcase UX/UI topics: Hold UX/UI talks and demos every second week during the SDLC process to ensure Developers do not deviate from the initial designs.

Prioritise UX: All systems are geared towards the user, therefore, UX should be made a priority or Developers will not be able to develop without the designs from UX experts.

Ensure the team has enough UX work: Teams should ensure they have enough UX tasks. This can be achieved during Sprint planning.

Monitor users in UAT: Monitor and guide users during UAT, so that we can make sure valid issues are picked up and raised.

Team collaboration and communication: Open communication and collaboration is encouraged throughout the software or product delivery process. Have open conversations with UX Designers and understand where they fit into the team and which meetings are relevant for their involvement.

Write proper user stories: Software development teams should write user stories inclusive of UX designs and not just focus on the functional requirements.

Flexible software development teams: Allow for flexibility within the development process by encouraging change, but still allow the process to continue effectively.

Include UX Designers and UX leads in software development teams: Invite UX Designers and their superiors to team meetings and include UX design tasks in Sprint planning. UX and Agile share similar values, therefore they should work in collaboration. It helps to have a 'Champion user' involved throughout the product delivery process.

Involve Technical Developers in UX design sessions: This aids to understand and resolve any technical constraints.

Assign UX tasks to each user story or ticket: All tickets should be assigned a related UX ticket or subtask so that UX is aware of all features and improvements. The UX Designers work is pulled into the Sprint so that it is visible, and they form part of the team ceremonies.

Streamlining design and developer tasks**:** Teams will often rush through a UX Sprint to not delay the following Sprint. Sometimes the Sprint can be delayed when working on UX tasks. Working upfront and having to integrate back into the current Sprint causes context switching. All these factors can lead to time wasted. Therefore, the UX team must be managed well in their Sprint as well as the development team in theirs. Often there is a Developer Manager managing the development team but no UX Manager managing the UX team. Where this happens, the UX and Developer Managers would ensure that both teams are aligned and work effectively. However, smaller teams are less likely to have UX Managers.

6.2.2.3 Discussion on challenges when integrating Agile and UX methods

The previous sub sub-section discussed the contribution of UX methods to the success Agile software projects. It is equally important to assess the disadvantages and find an effective SD approach. These disadvantages are discussed further through the responses to the question below.

What would you say are the cons of having UX experts working as part of an Agile software development team?

In sub-section 5.2.1, SD Team 2 identified UX designers working a sprint ahead and short tight sprints as a challenge.

The same question was answered in sub-section 5.2.2. 2 of the 35 participants couldn't identify any disadvantages in answer to this question. It is a multiple-choice question, and the results show that 14 participants agreed that the disadvantage is the UX experts' lack of technical knowledge, and 14 said that UX experts could find it difficult to adapt to Agile. 4 participants said that there is a lack of comprehension of the importance of integrating UX and Agile practices. Furthermore, it was mentioned that Sprints are very developer-focused and can waste UX Designer time in Sprint meetings that focus on development work and not what the UX Designer is currently working on. The results indicate that a process to streamline both processes, allows UX experts and Developers to work together effectively. Lastly, the results highlight the importance of a fundamental comprehension of the UX role in Agile. Therefore, it is important to get stakeholder buy-in as organisational decisions flow from top-down.

6.2.3 Netnography Results Discussion

As mentioned in section 5.2.3, the comments indicate the common challenges that UX experts face in Agile software development teams of integrating their tasks with them. This section also presents the best methods to tackle these challenges, which are discussed below:

a) Integrating UX into Agile is a challenge

In section 5.2.3, the users identified the challenge of including UX into Agile software development teams. A user highlights that UX Designers find it difficult to execute their tasks in a Sprint environment as Developers would have to wait on mockups/prototypes to begin with their development tasks. One other user identified another difficulty that UX Designers have in that most of the time is spent performing user research, hence the solution to work Sprints ahead. This solution, however, is seen as that of a Waterfall approach. Another user shares their experience and confirms that the Agile-UX method worked best when the team worked two Sprints ahead. This allowed for user feedback and for one Sprint to adapt to any UX changes before Developer involvement. Ultimately, UX Designers in Agile-UX teams would have to work a couple of Sprints ahead depending on how long the research phase of the process takes for it to work effectively.

b) UX experts working two Sprints ahead a 'Waterfall' way of work

According to section 5.2.3, a user was experiencing challenges with UX Designers working in the same Sprint as the rest of the development team. This is due to UX Designers having to change prototypes, a challenge if done in the same Sprint, as Developers would have to use those prototypes to build their code. The solution would be for UX Designers to work a Sprint or two ahead of the rest of the development team. Furthermore, UX Designers need to be available for Developers to clarify any uncertainties they may have regarding the prototypes. Ultimately, this user wanted to know if there was an alternative approach for an Agile-UX software development team. This would ensure that Developers use up-to-date user mock-ups or prototypes. The alternative approach would have to ensure that these prototypes are demonstrated. Lastly, the approach would have to ensure that Developers do not duplicate UX effort after UX Designers have worked Sprint(s) ahead, as this can be viewed as a 'Waterfall' approach. Another user has stated that according to their understanding, items in Scrum should start and end in the same Sprint. A user story sometimes cannot be completed in one Sprint. This can be due to issues not being resolved timeously, and over-estimation of effort on a story during the planning phase, and dependencies due to a shortage in resource capacity. In such situations, items or user stories would have to be carried over to the following Sprint until completed. Therefore, the working Sprints ahead solution can work effectively.

c) Inclusivity of UX experts in the software development team

According to section 5.2.3, a user mentioned that including UX experts in the development team would be an advantage in incrementing the Sprint and providing an overall view for the rest of the team when the Sprint has started. The outcomes of that phase of the UX process would normally be approved by the business stakeholders. The same user also highlights the risk of following inherited practices due to force of habit, as this doesn’t allow the team to adopt new, efficient, and effective ways of working, which ultimately enables team growth.

d) Cross-functional teams

In section 5.2.3, one user proposed that cross-functional teams can solve dependencies during the Sprint through the supervision of the Scrum Master and if the UX experts were inclusive of the development team. Team members with appropriate skills and knowledge would be paired on a blocked Sprint item. Likewise, UX experts can support Developers with testing, confirming mock-up data or reviewing the user interface during the development process. In this way, team members can learn a lot from each other, develop their skills and knowledge and ultimately become a stronger team.

e) Agile-UX relationship

According to section 5.2.3, a user recommended an article on understanding the Agile-UX relationship. It explains the differences between the two practices and their core purposes. The article provides motivating insight of the Agile-UX relationship from the UX perspective. It starts by re-evaluating the foundation of Agile and how it has advanced from traditional software development. The information it provides affords an opportunity to learn how Agile and UX can co-exist, which supports the purpose of this research study.

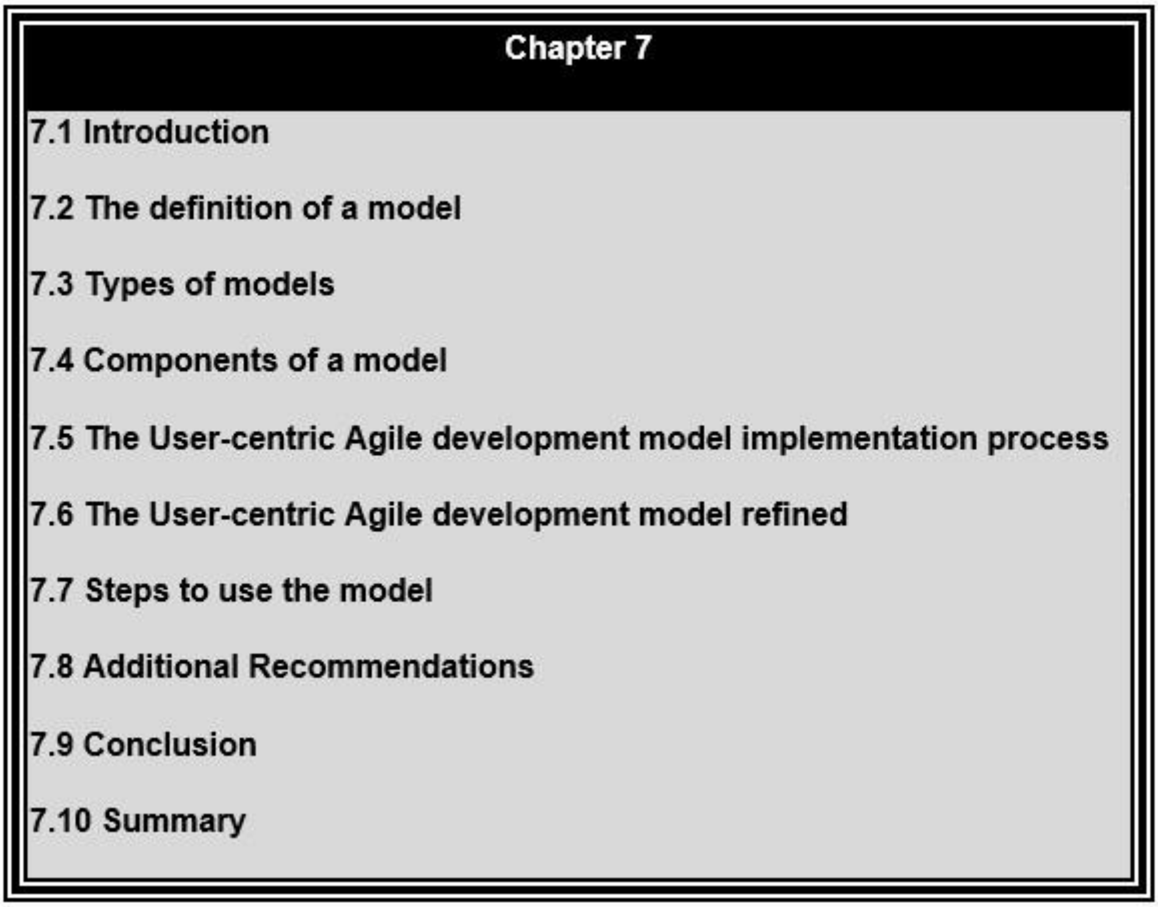
6.3 Conclusion

This chapter presented a discussion of the case study findings highlighted in sections 5.2.1 and 5.2.2 presenting results acquired from the survey and interviews conducted in 2021. Sub-section 3.3.6 described the literature review that addresses the impact and limitations of user-centred Agile development, and lastly, section 5.2.3 presented the comments obtained from online forums in netnography, also written in 2021. The results obtained from these various sources indicate common findings. For instance, a significant number of Agile software development team members are not aware of Agile-UX methods. Some are not aware of how team members contribute to ensuring user-centred software systems or products in their respective roles. Furthermore, it has been established that stakeholder buy-in is important to ensure that software development organisations are aware of the importance of UX inclusive Agile teams, as decision-making flows from the top-down. The comments highlighted through netnography results have described the challenge that exists with having UX and Agile tasks happening in the same Sprint. Lastly, cross-functional software development teams are encouraged to improve team and organisational culture. Agile-UX software development teams should strive to continuously improve and refine the process by ensuring that user-centred Agile software systems are developed.

6.4 Summary

This chapter has discussed the findings of the case study presented in Chapter 5 in detail (See sub subsection 6.21, 6.2.2 and 6.2.3). These findings were used to gain insight on the approach SD teams use to develop their software systems and formulate the model required for this study. The final model is presented in the following chapter together with the model development process used as guidance to formulate it.

Chapter 7: Formulation of model



Chapter 7

7.1 Introduction

The previous chapter presented the findings of the case study. The purpose of this chapter is to explore the model development process and propose a user-centred Agile model for this research study. Section 7.2 defines a model followed by types of models in section 7.3 and model components in section 7.4. The model implementation process is presented in section 7.5 followed by a refined version of the model in section 7.6. Section 7.7 describes the steps to use the model followed by additional recommendations in section 7.8. Lastly, the chapter conclusion and summary are presented.

7.2 The definition of a model

The Open University (2022) defines a model as a depiction or view of a system or object in the real world. Serc (2022) refers to a model as an abstract concept to aid in understanding true-life structures. Commonly, models have data input, processes, and output in the form of expected outcomes.

Often, software processes and software life cycle models are mistaken and considered to have the same meaning. The two can be used interchangeably as they both encompass the development of software systems (Kneuper, 2018).

Sulemani (2021) and Kneuper (2018) define software process models as a concept of the software development process. These models signify a structured set of events, entities, transformations, and actions that encompass approaches for implementing software solutions. These models are often used to create simple, formal details of software life cycle activities. Their value stems from their application of adequate influential representation and syntax, appropriate for process handling.

A software life cycle model, however, is a model that defines how to develop software. These can be descriptive categories (used to describe the history of how software was built) or prescriptive (used to define how to build new software systems). Prescriptive models are said to be easier and simpler for creating software, as they are suitable for encompassing development activities and practices for using various software engineering tools and environments during the development phase of a project.

Kneuper (2018) further shares that software life cycle models define the key steps in the software life cycle and their sequence, the software development life cycle in particular. Furthermore, both software process and software cycle provide a structure for different actions, outlining when and how to implement these actions, identifying the roles involved and the expected outcomes.

This research study addresses the Agile software methodology software life cycle model described in sub sub-section 2.2.3.1 in Chapter 2 and The User Experience Design process model identified in sub section 3.2.1 in Chapter 3. This study aims to identify a smart approach to successfully integrate the software life cycle with user experience practices to formulate a single efficient and effective model. Therefore, it is important to understand models in general, the software life cycle and process models and what it entails to construct them successfully. This chapter sets out to address all these elements.

The following section provides in-depth knowledge of the types of models that exist. Each type is structured specifically to provide a solution to an existing problem. The following section (7.3) presents knowledge on existing types of models.

7.3 Types of models

Each model should suit its intended purpose. In software development, the purpose of the project is driven by the type of project suited for the team heading it. The problem, project features, type of team and culture of the company heading the project are some factors to consider before selecting the type of model to adopt (Edmonds et al., 2019). Serc (2022) highlights four modelling types namely:

Conceptual models: These are qualitative models that identify relevant relationships in factual systems and procedures. They are utilised as the first phase in the creation of complex models.

Interactive lecture demonstrations: This model is a physical representation of a system that can be simply perceived and used. It has comparable features to the features of the system in reality. These aid to understand the difference between the conceptual and complex reality of systems.

Mathematical and statistical models: These models involve resolving the significant equivalence of a system constructed upon its statistical factors (e.g., mean, mode, variance of quantities). Some mathematical models are analytical and numeric. Statistical methods are beneficial for finding patterns, trends, and relations between data groups.

Visualisation models: These models help to envision how a system functions. A visualisation model can be a relationship between data and a specific, realistic, or graphic result, which, in some cases, can be associated with other model types to transform its output into a valuable visual format.

A conceptual model was utilised to formulate the model for this research study. It is a dependant model type with the purpose of highlighting appropriate connections and interactions between systems and the events that occur within them. Understanding the software process and software life cycle and the structures that make up these models is vital. It is important to understand the elements required to create these models. The following section (7.4) discusses the components that models comprise.

7.4 Components of a model

A model component defines the order of efficient and practical elements linked to related tasks, interactions and how all these elements integrate and interact with one another to meet the overall objective. Therefore, a model comprises of stock, flows and converters. These are explained below (Systems Thinking Handout, 2008):

Stock: This is referred to as the vessel for collecting something such as water, persons, or interest.
Flows: Flows update or interact with the stock. Flows vary and without flows a system would remain stagnant.
Converters: Converters update or interact with the flows. They direct the path of the activity symbolised by the flow.

How to distinguish between a stock, flow, or converter?

Systems Thinking Handout (2008) answers these questions in the three questions and answers below:

Q1. What would be the main actor in the story or problem being solved?

A1. The response should be a noun; hence the core component is referred to as stock

Q2. Which activities impact the main component or actor?

A2. The response should be several verbs. These then become the 'flows' of the model Q3. What challenges influence the path of the actions?

A3. The response should be an adverb which is a 'converter'. Converters guide the actions characterised by the flows in the model.

The model needs to be scoped in a way that supports its purpose. Furthermore, its quality is determined by the terms of its adherence to modelling strategies and the degree to which it meets its intended purpose.

Models may provide a great deal of information, for example:

Evaluate progress
Estimation of effort and fees
Evaluate technical quality and risks
Evaluate model value

Lastly, models may also be created to provide extra metrics associated with its resolution (Sebokwiki, 2022). The model for this research study comprises the elements defined below:

Stocks: The software development team; Developers, Business Analysts, Test Analysts, Project Managers, Product Owners and UX experts.

Flows: All the activities required to be completed in the development of a software system. These are generally iterative activities, from requirements gathering, design, coding, testing and deployment, planning, research, testing and launching UX designs. Both are continuous processes, and once the process ends, it then restarts all over again.

Converters: The predecessor of each activity needs to be completed before the following one can start. Furthermore, dependencies such as working with third parties may also delay the process should issues arise from a third party.

This section has defined and discussed software processes and the software life cycle, model types and model components. The following section (7.5) discusses the model development process adopted for this research study.

7.5 The User-centric Agile development model implementation process

As already established in the previous sections, a software process model is an abstract representation of the software development process (Sulemani, 2021). Research findings by Biecek (2019) defines model development as a repetitive process where models are formulated, verified, and redeveloped until a proper model is formulated. Based on findings identified by these authors: Iseesystems (2021) & Biecek (2019), basic steps for creating models are the same across all modelling techniques.

