Agile User Experience:

Integrating Good User Experience Development Practices into Agile within the South African Context

By

Narendren Pillay

Submitted in fulfilment of the requirements of the degree of

Masters Degree in Information and Communications Technology

In the Faculty of

Accounting and Informatics

at the Durban University of Technology

September 2020

Supervisor: _ _____ Date: 7 September 2020
Dr. Jeanette Wing - PhD: IT (DUT)

Co-Supervisor: Date: 7 Sep 2020
Dr. Alveen Singh – D. Tech: IT (DUT)

DECLARATION

I, Narendren Pillay, declare that this research dissertation which I hereby submit for the Masters in Information and Communications Technology (MICT) at Durban University of Technology is my own work and all relevant references that have been quoted or used, have been included in the reference list. This study has not been previously submitted by me for a qualification at this or any other tertiary institution.

The framework for integrating Agile and User Experience proposed in this thesis was presented at the Information Communication Technology and Society Conference (ICTAS) on the 6th March 2019 in Durban, South Africa. Parts of this paper have been re-interpreted and rewritten in this thesis; some passages have been quoted verbatim from the paper.

The paper has been published as: Pillay, N. and Wing, J. 2019. Agile UX: Integrating good UX development practices in Agile. In: Proceedings of 2019 Conference on Information Communications Technology and Society (ICTAS). IEEE, 1-6.

Student: Narendren Pillay — 13/01/2020 Date

Supervisor: Dr Jeanette Wing — 13/01/2020 Date

Co-Supervisor: Dr Alveen Singh — 13/01/2020 Date

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude and deepest appreciation to my supervisor, Doctor Jeanette Wing. Thank you for the guidance, patience and motivation throughout this study. You pushed me to reach new heights and I am eternally grateful for having a perfectionist as my mentor. I will certainly miss our frequent chats and meetings. On a personal level, you were supportive, caring and compassionate through a very difficult time in my life. Your valuable time and wisdom has made this thesis a reality.

Second, I would like to thank my co-supervisor, Doctor Alveen Singh for his invaluable input and for steering this research study in the direction that it required. Your support and advice gave me the courage to step outside the box.

This research could not have been completed without the support from a host of friends. Thank you Yuvir Ramsager for convincing me to register for my Masters, Ashika Jugmohan for encouraging me every step of the way, Kathleen Naidoo who shared this academic journey with me and was able to relate to the personal sacrifices and hard work that was required, and Leonie van der Westhuizen who always checked up on my progress and cheered me on. It is with a heavy heart that I remember my late friend, Katherine George for her assistance and guidance during the early stages of this study.

To my manager, Shaygen Arumugam, thank you for allowing me the time whenever required to focus on this study and to meet with my Supervisors. I will always be grateful for your understanding in a personal and professional capacity. My work colleagues have also been supportive and shared every milestone with me. Thank you, team.

I further express my love and gratitude to my parents and my family for instilling faith, courage and perseverance in me to achieve my goals and aspirations. To my late brother, Kavi, the last year and a half was a tough journey for you. I was there with you every step of the way. I wish you were here to celebrate this very special moment with me. I know that you are shining down on me from heaven,

until we meet again. A special thank you to my aunt Margie, for pushing me to further my studies and to reach new heights.

It was indeed a great achievement for me to complete this research. I thank God for guiding me and gracing me with good health and strength during this study period.

ABSTRACT

Agile software development has proliferated over the last two decades and become one of the dominant frameworks used by software development companies. Agile development methods and User Experience (UX) both strive towards providing software that meets the users' needs.

The purpose of this study is twofold; firstly, to study the current literature and approaches of integrating UX into Agile software development with the intention of combining it into good development practices for use by Agile and UX practitioners. Secondly, to investigate how UX is integrated into Agile in the South African context thus confirming if literature from studies conducted abroad apply within the South African context. A review of the literature confirms that there are a significant number of publications on Agile software development from a South African perspective (Joseph and Santana 2016; Chiyangwa and Mnkandla 2017; Sebega and Mnkandla 2017; Mudarikwa and Grace 2018). However, there are no publications that have explored UX integration within an Agile software environment. Recent research attempts, such as those by Coleman (2018) and Brosens (2018), are evidence of the growing interest in UX; however, these studies do not provide a higher level of abstraction on Agile UX integration practices. This research presents results of a qualitative study on how UX and Agile can be integrated. UX cannot be quantified or adequately explained by using variables or experiments. The grounded theory research method has been used. It is qualitative in nature and the theory consists of iterative data collection and analysis with an aim of producing a theory.

The results of this study highlighted approaches for increasing user involvement in Agile and good development practices to integrate UX into Agile. It also highlighted issues and challenges experienced. This research offers insight for UX/Agile practitioners and adds academic value in the form of a generic framework for the integration of UX into Agile. The framework has been developed through the lens of the Design Thinking paradigm.

Keywords: User Experience; UX; User Centred Designs; Agile; Agile UX; Usability.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

CHAPTER 1: INTRODUCTION

This thesis aims to investigate the integration of good UX development practices into Agile software development within the South African context. This chapter provides an overview of what is covered in the rest of the thesis. It begins with a background to the problem of integrating UX into Agile before providing an overview of the research problem. The chapter continues with an overview of the aims and objectives, the research questions and the academic value and contribution of this study. The chapter concludes with a roadmap on what is expected in the remaining chapters of this thesis.

1.1 Background

In the kingdom of buzzwords, “agile” has evoked a lot of attention. Authors such as Creasey (2019) shed some light on the difference between the agile organization (agility) and the Agile method – lower case “agile” and upper case “Agile”. The differentiation, according to Creasey (2019), is the lower case “agile” refers to durability, speed, flexibility and preparedness of an organization. On the other hand, the upper case “Agile” refers to the iterative process of software development and the associated principles, values, methods, roles, processes and tools (Creasey 2019). The researcher adopted the upper case “Agile” for this study; the aim of this research is to investigate better ways of developing software by integrating good UX development practices into Agile. This involves an in-depth analysis of the principles, values, methods, roles, processes and tools that influence the integration practices. For this reason, Agile appears in uppercase throughout this thesis.

The literature is amassed with studies that use Agile as a software development framework. The principles of Agile software development, as established in the Agile manifesto (Agile Manifesto 2001), recommends techniques and development guidelines that are guided by a set of principles. However, Agile on its own does not address the usability of the software product. The need for good User Experience (UX) has become more evident in today’s software development industry. The success or failure of any software application depends on whether the end-user finds it useful and whether it meets their expectations (Jurca,

Hellmann and Maurer 2014). User Experience can have major implications on a project as well as an organization if it is not managed appropriately.

Nielsen Norman Group (2019) has defined UX as a broad term that encompasses all interactions that a user has with products or services that a company has to offer. The same authors opine that UX development approaches prioritize the needs of the user over the needs of the system. A review of the literature frequently reveals that both Agile software development and user-centred design are two important processes for ensuring good UX (Sy 2007; Al Thunibat, Zin and Sahari 2011; Jurca, Hellmann and Maurer 2014; Verdiesen 2014). However, integrating UX into Agile remains difficult as Agile methods do not adequately cater for UX methods (Jurca, Hellmann and Maurer 2014). Emphasis on UX work begins in the early phases of projects before starting development which results in a product or service that does not meet the users' needs.

Scrum has been recognised in the literature as one of the most widely used Agile frameworks (Cho 2008; Singh 2008; ScrumAlliance 2009; Akif and Majeed 2012; Lárusdóttir, Cajander and Gulliksen 2012). Despite the popularity of Scrum, a review of the literature reveals that there are a low number of publications that have explored the integration practices of UX into Agile. Recent publications by a few authors show a growing interest in the Agile UX space to explore well accepted principles and good practices for integrating these two disciplines (Nadikattu 2016; Convertino and Frishberg 2017; Kieffer, Ghouti and Macq 2017; McKendrick 2018; Spool 2018).

Authors such as Nadikattu (2016) advocate that research needs to be conducted to get a deeper understanding of the challenges that are experienced when integrating UX in Agile. The same author opines that research needs to be conducted on across small, medium and large organizations that have varying development methodologies. This argument is well supported by Da Silva et al. (2011) who have presented 58 research articles on Agile UX and opine that most of the studies are exaggerated in their descriptions and conclusions of experiments rather than exploring the best practices of UX in Agile. It is also widely recognized by authors such as Fox (2010) that there are social

ramifications when the UX team joins the software project team. Further investigations need to be carried out to find an effective way of integrating UX into Agile software development projects to minimize social out casting and maximize team-gelling (Fox 2010).

This study contends that there is a lack of good development practices for integrating UX into Agile. There is also limited knowledge and no empirical proof of integrating UX in Agile within the South African context. The objective of this research is thus to investigate the relationship between UX and Agile in the South African context, the current approaches of integration and to confirm if the results from abroad applies within the South African context.

1.2 Research Problem

The Agile framework and UX have the same goal: to develop a software product that the user wants and needs. The literature frequently promotes the benefits of using Agile methods together with UX, as software is developed by and for people (Sy 2007; Fox 2010; Da Silva, Martin, Maurer and Silveira 2011; Abrahamsson, Salo, Ronkainen and Warsta 2017). In the context of software development, Agile methods apply time boxed iterative development, planning and delivery. Moreover, it includes values and practices that promote rapid and flexible response to change (Larman and Basili 2003). Agile development values are similar to the fundamental principles of UX development, e.g. focus is on individuals and interactions and frequent collaboration with the customer (Da Silva *et al.* 2011).

As advocated by Da Silva *et al.* (2011), Agile methods are lightweight; the team should focus on what is necessary to bring value to the customer. UX, on the other hand, shows the importance of using real users in the development of the design of the system or product (Da Silva *et al.* 2011).

UX designers and Agile developers approach development of software from different perspectives. According to Fox (2010) in terms of design, Agile software development addresses development from the perspective of the code. UX development, on the other hand, concentrates on the design in terms of how the

user interface is being used to accomplish the required task by the user (Fox 2010). Supporting arguments by authors such as Leffingwell (2016) and Nadikattu (2016) allude that Agile methods focus more on delivering software quickly with minimal user input and upfront design. However, UX methods focus on allocating more resources and time for user research and testing before commencing with development. Leffingwell (2016) makes a rather strong statement that Agile teams spend more time resolving user interactions instead of developing and testing incremental deliverables, which results in more iterations. Beck (2000) traces these pertinent challenges to when the Agile manifesto was released in 2001 as it focused more on the engineering aspect of software development and failed to identify UX or usability as essential attributes. Many organisations face issues with integrating UX in Agile, due to a lack of formal guidelines or best practices in the Agile manifesto (Beck, Beedle, Van Bennekum, Cockburn, Cunningham, Fowler and Jeffries 2001).

Literature concerning the challenges on how to incorporate UX into rapid iteration cycles, including implementation of visual design, navigation, and user interface (UI) elements, remains largely unsettled and under theorized. It is thus the view of this study that the current Agile methodology does not provide formal integration methods/good development practices for UX in the software development cycles. Moreover, there is a gap in terms of well-established and well accepted integration practices that both Agile and UX practitioners can understand and apply.