A model is designed to create and define a method to effectively incorporate the Agile software development model with the UX design process. The latter is generally isolated from the rest of the software development life cycle. The incorporation ensures that user-centred systems that meet user requirements are built. Furthermore, UX Designers are integrated into the development process to ensure that Developers know about UX tasks throughout the development process. Likewise, UX Designers simultaneously have visibility of Developer tasks and progress as highlighted in Section 1.2 in Chapter 1. Ultimately, the nature of the integration ensures that successful software development systems are created. Owing to the success of software development systems being measured by how effectively users can utilise them, it is important to ensure that these systems are user-centred (Biecek 2019).

As already established, the model follows a conceptual approach, which is a representation of the key relationships between real-life systems. Given that the software development life cycle is the type of software process dealt with within this research study, the factors mentioned above align with the model created for this research study.

The model development process is a repetitive one as it is necessary to verify that the model meets its intended purpose. This is the first and most crucial step in the model development process. The process itself defines the criteria for developing a beneficial, successful model. The conception of the *User-centric Agile development model* is described in the sub-section (7.5.1).

7.5.1 The Model development process

This research study followed the process highlighted below, informed by the seven guidelines for creating effective models to create the *User-centric Agile development model* (Iseesystems, 2021):

Step 1: Purpose of the model

As established in the section 7.5, the purpose of this model is to outline an approach to effectively incorporate the Agile software development model with the UX design process for use by software development teams. This model ensures that user-centred systems are built with Agile software development and UX design processes running concurrently. It is suggested in section 1.2 that Agile and UX are equally important processes that warrant effective UX. It also encourages a culture of learning and development through cross-functional teams as UX and the development team would be working closely together as mentioned in the case study findings in sections 5.2.1, 5.2.2 and 5.2.3 in Chapter 5. Emphasis is on the outcomes of the system rather than focusing solely on the deliverables. Babich (2018) states that the purpose of Agile-UX is to introduce a continuous method to design and develop features through teamwork and the supervision or stakeholders of users providing feedback. Furthermore, it has been identified in section 5.2.2 that buy-in from stakeholders is crucial for Agile-UX implementation.

Step 2: Define model limits

The elements that this model comprises are those defined in both the Agile and UX design processes in sections 2.2.3.1 and 3.2.1 respectively. However, the aim is to integrate the two effectively to meet the intended purpose as indicated in section 7.3. The purpose of this model is to include UX designers and intended users in SD process (section 1.2 and 1.3) to ensure user-centred systems are built. A user-centred approach places the user at the centre of design and development processes allowing developers to understand user needs while building useful systems (section 2.3.1).

The final model will be a combination of the elements depicted in the processes in Figure 7.1: the UX design process utilized by UX Designers and Figure 7.2: the Agile development process which is an SDLC approach that involves the rest of the software development team. The type of members can differ per organization and per project.

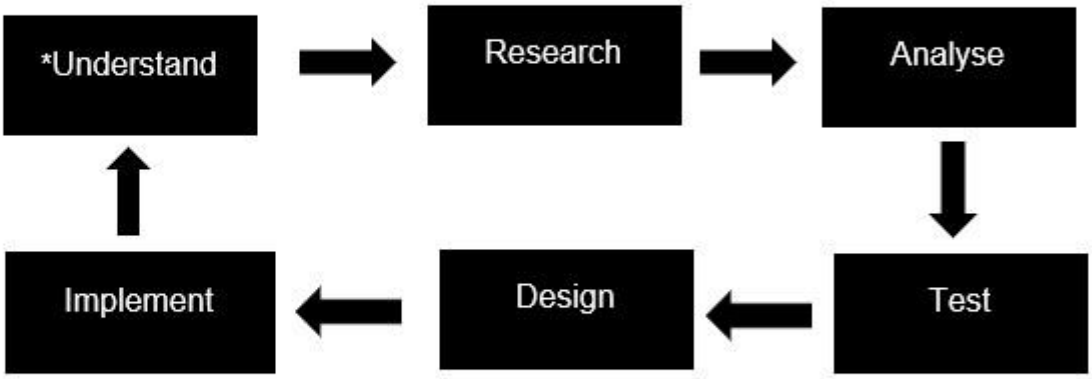


Figure 7.1: UX design process (Source Researcher)

*Marks the beginning of the process

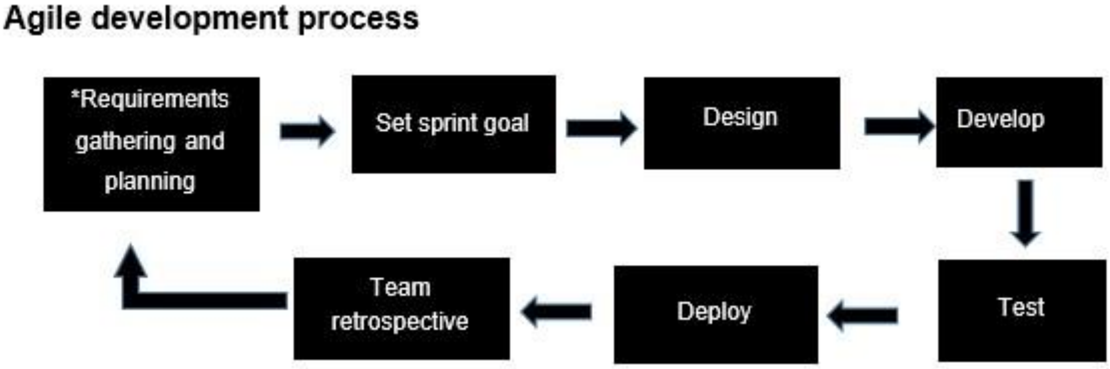

Agile development process
*Requirements gathering and planning
Set sprint goal
Design
Develop
Test
Deploy
Team retrospective

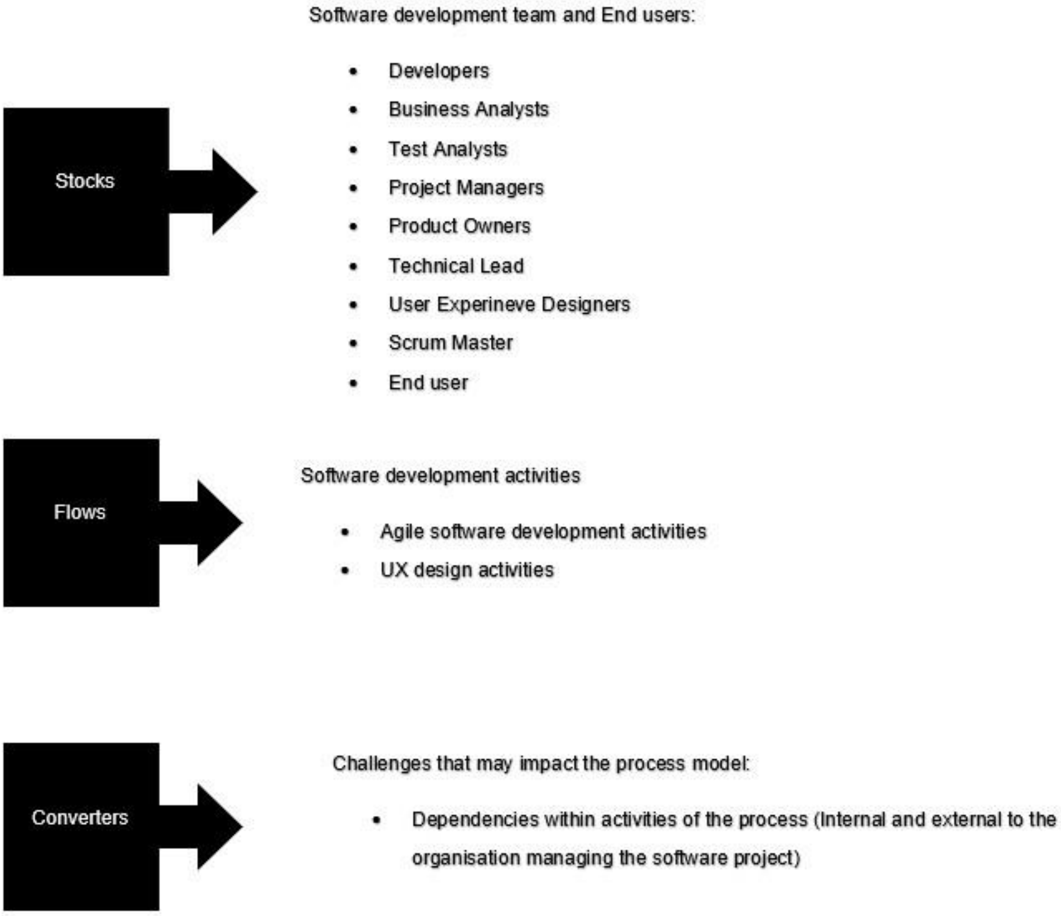
Figure 7.2: Agile development process (Source Researcher)

Step 3: Diagram and plan the model

Figure 7.3 outlines the elements of which the final model comprises. As mentioned in the section above, these elements are from the Agile and UX processes, which emphasise the inclusion of users by integrating UX experts in the Agile development process.

These elements are grouped into:

- Stocks: Persons or Users. In this case these are members of the software development team
- Flows: Process flows that the users interact with to keep the system active. These can be identified as the activities that form part of the Agile and UX processes
- Converters: These direct the path between flows. These are identified as successful completion of tasks within an activity that allow the flow to continue or dependencies or any impediments that hinder process flow.



Software development team and End users:

- Developers
- Business Analysts
- Test Analysts
- Project Managers
- Product Owners
- Technical Lead
- User Experineve Designers
- Scrum Master
- End user

Software development activities

- Agile software development activities
- UX design activities

Challenges that may impact the process model:

- Dependencies within activities of the process (Internal and external to the organisation managing the software project)

Figure 7.3: Model components (Source Researcher)

Model components:

The process is grouped into two Sprints, which are defined by Rehkopf (2019) as short-term periods where a Scrum team decides to complete a set amount of work. He describes these as the centre of Scrum and Agile methodologies. The more efficient the implemented Sprints are, the better the team implements the rollout of software systems to end users.

Design Sprint

As outlined in sub-sections 3.2.1 and 7.5.1 in Chapter 3 and 7 respectively, the first step is understanding the requirement and is the beginning of the design Sprint (referred to as Sprint 0). UX Designers need to **understand** the proposed problem. This can be done by collecting information from users in the form of interviews and brainstorming meetings to give the UX Designer a clear design direction. The second step is market **research** where the UX design team aims to find current UI and UX trends to design advanced products that meet user needs. The third step focuses on **analysing** the required features and how users experience and interact with the product through a continuous process of brainstorming and creating mock-ups. The outputs are used to enhance user experience. **Testing** happens in the following step, where outputs from the previous step are tested and shared with the business stakeholders for approval. **Design** is the next step where the final designs are created, and design specifications shared with the development team for insight. In this case it would be the UX team, technical lead and/or the Developers taking part in this step. Implementation is the last step within the design Sprint where the designs are officially handed over and are ready for Developers to use in the development Sprint to build code. This step involves both Developers and UX Designers as they need to support the Developers in this step should any issues or confusion arise.

Development Sprint

The following step focuses on the **requirements gathering and planning**. This is the beginning of the development Sprint (referred to as Sprint 1). As outlined in Chapter 3 and 7, sub-sections 3.3.1 and 7.5.1 respectively, this step involves deliberating and discussing project objectives in detail by the entire team including UX Designers. As mentioned in section 1.2 of Chapter 1, UX Designers should not be isolated from the rest of the software development life cycle. The requirements are discussed in detail for the development team to be clear on what needs to be done and converted into backlog story items that will be taken into the Sprint and worked through by the development team. These stories are analysed for risks, prioritized based on deadlines and estimated by the entire team and assigned to each team member based on their roles. This is the planning step which is followed by setting a Sprint goal that the team will work towards to ensure that the Sprint has been completed. The **design** step follows and focuses on discussing and analysing the technical requirements including the design specifications by Developers, Technical Leads, UX Designers and sometimes Test Analysts when required. These design sessions are important for Developers to understand and convert them into code. Once the requirements have been defined and understood, the following step is **development**. This step includes Developers and sometimes technical leads for support e.g., code review. Developers start building the first iteration of the code and are expected to deliver the entire system by the end of the Sprint. They are required to test their code before sending it through to the following step, **testing** (quality assurance). Testing is required after each iteration and Developers are responsible for resolving any issues that may come out the testing step until the software product is working as expected**. User acceptance testing** can also be included in this step where the expected users are given the opportunity to use the product and ensure it is working as expected. Once the testing step is completed, all issues fixed, and testers/users sign off the product to be working as expected then the software product is **deployed** to the intended users. No issues are expected to come up after this step, however, should any arise, they will be treated as production support and fixed by the development team in the following Sprint. The final step of this process is the **team retrospective**. This step incudes the entire team. The team discusses the past Sprint. The team unpacks the pros and cons of the Sprint. The outcome of the session is to find ways to work better as a team as well as continue with the processes that work. The team will also reflect on whether the Sprint goal has been met. This now marks the end of the development Sprint. Thereafter the design Sprint can start again marking the beginning of the following Sprint. See highlighted elements below in Figure 7.4 for reference:

Agile-UX design process

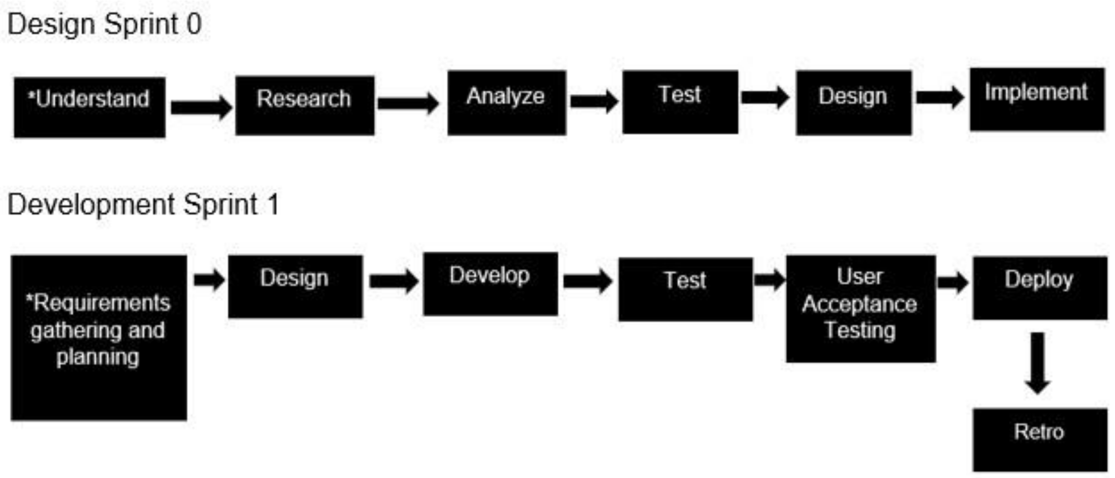


Figure 7.4: User-centric Agile development process (Source Researcher)

Step 4: Formulate final model: Figure 7.5 below is a representation of the first version of the proposed process model based on previous step (7.4) that has integrated Agile and UX processes.

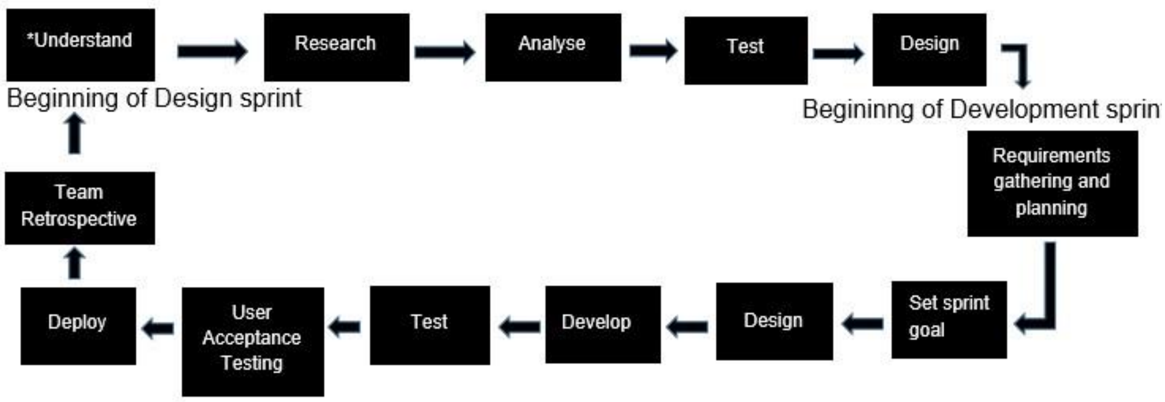


Figure 7.5: Formulated User-centric Agile development model (Source Researcher)

Step 5: Evaluate model

The model is distributed to UX experts, who form part of Agile teams, to test and evaluate before being distributed to Agile software development teams.

Step 6: Develop an interface

As defined in the sections above, the Agile-UX process is a combination of the UX and Agile process. The general actors participating in the entire process and each of their identified roles are described below. Figure 7.6 below depicts the second version of the proposed model process and the actors who interact within each phase. The software development team may be different depending on the project or type of organisation.

Scrum Master: The role of a Scrum Master is to facilitate Scrum activities and ensure that the team follows the Agile process that it has defined and decided to practise (Agile Alliance, 2021).

UX Designer: A UX Designer's role is to campaign for the end user during the development of software systems. UX Designers consider the overall user experience of the software system under development. They're there to ensure that the final product meets the users' needs. (White, 2020).

Project Manager: Miller (2019) defines the role of a Project Manager as that of preparing, organising, and guiding the completion of organisational projects, while ensuring that they are completed according to set deadlines.