There is also no proof in the South African context of identifying and implementing good UX development practices into Agile. A review of the literature confirmed that there are a significant number of publications on Agile software development from a South African perspective (Joseph and Santana 2016; Chiyangwa and Mnkandla 2017; Sebega and Mnkandla 2017; Mudarikwa and Grace 2018). There is a growing interest in UX as evidenced by Brosens (2018) and Coleman (2018). However, the studies by these authors do not provide a higher level of abstraction on Agile UX integration within an Agile software development environment and call for further research to be conducted within the South African context.
The researcher aims to investigate Agile UX integration practices with the intention of providing academic value to the Agile/UX community.

1.3 Aim and Objectives of This Study

The research question, aim and objective of a study is inextricably linked to each other (Doody and Bailey 2016). For this study the researcher considered the following factors, as proposed by Doody and Bailey (2016), when determining the aim:

- Adding to the knowledge base
- Measure change
- Understanding complex phenomena
- Generating new ideas
- Examining the past

The major proposition of this study is to investigate how UX is integrated into Agile software development in South Africa as there appears to be a gap in knowledge. This line of argument has resulted in the following research aim:

To investigate how UX development is currently integrated into Agile software development within the South African context.

In order to elucidate this overarching aim, it is incumbent for the researcher to cover all aspects of the problem ordered in a logical sequence and should also take into consideration the available resources (Doody and Bailey 2016).

This study's research objectives (RO) can therefore be listed as follows:
RO1: To study how UX developers collaborate with members of the Agile team in the software development process.

RO2: To understand how UX is perceived by software development organisations, in order to identify the extent at which organisations have been successful in incorporating UX into Agile

RO3: To evaluate the existing Agile UX software development process in the South African context and confirm if approaches from previous studies applies to the South African context.

RO4: To propose a generic framework for integration of UX in Agile and to provide further recommendations for efficient integration of UX in Agile.

1.4 Research Questions

The formulation of the research question involved a fair degree of mechanics. This study adopted the process of research question formulation proposed by Thabane, Thomas, Ye and Paul (2009), as follows:

1. Identify the research problem.
2. Consider options for factors such as the population, intervention, control, outcome and time frame.
3. Formulate the research question.
4. Check if the question follows the FINER criteria (i.e. is it **F**easible, **I**nteresting, **N**ovel, **E**thical and **R**elevant?)

Question one and two were achieved through the review of literature review and investigation of current studies in the South African context. Despite the researcher's efforts there were limited studies in South Africa. The lack of theoretical contributions in the research area required further refinement for the research questions during the course of the fieldwork. After using the process above, the research question that was formulated is as follows:

"How can good UX development practices be integrated into Agile within the South African context?"

The following sub questions will assist in investigating a conclusion to the main question:

1. "What Agile development techniques are best suited to ensure UX requirements are met?"
2. "What are good development practices for creating good UX?"
3. "What are the roles of team members that are directly involved in the process of integrating UX in Agile?"
4. "What are the problems when integrating UX in Agile?"

1.5 Academic Value and Contribution of Study

Agile is a dominant framework in the software development industry and the challenges surrounding the integration of UX proved to be a relevant topic for investigation. This study provides an opportunity to learn more about Agile and UX and to observe it in practice. It also provides an opportunity to discover the challenges faced by Agile and UX practitioners.

The theoretical contribution of this research is the development of a generic framework for integrating UX in Agile. The proposed framework was developed using the principles of design thinking. The academic value of the proposed framework is the result of exploiting the processes of the design thinking methodology coupled with the results from existing literature (Chapter Two) and the results and analysis from this study (Chapter Four). The theoretical contributions fall in category five of the 'Five Types of Theory in Information Systems' (Gregor 2006). Type five relates to *Theory for design and action*; according to Gregor (2006) it justifies how to do something by taking into account the principles of form and function, methods and theoretical knowledge. The criteria for theory for design and action include "utility to the users, the novelty of the artifact and the persuasiveness of the claims that it is effective" (Gregor 2006: 629).

The practical contribution of this research is the evaluation of the framework. The proposed framework was evaluated by a focus group comprising of Agile UX experts. The academic value of using experts to evaluate the framework aimed to translate the framework back into practice. Contributions by the focus group assisted in refining the framework. To the researcher's best knowledge, the combination of good integration practices for Agile UX that have been proposed in the framework has not been used before in the development of software using Agile in the South African context.

The analysis of results will assist Agile and UX practitioners to identify good development practices for integrating UX in Agile. Furthermore, this study contributes to academic knowledge concerning UX and Agile within a South

African context as there is limited knowledge in this field. Ultimately, the vision of this research is to assist both UX and Agile practitioners to improve the current state of integration.

1.6 Limitations

The study focuses on organizations based specifically in Kwazulu-Natal, South Africa. However, it is important to note that most of the companies used for the purpose of this research are national companies, with divisions in other provinces in South Africa. The development framework adopted in one division impacts the organization as a whole. Hence, the assumption is that results from the study (although conducted in Kwazulu-Natal), applies to the South African context.

In addition to the above, the results from this study are only valid for the period in which it was conducted, as new developments in the Information Technology environment will have an impact on the results.

1.7 Chapter Outline

The remaining chapters in this thesis are organised as follows:

Chapter Two is a synthesis of the two areas of interest, namely Agile software development and UX, in terms of the history, definition and the interplay between UX and Agile software development. The Scrum methodology is identified as most commonly used methodology. However, the similarities demonstrated by Scrum indicate that the integration of UX is possible in the other Agile methodologies as well. A synthesis of the practices of Scrum is uncovered to provide a foundation for the fieldwork. The presentation of the literature aims to focus the research effort on the known difficulties, good practices and implications for this study. A suitable lens was sought to investigate the integration between Agile and UX.

Chapter Three describes the research methodology. A qualitative research approach was appropriate as this study aims to understand the human and social factors in the Agile UX community. Qualitative research allowed the researcher to elicit responses from participants in order to identify themes in the data.

Moreover, the human element in UX cannot be described using numbers neither can it be easily manipulated or measured using controlling variables (Verdiesen 2014). The researcher attempted to understand the varying perceptions of Agile and UX practitioners and their projects teams. The interpretative research stance played a vital role in the data collection. The choice of the associated data collection methods, which comprised of interviews and a questionnaire, proved to be adequate for this study. The chapter briefly explains the key concepts of transcription and the coding procedures of grounded theory that were utilized in the data analysis.

Chapter Four provides the results and more in-depth details on the data analysis of the interviews and questionnaires. First, the results from the literature review in terms of classification of previous studies is presented in order to identify known difficulties when integrating UX into Agile and the good practices that can be applied in bridging the gap. The key inferences drawn from the literature review influenced the design and the execution of this study. Second, a more in-depth analysis of the interviews and questionnaires is presented. The researcher used pseudonyms to protect the identity of the participants and their companies. The process of data analysis using grounded theory is illustrated. The chapter concludes by presenting the results from the interviews and questionnaires. The level of detail that is evident in this sub-section of the chapter was necessary to reveal the relationships between the categories. It was also helpful in unearthing the two major themes of this study, namely '*Agile UX Integration Practices'* and '*UX Vision'*.

Chapter Five presents the theoretical constructs from the two themes and the findings from the literature review which were used to shape the generic framework for integrating UX in Agile. Design thinking provided the basis for the development of the framework which allowed the researcher to design an artifact to solve the wicked problem of integrating UX in Agile. The framework conformed to the proposal provided by e Sá (2016) for design thinking. Evaluation of the model was facilitated by using a focus group. The participants were experts in the field of Agile and UX. The attitudes, beliefs and opinions of the participants in the focus group provided a greater understanding of the study area. This research

has culminated in findings that are believed to be relevant in understanding the nature of Agile UX and the integration practices that can support or hinder the process.

Chapter Six reflects on the results on this study. The research objectives and research questions are revisited to ascertain if they were attained. Possible areas for future research are also proposed.

The figure below illustrates the roadmap and serves as an overview of the whole study.

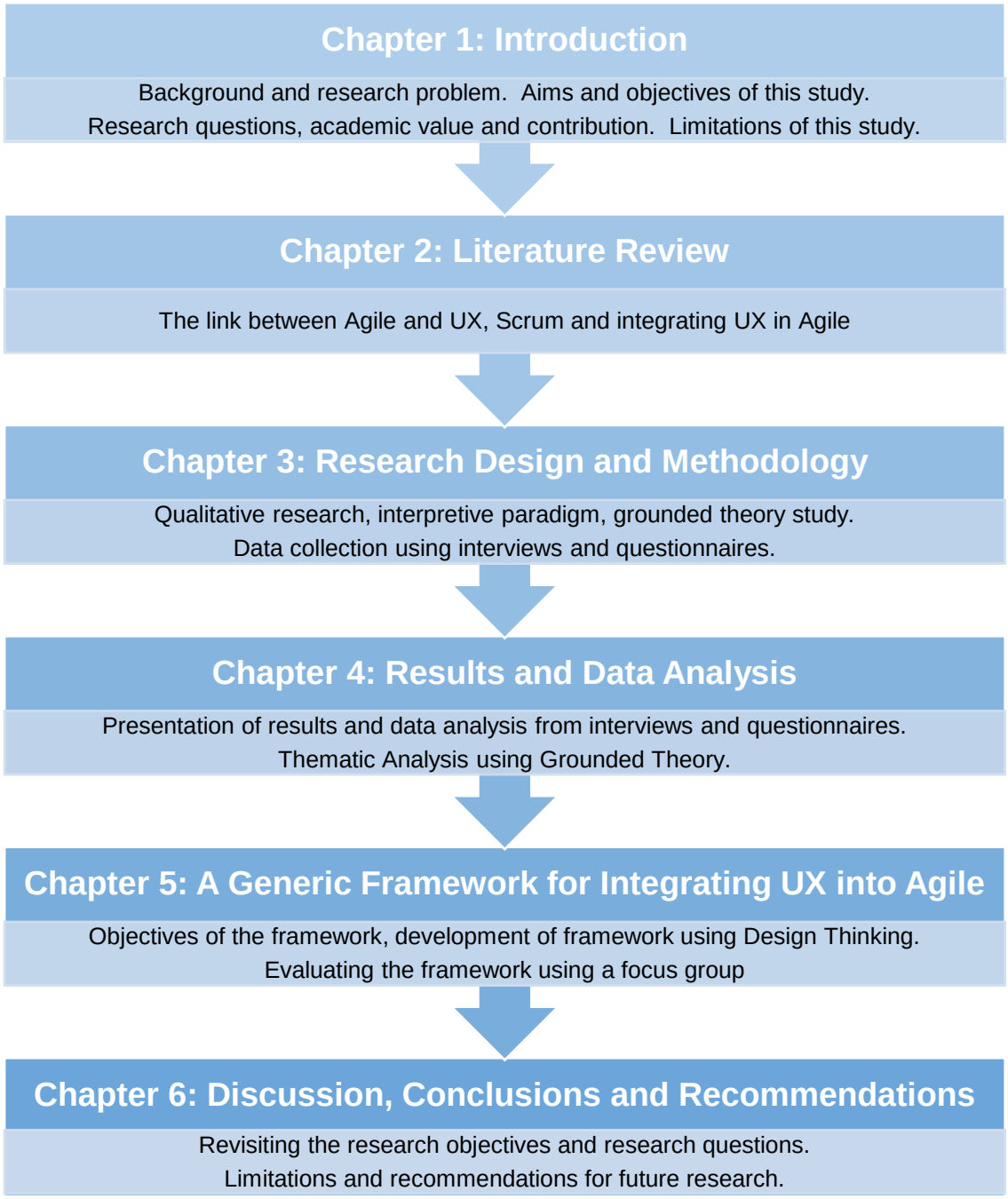


Figure 1: Roadmap of this thesis

CHAPTER 2: LITERATURE REVIEW

This chapter provides a detailed study of the existing literature relating to the three key research areas; these being Agile software development, User Experience and the integration of User Experience into Agile. The digital libraries JSTOR, Springerlink, Association for Computer Machinery (ACM), Scopus and IEEE Xplore were searched to locate papers. ResearchGate and Google Scholar were also used as resources to locate papers. The literature review was content based, and a combination of the following keywords was used in the search:

Table 1: Keywords used for search

Area of study	**Keywords**
Agile	Agile Scrum Kanban Lean
UX	User Experience Usability User-Centred Human-Computer Interaction

To formulate the search string, each of the keywords listed above were combined to broaden the search results and cast a wide net, e.g. Agile User Experience. The researcher also ensured that the search string included ‘South Africa’ to identify studies that were conducted in the South African context.