Developer: Developers are responsible for building the actual software system and collaborate with the rest of the team about the progress and status of the software project (Swords, 2017).

Test Analyst: Swords (2017) also defines the Test Analysts' role as ensuring that the software system solution meets user requirements by analysing and scrutinising the software thoroughly. The final software solution should be bug-free after the testing phase.

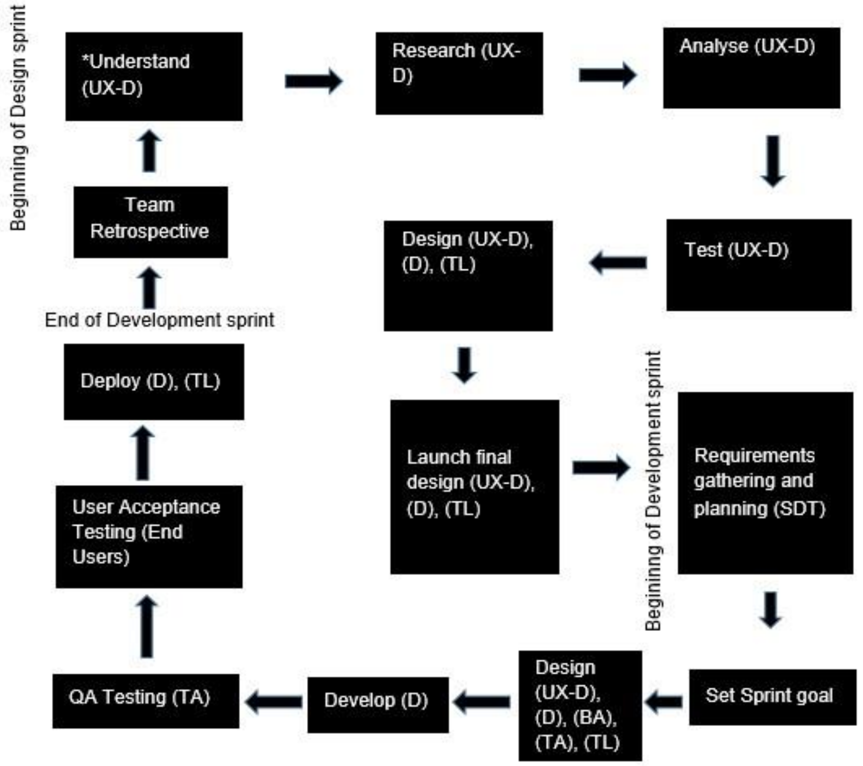
Business Analyst: Business Analysts interact with the main stakeholders to identify the project objective and requirements. They identify the problem and the solution to attempt to resolve it (Kowa, 2021).

Domain Owner/Product Owner: The Product or Domain Owners have similar responsibilities. They represent either the business or the end users. They are responsible for working with the users to determine which features or items need to be included in each Sprint for software development projects. Domain Owners are viewed as subject matter experts in specific fields and deal with defining and capturing technical requirements (Swords, 2017).

Technical Lead: The Technical Lead interprets the business requirements into the technical software solution (Swords 2017).

End user: The End User is the final solution provider or for whom the software solution is intended. Their role is to test out the built solution to ensure that it meets user needs. Testing takes place before the solution is deployed to all the intended users and would be the last phase of the development process (Swords, 2017).

Proposed User-centric Agile development Model


Beginning of Design sprint
*Understand (UX-D)
Research (UX-D)
Analyse (UX-D)
Test (UX-D)
Design (UX-D), (D), (TL)
Launch final design (UX-D), (D), (TL)
Begininng of Development sprint
Requirements gathering and planning (SDT)
Set Sprint goal
Design (UX-D), (D), (BA), (TA), (TL)
Develop (D)
QA Testing (TA)
User Acceptance Testing (End Users)
Deploy (D), (TL)
End of Development sprint
Team Retrospective

*Marks the beginning of a process

Actors:

- Scrum Masters (SM)
- UX- Designers (UX-D)
- Developers (D)Technical Lead (TL)
- Business Analysts (BA)
- Test Analysts (TA)
- Project Manager (PM)
- Product Owner/Domain Owner (PD/DO)
- End Users

Software Development Team (SDT) – consists of all the actors mentioned in above list except end users

Figure 7.6: Proposed User-centric Agile development model (Source Researcher)

Step 7: Distribute model

The final model will be shared via email to software development teams.

This section has defined in depth the process to formulate and implement the *Usercentric Agile development model* prescribed in the objective of this research study. It also describes the activities within the process and the actors who interact in each. The actors form part of the software development team. The model puts users at the centre of the software development process. The following sub-section will describe the steps within the model, demonstrating the flow from start to finish.

7.6 The User-centric Agile development model refined

Sub-section 7.5.1 described the model development process for this research study. Furthermore, Step 6 of this process, indicated in Figure 7.7 below, proposes the *Usercentric Agile Development Model* that user-centred Agile software development teams will use. The model below is the final refined version of this model.

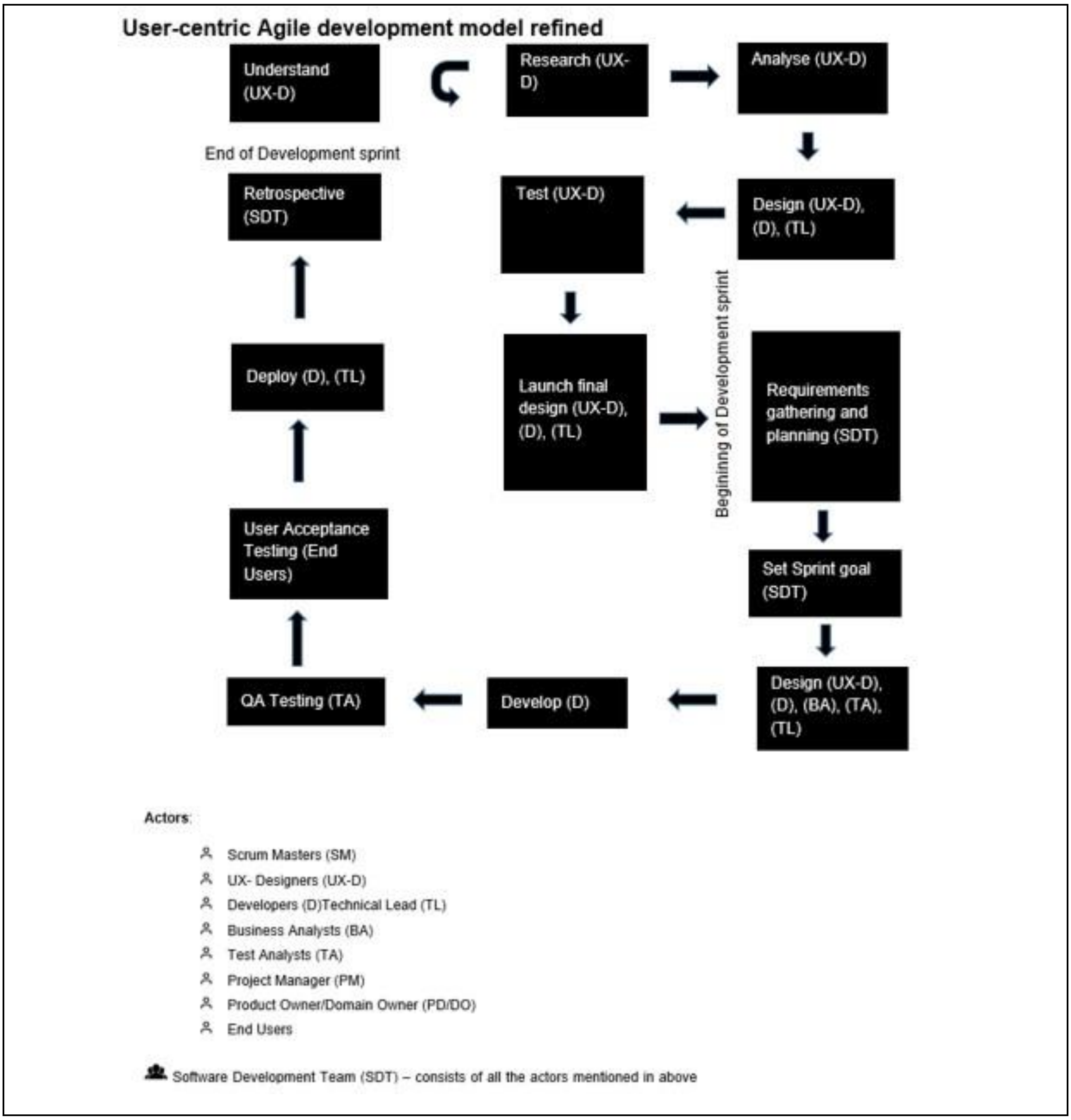

User-centric Agile development model refined
Understand (UX-D)
Research (UX-D)
Analyse (UX-D)
End of Development sprint
Retrospective (SDT)
Test (UX-D)
Design (UX-D), (D), (TL)
Beginінg of Development sprint
Deploy (D), (TL)
Launch final design (UX-D), (D), (TL)
Requirements gathering and planning (SDT)
User Acceptance Testing (End Users)
Set Sprint goal (SDT)
QA Testing (TA)
Develop (D)
Design (UX-D), (D), (BA), (TA), (TL)
Actors:
Scrum Masters (SM)
UX- Designers (UX-D)
Developers (D)Technical Lead (TL)
Business Analysts (BA)
Test Analysts (TA)
Project Manager (PM)
Product Owner/Domain Owner (PD/DO)
End Users
Software Development Team (SDT) – consists of all the actors mentioned in above

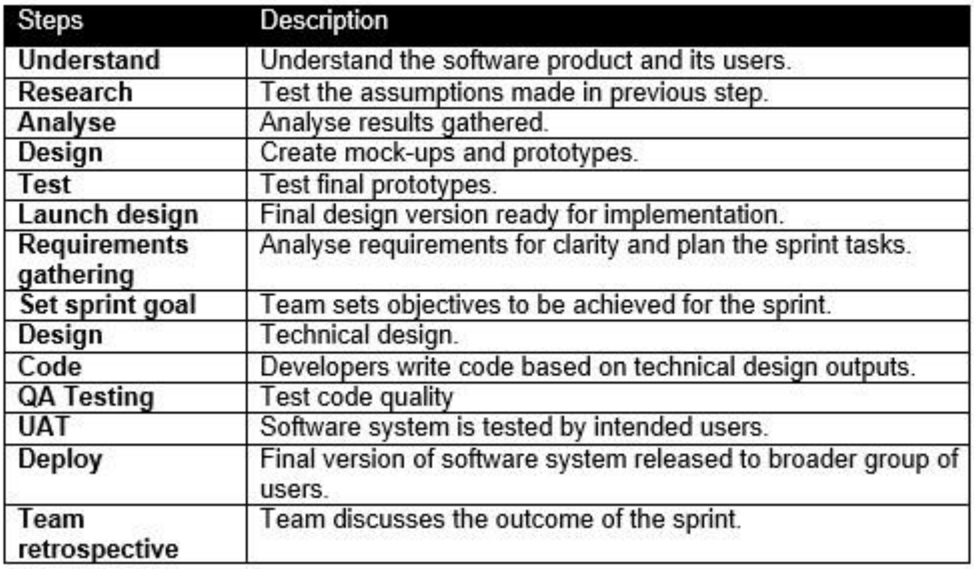
Figure 7.7: User-centric Agile development model refined (Source Researcher)

7.7 Steps to use the model

This section presents detailed steps on how to use the *User-Centric Agile Development Model* defined in the section above. As already established, the intended users for this model are user-centred Agile development teams. These teams are made up of the following IT professionals and users:

Scrum Masters
Project Managers
Developers
Business Analysts
Test Analysts
Domain Owners/Product Owners
User Experience Designers
End users or consumers of the software systems or products being created. Follow

the steps in Table 7.1 below to utilise this model.

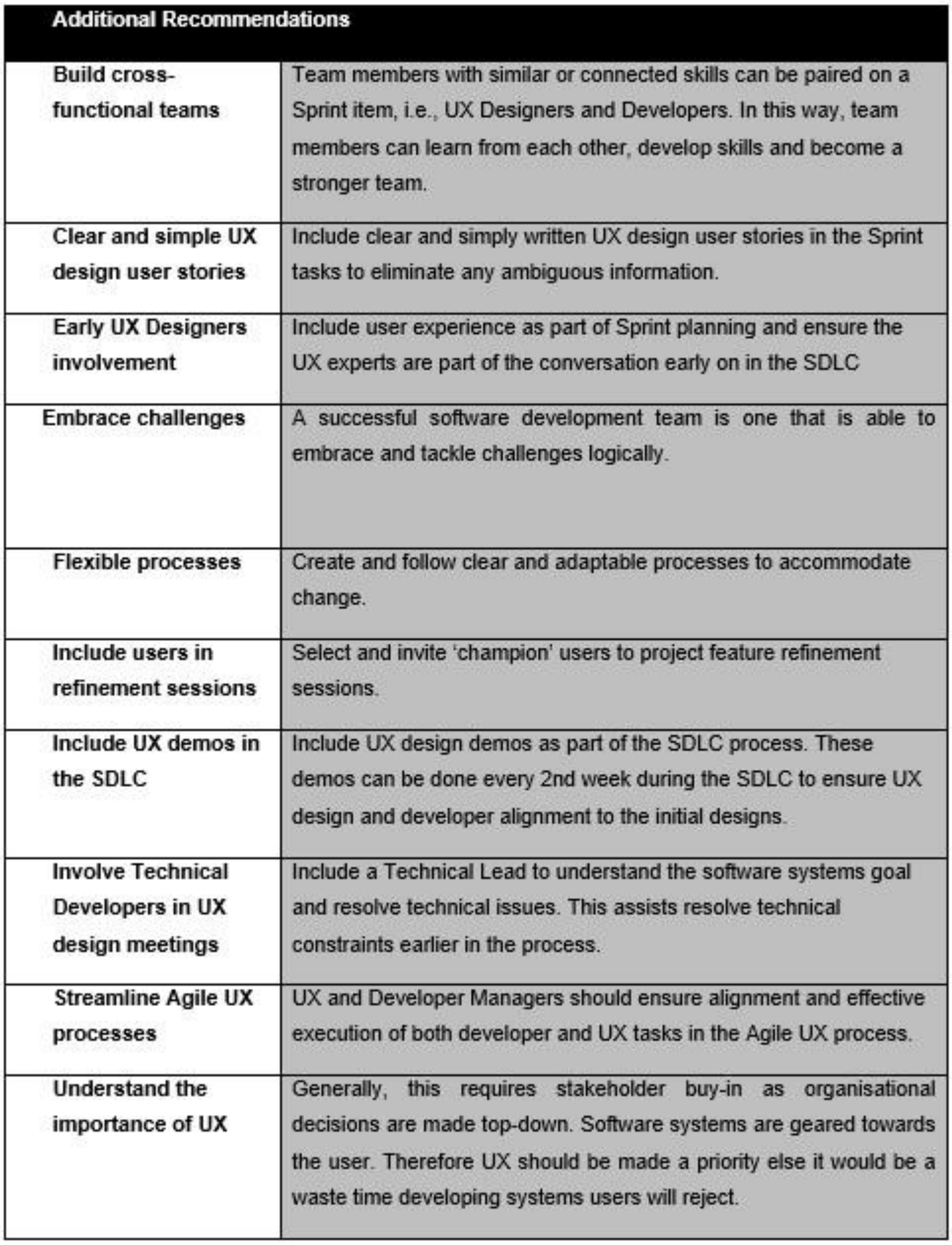


Steps	Description
Understand	Understand the software product and its users.
Research	Test the assumptions made in previous step.
Analyse	Analyse results gathered.
Design	Create mock-ups and prototypes.
Test	Test final prototypes.
Launch design	Final design version ready for implementation.
Requirements gathering	Analyse requirements for clarity and plan the sprint tasks.
Set sprint goal	Team sets objectives to be achieved for the sprint.
Design	Technical design.
Code	Developers write code based on technical design outputs.
QA Testing	Test code quality
UAT	Software system is tested by intended users.
Deploy	Final version of software system released to broader group of users.
Team retrospective	Team discusses the outcome of the sprint.

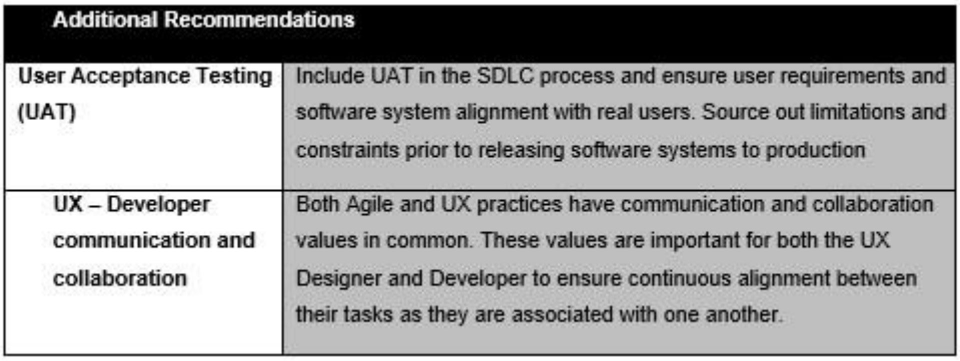
Table 7.1: Steps to use the model (Source Researcher)

7.8 Additional Recommendations

This section presents future recommendations for Agile-UX software development teams. Some of these recommendations have been taken from the survey and netnography results in sub-sections 5.2.2 and 5.2.3. The rest have been gathered from the literature review in sub-section 3.3.6. These recommendations are presented in Table *7.2* below.