The process followed in this chapter is firstly to study the history of Agile, the methodologies and techniques. This is followed by a study on User Experience which focuses on the key principles of User Experience. Thirdly the study focuses on integrating Agile and User Experience, which examines the similarities, difficulties and good practices for integrating these two methodologies.

The findings will guide and assist in identifying a clearer understanding of how User Experience is integrated with Agile with the aim of answering the research

question of this study: *"How can good UX development practices be integrated into Agile within the South African context?"*

In the next sub-section, the literature has been categorised firstly to describe the context of Agile software development through a chronological history. This is followed by the various flavours of the Agile methodology in order to understand the impact they have on UX related integration in Agile. The link between Lean and Agile software development is then explored which lays the foundation for a better understanding of the Scrum methodology.

2.1. An Overview of Agile Software Development

2.1.1. A Brief History of Agile

Almost two decades ago, Agile software development took on a new meaning with the publication of the Agile manifesto by seventeen independent-minded software practitioners in 2001 (Boehm 2002; Highsmith 2002). Prior to the advent of Agile software development, software systems were developed using various models such as the Waterfall, Incremental and Spiral Models. These 'classical process models' offered a more formal and sequential approach to developing software (Kollmann 2008). According to Schach (2007), software was developed sequentially and consisted of two distinct activities: *development* and *maintenance.* The same author states that any changes to the software after it was developed were handled in the post-delivery maintenance phase. With classical process models, software can only be developed once changes have been finalised (Schach 2007). This argument is supported by Sohaib and Khan (2010), who report that traditional software models advocate that the next phase should not start until the last phase is complete.

The *development-then-maintenance* model is unrealistic in today's software environment (Schach 2007). With traditional development model's software is developed from scratch. In contrast, developers in today's software development environment try to reuse components of existing software products in the software that is being developed (Schach 2007).

Expanding on the argument from Schach (2007), evidence from the literature also suggests that there are problems with requirements elicitation in traditional

models. Authors such as Spence (2005) suggest that there are specific problems related to plan driven, document-centric traditional models. For example, it is difficult to define and document user requirements prior to implementation and to accurately estimate project timelines in advance of product delivery (Spence 2005). The difficulty with defining and documenting user requirements upfront is due to client's requirements changing all the time (Spence 2005; Schach 2007). Schach (2007) adds that it is not unusual for client's requirements to change; changes to requirements results in changes to specifications documents and to the design as well. Adding to this argument, Edeki (2015) opines that software development is notorious for poor upfront planning in terms of the amount of time required for completing the provided requirements. The same author reports that development teams should break down large, overarching requirements into smaller tasks in order to estimate them individually. This improves the team's ability to estimate completion time accurately (Edeki 2015). Another problem highlighted by Schach (2007) is that the customer is only involved during the requirements gathering phase and only sees the finished product at the end.

The issues highlighted above with regards to requirements elicitation are well known by many authors and have been explained eloquently by various sources (Spence 2005; Schach 2007; Kollmann 2008; Sohaib and Khan 2010; Olsson, Alahyari and Bosch 2012; Edeki 2015). Olsson, Alahyari and Bosch (2012) make a very strong statement that complex and ever-changing customer requirements, pressures of shorter time-to-market and fast changing and unpredictable markets require a development model that addresses this increasingly "turbulent business environment". "Turbulent business environment" has been a popular catch phrase prior to the advent of Agile which was used to describe the fast changing and unpredictable process of requirements elicitation (Boehm 2002; Olsson, Alahyari and Bosch 2012; Abrahamsson *et al.* 2017). It was felt that business required a new software approach that could address and accommodate this turbulence.

Almost two decades ago, Boehm (2002) and Highsmith (2002) predicted that software development using Agile practices would be the future dominant approach and would address the *turbulent business environment*. Boehm (2002) clearly points out that Agile intends to discourage upfront plans and

documentation. Moreover, he stresses that it will be more expensive to change requirements when they are required later in the software development due to developing software based on big upfront designs. Boehm (2002) advocates breaking down requirements into smaller tasks in order to facilitate change. The turbulent business environment or "*the change-driven economy*" as the agilists refer to it, suggests that changes are can be expected very late phase in the project (Highsmith 2002). Adding to this argument, authors such as Beck *et al.* (2001), Charvat (2003), Ambler (2008) and Fox, Sillito and Maurer (2008) felt that the classical process models were not suited to the evolving software development environment. According to Larman and Basili (2003), many view Agile as a modern practice. However, its concept dates back as far as the 1950's when iterative and incremental development (IID) became popular in engineering. 'Agile process models' encourage incremental development of software. The emphasis on flexibility, speed, efficiency and cost has led to a paradigm shift on how software is developed.

Spence (2005) reports that although the Agile approach to software development is not a silver bullet, it has constructive and measurable effects. The same author strongly evokes that by applying Agile principles and values while respecting the constraints and cultures of the business will improve the ability and reliability to deliver high value software. The popularity of Agile is evidenced in the literature and there has been a surge in Agile software development following the publication of the Agile manifesto. Results from a 2005 survey in the US and Europe indicated a 14% adoption of Agile methods (Dahl 2012). Moreover, 49% of responders indicated that they were aware of Agile and were interested in adopting it. Fast forward to 2009, a US survey indicated a 35% adoption rate. ScrumAlliance (2009) also confirmed the exponential growth in a report that indicated that there were over 50,000 certified Scrum Masters in the world.

The literature frequently portrays that there are many benefits to using the Agile approach to develop software. For example, Agile software development addresses the problems with requirements elicitation in traditional models by promoting flexibility, efficiency and the ability for software development companies to accommodate fast changing customer requirements (Olsson, Alahyari and

Bosch 2012). Sohaib and Khan (2010) report that one of the primary causes of software project failures is the lengthy period that it takes to develop systems using the classical process models. The same authors report that Agile methods ensure that systems are developed quickly with limited time spent on analysis and design. Tasks are broken down into increments and increments are built in different iterations; working in increments minimizes the risk of not delivering to stakeholders and also makes the development process quick to adapt to change (Jönsson 2013).

It is the view of this study that the Agile software development method has been well publicised and promoted since its inception to improve the quality of the software product (Sohaib and Khan 2010; Dahl 2012; Olsson, Alahyari and Bosch 2012). A myriad of journals, papers, books and websites disseminate the implementation, practices and technicalities of Agile software developments projects.

In order to understand how UX processes can be integrated into Agile processes, it is important to firstly understand the background and goals of Agile. The next sub-section proffers the common traits of Agile and the underlying philosophies of this methodology. This is followed by an overview of UX which lays the foundation for a better understanding of the interplay between UX and Agile software development.

2.1.2. A Definition of Agile Software Development Approaches

The founders of the Agile Alliance met to bring together various methodologies that they were using. There are many flavours of the Agile methodologies available, each offering a different approach and purpose. However, they follow the underlying philosophy of the four key values and twelve principles of Agile as outlined in the Agile manifesto (refer to section 8.1. in the Appendix for the Agile manifesto). Abrahamsson *et al.* (2017) provide an apt definition of what makes a development method an Agile one:

"[…] software development is

- incremental (small software releases, with rapid cycles)

- cooperative (customer and developers working constantly together with close communication)
- straightforward (the method itself is easy to learn and to modify, well documented), and adaptive (able to make last moment changes)." (Abrahamsson *et al.* 2017: 19)

While it is beyond the scope of this study to showcase all the Agile methodologies that are available, a few of the commonly used methodologies that have an impact on UX related integration in Agile merits discussion for completeness. Some of the commonly known methodologies are *Dynamic Systems Development Method (DSDM)*, *Scrum*, *Feature Driven Development (FDD), Adaptive Software Development (ASD), eXtreme Programming (XP), Lean Software Development and Kanban.* The similarities in the Agile methodologies are explained briefly in the table below.

Table 2: Commonly known Agile Methodologies

Methodology	**Features**
eXtreme Programming (XP)	According to Beck (2000), XP advocates frequent releases to cater for change in requirements from customers. XP also advocates a technique called "pair programming" where two developers sit behind the same screen and code together. This aims to enhance the quality of the code. Another practice of XP is having an on-site customer. However, it is not clearly defined if the on-site customer is the "user".
Adaptive Software Development (ASD)	Highsmith (2002) outlines ASD as a being designed for projects in which high-speed, change and uncertainly reign. ASD replaces the traditional waterfall cycle with a series of *speculate, collaborate and learn cycles* (i.e. adaptive life cycle), which is based on incremental/iterative development.

Methodology	Features
Feature Driven Development (FDD)	FDD focuses on short iterations that deliver tangible functionalities. Khramtchenko (2004) states that FDD consists of two main stages: (1) Discovering list of features and (2) Develop list of features. The discovery phase is critical as the quality of this step defines how precise the project will be tracked and requires full-time participation from the customer. As with the XP methodology, it is not clear as to what qualifies a "user" and what differentiates a user from a customer. The outcome of the discovery step is UML diagrams which is used to develop features (Khramtchenko 2004: 2)
Lean Software Development	Lean Software Development is more a set of project management practices than a process. According to Verdiesen (2014), Lean suggests that the pursuit of perfection can be achieved by eliminating waste. Lean has core values (similar to the Agile manifesto) that determine whether the development process is Lean or not.
Dynamic Systems Development Method (DSDM)	DSDM is more a framework than a methodology. It is a formalization of Rapid Application Development (RAD). The DSDM process places strong emphasis on project management and solution delivery and embraces Agile principles by using an iterative and incremental approach (Highsmith 2002).
Kanban	Kanban adopts Lean ideology by managing and limiting the work in progress. The Kanban board "signal cards" are used to visualize the workflow (Kuusinen and Väänänen-Vainio-Mattila 2012). Kanban systems are "pull" systems as they limit the work based on capacity thus ensuring that the system cannot be overloaded (Dahl 2012).

Methodology	Features
Scrum	Scrum is a framework based on the best practices of Agile software development which organizes project work into iterations called sprints. Requirements are organised into artifact called a *product backlog.* The product owner creates and updates the product backlog. Requirements are documented into user stories. A typical project team consists of roles such as the product owner, scrum master and developers (Schwaber 1997).