Additional Recommendations	
Build cross-functional teams	Team members with similar or connected skills can be paired on a Sprint item, i.e., UX Designers and Developers. In this way, team members can learn from each other, develop skills and become a stronger team.
Clear and simple UX design user stories	Include clear and simply written UX design user stories in the Sprint tasks to eliminate any ambiguous information.
Early UX Designers involvement	Include user experience as part of Sprint planning and ensure the UX experts are part of the conversation early on in the SDLC
Embrace challenges	A successful software development team is one that is able to embrace and tackle challenges logically.
Flexible processes	Create and follow clear and adaptable processes to accommodate change.
Include users in refinement sessions	Select and invite 'champion' users to project feature refinement sessions.
Include UX demos in the SDLC	Include UX design demos as part of the SDLC process. These demos can be done every 2nd week during the SDLC to ensure UX design and developer alignment to the initial designs.
Involve Technical Developers in UX design meetings	Include a Technical Lead to understand the software systems goal and resolve technical issues. This assists resolve technical constraints earlier in the process.
Streamline Agile UX processes	UX and Developer Managers should ensure alignment and effective execution of both developer and UX tasks in the Agile UX process.
Understand the importance of UX	Generally, this requires stakeholder buy-in as organisational decisions are made top-down. Software systems are geared towards the user. Therefore UX should be made a priority else it would be a waste time developing systems users will reject.



Additional Recommendations	
User Acceptance Testing (UAT)	Include UAT in the SDLC process and ensure user requirements and software system alignment with real users. Source out limitations and constraints prior to releasing software systems to production
UX – Developer communication and collaboration	Both Agile and UX practices have communication and collaboration values in common. These values are important for both the UX Designer and Developer to ensure continuous alignment between their tasks as they are associated with one another.

Table 7.2: Additional recommendations (Source Researcher)

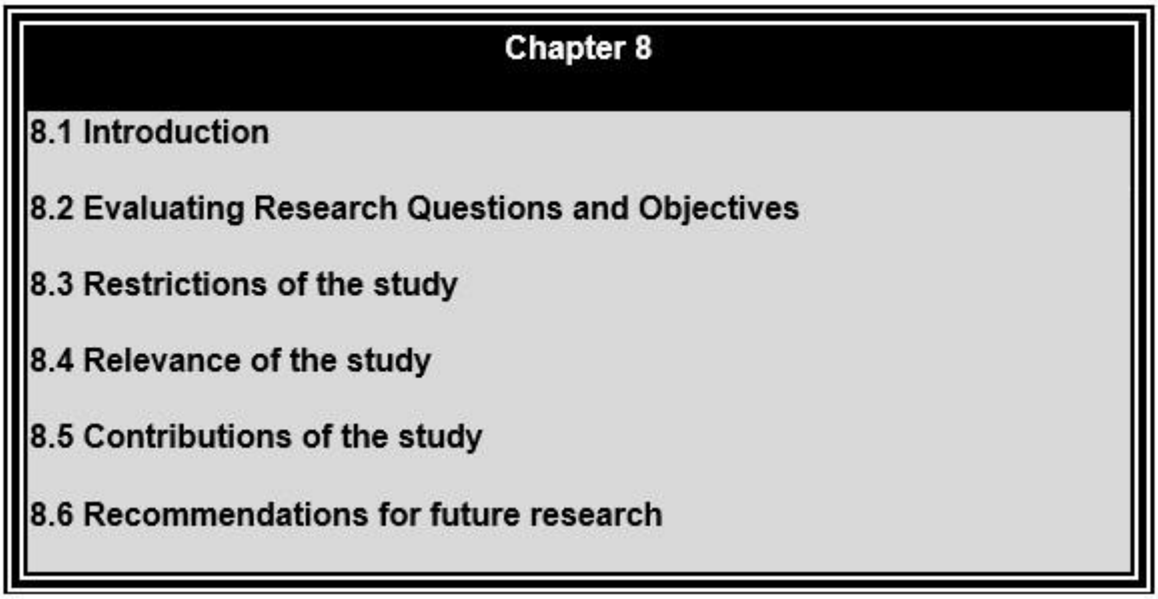
7.9 Conclusion

It is of utmost importance to assess the approach used by SD teams to implement software projects for inclusion of user needs as software systems are geared towards users. Therefore, it is vital for Agile software development teams to ensure they create and follow a streamlined software development process that is user centred. It would be purposeless to build software systems that the intended users will eventually reject. The best way to ensure this doesn’t happen is to include UX Designers and end users in the software development process. The final model has been proposed together with additional recommendations.

7.10 Summary

This chapter has discussed model development (section 7.1, 7.2, 7.3, and 7.4) and the process behind it (sub-section 7.5.1) and presented the *User-centric Agile development model* (section 7.6) together with steps to use the model (section 7.7). Furthermore, recommendations are presented in section 7.8 for Agile SD teams that wish to follow a user-centred approach when creating software systems and improve team and organisational culture. The following chapter provides the conclusion and analysis of findings.

Chapter 8: Conclusion



Chapter 8

8.1 Introduction

8.2 Evaluating Research Questions and Objectives

8.3 Restrictions of the study

8.4 Relevance of the study

8.5 Contributions of the study

8.6 Recommendations for future research

8.1 Introduction

This study investigated the best approach to effectively combine Agile and User Experience Design practices for software development projects. The literature identified the importance of understanding the role UX plays in software development through stakeholder buy-in and the inclusion of UX Designers and end users in the software development process. Similarly, the factors mentioned above were highlighted as the challenges that hinder the integration of these practices.

There are multiple existing Agile-UX methods, but Agile software development teams scarcely use these. This could be due to a lack of awareness or need to utilise them, as UX Designers often work in isolation from the rest of the team. The need to address the importance of UX in Agile projects and create an awareness of Agile-UX methods increases the need to develop a model that will encompass the best approach for Agile teams to create user-centred software systems.

This chapter evaluates the research questions and objectives highlighted in Chapter 1 in sections 1.4 and 1.5 respectively.

8.2 Evaluating Research Questions and Objectives

Sub-section 1.3. in Chapter 1 of the research study highlights the problem statement that needed to be addressed. This problem statement is reiterated below:

Current research shows that there are insufficient existing models to integrate user experience design processes in Agile software development effectively.

Research questions and objectives were deduced from the problem statement. The primary research question for this study is:

- What will a model for integrating user experience design processes into Agile software development consist of?

The secondary research questions are indicated below:

- What best practices currently exist to integrate user experience into Agile software development?
- What challenges exist for integrating user experience into Agile software development?
- What is the impact of integrated UX practices in Agile software projects?

The primary research question is aligned to the main objective of this research study indicated below:

- To develop a model for integrating user experience design processes into Agile software development

Similarly, the secondary research questions for this study also align to the secondary research objectives indicated below:

- To identify existing best practices to integrate user experience design processes into Agile software development
- To identify challenges that exist for integrating user experience design processes into
- Agile software development
- Investigate the impact of integrated UX practices in Agile software projects

8.2.1 Secondary Research Question 1

What best practices currently exist to integrate user experience into Agile software development?

8.2.1.1 Findings: Secondary Research Question 1

A literature review was conducted in sub-sections 2.4.1, 2.4.2 and 2.4.3 of Chapter 2 and the interviews in sub-section 5.2.1 of Chapter 5 to address the secondary research question. The findings validate why Agile is the most used SDM. The main reason is that it has an advantage of following a continuous, iterative, flexible, and transparent approach. Agile is identified as a developer-focused methodology with little focus on the rest of the software development team, let alone UX Designers. Agile through the Agile manifesto fails to embrace UX as it does not consider the time and tasks required to complete designs for software development upfront before the Sprint starts. However, UX Designers are encouraged to work through these challenges successfully by integrating Agile and UX practices.

Sub-section 2.4.1 identifies the characteristics required to effectively integrate UX and Agile software development teams, while sub-section 2.4.2 highlights the existing Agile-UX methods. Lastly, sub-section 2.4.3 describes factors to consider when selecting an appropriate Agile-UX method. Furthermore, sub-sections 5.2.1 and 5.2.2 identify communication and collaboration and the inclusion of UX Designers and end users in the team, team alignment, flexibility and streamlining the process as such factors. However, best practices required to successfully integrate user experience into Agile software development have been accurately deduced from sub-sections 2.4.1, 2.4.3 in Chapter 2 and sub-sections 5.2.1 and 5.2.2 in Chapter 5 respectively.
These best practices are described below:

- Awareness and importance of Agile-UX methods.
- Superiors and organisation leaders should understand the benefit of UX.
- UX Designers should possess leadership qualities to thrive in Agile.

- A flexible Agile development process is required for UX to work successfully in Agile Streamlined UX and development tasks managed and coordinated by Developer and UX Mangers or Project Managers.
- UX Designers and Developers should work collaboratively in one team.
- The team needs to find the most suitable Agile-UX approach that ensures flexibility, transparency, continuous delivery, easy to learn, research and design, user involvement, team management, the right skills and knowledge, tractability, influence, and innovation.
- UX Designers should be clear about their tasks and understand the Agile team approach to SDLC.
- UX Designers and end users should be included into the SDLC.
- A team can select an accustomed approach and can modify it to fit the team objectives.

Based on the literature review in Chapter 2 and the interviews discussed in Chapter 5 from the case study, this research question was answered and the best practices stated above can be used to effectively integrate Agile and UX practices.

8.2.2 Secondary Research Question 2

What challenges exist for integrating user experience into Agile software development?

8.2.2.1 Findings: Secondary Research Question 2

The research question was addressed by the literature review highlighted in Chapter 3, sub-section 3.3.6 and the survey results depicted in Chapter 5 in sub-section 5.2.2. The survey was distributed to the software development team members. Various challenges were identified based on the responses obtained from it. These challenges include:

- Lack of understanding the importance of UX in Agile projects
- No stakeholder buy-in
- Understanding roles and responsibilities

- Company attitude

These were some of the disadvantages solicited from the findings of the case study.

The literature review conducted in sub-section 3.3.6 in Chapter 3 states that user-centred software development projects may encounter challenges due to the differences that exist between Agile Development (AD) and User-Centred Design (UCD). According to the findings UCD relies on producing design artefacts and AD highlights minimal documentation. Furthermore, UCD attempts to understand the users before software product creation and AD avoids performing any activities before development commences.

The findings also reveal that UX Designers have less time to design than required. The focus is on meeting deadlines rather than the quality of the software product. Similarly, short iterations within the Agile process may lead to less testing time or have it abandoned completely.

Insufficient user feedback is also highlighted as a disadvantage. As mentioned above, limited testing time may also lead to abandoned user feedback. Important user feedback can assist meet user requirements. Ultimately, UX ends up being excluded should feedback not be accounted for in the process.

A lack of communication and collaboration can hinder the success of software projects. It is very important to maintain effective collaboration and communication to ensure the entire team is up to date with the progress of the software product. Furthermore, team members working remotely may potentially cause communication issues.

Lastly, the involvement of other stakeholders could potentially hold the process back if the team awaits stakeholder feedback or involvement for lengthy periods.

Based on the literature review conducted in Chapter 3 and the responses captured from various software development team members, this research question has been answered.

8.2.3 Secondary Research Question 3

What is the impact of integrated UX practices in Agile software projects?

8.2.3.1 Findings: Secondary Research Question 3

Sub-section 3.3.6 of Chapter 3 highlighted the literature review, and sub-section 5.2.1 discussed the semi-structured interviews conducted in a case study designed to answer this research question. Sub-subsection 5.2.2 reviewed a survey sent to software development teams for the same purpose. All the data collection methods identified the factors highlighted below as the result of the impact UX practices have when integrated into Agile software development projects:

- End product is built as intended.
- Integrating both practices ensure cross-skilling, team development and enhanced company culture.
- User-centred software development gives the organisation a slight competitive advantage.
- Both UCD and AD support iterative activities in their process.
- Both practices advocate for user involvement in the process.
- Both practices highlight team collaboration.
- Test-driven development, decreasing project risk, flexibility, transparency within the processes and a clear understanding of the objective are some of the reasons why UCD and AD fit well together.
- UCD enhances the method by not just focusing on the creation of the code but also on the quality that the solution provides to the user UCD integration accounts for clear requirements.

This research question has been answered by the literature review, interviews and the survey, and the responses captured from various software development team members.

8.2.4 Primary Research Question

What will a model for integrating user experience design processes into Agile software development consist of?

The primary research question was answered in Chapter 7, in sections 7.4, 7.5, 7.6 and 7.7 respectively. A model (as indicated in Figure 8.1 below) was created and corresponding steps for implementation were described. Lastly, additional recommendations to improve Agile-UX software development projects have been provided. The recommendations below were some of the most frequently recognised for this study:

Understand the importance of UX: UX should be made a priority, else it would be a waste of time developing systems users will reject

Early UX Designers involvement: User experience should be included as part of Sprint planning and ensure that the UX experts are part of the conversation early in the SDLC

Communication and collaboration: Both UX Designers and Developers need to ensure continuous alignment between their tasks as they are associated with one another

User Acceptance Testing: Include UAT in the SDLC process and ensure user requirement and software system alignment with real users

Clear user stories: Clear and simply written UX design user stories should be included in the Sprint tasks to eliminate any ambiguous information

Flexible processes: Develop and follow clear and flexible processes to enable change

Build cross- functional teams: Team members with similar or connected skills and knowledge should be encouraged to work together. Ultimately, they will develop their skills and become a stronger team

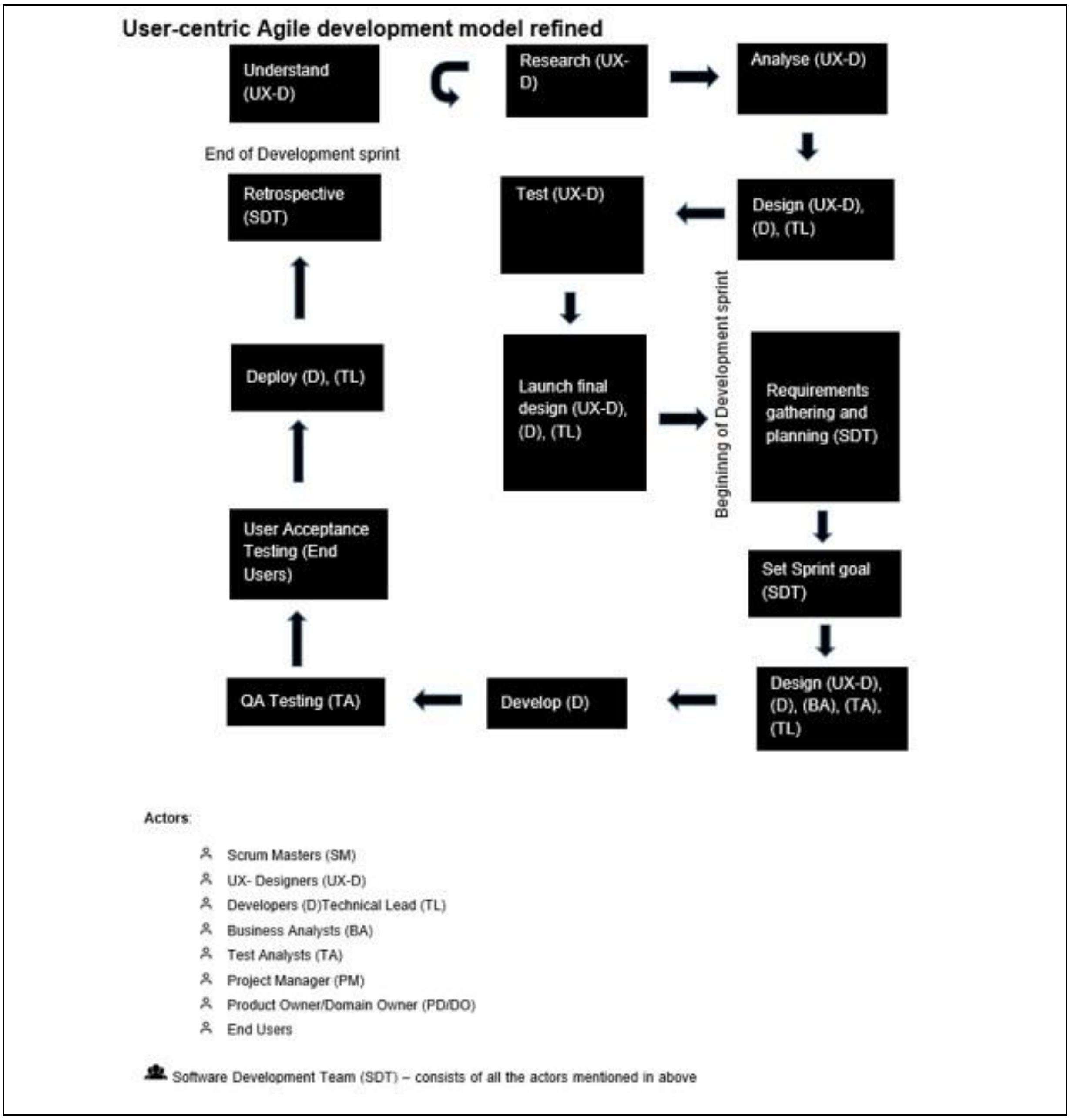

User-centric Agile development model refined
Understand (UX-D)
Research (UX-D)
Analyse (UX-D)
End of Development sprint
Retrospective (SDT)
Test (UX-D)
Design (UX-D), (D), (TL)
Beginінng of Development sprint
Deploy (D), (TL)
Launch final design (UX-D), (D), (TL)
Requirements gathering and planning (SDT)
User Acceptance Testing (End Users)
Set Sprint goal (SDT)
QA Testing (TA)
Develop (D)
Design (UX-D), (D), (BA), (TA), (TL)
Actors:
Scrum Masters (SM)
UX- Designers (UX-D)
Developers (D)Technical Lead (TL)
Business Analysts (BA)
Test Analysts (TA)
Project Manager (PM)
Product Owner/Domain Owner (PD/DO)
End Users
Software Development Team (SDT) – consists of all the actors mentioned in above

Figure 8.1: User-centric Agile development model (Source Researcher)

8.3 Restrictions of the study

This research study is restricted mostly to Agile software development teams in the South African region. The case study was conducted in the Western Cape province to where semi-structured interviews were conducted, and surveys were sent out to software development teams to determine their approach to the implementation of software projects.