Scrum is chosen as the focus Agile methodology for integration of UX as it is the most commonly used methodology (Highsmith 2002; Kuusinen and Väänänen-Vainio-Mattila 2012; Verdiesen 2014). However, the similarities demonstrated by Scrum indicate that the integration of UX is possible for the other Agile methodologies as well. It is evident in the literature that Agile principles are honed from more mature frameworks such as Lean. The link between Lean principles and Agile principles is highlighted in the next sub-section. For example, the literature frequently points to improved communication using product backlogs and Kanban boards (Abrahamsson *et al.* 2017). Communication is a principle of Agile, which is derived from Lean (Poppendieck and Poppendieck 2003); however, all Agile frameworks promote communication which assists in defining clearer requirements from the client.

Below is a timeline of the various Agile methodologies:

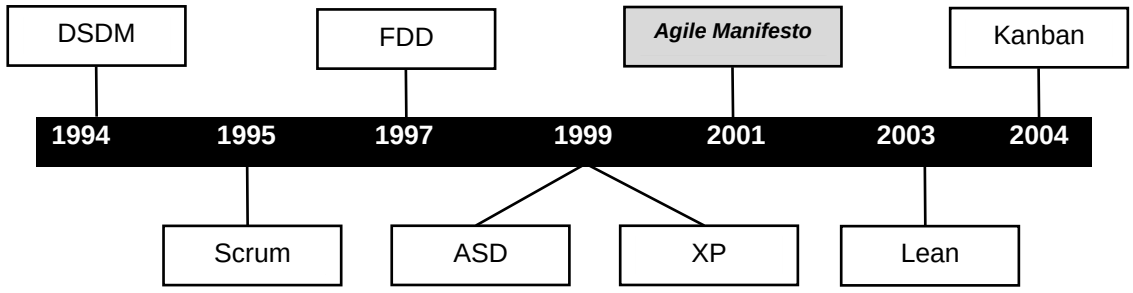


Figure 2: Timeline of Agile Methodologies
Source: Verdiesen (2014: 16)

Boehm and Turner (2003) have juxtaposed each of the Agile methods in terms of their levels of concern, sources of constraint and life cycle activities. The authors

rank the methods based on the number and strength of the constraints that they place on the implementer. A similar approach is used by Abrahamsson *et al.* (2017), who have compared each of the discussed Agile methods and given them an "Agile rank" which places them on a scale that dictates whether they are "more prescriptive" or "more adaptive". This comparison is depicted in the figure below, which shows some of the most commonly used methods on scale of their Agility.

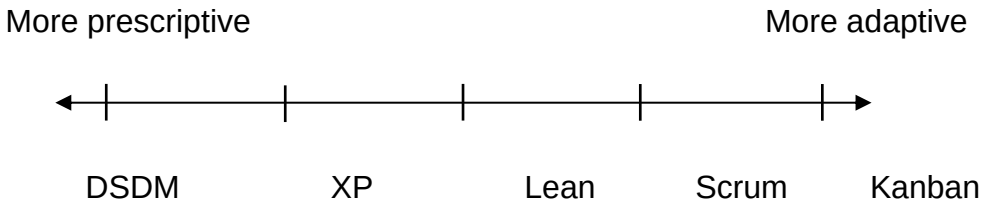


Figure 3: Comparison of Agile Methods - Prescriptive vs Adaptive
Source: Adapted from Abrahamsson et al. (2017); Boehm and Turner (2003)

Abrahamsson *et al.* (2017) and Boehm and Turner (2003) have emphasised that "less prescriptive" does not mean that the method is less difficult to implement. It does, however, mean careful consideration in terms of added contextual variables such as the project size, type etc. The same authors indicate that prescriptive methods focus more on practices (XP, Lean), while adaptive methods focus on managing the software project (Scrum, Kanban). At the same time, Agile teams need to have the ability to be adaptive for the right reasons and with minimal guidance. Agile methods have been designed to accommodate "change, speed and uncertainty" (Chamberlain, Sharp and Maiden 2006) that trouble projects that follow a plan-driven approach. As articulated by the Agile Manifesto (2001), the emphasis in Agile methods is on people, working software and responding to change.

Agile methods are often referred to as iterative and incremental models (Larman and Basili 2003). The concept of iterative and incremental development (IID) dates back as far as the 1950's when it became popular in engineering. The primary characteristics of IID is that the product is developed *iteratively* – multiple iterations to improve the quality of the product – and *incrementally*, or piece by piece (Larman and Basili 2003). IID seeks to avoid "traditional" or "plan-driven" approaches. However, many authors have argued that IID represents a different

strategy to that of software development (Schwaber 1997; Boehm and Turner 2003; Dahl 2012), as it is based on engineering disciplines rather than software projects that Agile is based on (Boehm and Turner 2003).

The IID approach makes sense as the roots of Agile can be traced to “Lean” production and manufacturing. As stated by (Dahl 2012), Lean software development owes its popularity to Toyota’s product manufacturing process in the 1980s. The main goal of “Lean Development”, as adopted by Toyota, is to maximize customer **value** and minimize **waste** – the latter can refer to anything – activities, tools, roles, processes etc. (Poppendieck and Poppendieck 2003). Toyota’s aim was to drive the manufacturing process closer to the production process by introducing principles such as ‘reduce wastage along the production line’, ‘provide products that only add value’ and ‘continuous improvement of the production process’ (Poppendieck and Poppendieck 2003). Authors such as Liker and Morgan (2006) outline and illustrate that the management principles of Toyota’s Production System (TPS) can be applied beyond manufacturing and the lean movement has gone beyond the shop floor to white-collar offices. The same authors advocate that the theory behind TPS is represented as a house – the stability depends on a strong foundation and the parts work together to create the whole structure. The simplest version of TPS is shown in the figure below:

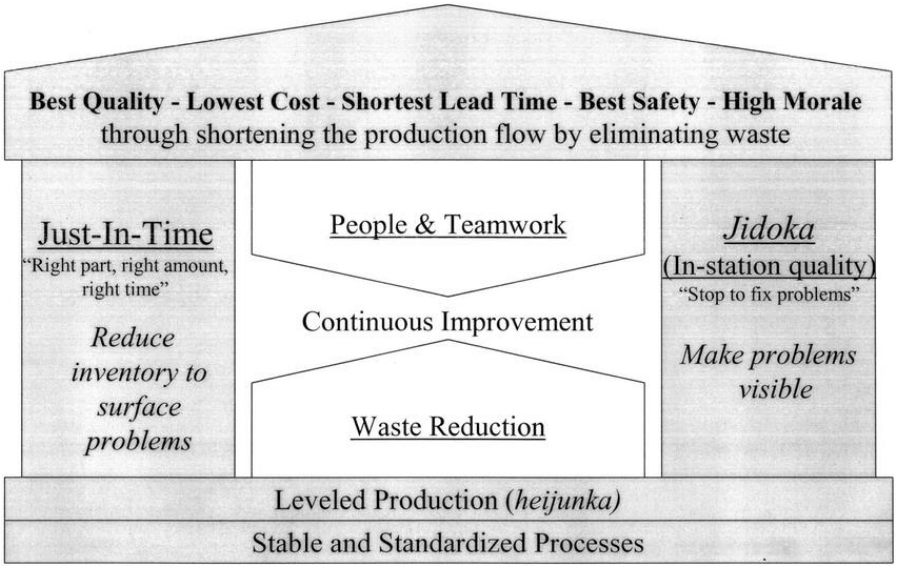


Figure 4: Toyota's Lean Production System
Source: Liker and Morgan (2006: 7)

It is beyond the scope of this thesis to provide a detailed deliberation of how Lean has impacted and influenced Agile values and principles. Instead, the key concepts that have an impact on UX integration into Agile development, merits attention and will be discussed next. Scrum ranks as the most popular permutation of Agile and the literature frequently represents a rich source of contributions for Scrum in areas of UX integration (Cockburn 2002; Singh 2008; Budwig, Jeong and Kelkar 2009; Fox 2010; Akif and Majeed 2012; Olsson, Alahyari and Bosch 2012; Cockton, Lárusdóttir, Gregory and Cajander 2016). The next sub-section synthesizes the link between Lean and Agile. Inherently this lays the foundation for a better understanding of the Scrum methodology, which is discussed in the sub-section that follows.

2.1.2.1. Lean and Agile

The Agile Manifesto (Agile Manifesto 2001) was created to spread the philosophies of Agile software development in order to guide software practitioners by providing principles and values which are present in most of the modern Agile models. Whilst investigating the literature, it is evident that the roots of Agile principles and values can be traced to “Lean” production and manufacturing (Poppendieck and Poppendieck 2003; Dahl 2012).

First and foremost, the concept of Lean seeks to adopt the principles articulated by the Agile manifesto with emphasis on the customer. Agile value 4 makes mention of “***Customer*** *collaboration over contract negotiation*” and principle 1 states “*our highest priority is to satisfy the* ***customer*** *through early and continuous delivery of valuable software*” (refer to section *“8.1. The Agile Manifesto”* in the appendix for the full list of the 12 principles). This perspective regarding the customer may be interpreted as conflicting when compared to UX design principles and will be discussed further in sub-section 2.3.1.

Secondly, the balance between *time* and *productivity* has been promoted in Lean, whereby the team has to “do more with less”. Ballé and Ballé (2005) report that the Toyota Production System was able to complete development in half the time, with four times the productivity when compared to the US equivalent. Dahl (2012) compares the *time* and *productivity* factors to the eighth Agile principle in the Agile

Manifesto, “Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.” The short timescales and early delivery advocated in Agile combined with *time* and *productivity* factors of Lean may result in UX related activities being overlooked (Dahl 2012).

Thirdly, the concept of roles is important in Agile in order to adopt the values and principles. For example, the role of *Product Owner* can be linked to Toyota’s “heavy weight project manager” (Ballé and Ballé 2005). In Toyota, responsibilities and the skills required for this role was clearly defined and understood – amongst them, the skills to “*advocate the product concept” and “possess market imagination, or the ability to forecast future customer expectations based on ambiguous and equivocal clues in the present market*” (Ballé and Ballé 2005: 19). In Lean (and Scrum), this responsibility lies with the Product Owner. However, the responsibilities, skills and competencies necessary to advocate the concept and to communicate it to the team is not clearly defined or emphasised in the literature. Dahl (2012) reports that this can be identified as one of the root causes of why UX practitioners often play a facilitator role instead of working closely in the team.

Having provided the foundation in terms of the link between Lean and Agile, the next sub-section proffers a definition of the Scrum Methodology; synthesizes the Scrum lifecycle and looks at the key roles.

2.1.2.2. Overview of the Scrum Methodology

In this section, a very brief and summarised form of Scrum is presented. The central aspects of Scrum, mainly those that are relevant in understanding the role of Agile to answer the research question “*How can good UX development practices be integrated in Agile within the South African context?”* are presented.
The presentation of Scrum that follows is by no means meant to be exhaustive. Instead, the aim is to provide enough evidence from the literature to allow for better understanding of this methodology and to have a better understanding of the chapters that follow.

The Scrum methodology was named after the Scrum in rugby: “A Scrum occurs when players from each team clump together in an attempt to advance down the playing field.” (Highsmith 2002: 50) This methodology was made popular by founders Schwaber, Sutherland and Beedle and is described as a project and requirements management methodology (Schwaber 1997). As explained by co-founder Schwaber (1997), it is difficult to predicted or plan what will be delivered when, and what the quality or cost will be. Scrum assumes that the development process is unpredictable, and it can only be described as an overall progression. Scrum therefore defines system development as a loose set of activities that combines tools and techniques with the best that a development team can contrive to build systems. The three essential roles of a Scrum team are outlined as follows by Schwaber (1997):

- Product Owner - represents the key stakeholders
- Development team - creates the end-product/s
- Scrum Master - remove impediments for the team to work effectively as possible

- Scrum Life Cycle:

The Scrum Methodology is made up of 2-4 week sprints whereby a working increment of software is delivered. Schwaber (1997: 14) states that a sprint comprises of a set of activities which are created over a period of time, the speed and intensity of which are driven by factors such as the complexity of the product being developed, the risk assessment and the degree of oversight desired.
The three main phases of sprints are outlined as follows:

- Pre-sprint Planning (Pregame):

This phase comprises of planning and system design activities. All work to be completed is stored in the “*product backlog*”. Functional, non-functional and technical requirements are all included in the product backlog. The Product Owner is the person that creates the product backlog, as well as prioritising its items. During the pre-sprint planning phase, all features and functionalities are selected from the product backlog and put into the “*sprint backlog*” which is done during the “*sprint planning meeting”* held by the product owner, the development

team and relevant stakeholders. The sprint backlog therefore has the collection of tasks that are of the highest priority to be completed in the next sprint, and it also has a specific and measurable goal for that sprint (Highsmith 2002).

- Sprint (Game):

At the end of the pre-sprint planning, the sprint backlog is handed over to the team for development of the new release functionality. The Scrum Master is the person that ensures that the team can focus on the current sprint by not allowing changes to the sprint backlog (Highsmith 2002). Short daily stand-up meetings are critical to the success of Scrum, as they enhance communication and inform all stakeholders on the status of the project, and identify issues encountered. There are multiple, iterative sprints that are used in the project to achieve the end-product (Schwaber 1997).

- Post-sprint Planning (Postgame):

At the end of every sprint, a post-sprint meeting or sprint review is held to analyse progress. According to Highsmith (2002), reflection and improvement is an imperative part of Scrum whereby the Scrum team can reflect on successes of the previous sprint, as well as what could be improved upon.

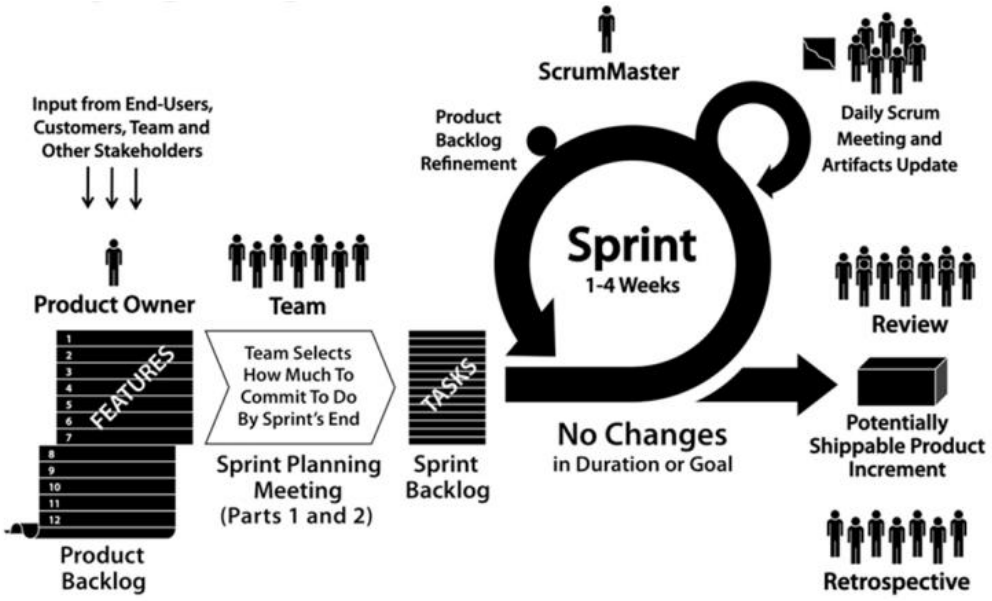


Figure 5: Scrum Methodology
Source: Lin, Hilaire, Gaud and Koukam (2011: 198)

Having presented the core features of Scrum, the next subsection will present literature on the effectiveness of Scrum.

2.1.2.3. Effectiveness of Scrum Methodology in software development

According to a recent survey in 2018 involving 1,319 from a broad range of industries in the global software community, it was concluded that Scrum and Scrum/XP (64%) continues to be the most common Agile methodology used by respondents' organizations (Version One 2019). It is interesting to note that the study also shows that 27% of companies indicated that they have adopted Scrum for 5+ years which indicates an impetus towards the maturity of this field. There are quite a few reports that highlight that correct implementation of Scrum could result in the realization of it being an effective methodology, bringing in the product closer to the customers' requirements, result in shorter delivery times, reduced costs and better management of the development process (Beck 2000; Boehm 2002; Cockburn 2002; Ambler 2008; Singh 2008; Version One 2019). This growing popularity and maturity of Scrum warrants further consideration and research in this study.

However, substantial research suggests that there are several issues and challenges with Scrum (Schwaber 1997; Boehm and Turner 2003; Cho 2008; Akif and Majeed 2012). Akif and Majeed (2012) opine the limitations of Scrum in respect of the nature of the project and work and have highlighted processes such as management, training, access to resources and organization size as some of the pertinent issues. These issues also resonate with authors such as Cho (2008) and Schwaber (1997) who have highlighted that Scrum (and Agile in general) significantly reduces the amount of documentation which results in issues for developers as they have to place more comments in the code. The reduction in documentation also results in software maintenance issues when there are changes to the team (Cho 2008).

Boehm and Turner (2003) discuss a wide array of limitations such as building reusable artifacts, working in larger teams, building safety critical software and distributed environments. The literature also suggests that ineffective communication is the root of failure in software development, and Cho (2008)

makes a strong statement that the lack of communication between teams could cause problems such as duplicated work. The author goes on to highlight that communication with the customer can also be problematic. A possible reason for limited feedback from the customer is due to them being busy with their daily jobs. This issue relates to the customer involvement issue in the lifecycle of the project and is discussed in sub-section 2.3.1. A rather strong supporting claim by Akif and Majeed (2012) suggests that some of the Scrum ceremonies are inefficient, as a lot of time goes into planning and review. The authors go on to stress that the Scrum ceremonies should be adjusted based on the complexity of the project.

The literature also claims that there is a research gap in terms of integrating user experience in Scrum (or other Agile methods). Some authors contend that there have been a few attempts to interpret how user experience is being integrated into Scrum; examples include Kuusinen, Mikkonen and Pakarinen (2012), Ferreira, Sharp and Robinson (2012), Jönsson (2013) and Kashfi, Nilsson and Feldt (2017). It has been recognised by these authors that there is a shortfall in the rigor of integrating UX in Scrum and there is a need for more clarity in this area. Lárusdóttir, Cajander and Gulliksen (2012) report that many people associate Scrum with user experience. However, there is no evidence in the process that stipulates that user experience is taken into consideration. Attempts to address the deficiencies in this area has resulted in professional bodies such as Nielsen Norman Group (2019) and Scaled Agile Academy (2019) aiming to bridge the gap by offering training for Agile UX adoption.

In summary, the literature portrays that Scrum is a relevant and widely used software development method. Scrum has also proven to be the most widely used Agile method, as confirmed by a myriad of authors and by a recent survey conducted by Version One (2019). The growing interest in Scrum shown by both software development practitioners and academics warrants the necessity for research and theoretical contribution in this area. It is also worth noting that the term “Agile” was chosen for this thesis as it allows for a wider scope than that of “Scrum”. Hence, it is the view of the author to use an all-encompassing term such as “Agile” to conduct a broader research on the topic of Agile UX. The similarities, difficulties and best practices demonstrated in the various permutations of Agile

promote the concept that recommendations and best practices are not limited to the Scrum methodology and can be used in other Agile methods integration (Cockburn 2002; Singh 2008; Budwig, Jeong and Kelkar 2009; Fox 2010; Akif and Majeed 2012; Olsson, Alahyari and Bosch 2012; Cockton *et al.* 2016). The literature is rich with well cited papers confirming the maturity of Scrum over the past decade. Arguments by authors such as Oprins, Frijns and Stettina (2019) describe the transformation and maturity of roles, teams and practices of Scrum in their seminal study conducted in eleven well established business divisions that have applied Scrum over a period of up to the three years.

In the next section, User Experience will be discussed. Here the study focuses on what User Experience is, and what some of the methods and techniques are.

2.2. Good development practices for creating good User Experience (UX)

This section starts with the definition of User Experience (UX), followed by why it has become more and more important to focus on UX in human-computer interaction. UX terminology is discussed, followed by the 6 key principles to ensure that designs are user-centred, as defined by ISO (2009).
The findings will guide and possibly assist in identifying good UX development practices in relation to research sub-question 2 of this study: *"What are good development practices for creating good UX?"*

2.2.1. A Brief History of UX

The term 'user experience' was coined by cognitive psychologist and designer Don Norman at Apple Computer Incorporated in the 1990's.

"Apple Computer is known for its innovation in the field of human interface. This is a result of considerable research effort and attention to detail in the execution of products. In this organizational overview, we cover some of the critical aspects of human interface research and application at Apple or, as we prefer to call it, the **user experience**" (Norman, Miller and Henderson 1995: 155).

Although the term *'user experience'* was formally introduced in the 90's, the history of UX predates its name by quite a few decades (Stevens 2018). According to Stevens (2018), evidence suggest that the origins of UX can be traced back to the 5th century whereby the Greek civilisation designed tools and their workplace based on ergonomics or human factors. Fast forward a thousand years and here are some of the major breakthroughs that have been made in UX, as outlined by Stevens (2018):

- Frederick Winslow Taylor wrote the "Principles of Scientific Management" in 1911 which focused on optimizing the relationship between humans and their tools.
- In the 1940's, car manufacturer Toyota continued the quest for UX efficiencies by introducing the human-centred production system. This was a milestone in UX history as it highlighted the importance of how humans interact with machines.
- In 1955, Henry Dreyfuss, an American industrial engineer wrote "Designing for People" which explained UX practices on how to succeed as a designer by focusing on making the user feel more safe, comfortable and eager to purchase.
- In 1995, Donald Norman, who was a User Experience Architect at Apple, invented the term *'user experience'*. Norman was the first person to have UX in his job title and he found ways of encompassing all aspects of UX such as industrial design, graphics, the interface and the physical/manual interaction with the system.