The various challenges identified during this study were due to the scarcity of available studies regarding Agile-UX methods in South Africa. The researcher studied literature globally to fully understand the perception of Agile-UX methods. The findings of the literature can be used by any Agile software development team.

Semi-structured interviews were conducted online due to the COVID-19 pandemic in 2020. A few challenges were encountered such as internet connection issues and participants not being comfortable when recorded during the interview for privacy reasons although interviews were not recorded without the participant's' consent. Due to these restrictions few interviews were conducted for this study.

Lastly, the findings of the literature reviews in this study indicated several existing Agile-UX methods which were like the process required to develop the model for this research study. However, the study was not limited by this challenge. Ultimately, these existing Agile-UX methods were used as an advantage by their contribution to develop a more user-centred Agile model for this study.

The study was conducted using literature reviews, survey, interviews and focus groups in the case study, which was conducted with software development teams. The purpose of the study was restricted to gaining insight on the factors that influence the impact of integrating Agile and UX methods to develop a model that enables Agile software development teams to effectively integrate UX in their software projects.

8.4 Relevance of the study

Regardless of the restrictions highlighted above, the study proceeded successfully. The relevance of this study is highlighted below:

- The study identified best practices for integrating Agile and UX.
- Factors that contribute to the impact of integrating Agile and UX methods were identified through this study.

- The identified best practices and factors contributing to the impact of combining the practices were used to develop recommendations for Agile-UX software development teams.

Based on the findings of this study, a *User-centric Agile development model* was developed to effectively implement software development projects. This study can contribute to software development teams that utilise the traditional approach for software development and desiring transition to Agile. Lastly, this study will contribute to the restricted amount of existing literature published in South Africa regarding Agile-UX methods.

8.5 Contributions of the study

The important outputs of this research study are knowledge contribution to the existing knowledge base of the area of study. Knowledge was gathered and shared throughout the research process to the software development environment. Figure 8.2 and Table 8.1 present the various contributions of this study. These also involve the processes followed, research instruments and tools used.

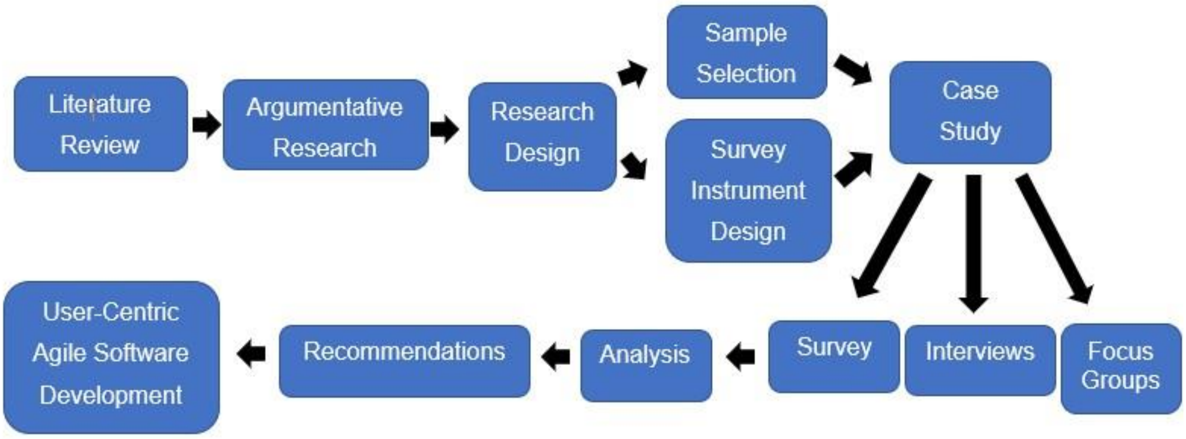


Figure 8.2: The process of creating A Model for integrating User Experience into Agile software development (Source Researcher)

Figure 8.2 illustrates key factors of the processes followed to create a model for integrating user experience into Agile software development.

Contribution	Description
Survey as a data collection instrument	Survey to confirm the relevance of the proposed model.
Additional Recommendations	Proposed recommendations for Agile software development teams on how to build strong successful teams that are UX inclusive.
The process of creating a model	Key steps to be followed when creating the proposed *User-centric Agile Software development model.*
User-centric Agile Software development model	*The User-centric Agile Software development model* is the main contribution in artefact form for this study. The proposed model is meant to provide a clear, user-centric, inclusive, and simple Agile-UX process to be used by software development teams. The model has been fully documented to indicate activities, tools, and techniques within each step of the process. The need for this model has been evaluated through a case study with software development teams to confirm its significance. Researchers can improve the model by refining or adding to the existing process.

Table 8.1: Contributions of the study (Source Researcher)

Table 8.1 outlines the key contributions of this study. Instruments and tools used, knowledge shared and the proposed model. Appendix A and B illustrates the survey used as a contribution to achieve the aim of this study.

8.6 Recommendations for future research

This study targeted Agile software development teams mostly in the Western Cape Province in South Africa. Recommendations for future research comprise the outline in the next paragraph.

This study was focused on Agile teams mostly. Future research should include any of the other existing software development methodologies, and especially Waterfall, as the findings of the study have indicated that some teams still use it. This study was conducted in the Western Cape Province, therefore the inclusion of the broader provinces of South Africa is recommended for future research.

Online interviews were conducted for this research study due to the COVID-19 pandemic. These encountered challenges such as internet connection issues and some participants not being comfortable with recorded interviews. Therefore, face-to-face interviews where participants are more comfortable and where there are no technical difficulties are recommended for future research if possible.

REFERENCES

Agile Alliance. (2021, September 24). *What is a scrum master?* Agile Alliance. Retrieved October 8, 2022, from https://www.agilealliance.org/glossary/scrum-master/

Alexander, M. (2021, July 5). *What is a project manager? The lead role for Project Success*. CIO. Retrieved October 8, 2022, from https://www.cio.com/article/230682/what-is-a-project-managerthe-lead-role-for-project-success.html

Allabarton, R. (2022, December 6). What Is the UX Design Process? A Complete, Actionable Guide. Retrieved December 18, 2022, from https://careerfoundry.com/en/blog/ux-design/the-ux-designprocess-an-actionable-guide-to-your-first-job-in-ux/.

Altexsoft. (2021, January 18). Extreme Programming: Values, Principles, and Practices [web log]. Retrieved October 6, 2022, from https://www.altexsoft.com/blog/business/extremeprogramming-values-principles-and-practices/.

Antunes, P. (2020, April 20). *User experience design for business students - part 2*. ResearchGate. Retrieved October 8, 2022, from https://www.researchgate.net/publication/333828576_User_Experience_Design_for_Business_Students_-_Part_2

Arasu, D. (2019, June 17). *Choosing the right UX process for your software-development model*. UXmatters. Retrieved October 8, 2022, from https://www.uxmatters.com/mt/archives/2019/06/choosing-the-right-ux-process-for-yoursoftware-development-model.php

Babich, N. (2018, January 31). *A comprehensive guide to product design*. Smashing Magazine. Retrieved October 8, 2022, from https://www.smashingmagazine.com/2018/01/comprehensiveguide-product-design/

Babich, N. (2019, October 18). *User centred design principles & methods: Adobe XD ideas*. Ideas. Retrieved October 8, 2022, from https://xd.adobe.com/ideas/principles/human-computerinteraction/user-centered-design/

Benyon, D. (2019). An introduction to user experience. In *Designing user experience. A guide to HCI, UX and interaction design* (4th ed., pp. 5–6). essay, Pearson. Retrieved October 8, 2022, from https://pdfuni.com/sample/PoliticsSociology/PS12011300/PS1271/sample%EF%BC%8DDesigning%20User%20Experience%204th%204E%20Davi d%20Benyon.pdf.

Biecek, P. (2019, July 9). *Model development process*. arXiv.org. Retrieved October 8, 2022, from https://arxiv.org/abs/1907.04461

Boehm, B., & Belz, F. (1990). Experiences with the spiral model as a process model generator. In *ISPW '90: Proceedings of the 5th International Software Process Workshop on Experience with Software Process Models*. CA. Retrieved from https://dl.acm.org/doi/pdf/10.5555/317498.317694.

Boru, T. (2018). *Chapter Five Research Design and Methodology* (thesis). *ResearchGate*. ResearchGate. Retrieved October 8, 2022, from https://www.researchgate.net/publication/329715052_CHAPTER_FIVE_RESEARCH_DESIGN_AND_METHODOLOGY_51_Introduction_Citation_Lelissa_TB_2018_Research_Methodology_University_of_South_Africa_PHD_Thesis.

Brunskill, V.-L. (2019, August 21). *What is a sprint (Agile Software Development)? definition from whatis.com*. SearchSoftwareQuality. Retrieved October 8, 2022, from https://www.techtarget.com/searchsoftwarequality/definition/Scrum-sprint

Bruun, A., Larusdottir, M. K., Nielsen, L., Nielsen, P. A., & Persson, J. S. (2018, September 29). The role of UX professionals in agile development: a case study from industry. In *The role of UX professionals in agile development: a case study from industry*. ResearchGate. Retrieved October 8, 2022, from https://www.researchgate.net/publication/327866070_The_role_of_UX_professionals_in_agile_development_a_case_study_from_industry.

Bulajewsk, M. (2015, April 21). *Crossing the Agile-UX divide*. Agile Buzz Forum - The back story on chips. Retrieved October 8, 2022, from https://www.artima.com/forums/flat.jsp?forum=155&thread=105742

Chaione. (2017, January 19). UX is not UCD. Retrieved October 8, 2022, from https://www.chaione.com/blog/ux-ucd.

Cockton, G., Lárusdóttir Marta, Gregory, P., & Cajander Åsa. (2016). Integrating User-Centred Design in Agile Development. In *Integrating user-centred design in Agile Development* (pp. 1–46).

essay, Springer International Publishing. Retrieved October 8, 2022, from https://doi.org/10.1007/978-3-319-32165-3_1

Debever, B. (2019, March 29). Marketer’s Guide: Key Differences Between Qualitative Data and Quantitative Data [web log]. Retrieved October 8, 2022, from https://www.convert.com/blog/growth-marketing/qualitative-data-quantitative-data/.

Dudovskiy, J. (2022a). *Reliability and validity - research methodology*. Business Research Methodology. Retrieved October 8, 2022, from https://research-methodology.net/researchmethodology/reliability-validity-and-repeatability/

Dudovskiy, J. (2022b). *Research approach - research-methodology*. Business Research Methodology. Retrieved October 8, 2022, from https://research-methodology.net/researchmethodology/research-approach/

Dudovskiy, J. (2022c). *Research philosophy - research methodology*. Business Research Methodology. Retrieved October 8, 2022, from https://research-methodology.net/researchphilosophy/

Edmonds, B., Le Page, C., Bithell, M., Chattoe-Brown, E., Grimm, V., Meyer, R., Montañola-Sales, C., Ormerod, P., Root, H., & Squazzoni, F. (2019). Different modelling purposes. *Understanding Complex Systems*, *22*(3), 39–58. https://doi.org/10.1007/978-3-319-66948-9_4

Experience UX. (2022). *What is user-centred design*. Experience UX. Retrieved October 8, 2022, from https://www.experienceux.co.uk/faqs/what-is-user-centred-design/

Farebo, S. (2015, December 14). *User-centered design in agile software development for in-house enterprise tools*. DIVA. Retrieved October 8, 2022, from http://www.divaportal.org/smash/record.jsf?pid=diva2%3A878385&dswid=-715

Follett, J. (2017, May 2). *Integrating Agile and UX Design*. O'Reilly Media. Retrieved October 8, 2022, from https://www.oreilly.com/radar/integrating-agile-and-ux-design/

Forbrig, P., & Herczeg, M. (2015). Managing the Agile Process of Human-Centred Design and Software Development. Bamberg. Retrieved October 8, 2022, from https://www.irit.fr/recherches/ICS/events/conferences/workshop-IFIPWG13.2Bamberg/contributions/W04-04-forbirg.pdf.

Fustik, V. (2017). The advantages of Agile methodologies applied in the ICT development projects. *International Journal on Information Technologies & Security*, *9*(4), 51–62. Retrieved October 8, 2022, from https://ijits-bg.com/contents/IJITS-No4-2017/2017-N4-05.pdf.

Gartenstein, D. (2018, August 3). *What are the duties of a manager in the workplace?* Small Business - Chron.com. Retrieved October 8, 2022, from https://smallbusiness.chron.com/duties-managerworkplace-24280.html

Gilley, C. (2021). The Pros and Cons of Agile Product Development [web log]. Retrieved October 8, 2022, from https://uservoice.com/blog/agile-product-development.

Gladkiy, S. (2018, June 14). *User-centered design: Process and benefits*. Medium. Retrieved October 8, 2022, from https://uxplanet.org/user-centered-design-process-and-benefits-fd9e431eb5a9

Gothelf, J., & Seiden, J. (2016). Lean Ux: Designing great products with agile teams. *Google books* (2nd ed.). O'Reilly. Retrieved October 8, 2022, from https://books.google.co.za/books?id=QnQNDQAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false.

Handa, A., & Vashisht, K. (2016, November 21). *Agile development is no excuse for shoddy UX research*. UXmatters. Retrieved October 8, 2022, from https://www.uxmatters.com/mt/archives/2016/11/agile-development-is-no-excuse-for-shoddy-uxresearch.php

Hartson, R., & Pyla, P. (2019). The UX book: Agile UX design for a quality user experience. *Google books*. Elsevier. Retrieved October 8, 2022, from https://books.google.co.za/books?hl=en&lr=&id=RHIGCwAAQBAJ&oi=fnd&pg=PP1&dq=+Agile+ +
and+UX+combined+&ots=ENGfKmg0vP&sig=wzkMaxZSETNVuDlK34jLHKaesMs&redir_esc=y#v=onepage&q=Agile%20and%20UX%20combined&f=false.

Hassan, H. M., & Galal-Edeen, G. H. (2017). 2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS). In *From usability to user experience*. IEEE. Retrieved October 8, 2022, from https://ieeexplore.ieee.org/document/8279761.

Hayes, A. (2022, November 7). *End user: Definition, examples, vs. customer*. Investopedia. Retrieved December 21, 2022, from https://www.investopedia.com/terms/e/end-user.asp

Hayes, H., & Sakoda, L. (2013, March 1). What is Netnography and How Can It Benefit You? Retrieved December 21, 2022, from https://harrisonhayes.com/blog/31/What-is-Netnographyand-How-Can-It-Benefit-You-/.

Heap. (2022). *What is a tech stack? choosing what goes in yours*. Heap. Retrieved October 8, 2022, from https://heap.io/topics/what-is-a-tech-stack

Hoda, R., Salleh, N., & Grundy, J. (2018). *The Rise and Evolution of Agile Software Development*. ResearchGate. Retrieved October 8, 2022, from https://www.researchgate.net/profile/RashinaHoda/publication/326339847_The_Rise_and_Evolution_of_Agile_Software_Development/links/5 c5353aa299bf12be3f1026b/The-Rise-and-Evolution-of-Agile-SoftwareDevelopment.pdf?origin=publication_detail

Holishevska, A. (2022, September 9). UX Expert: Hiring tips, Rates and Team integration [web log]. Retrieved October 8, 2022, from https://northell.design/blog/ux-expert-where-and-how-to-hireright-tips-for-startups.

Ibanez, L. (2017, November 9). A quick overview of Agile [web log]. Retrieved October 8, 2022, from https://www.lazaroibanez.com/blog/2017/11/09/agile-development-a-quickoverview?rq=agile%20development.

Ilievska, S. (2019, March 29). Making software lovable - The importance of UX design in the SDLC [web log]. Retrieved October 8, 2022, from https://www.iborn.net/blog/making-software-lovableimportance-ux-design-sdlc.

Indeed. (2020, October 5). *How to develop cross functional teams | indeed.com - indeed career guide*. Indeed. Retrieved October 8, 2022, from https://www.indeed.com/career-advice/careerdevelopment/develop-cross-functional-teams

Indeed. (2021a, March 9). *What is software development: Definition, processes and types*. Indeed. Retrieved October 8, 2022, from https://www.indeed.com/career-advice/careerdevelopment/what-is-software-development

Indeed. (2021b, October 6). *A guide to technical leads: What they do and how to become one*. Indeed. Retrieved October 8, 2022, from https://www.indeed.com/career-advice/finding-ajob/technical-lead

Iseesystems. (2021). *The model building process*. The Model Building Process. Retrieved October 8, 2022, from https://www.iseesystems.com/resources/help/v10/Content/The_model_building_process.htm

Ishtiaq, M. (2019.). *ResearchGate* (2nd ed., Vol. 12). Canadian Center of Science and Education. Retrieved October 8, 2022, from https://www.researchgate.net/publication/332246566_Book_Review_Creswell_J_W_2014_Research_Design_Qualitative_Quantitative_and_Mixed_Methods_Approaches_4th_ed_Thousand_Oaks_CA_Sage.