The timeline of major breakthroughs in UX is shown below:

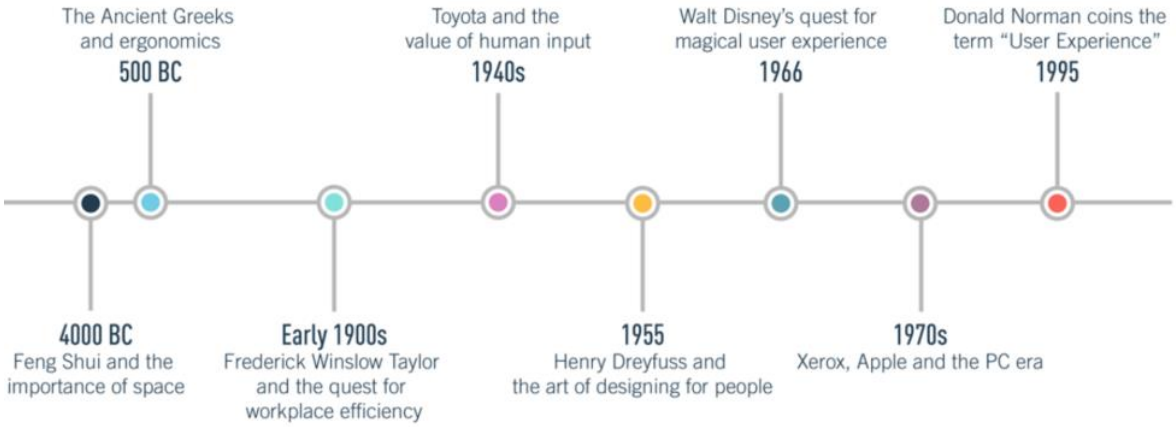


Figure 6: Timeline of major breakthroughs in UX
Source: Stevens (2018)

The timeline above indicates that UX design is constantly evolving and the importance has been emphasized in various studies over the past decade. The success or failure of any software application depends on whether the user finds it useful, and whether it meets the users' expectations. The next sub-section looks at the definition of UX which lays the foundation for the interplay between UX and Agile.

2.2.2. A Definition of UX

Previous studies suggest that many experts have attempted to define "User Experience", for instance Gulliksen, Göransson, Boivie, Blomkvist, Persson and Cajander (2003) states that there is no agreed upon definition, consequently there is a great variety in the ways in which it is applied, which may lead to poor quality and usability of the system. Hassenzahl and Tractinsky (2006) define User Experience as a result of the user's internal state (for example the expectations, motivations and needs), the features of the designed system (for example the usability, functionality, complexity) and the environment where the interaction occurs (for example organizational or social setting).

Lodato (2015) explains that it is important to note that User Experience is two words. It combines two key concepts in the design and development of software. The first concept is the ***user***, and this relates to "the current, proposed, eventual,

or speculative person who interacts with a product, service, or system" Lodato (2015: 6).

The second concept is ***experience***, and this relates to "the current, proposed, eventual, or speculative character of using a product, service, or system" Lodato (2015: 8).

These two terms collectively define User Experience and according to the International Organization for Standards (ISO), User Experience is defined as "A person's perceptions and responses that result from the use or anticipated use of a product, system or service." (Law, Roto, Hassenzahl, Vermeeren and Kort 2009: 727). Verdiesen (2014) dissects this definition by ISO as follows:

- "Perceptions and responses" covers everything from smell to touch to sentiment hence the confusion in the definition.
- "Anticipated use" creates the impression that UX plays an essential role in the early development stages hence studying UX before the working product is presented can provide valuable input.

According to Verdiesen (2014), technological and social advancements in computing or systems have created an imperative for human-computer interaction to be a major part of our lives. The sentiment and motivation of the user when using the system are given a more central role. However, he notes that the term User Experience is ambiguous. This is outlined in previous studies conducted by Law *et al.* (2009) to define User Experience amongst UX professionals which have proven to be inconclusive. Surveys were conducted on the views on UX by 275 researchers and practitioners. The results of the surveys indicated that the responders understood the concept of User Experience very differently and there were no patterns identified. Hence, implications for future work. Some of the key differences identified from the surveys were that the researchers did not want to restrict UX to interaction with a product/artifact and showed the need to define the relation between product, brand or service experience. The recommendation made by the study indicates that the term User Experience be scoped in products, systems, services, and objects that a person interacts with through a user interface, as per figure below (Law *et al.* 2009).

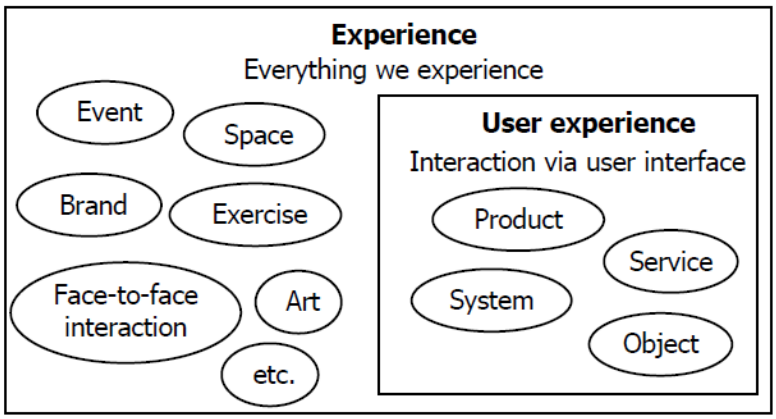


Figure 7: Experience versus User Experience
Source: Law et al. (2009: 727)

Nadikattu (2016) argues that User Experience, Usability and User Centred Designs are pieces of the same puzzle. It can be broadly defined as follows:

Usability
The degree of ease with which a user can interact with a system, meeting the user's needs while maintaining a high level of user satisfaction.

User Experience (UX)
Net total of all the user's feelings and perceptions resulting from interacting with the system.

User Centred Designs
Set of procedures and processes that is built around user needs and expectations. Includes activities such as evaluating user requirements, building prototypes, exploring design alternatives, analysing tasks and functions and testing usability.

Figure 8: User Experience, Usability and User Centred Designs
Source: Adapted from Nadikattu (2016: 14)

Although the term UX seems to be on the increase, the literature also seems to pay attention to User Centred Designs (UCD) within the field of Agile UX. Kuusinen and Väänänen-Vainio-Mattila (2012) considers that Agile UX development involves a combination of aspects from software engineering, human-computer interaction and management. The term “UX” was chosen for this study as it allows for a wider scope than that of UCD.

The next section outlines the need for User Experience and why designs should be user-centred. It also describes the link between UX principles and Agile principles.

2.2.3. Addressing the interplay between UX and Software Development

Designing for better UX is much more difficult than clearer usability goals. The software development community was once concerned more with usability. However, it has since become more interested in understanding, designing and evaluating a wider range of UX needs. UX professionals aim to make the product, system or service more user-centred. According to Verdiesen (2014) during the development of the end-product, the user needs to always be the focus point and drive the development process. The International Standards Organization (ISO) lists 6 key UX principles which ensure that the design is user-centred (ISO 2009):

1. The design is based upon an explicit understanding of the users, tasks and environments.
2. Users are involved throughout the design and development.
3. The design is driven and refined by user-centred evaluation.
4. The process is iterative.
5. The design addresses the whole user experience.
6. The design team includes multidisciplinary skills and perspectives.

Law and van Schaik (2012) acknowledge that the field of HCI (in which UX is rooted) has inherited a diversity of disciplines (e.g. theoretical concepts, epistemological assumptions, values and methodologies). These disciplines range from engineering where measures are strongly embraced, to humanities where measures can be easily seen as being over-simplistic especially when the concepts are not clearly defined, which leaves too much for interpretation (Law and van Schaik 2012). According to Bevan (2009), there are many different interpretations of UX. However, ISO suggests that the measures of UX are similar to that of satisfaction in usability. Bevan argues that ISO does not explicitly define the user, learnability (i.e. usability changing over time) or the way UX evolves from expectation, actual interaction to a total experience (including reflection on the experience).

A supporting claim is shown in a recent survey conducted across 429 UX professionals on *UX research in practice* by the Nielsen Norman Group (2019), which reports that 71% of teams perform some sort of quantitative UX research and almost everyone reported challenges to get quantitative research done. The survey results also exhibit mixed reactions to which UX research methodologies they used. Predictably, the relatively lower cost methodologies were used more often (example interviews, qualitative usability testing and analytics), whilst more expensive methodologies ranked lower. The survey goes on to highlight the lack of success evaluation (18% of respondents admitted that they do not know if the design changes resulted in improvements). Other challenges reported related to cost, lack of knowledge, lack of team coherence, difficulty interpreting and reporting on research findings (Nielsen Norman Group 2019). These challenges were also confirmed by authors such as Ardito, Buono, Costabile and Lanzilotti (2012) and Law, Abrahão, Vermeeren and Hvannberg (2012).

Elaborating further, Law *et al.* (2012) argue that UX metrics are yet to be defined. These authors' contest that both usability and UX measures should be utilised to benchmark competitive design artifacts in order to select the correct design options. The literature also hints that although the gap between human-computer experience and software engineering has been narrowed, there is a gap due to the emergence and relevance of UX (Law *et al.* 2012), for example, efforts have been made by the software engineering community to recognise the relevance of usability. Can such recognition be taken for granted for UX evaluation methodologies and measures? Lindell (2012) opines that UX and software engineering have two different epistemologies; UX is a design practice, while software engineering is somewhere between engineering and science. During a series of workshops and surveys on the interplay between UX evaluation and software development, Ardito *et al.* (2012) argue that methods addressing HCI and UX are always mentioned in research papers but seldom applied in practice. Some of the findings indicate that UX approaches are applied only to a limited extent by practitioners. These authors call for follow up studies to be conducted to address activities such as *defining business impact of UX*, *understanding of the system and users' world*, *defining UX targets for design, transforming UX requirements into design solutions* and *defining UX metrics* as perceived by a

multitude of organisations (Ardito *et al.* 2012; Law and van Schaik 2012; Lindell 2012; Nielsen Norman Group 2019).

2.2.4. An overview of UX Dimensions in relation to Agile

The development of software has been described as an activity with a wide range of intrinsic properties. Several authors such as Da Silva *et al.* (2011), Budwig, Jeong and Kelkar (2009), Humayoun, Catarci, Kimani and Dubinsky (2017) and Kashfi, Nilsson and Feldt (2017) have provided comprehensive overviews of UX in relation to Agile. Brhel, Meth, Maedche and Werder (2015) report that UX is not considered as a central topic in software engineering, in which it is seen as a non-functional requirement and quality attribute. However, it has become crucial in recent years for economic success in highly competitive markets (Brhel *et al.* 2015). This is confirmed by authors such as Cockton *et al.* (2016) who advocate that UX can deliver on a wide range of values such as effectiveness, efficiency, brand loyalty, employee retention and competitiveness.

Work by Kuusinen, Väätäjä, Mikkonen and Väänänen (2016) suggests that UX follows the user-centred design approach and mechanisms for integration in Agile still remain largely un-established. The authors go on to elaborate that the most widely used approaches include design upfront and developing iterations with user testing. Supporting claims by Cockton *et al.* (2016) indicate that some of the key characteristics of compatibility between UX and Agile is that they both rely on evolving requirements iteratively across a series of iterations, incremental development for early delivery of software, and close collaboration with the customer. Other less favourable characteristics highlighted by Cockton *et al.* (2016) were *reduced time for user testing* and *less upfront planning*. The authors also highlight that these adverse factors can result in UX being excluded from the Agile practices or having the UX responsibilities transferred to other roles like the business analyst or developer. Kuusinen *et al.* (2016) make a strong statement that in order to successfully include UX work in an Agile environment, lightweight methods are needed.