Jamsheer, K. (2018, June 1). 12 Best software development methodologies with pros and cons [web log]. Retrieved October 8, 2022, from https://acodez.in/12-best-software-developmentmethodologies-pros-cons/.

Johnson, J. (2021, August 24). *What is a head of product?* USChamber. Retrieved October 8, 2022, from https://www.uschamber.com/co/run/technology/head-of-product-job-description

Jones, A. J. (2019). *Factors for successful agile collaboration between Ux designers and software developers in a complex organisation* (thesis). *University of Chester digital repository*. University of Chester. Retrieved October 8, 2022, from https://chesterrep.openrepository.com/handle/10034/622425.

Joseph, N., Marnewick, C., & Santana, M. J. (2016). *Agile Software Development and IT project performance in South Africa: A positive relationship*. ResearchGate. Retrieved October 8, 2022, from https://www.researchgate.net/publication/318206568_Agile_software_development_and_IT_project_performance_in_South_Africa_a_positive_relationship

Justinmind. (2018, April 17). *Designer's Guide to Scrum*. Justinmind. Retrieved October 8, 2022, from https://www.justinmind.com/blog/designers-guide-to-Scrum/

Justinmind. (2021, May 19). *The UX design process explained*. Justinmind. Retrieved October 8, 2022, from https://www.justinmind.com/ux-design/process

Keatinge, S. A. (2021, March 20). *Scrum master*. Scrum.org. Retrieved October 8, 2022, from https://www.scrum.org/forum/scrum-forum/47396/scrum-master

Khoza, L., & Marnewick, C. (2020). Waterfall and Agile Information System project success rates – A South African perspective. *South African Computer Journal*, *32*(1). https://doi.org/10.18489/sacj.v32i1.683

Kieffer, S., Ghouti, A., & Macq, B. (2017). The Agile UX Development Lifecycle: Combining Formative Usability and Agile Methods. In *Proceedings of the 50th Hawaii International Conference on System Sciences*. Semanticscholar.org. Retrieved October 8, 2022, from https://pdfs.semanticscholar.org/0b0b/18f89944c7d14a6a8edc37269c2c10a0155e.pdf.

Kienitz, P. (2017, February 27). *The Pros and cons of Waterfall Software Development*. One Beyond. Retrieved October 8, 2022, from https://www.dcsl.com/pros-cons-waterfall-softwaredevelopment/

Kissflow. (2022, September 22). *Rapid application development (RAD): Definition, Steps & Full Guide*. Rapid Application Development (RAD) | Definition, Steps & Full Guide. Retrieved October 8, 2022, from https://kissflow.com/application-development/rad/rapid-applicationdevelopment/

Kneuper, R. (2018). Software Process Definition and Modelling. In *Software Processes and Life Cycle Models* (pp. 41–68). essay, Springer, Cham. Retrieved October 8, 2022, from https://link.springer.com/chapter/10.1007/978-3-319-98845-0_2.

Kowa, N. (2021, September 30). *The importance of Business Analysis in software development*. QAT Global. Retrieved October 8, 2022, from https://www.qat.com/importance-business-analysissoftware-development/

Kozinets, R. (2010, January). *Netnography - Researchgate*. Researchgate. Retrieved December 21, 2022, from https://www.researchgate.net/profile/Robert-Kozinets/publication/267922181_Netnography_Doing_Ethnographic_Research_Online/links/5c575bb3a6fdccd6b5e0f5fd/Netnography-Doing-Ethnographic-Research-Online.pdf

Kozinets, R. V., Dolbec, P.-Y., & Earley, A. (2022, April 17). *Netnographic analysis: Understanding culture through social media data*. Digital Fieldwork. Retrieved December 21, 2022, from https://digitalfieldwork.iu.edu/bib-entry/netnographic-analysis-understanding-culture-throughsocial-media-data/

Kuizinas, G. (2021, August 11). *CTO vs head of engineering*. Medium. Retrieved October 8, 2022, from https://gajus.medium.com/cto-vs-head-of-engineer-8845da32ea67

Kukhnavets, P. (2018, March 18). Advantages and disadvantages of Extreme Programming (XP) [web log]. Retrieved October 8, 2022, from https://hygger.io/blog/disadvantages-andadvantages-of-extreme-programming/.

Kumana, D. (2009, July 16). Agile and UCD [web log]. Retrieved October 8, 2022, from https://www.thoughtworks.com/insights/blog/agile-and-ucd.

Laubheimer, P. (2017, September 24). *Agile is not easy for UX: (how to) deal with it*. Nielsen Norman Group. Retrieved October 8, 2022, from https://www.nngroup.com/articles/agile-not-easy-ux/

Leavy, P. (2017). *Research design: Quantitative, qualitative, mixed methods, arts-based, and community-based... participatory research approaches*. *Google books*. GUILFORD. Retrieved October 8, 2022, from https://books.google.co.za/books?id=hxyDDgAAQBAJ&printsec=frontcover#v=onepage&q&f=fal se.

Levitt, D. (2022). Agile UX: How to Incorporate UX and Product Design Into Agile [web log]. Retrieved October 8, 2022, from https://www.toptal.com/product-managers/agile/agile-ux-designexplained.

Lo, G. (2017, November 11). *What's a design sprint and why is it important?* Medium. Retrieved October 8, 2022, from https://uxplanet.org/whats-a-design-sprint-and-why-is-it-importantf7b826651e09

Lodato, T. J. (2015). *The work of user experience design: materiality and cultures in designing* (dissertation). *Smartech.gatech.edu*. Smartech.gatech.edu. Retrieved October 8, 2022, from https://smartech.gatech.edu/bitstream/handle/1853/53940/LODATO-DISSERTATION2015.pdf?sequence=1&isAllowed=y.

Mahapatra, H. B., Lodato & Goswami, B. (2015). *Selection of Software Development Methodology (SDM): A comparative approach: Semantic scholar*. Semantic scholar. Retrieved October 4, 2022, from https://www.semanticscholar.org/paper/Selection-of-Software-Development-Methodology-A-Mahapatra-Goswami/6935251e44ebe9652ebb6579c672132fc191e8db

Malhotra, G. (2017). Strategies in Research. *International Journal of Advance Research and Development*, *2*(5), 172–180. Retrieved October 8, 2022, from https://www.ijarnd.com/manuscripts/v2i5/V2I5-1220.pdf.

Martinig, F. (2015, April 21). *Agile Buzz Forum -Crossing the Agile-UX Divide*. Aritma. Retrieved October 8, 2022, from https://www.artima.com/forums/flat.jsp?forum=155&thread=370995

Melnikovas, A. (2018). Towards an Explicit Research Methodology: Adapting Research Onion Model for Futures Studies. *Journal of Futures Studies*, *23*(2), 29–44. https://doi.org/10.6531/JFS.201812_23(2).0003

Mendix. (2022, September 22). *Rapid application development*. Mendix. Retrieved October 8, 2022, from https://www.mendix.com/rapid-application-development/

Merriam-Webster. (2022, September 16). *Integrate definition & meaning*. Merriam-Webster. Retrieved October 8, 2022, from https://www.merriam-webster.com/dictionary/integrate#:~:text=1%20%3A%20to%20form%2C%20coordinate%2C

Miller, K. (2019, February 4). *What does a project manager do?* Northeastern University Graduate Programs. Retrieved October 8, 2022, from https://www.northeastern.edu/graduate/blog/projectmanager-responsibilities/

Minhas, S. (2018, April 23). *User experience design process*. Medium. Retrieved October 8, 2022, from https://uxplanet.org/user-experience-design-process-d91df1a45916

MJV Technology & Innovation. (2021, January 6). *User experience: Roi and methods to measure your investment in UX*. MJV Technology & Innovation. Retrieved October 8, 2022, from https://www.mjvinnovation.com/blog/roi-in-ux/

Mohajan, H. K. (2017). Two criteria for good measurements in research: Validity and Reliability. *Annals of Spiru Haret University. Economic Series*, *17*(3), 59–82. https://doi.org/10.26458/1746

Moloko, G. (2022, July 27). *Gender gap: Tech start-ups in SA still ruled by men*. Ventureburn. Retrieved October 8, 2022, from https://ventureburn.com/2022/07/gender-gap-tech-start-ups-insa-still-ruled-by-men/

Moyo, S. (2020). *A software development methodology for solo software developers: Leveraging the product quality of independent developers* (dissertation). *A software development methodology for solo software developers: levering the product quality of independent developers*. University of South Africa. Retrieved October 8, 2022, from https://uir.unisa.ac.za/bitstream/handle/10500/27292/thesis_moyo_s.pdf?sequence=1&isAllowed=y.

Nightingale, A. J. (2022). Triangulation. In *International Encyclopaedia of Human Geography* (2nd ed., pp. 477–480). essay, Elsevier. Retrieved October 8, 2022, from https://www.sciencedirect.com/science/article/pii/B9780081022955104378.

Norman, D. A., & Draper, S. W. (1986). *User Centred System Design: New perspectives on humancomputer interaction*. *Taylor & Francis Group* (1st ed.). Lawrence Erlbaum. Retrieved October 8, 2022, from https://www.taylorfrancis.com/books/edit/10.1201/9780367807320/user-centeredsystem-design-donald-norman.

Norman, D., & Nielsen, J. (2022). *The definition of user experience (UX)*. Nielsen Norman Group. Retrieved October 8, 2022, from https://www.nngroup.com/articles/definition-user-experience/

Novoseltseva, E. (2019, September 8). *User-centered design: An introduction*. Usability Geek. Retrieved October 8, 2022, from https://usabilitygeek.com/user-centered-design-introduction/

Opoku, A., Ahmed, V., & Aziz, Z. (2016). *Research methodology in the built environment: A selection of case studies*. *ResearchGate*. Routledge. Retrieved October 8, 2022, from https://www.researchgate.net/publication/299593898_Choosing_an_appropriate_research_methodology_and_method.

O'Reilly. (2022). *Effective software project management*. O'Reilly. Retrieved October 8, 2022, from https://www.oreilly.com/library/view/effective-softwareproject/9780764596360/9780764596360_ch01lev1sec1.html

Paradis, E., O'Brien, B., Nimmon, L., Bandiera, G., & Martimianakis, M. A. (2016). Design: Selection of data collection methods. *Journal of Graduate Medical Education*, *8*(2), 263–264. https://doi.org/10.4300/jgme-d-16-00098.1

Phair, D., & Warren, K. (2021, January). *Saunders' Research Onion: Explained simply (+ examples)*. Grad Coach. Retrieved October 8, 2022, from https://gradcoach.com/saunders-research-onion/

Pillay, N., & Wing, J. (2019). In *2019 Conference on Information Communications Technology and Society (ICTAS)*. Durban; IEEE. Retrieved October 8, 2022, from https://ieeexplore.ieee.org/document/8703607.

Pollard, J. (2018, April 13). *The principles and process of Lean Ux*. Medium. Retrieved October 8, 2022, from https://medium.com/@jpolldesign/the-principles-and-process-of-lean-ux-5fa0865c8767

Pratt, M. K., & White, S. K. (2019, April 17). *What is a business analyst? A key role for business-it efficiency*. CIO. Retrieved October 8, 2022, from https://www.cio.com/article/2436638/projectmanagement-what-do-business-analysts-actually-do-for-software-implementation-projects.html

Pubrica. (2022, March 3). *Experimental study design: Research, types of design, methods and advantages*. Pubrica. Retrieved October 8, 2022, from https://pubrica.com/academy/experimental-design/experimental-study-design-research-types-ofdesign-methods-and-advantages/

Ramírez, V. (2018, August 4). *What is a prototype?* Medium. Retrieved October 8, 2022, from https://medium.com/nyc-design/what-is-a-prototype-924ff9400cfd

Rehkopf, M. (2019). *Scrum sprints: Everything you need to know*. Atlassian. Retrieved October 8, 2022, from https://www.atlassian.com/agile/scrum/sprints

Riaz, M. N. (2019). Implementation of kanban techniques in software development process: An empirical study based on benefits and challenges. *Vol 3 No 2*, *3*(2), 25–36. https://doi.org/10.30537/sjcms.v3i2.356

Saeed, S., Jhanjhi, N. Z., Naqvi, M., & Humayun, M. (2019). Analysis of Software Development Methodologies. *International Journal of Computing and Digital Systems*. Retrieved October 8, 2022, from https://journal.uob.edu.bh/bitstream/handle/123456789/3583/paper%202.pdf?sequence=1&isAllowed=y.

Sahay, A. (2016, July). *Peeling Saunders's Research Onion.* ResearchGate. Retrieved October 8, 2022, from Peeling Saunders's Research Onion

Sauer, J., Sonderegger, A., & Schmutz, S. (2020). *Usability, user experience and accessibility: Towards an integrative model*. ResearchGate. Retrieved October 8, 2022, from https://www.researchgate.net/publication/341653572_Usability_user_experience_and_accessibility_towards_an_integrative_model

Saunders, M. N. K., Lewis, P., Thornhill, A., & Bristow, A. (2019, March). *"Research methods for business students" Chapter 4: Understanding research philosophy and approaches to theory development*. ResearchGate. Retrieved October 8, 2022, from https://www.researchgate.net/publication/330760964_Research_Methods_for_Business_Students_Chapter_4_Understanding_research_philosophy_and_approaches_to_theory_development Scaled Agile. (2022, February 3). *Product owner*. Scaled Agile. Retrieved October 8, 2022, from https://www.scaledagileframework.com/product-owner/

Sebokwiki. (2022). *Why model?* Guide to the System Engineering Body of Knowledge. Retrieved October 8, 2022, from https://www.sebokwiki.org/wiki/Why_Model%3F

Serc. (2022, September 20). *What is a model?* Serc. Retrieved October 8, 2022, from https://serc.carleton.edu/sp/library/models/WhatIsAModel.html

Seth, G. (2020, December 4). Insulation from the user experience [web log]. Retrieved October 8, 2022, from https://seths.blog/2020/12/insulation-from-the-user-experience/.

Sharma, K. (2020, December 25). Top 12 Software Development Methodologies [web log]. Retrieved October 8, 2022, from https://www.tatvasoft.com/blog/top-12-software-developmentmethodologies-and-its-advantages-disadvantages/.

Shiff, L. (2020, March 2). What Does a System Architect Do? [web log]. Retrieved October 8, 2022, from https://www.bmc.com/blogs/system-architect/.

Signoretti, I., Salerno, L., Marczak, S., & Bastos, R. (2020). Combining User-centered Design and Lean Start-up with Agile Software Development: A Case Study of Two Agile Teams. In *Agile Processes in Software Engineering and Extreme Programming*. Springer. Retrieved October 8, 2022, from https://link.springer.com/content/pdf/10.1007/978-3-030-49392-9.pdf.

Silva-Rodríguez, V., & Nava-Muñoz, S. E. (2018). Identifying Human-Computer Interaction Patterns in Support of the User Interfaces Design. In *MexIHC '18: Proceedings of the 7th Mexican Conference on Human-Computer Interaction*. ACM Digital Library. Retrieved October 8, 2022, from https://dl.acm.org/doi/10.1145/3293578.3298781.

Stevens, E. (2022, June 22). What Is User Experience (UX) Design? Everything You Need to Know [web log]. Retrieved October 8, 2022, from https://careerfoundry.com/en/blog/ux-

Sulemani, M. (2021, January 4). What is a software process model? Top 7 models explained [web log]. Retrieved October 8, 2022, from https://www.educative.io/blog/software-process-modeltypes.

Swords, S. (2017, November 21). Software Development Project Roles and Responsibilities [web log]. Retrieved October 8, 2022, from https://www.atlascode.com/blog/software-developmentproject-roles-and-responsibilities/.

Systems thinking Handout. (2008). *What is a model?* Systems thinking Handout. Retrieved October 8, 2022, from https://fhm.fs.fed.us/emc/nepa/ecr2008/sessions/materials/18/Modeling%20&%20System%20Handout.pdf

Techopedia. (2017, December 12). *What is a developer?* Techopedia. Retrieved October 8, 2022, from https://www.techopedia.com/definition/17095/developer

Techopedia. (2022). *What is software development life cycle model (SDLC)?* . Techopedia. Retrieved October 8, 2022, from https://www.techopedia.com/definition/25976/software-development-lifecycle-model-sdlc

TechTarget. (2022). *UI and UX Design*. TechTarget. Retrieved October 8, 2022, from https://www.techtarget.com/searchsoftwarequality/resources/UI-and-UX-design

The Career Coach Company. (2022). *Software testing*. The Career Coach Company. Retrieved October 8, 2022, from https://www.careerco.ca/software-testing-training/?v=707f3a40153b

The Interaction Design Foundation. (2016). *Integrating UCD and agile methodologies*. The Interaction Design Foundation. Retrieved October 8, 2022, from https://www.interactiondesign.org/literature/article/integrating-ucd-and-agile-methodologies

The Interaction Design Foundation. (2022). *What is user experience (UX) design?* The Interaction Design Foundation. Retrieved December 18, 2022, from https://www.interactiondesign.org/literature/topics/ux-design

The Open University. (2022). *Models*. The Open University. Retrieved October 8, 2022, from https://www.open.edu/openlearn/science-maths-technology/computing-ict/models-andmodelling/content-section-2.1

The Standish Group. (2015). *Chaos report 2015*. The Standish Group. Retrieved October 8, 2022, from https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf

Tran, T. H. (2019, June 24). User-centered design: Definition, examples, and tips [web log]. Retrieved October 8, 2022, from https://www.invisionapp.com/inside-design/user-centered-designdefinition-examples-and-tips/.