McKendrick (2018) advocates that superior UX is both a qualitative and quantitative exercise. The same author highlights that many times in designing

and developing UX, an important component gets left out of the equation: the user. When comparing UX dimensions in relation to Agile, Spool (2018) reports that Agile is not supposed to be *UX hostile*. These methodologies are both kindred spirits that share common principles and can work together. He opines that if one has to trace back to Agile's roots, it is evident that it embodies UX design core principles. The author explains that there is a strong influence of good practice and smart design in the Agile manifesto. Spool (2018) links the four key values from the Agile manifesto to UX dimensions as follow:

Table 3: Four Key Agile Values in relation to UX

Agile Key Values	**Application in UX**
Individuals and interactions over processes and tools.	Design leaders know the way they interact with their teams that affect the designs their teams' produce. Everyone, including developers and product managers, work best when they collectively understand what the user needs and how the design can meet those needs.
Working software over comprehensive documentation.	This Manifesto principle embodies the show, don't tell ethic behind great design collaboration. Sitting together and demonstrating a working idea can push a design forward much faster than any amount of specifications or non-interactive diagrams.
Customer collaboration over contract negotiation.	The pressure to deliver on time focuses our efforts into the negotiation of what-by-when. Delivering something our customers will use and get value from can take second priority if we let those negotiations push the wrong priorities.
Responding to change over following a plan.	As we put our thinking in front of our users, we learn where we made the wrong assumptions. Solid design practice teaches us to respond to an improved understanding, keeping our practice flexible.

Source: Adapted from Spool (2018: 1)

Practitioners such as Cockton *et al.* (2016) believe that the Agile manifesto (section 8.1 in the appendix) itself does not outlaw the processes, tools, documentation, contracts or plans that appear on the *right*:
"That is, while there is value in the items on the *right*, we value the items on the *left* more." (Agile Manifesto 2001: 1)

Cockton *et al.* (2016) states that the tensions are due to the extent to which interpreters of the manifesto value items on the *left* more than the items on the *right*. Supporting arguments by Brhel *et al.* (2015) and Boehm and Turner (2003) indicate that over half of the 12 Agile principles in the Agile manifesto (refer to section *"8.1. The Agile Manifesto"* in the appendix for the full list of the 12 principles) should not block good support for UX (i.e. principle 2, 4, 5, 8, 9, 11 and 12). However, the same authors highlight that conflicts could arise from principles 1, 3, 6, 7 and 10. Principle 1, 3 and 10 could eliminate UX (with reference to 'early' 'shorter' and 'work not done' in the principles). Principle 6 also devalues documentation and places emphasis on face to face conversations (Boehm and Turner 2003; Brhel *et al.* 2015; Cockton *et al.* 2016).

Kuusinen *et al.* (2016) states that the dimensions of UX in Agile projects involve a combination of aspects from the fields of software engineering, human-computer interaction and management. A few authors state that UX work and software engineering practices mostly concentrate on organisational level aspects (Lárusdóttir, Cajander and Gulliksen 2012; Olsson, Alahyari and Bosch 2012; Schwartz 2013; Subair 2014; Nadikattu 2016). These authors discuss the interplay between software engineering and UX and the extent to which daily work of these disciplines should be connected. The same authors stress that the problem with integrating UX in Agile is multidimensional. Several aspects from an individual practitioner to the organizational culture needs to be addressed. Kuusinen *et al.* (2016) considers four dimensions that are enabling factors for evolution of performance management systems: *process, people, systems* and *culture.* The dimensions are supported by authors such as Brhel *et al.* (2015) and Chow and Cao (2008). The same authors report that all these dimensions are extensive and abstract and have to be operationalized to be measured.

Unfortunately, there are no established metrics for measuring the performance of the development work (Kuusinen *et al.* 2016).

In summary, the literature portrays that UX is constantly evolving and the importance has been emphasized in various studies over the past decade. Moreover, there is a growing demand for approaches that offer guidance on how to develop for good UX and for good usability. The growing interest in UX has also created an impetus towards research for approaches to integrate UX in Agile software development. Interest shown by software development practitioners warrants the necessity to conduct research on good practices for integrating UX in Agile.

In the next section, integrating UX in Agile will be discussed. Here the study focuses on what are some of the similarities, good practices and challenges when integrating these two methodologies.

2.3. Integrating UX in Agile

In this section, the literature review explores the discussion on how UX and Agile can be integrated. We look at the similarities, potential challenges that have been identified in the literature and different approaches of addressing these challenges.

Ostensibly, Agile and UX seems to be the perfect match. The buzzword, “Agile UX”, can be described as the assimilation of the Agile methodology to include UX design principles and methods. The goal of Agile UX is to unify UX designers and Agile developers during the product development process (Treder 2013). The advent of Agile UX has resulted in more and more UX practitioners dipping their toes into Agile waters and it has become a topic that has continued to evolve (Kuusinen, Mikkonen and Pakarinen 2012; Brown 2013). Recent studies have examined Agile and UX methods to determine if and how these two methodologies can be integrated in the development process. As discussed by Nielsen Norman Group (2019), Agile promises to address several issues that have vexed UX practitioners in traditional “waterfall” approaches. Authors such as Dahl (2012) and Kuusinen *et al.* (2016) advocate that short cycles may reduce time for requirements and implementation thus alleviating the issue of requirements

changing before or during development. Moreover, direct communication and collaboration replacing documentation also increases the chances of what is being developed versus what is documented (Kuusinen *et al.* 2016). Developing iteratively and incrementally also provides opportunity for continuous user research, testing and feedback, thus improving UX in every iteration (Dahl 2012; Kuusinen *et al.* 2016).

Building on the findings from Fox (2010), Dahl (2012), Kuusinen *et al.* (2016) and Nielsen Norman Group (2019), other authors such as Schwaber (1997), Beck (2000) and Fox (2010) hint that there are many similarities between UX and Agile.

Firstly, both rely on *iterative development*, building on information from previous cycles. In UX, one of the key principles is iterative design. This is confirmed by Fox (2010); both methodologies are iterative in nature. However, they iterate over different artifacts – Agile being the application code, and UX being the user interface design (Fox 2010).

Secondly, these two methodologies place *emphasis on the user*. User involvement during the development process is encouraged. In Scrum, the user is involved during sprint reviews to evaluate the product. However, the “Product Owner” is responsible for the requirements and the product feature prioritization. In comparison, UX advocates early involvement of the user (Schwaber 1997; Fox 2010). Which raises the question: how does the definition of the user in UX compare to Agile? One of the pertinent issues identified in the Agile methodology, is the role of the customer and the user as it is not clearly defined as to who the customer is and who the user is, and what their functions are in Agile UX. More on this topic is covered in the next sub-section.

Thirdly, team coherence is vital in Agile software development. One of the purposes of the “Pre-game” phase in Scrum is to ensure that the team is aligned and brought together (Schwaber 1997; Beck 2000). In comparison, one of the key features in UX is that the team should always keep the user in mind when developing a product.

Several literature reviews on Agile UX research have been conducted. Although most of the literature is largely positive to the integration of Agile software development and UX related activities (Kuusinen *et al.* 2016), many authors have found that studies of Agile UX 'in the wild' focuses more on what happens rather than what should happen (Cho 2008; Chow and Cao 2008; Dahl 2012; Cockton *et al.* 2016). Cockton *et al.* (2016) advocates that the disparity between theory and practice in the literature calls for extensive qualitative research expertise to bridge the gap between these two methodologies. Authors such as Sohaib and Khan (2010) confirm this argument and state that post the release of the Agile manifesto, UX has made its way into the software mainstream. The same authors report that despite UX and Agile having different perspectives on developing software, both these methodologies have a significant role in creating good software - UX places emphasis on how the end-user will work with the software, and Agile software development focuses on how the software should be developed.

A myriad of authors have expressed concerns on how these two disciplines are being combined in practice as it is still not well understood (Sohaib and Khan 2010; Akif and Majeed 2012; Dahl 2012; Kuusinen, Mikkonen and Pakarinen 2012; Lárusdóttir, Cajander and Gulliksen 2012; Olsson, Alahyari and Bosch 2012; Brhel *et al.* 2015; Cockton *et al.* 2016). This line of argument has led to the following research question:
"How can good UX development practices be integrated into Agile within the South African context?"

The next sub-section looks at the difficulties of integrating UX in Agile, as identified in the literature.

2.3.1. Known Difficulties of Integrating UX into Agile

The aim of this sub-section of the chapter is to examine the literature to discover: "What are the problems when integrating UX in Agile?" (Research sub-question 4). Thematic analysis using concept analysis formed the basis of analysis to synthesize data from the literature. Concept analysis allows for theories to be

developed by using the following phases, as defined by Vaismoradi, Turunen and Bondas (2013):

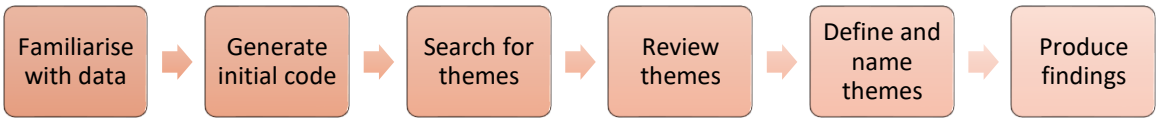


Figure 9: Phases of Concept Analysis
Source: Adapted from Vaismoradi, Turunen and Bondas (2013: 402)

The process of concept analysis was suitable for this study as it can be applied to data in the fieldwork as well. Although grounded theory was used as the research method for this study, the additional refinement provided by concept analysis assisted in generating codes and themes.

There are a few empirical studies on the integration of Agile UX or UCD work. The earliest one that was identified was from Fox, Sillito and Maurer (2008). In this empirical study, these authors highlighted that Agile methods alone cannot ensure that the user interface is usable. The *Generalist, Specialist* and *Hybrid* approaches were recommended by Fox, Sillito and Maurer (2008). The commonality between these approaches was the inclusion of a UX team member. These researchers found that teams used the parallel track approach, which is also recommended by Kollmann (2008), Budwig, Jeong and Kelkar (2009) and Federoff and Courage (2009). Issues with this approach, as outlined by these authors, were that the UX team experienced that the focus on delivering functionality for every sprint led to overall goals being overlooked in development. Moreover, additional processes were required in order to handle stakeholder alignment and to keep track of UX work needed for development.

Authors such as Kollmann (2008), Federoff and Courage (2009) and Verdiesen (2014) opine that due to the lack of stakeholder alignment and keeping track of UX work, developers have to wait on designs. It is difficult to plan upfront as changes are inherent in Agile methods, therefore planning upfront is discouraged. There is also a lack of time for planning design activities, end-user workflows, holistic coherent designs and user research as Agile methods focus on delivery after each sprint. The *lack of time in sprints* also results in developers waiting on designs.

The lack of time also results in designs being completed in a hurry which affects the quality of the design (Kollmann 2008; Federoff and Courage 2009; Verdiesen 2014). Supporting claims by Salah, Paige and Cairns (2015) indicates that "feature by feature" development in Agile results in a UI that is disjointed and lacks holistic and overall structure and vision.

Prioritization of UX or usability activities into sprints was evident in a few other studies as the developers focus more on developing functional requirements rather than non-functional UX requirements (Miller 2005; Sy 2007). Singh (2008) makes a rather strong statement that UX tasks are often included in the backlog but usually do not get prioritized to be added to the current sprint.

The Scrum development process has been heavily criticized for not involving real users and for not addressing their usability needs (Schwaber 1997; Lárusdóttir, Cajander and Gulliksen 2012). As noted by several authors, there were no UX representatives when the Agile Alliance was formed (Kollmann 2008; Dahl 2012; Nielsen Norman Group 2019). The signees of the Agile manifesto were all software engineers.