University of Minnesota. (2022). *Collecting data*. University of Minnesota. Retrieved October 8, 2022, from https://cyfar.org/collecting-data

Usertesting. (2019, March 27). What is UX design? 15 user experience design experts weigh in [web log]. Retrieved October 8, 2022, from https://www.usertesting.com/blog/what-is-ux-design-15user-experience-experts-weigh-in.

UX Design Institute. (2019, July 2). Moving from graphic design to UX: a practical guide [web log]. Retrieved October 8, 2022, from https://www.uxdesigninstitute.com/blog/graphic-design-to-ux/.

Ventera. (2021). What is User-centered Agile Development? Retrieved October 8, 2022, from https://www.ventera.com/what-is-user-centered-agile-development/.

White, C. (2020, August 19). What Does a UX Designer Actually Do? [web log]. Retrieved October 8, 2022, from https://careerfoundry.com/en/blog/ux-design/what-does-a-ux-designer-actually-do/.

Wilson, V. (2014). Research Methods: Triangulation . *Evidence Based Library and Information Practice*, *9*(1), 74–75. Retrieved October 8, 2022, from https://journals.library.ualberta.ca/eblip/index.php/EBLIP/article/download/21469/16225.

Windsor, G. (2020, February 28). 5 Stages of the Agile System Development Life Cycle [web log]. Retrieved October 8, 2022, from https://medium.com/brightwork-collaborative-projectmanagement-blog/5-stages-of-the-agile-system-development-life-cycle-brightwork-coma207bdf61696.

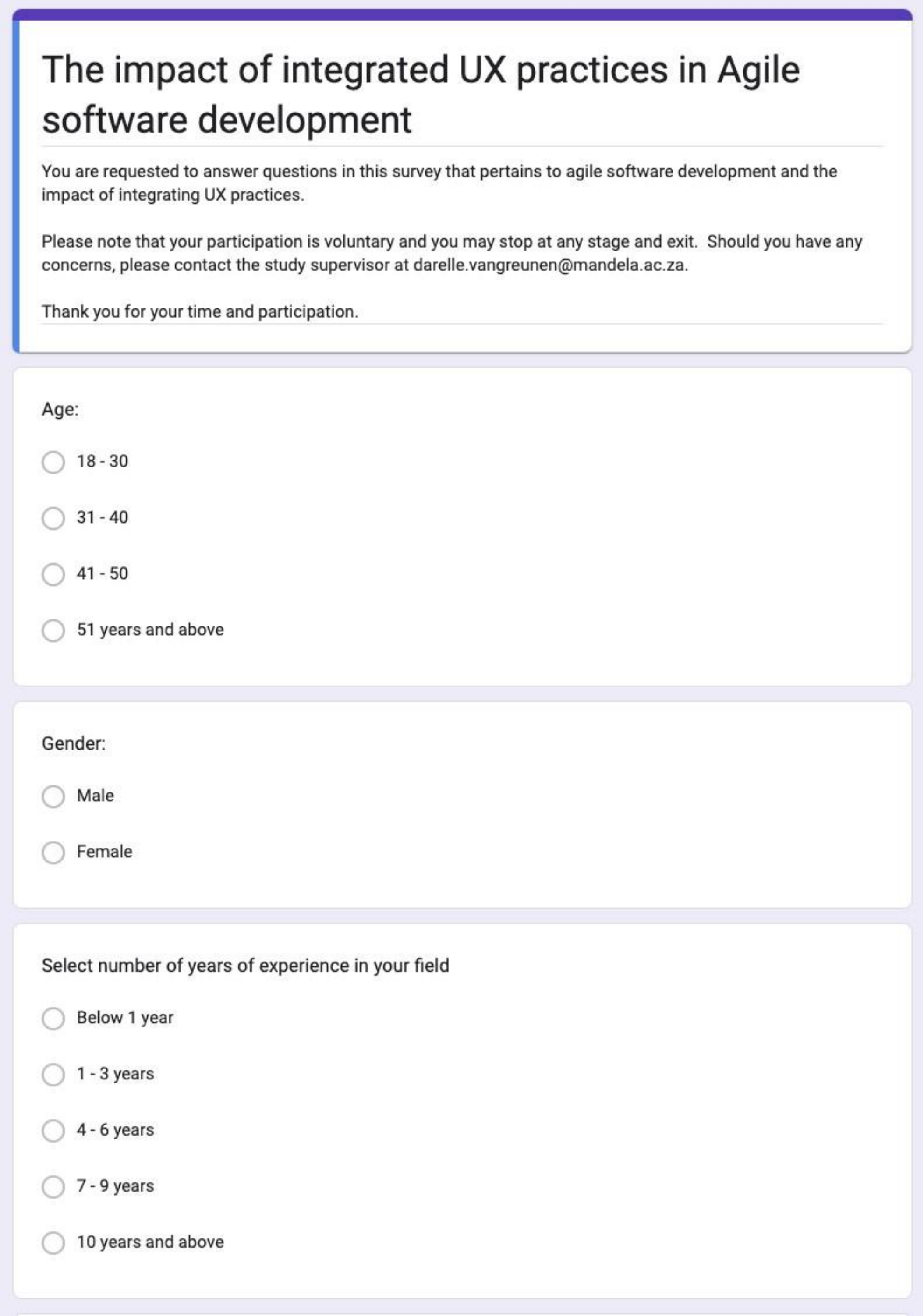
Wordpress. (2016, November 8). *Waterfall method*. Wordpress. Retrieved October 8, 2022, from https://miguelcbrm.wordpress.com/2016/11/08/waterfall-method

Zayed University. (2022). *Ethical Clearance for Research*. Zayed University. Retrieved October 8, 2022, from https://www.zu.ac.ae/main/en/research/for_researchers/research_integrity/ethical_clearance.aspx

Zorzetti, M., Salerno, L., Marczak, S., & Bastos, R. (2022). Improving Agile Software Development using User-Centered Design and Lean Startup. In I. Signoretti (Ed.), *Information and Software Technology* (Vol. 141). essay. Retrieved October 8, 2022, from https://www.sciencedirect.com/science/article/abs/pii/S0950584921001701.

Zorzetti, M., Vaccaro, M., Moralles, C., Prauchner, B., Signoretti, I., Pereira, E., Salerno, L., Bastos, R., & Marczak, S. (2020). Maturity models for agile, lean startup, and user-centered design in software engineering: A combined systematic literature mapping. *Proceedings of the 22nd International Conference on Enterprise Information Systems*, *2*, 145–156. https://doi.org/10.5220/0009582401450156

APPENDIX A: Demographic information and Awareness of UX Agile methods



The impact of integrated UX practices in Agile software development

You are requested to answer questions in this survey that pertains to agile software development and the impact of integrating UX practices.

Please note that your participation is voluntary and you may stop at any stage and exit. Should you have any concerns, please contact the study supervisor at darelle.vangreunen@mandela.ac.za.

Thank you for your time and participation.

Age:

○ 18 - 30

○ 31 - 40

○ 41 - 50

○ 51 years and above

Gender:

○ Male

○ Female

Select number of years of experience in your field

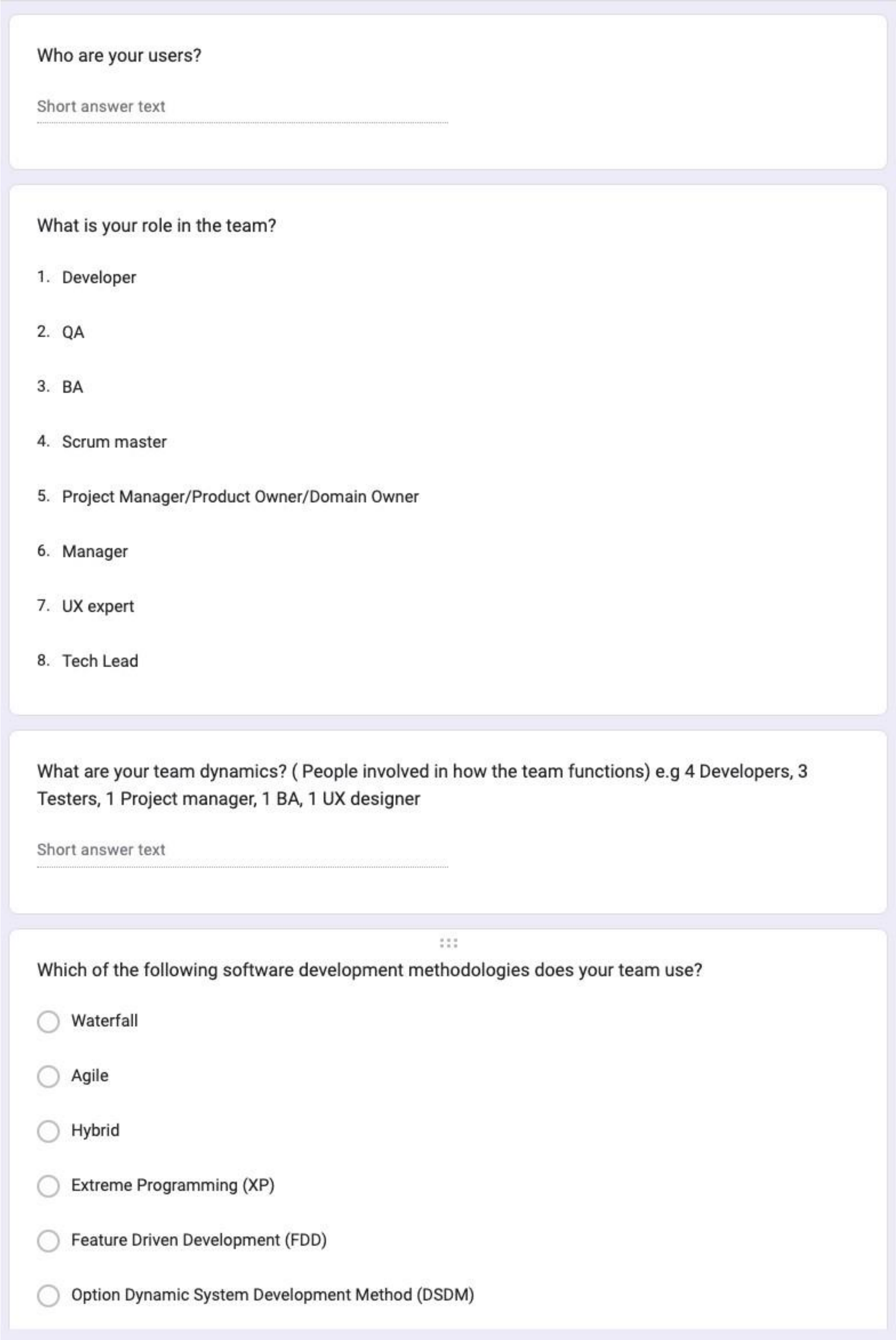
○ Below 1 year

○ 1 - 3 years

○ 4 - 6 years

○ 7 - 9 years

○ 10 years and above



Who are your users?

Short answer text

What is your role in the team?

1. Developer
2. QA
3. BA
4. Scrum master
5. Project Manager/Product Owner/Domain Owner
6. Manager
7. UX expert
8. Tech Lead

What are your team dynamics? (People involved in how the team functions) e.g 4 Developers, 3 Testers, 1 Project manager, 1 BA, 1 UX designer

Short answer text

Which of the following software development methodologies does your team use?

○ Waterfall

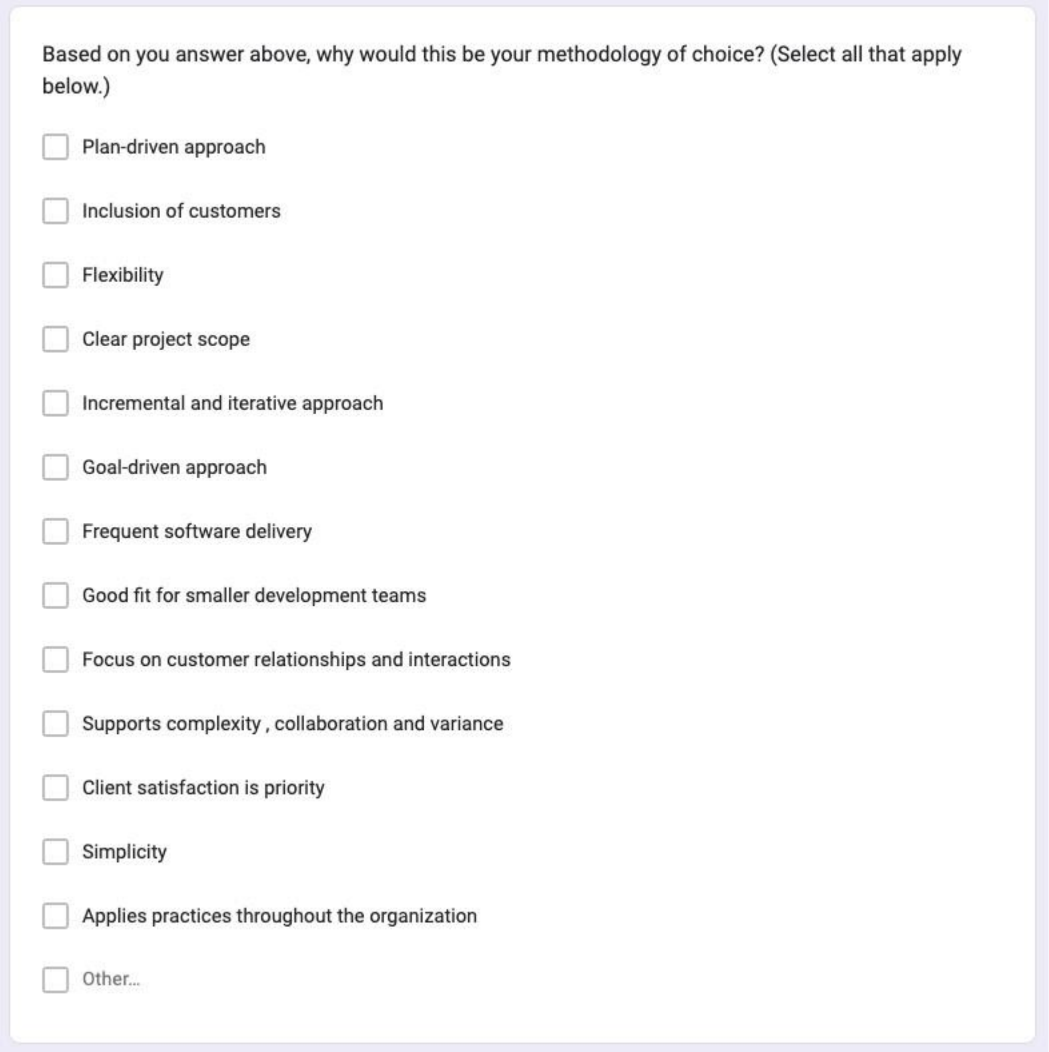
○ Agile

○ Hybrid

○ Extreme Programming (XP)

○ Feature Driven Development (FDD)

○ Option Dynamic System Development Method (DSDM)



Based on you answer above, why would this be your methodology of choice? (Select all that apply below.)

□ Plan-driven approach

□ Inclusion of customers

□ Flexibility

□ Clear project scope

□ Incremental and iterative approach

□ Goal-driven approach

□ Frequent software delivery

□ Good fit for smaller development teams

□ Focus on customer relationships and interactions

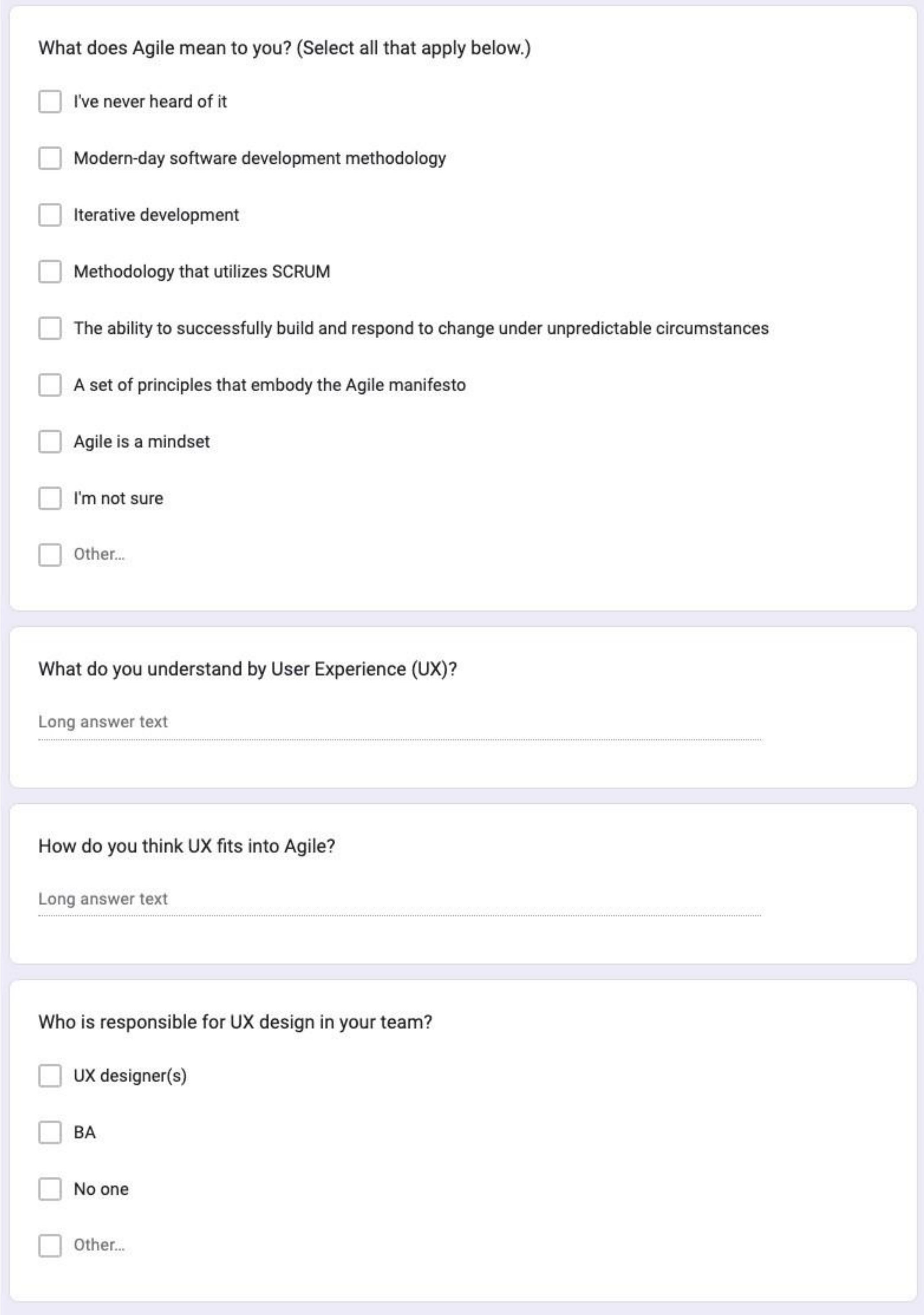
□ Supports complexity , collaboration and variance

□ Client satisfaction is priority

□ Simplicity

□ Applies practices throughout the organization

□ Other...



What does Agile mean to you? (Select all that apply below.)

□ I've never heard of it

□ Modern-day software development methodology

□ Iterative development

□ Methodology that utilizes SCRUM

□ The ability to successfully build and respond to change under unpredictable circumstances

□ A set of principles that embody the Agile manifesto

□ Agile is a mindset

□ I'm not sure

□ Other...

What do you understand by User Experience (UX)?

Long answer text

How do you think UX fits into Agile?

Long answer text

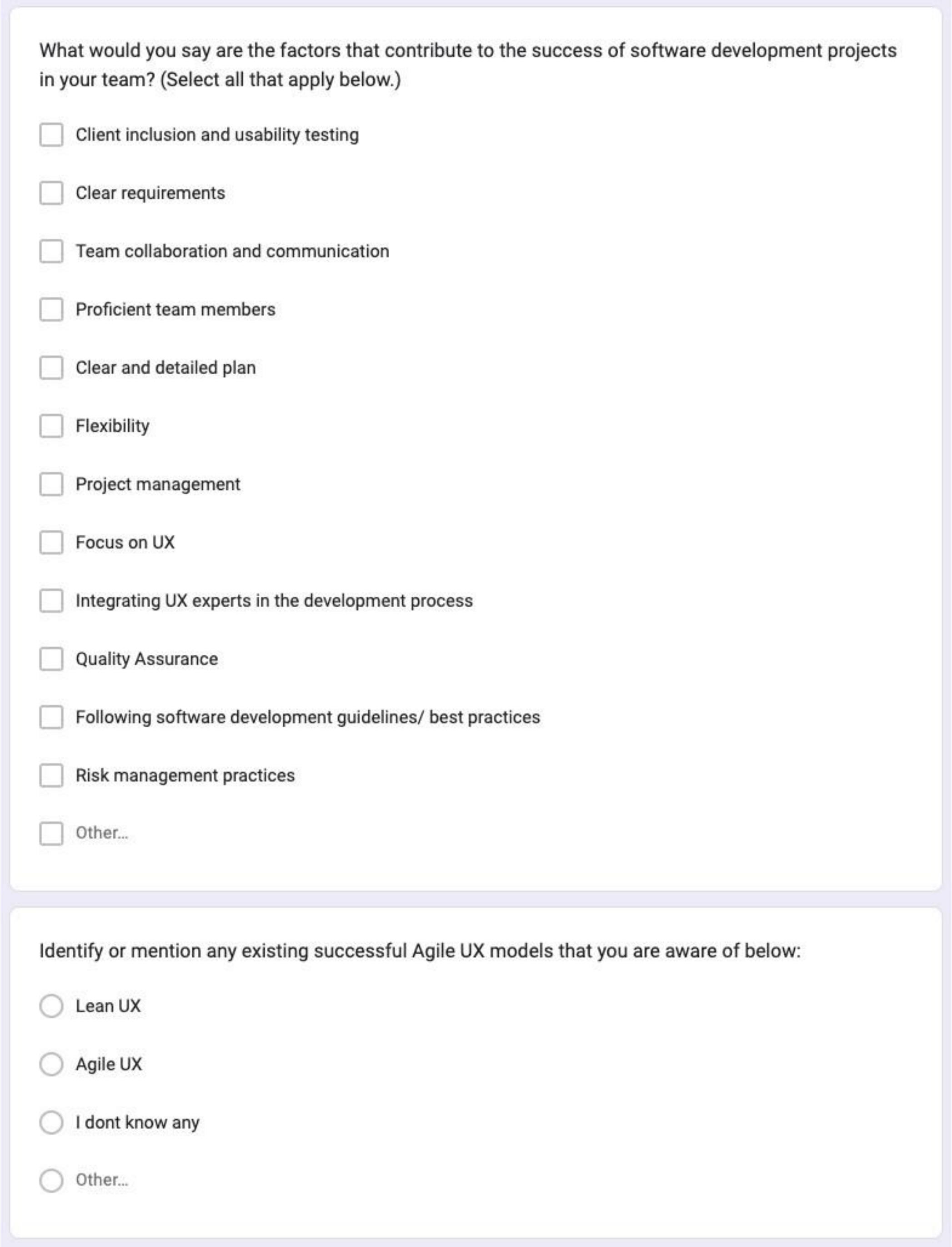
Who is responsible for UX design in your team?

□ UX designer(s)

□ BA

□ No one

□ Other...



What would you say are the factors that contribute to the success of software development projects in your team? (Select all that apply below.)

□ Client inclusion and usability testing

□ Clear requirements

□ Team collaboration and communication

□ Proficient team members

□ Clear and detailed plan

□ Flexibility

□ Project management

□ Focus on UX

□ Integrating UX experts in the development process

□ Quality Assurance

□ Following software development guidelines/ best practices

□ Risk management practices

□ Other...

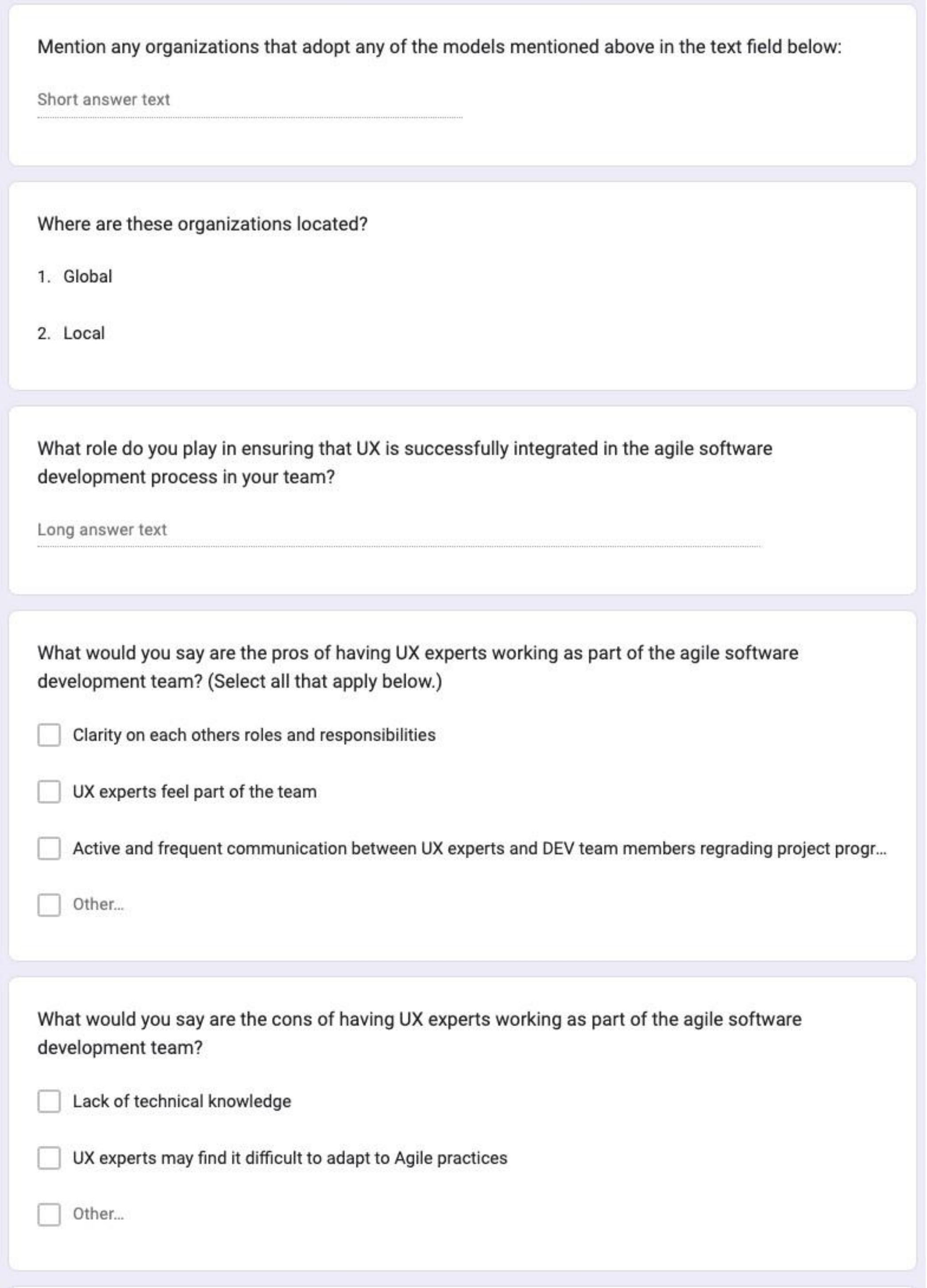
Identify or mention any existing successful Agile UX models that you are aware of below:

○ Lean UX

○ Agile UX

○ I dont know any

○ Other...



Mention any organizations that adopt any of the models mentioned above in the text field below:

Short answer text

Where are these organizations located?

1. Global
2. Local

What role do you play in ensuring that UX is successfully integrated in the agile software development process in your team?

Long answer text

What would you say are the pros of having UX experts working as part of the agile software development team? (Select all that apply below.)

□ Clarity on each others roles and responsibilities

□ UX experts feel part of the team

□ Active and frequent communication between UX experts and DEV team members regrading project progr...

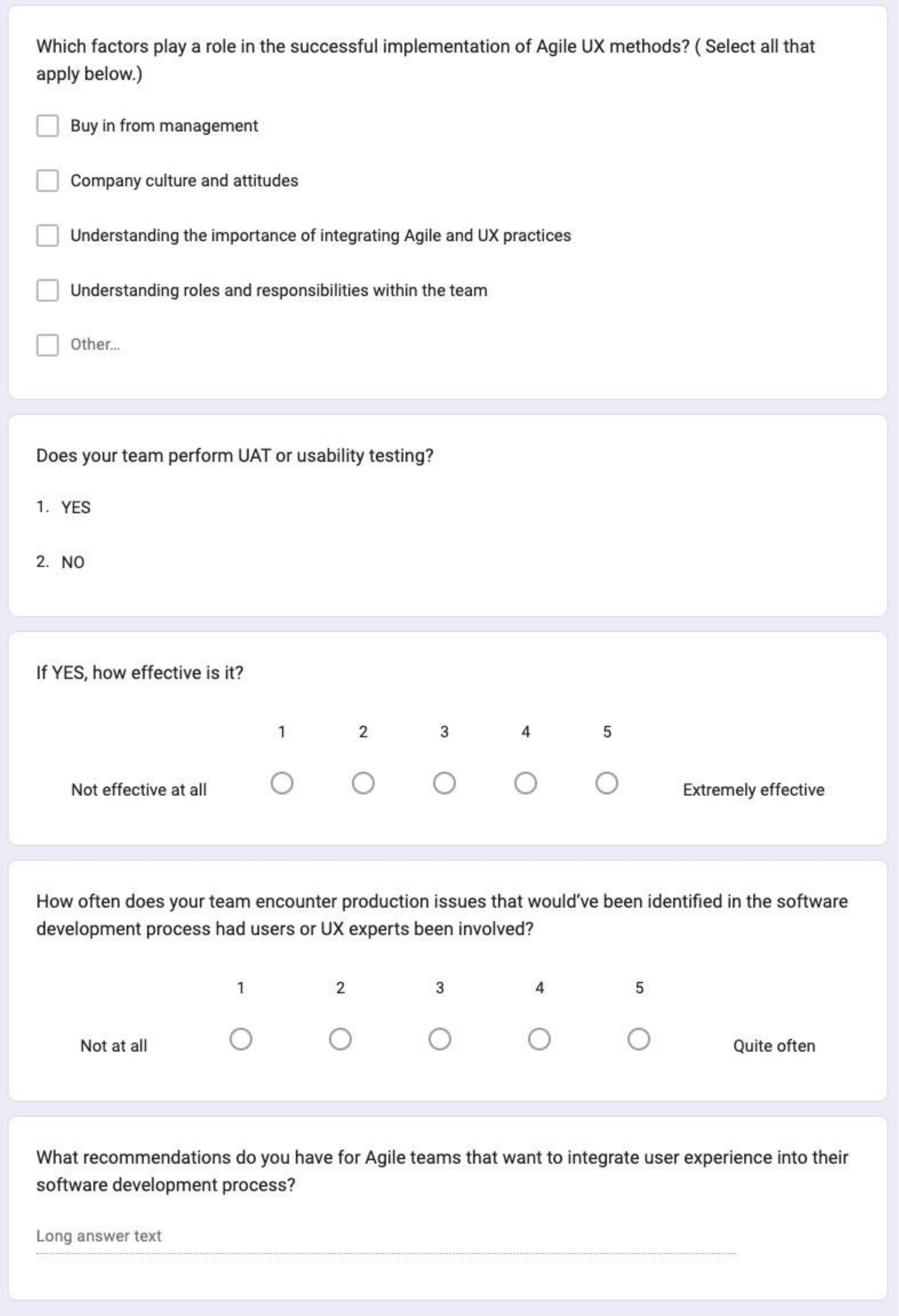
□ Other...

What would you say are the cons of having UX experts working as part of the agile software development team?

□ Lack of technical knowledge

□ UX experts may find it difficult to adapt to Agile practices

□ Other...



Which factors play a role in the successful implementation of Agile UX methods? (Select all that apply below.)

□ Buy in from management

□ Company culture and attitudes

□ Understanding the importance of integrating Agile and UX practices

□ Understanding roles and responsibilities within the team

□ Other...

Does your team perform UAT or usability testing?

1. YES
2. NO

If YES, how effective is it?

	1	2	3	4	5	
Not effective at all	○	○	○	○	○	Extremely effective

How often does your team encounter production issues that would've been identified in the software development process had users or UX experts been involved?

	1	2	3	4	5	
Not at all	○	○	○	○	○	Quite often

What recommendations do you have for Agile teams that want to integrate user experience into their software development process?

Long answer text

APPENDIX B: Guiding Interview questions

Identify existing best practices /model (name them) – research question: which Agile UX models exist?

1. What type of business or service does your organisation provide?
2. What are your team dynamics and what is your role?
3. Which software development methodology does your company or team use? And what is the process of that methodology? Which methodology do you prefer and why?
4. What do you understand by Agile?
5. What do you understand by UX?
6. How do you think UX fits into agile?
7. Who is responsible for UX design in your organisation? If at all
8. What type of model would you say your organisation has adopted to integrate users into the software development process? If any. Give it a name
9. Do you perhaps know any successful existing Agile UX models?
10. Do you think BAs contribute to UX as they're objective is to gather user requirements for the software development team to use to develop systems?
11. Do you think you would be able to perform your role in isolation to your team? Why?
12. Based on above question do you think it is important to have UX experts part of the software development team instead of having them work in isolation? Why?

Identify existing challenges of integrating Agile and UX (based on an existing model, name model)

1. Who drives the decision making behind the activities that are performed by your software development team? organisational heads/superiors -
2. Do they understand the importance of UX within the software development team
3. Do you think that the UX model used in your organisation or team works?
4. If yes why? If not why?

5. Do you have any suggestions on how to improve it?
6. What do you think is a hindrance to the success of Agile UX models in general?
7. How easy would it be to include users in your software development life cycle?
8. Who in your team would be responsible for that? And why?
9. Would you prefer to have them onsite or offsite? If onsite why? If offsite why?

Identify the impact of integrating Agile and UX (based on an existing model, name model)

1. What would you say are the factors that contribute to the success of your software development projects?
2. Identify any existing successful Agile UX models in genera. Explain why you think they are successful?
3. Do you know any companies that use these models?
4. Where are they located?
5. What role do you play in the successful integration of ensuring UX is included in agile software development process in your organisation?
6. What would you say are the pros and cons of having UX experts working as part of the agile software development team?
7. Does your team do UAT or usability testing?
8. How effective is it?
9. How often does your team receive production issues that would’ve been identified in the software development stage had users or UX experts been involved in the software development process?