There is ample literature from Sohaib and Khan (2010) that indicates that one of the main conclusions from an extensive literature survey is that the end-user is not sufficiently included in the Agile development process. Lárusdóttir, Cajander and Gulliksen (2012) explain that short sprints and emphasis on completing the development for each sprint results in Agile teams not having enough time for involving users or for conducting UX evaluations.

Moreover, the literature frequently labels challenges relating to *usability testing* within Agile. The short, iterative nature of Agile poses challenges in terms of conducting tests as there is limited time to conduct evaluation and testing with end-users (Federoff and Courage 2009; Ferreira, Sharp and Robinson 2012). Ferreira et al. (2012) reports that scheduling UX testing within an Agile context is a challenge due to a lack of clarity in terms of timing of the testing and evaluation within the iterative structure of the Agile development process. Kane (2003) elaborates that if UX testing is conducted at the end of the Agile development

process, there could be a lack of time and resources to respond to issues identified during testing. If UX testing is conducted as frequently as feature acceptance testing, this could lead to massive budget increases (Kane 2003). Access to the users is also a challenge. Scheduling users' involvement in advance for testing is difficult and may not fit within the Agile schedule (Ferreira, Sharp and Robinson 2012). According to Verdiesen (2014) it is also not feasible to schedule testing in the same iteration, even though the results gained from the testing would have benefited from a *just-in-time* approach. Furthermore, there is little or no time to integrate feedback from UX testing in the next sprint cycles. Feedback is acquired but there is no assurance that it will be used. Delays in usability testing can also result in delays of reporting the feedback from testing. Subsequently features can be developed whilst testing is being conducted, which results in feedback from testing being obsolete (Miller 2005; Sy 2007).

Agile Principle 1 under the Twelve principles of Agile mentions that the role of the customer varies according to the nature of the project e.g. domain experts, product manager, start-up founder (Martin, Biddle and Noble 2004). Due to this understanding or ambiguity of the "customer", Chamberlain, Sharp and Maiden (2006) argue that collaboration and design does not include user involvement which questions whether Agile also does not focus on the user. In Scrum, the product owner is responsible for gathering requirements from all stakeholders. According to Ambler (2008), the misconception of collecting requirements from only the customer with no end-user involvement often results in delivering a product that does not meet the end-users' needs. Requirements are therefore based on assumptions about what the end-users' needs are. Beyer, Holtzblatt and Baker (2004) confirm this misconception and state that neither customers nor users can articulate what they need. It is also impossible for one person to be a representative of the user population.

According to Kuusinen, Mikkonen and Pakarinen (2012), previous studies conducted suggest that there seemed to be a lack of responsibility in terms of UX issues and activities in a wider perspective, particularly around whether the customer or the end-user is responsible for UX activities. In a case study conducted by Kuusinen, Mikkonen and Pakarinen (2012), participants that were

interviewed advised that users were not involved in UX activities because the software was not being sold to them but to the customers therefore it is the customer's concern. Some of the findings from the case study suggest that UX goals were not considered in the beginning of the project, and the customer sent a request with functional requirements without any consideration of non-functional requirements. The adequacy of usability or quality of UX is compromised if only the customer's input is taken into consideration as the customer is not fully aware of all the needs of the users: "Sometimes we have to explain usability defects to the customer, like having to click ten times to complete a simple process" Kuusinen, Mikkonen and Pakarinen (2012: 6)

One of the key values in Agile is "Working software over comprehensive documentation". According to Verdiesen (2014: 33) Agile is a people-oriented methodology. Hence, there is more interaction between team members and less need for comprehensive documentation that needs to be maintained and updated. However, documentation is required for integrating Agile and UX. Previous studies conducted by Budwig, Jeong and Kelkar (2009) indicate that the confusion with regards to UX deliverables are due to the lack of documentation. Salah, Paige and Cairns (2015) also report that a number of pertinent issues need to be documented in order for integration to work, such as:

- Design rationale – to justify and make note of previous design decisions
- Source requirements – from customers, users, developers, and usability experts etc. which will assist in creating new user stories or modifying exist ones.
- Current designs – need to document current designs, expected delivery dates, test results, recommendations and fixes from test, user and task information, and UI designs.

Schön, Thomaschewski and Escalona (2017) note that in previous studies analysed on Agile Requirements Engineering, problems were faced regarding requirements documentation as it was difficult to establish the right type of artifacts to use which will ensure collaboration between developers, UX stakeholders and the Agile team therefore the need for further research.

At the start of this sub-section the following research sub-question was posed:
"What are the problems when integrating UX in Agile?"
The main problems that arise when integrating UX in Agile can be divided into the following problem areas:

- Sprints are too short
- Difficulty in prioritizing UX activities
- Difficulty in performing UX testing
- Customer versus User
- Lack of documentation

Despite the challenges identified in the literature, there is an opportunity to bring UX design closer to Agile thus enhancing the interaction between these two disciplines. The next section looks at some of the good practices for resolving these difficulties.

2.3.2. Good Practices for Resolving the Difficulties of Integrating UX into Agile

This sub-section focuses on the good practices that address some of the difficulties discussed in the previous sub-section with the following aims:

1. Identify good Agile development techniques to ensure that UX requirements are met as per research sub-question 1: "*What Agile development techniques are best suited to ensure UX requirements are met?"*
2. Identify good development practices for creating good UX as per research sub-question 2: *"What are good development practices for creating good UX?"*
3. The role of team members that are directly involved in the process of integrating UX in Agile as per research sub-question 3: *"What are the roles of team members that are directly involved in the process of integrating UX in Agile?"*
4. The above mentioned good practices will serve as an antecedent to confirm if these practices exist within the South African context for integration of UX in Agile. There is no proof in the South African context of identifying and implementing good UX development practices into Agile. A review of the literature within the South African context confirmed that there are a

significant number of publications on Agile software development from a South African perspective (Joseph and Santana 2016; Chiyangwa and Mnkandla 2017; Sebega and Mnkandla 2017; Mudarikwa and Grace 2018). There is also a growing interest in UX as evidenced by Brosens (2018) and Coleman (2018). However, these studies do not provide a higher level of abstraction on Agile UX integration practice and there is a call for well-established and well accepted integration practices that both Agile and UX practitioners can understand and apply.

As with the previous sub-section, thematic analysis using concept analysis formed the basis of analysis for synthesizing the good practices of Agile UX from the literature. The potential for combining UX and Agile is not a new idea, as previous studies suggest that there many advantages and strengths (Beck 2000; Kane 2003; Ambler 2008; Fox 2010). Schwartz (2013) also suggests that the reconciliation of UX and Agile is possible and has often been implemented even though some of the Agile concerns prevent a UX attitude (e.g. focus of programming techniques, short iterations, fast increments and automated tests). Cockton *et al.* (2016) reports that a simple yet pertinent question has been asked since the dawn of UCD: “What is good and bad in different methods?” The authors advocate a need to validate and evaluate methods reflecting software engineering and human science values.

There is ample literature that juxtaposes UX principles and Agile software development. However, none of them were identified within the South African context. The tabular comparison below endeavours to highlight these good development practices and to highlight the link between the philosophies of UX and Agile software development.

Table 4: Good practices for integrating UX in Agile

Known difficulties In the Literature	**Good practices for resolving difficulties**	**Authors**	**Research Question**
Sprints are too short	Upfront design	Salah, Paige and Cairns (2015)	1 and 2
	UX work being	Verdiesen (2014)	1 and 2

	desynchronised – UX work is one or two sprints ahead of development		
Difficulties in prioritizing UX activities	Assigning the responsibility to the UX designer/UX practitioner	(Miller 2005; Sy 2007; Budwig, Jeong and Kelkar 2009)	2
	Separate UX product backlog	(Budwig, Jeong and Kelkar 2009)	1 and 2
	UScrum	Singh (2008)	1
	Separate UX team to prioritize UX activities	(Budwig, Jeong and Kelkar 2009)	2
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	• Fox, Sillito and Maurer (2008) • Federoff and Courage (2009) • (Chamberlain, Sharp and Maiden 2006; Hussain, Slany and Holzinger 2009; Fox 2010)	2
	Scheduling of testing: UX testing done during Agile development tests.	Kane (2003)	1 and 2
	Access to users: • Planning in advance for user inclusion. • Use an existing user pool to act as development and design partner and conduct testing	• (Kollmann 2008) • (Federoff and Courage 2009)	3
	Shorter time to iterate design:	(McInerney and Maurer 2005)	1 and 2

	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	• Convertino and Frishberg (2017) • (Ambler 2008; Kieffer, Ghouti and Macq 2017) • Kollmann (2008)	3
Lack of documentation	User stories were reported as being the most prevalent. However, there are a wide range of artifacts.	(Näkki, Koskela and Pikkarainen 2011; Harbers, Detweiler and Neerincx 2015; Salah, Paige and Cairns 2015; Schön, Thomaschewski and Escalona 2017)	4

This tabular comparison presented above does not offer a definitive list of the good practices on the knowledge area. However, it serves as a foundation for the researcher and provides clues to the known difficulties and possible solutions that might be encountered during the research process.

2.3.3. Summary and implications for this study

Agile UX is an emerging research area, as can be seen by the growing number of publications in the literature. Authors such as Brhel *et al.* (2015) and Sohaib and Khan (2010) have reported that most of the research has addressed the use of usability practices in Agile development. Although there are several studies that have attempted to elucidate the Agile UX knowledge areas, further empirical studies are required in this research area (Brhel *et al.* 2015; Kuusinen *et al.* 2016). Moreover, the literature is fairly silent in terms of empirical studies conducted

within the South African context. Despite the researcher's efforts, it was found that there was a lack of academic literature on Agile UX. To accommodate for this deficiency, the researcher sourced literature from international counterparts.

Process models that have been created by the likes of Fox, Sillito and Maurer (2008), Budwig, Jeong and Kelkar (2009) and Federoff and Courage (2009) advocate parallel tracks and UX design upfront by separating development and UX activities, which leaves room for problems in development work. Moreover, these approaches decrease agility. Previous research also reports several challenges with organizing and conducting UX work in Agile software development (Kollmann 2008; Federoff and Courage 2009; Verdiesen 2014). Authors such as Salah, Paige and Cairns (2015) and Brhel *et al.* (2015) separate these challenges into process-oriented challenges and people-oriented challenges. Process-oriented challenges relate to how and where UX design work fits into Agile iterations (Salah, Paige and Cairns 2015), while people-oriented challenges relate to the work dynamics between the Agile and UX team members (Brhel *et al.* 2015). According to the latter, there has been an increase in the number studies relating to people-oriented challenges.

This thesis aims at increasing the body of knowledge and structuring a new approach of integrating UX in Agile. It aims at supporting Agile UX work and is expected to mitigate any challenges related to the integration. Moreover, the thesis aims at understanding the good practices and work dynamics between Agile and UX practitioners. More specifically, it aims to investigate the research question: *"How can good UX development practices be integrated into Agile within the South African context?"*

2.4. Conclusion

This chapter set out to review literature on integrating UX in Agile. It first started with a history of Agile software development and the key concepts, key values and principles and the reasons why this methodology is applicable to this study (section 2.1). A brief overview of commonly known Agile Methodologies was then discussed. Next, the link between Agile principles and values and "Lean" production and manufacturing were explained which provided the foundation for

the next sub-section. A definition of the Scrum Methodology was provided which briefly synthesized the Scrum lifecycle and looked at the key roles. The literature revealed that Scrum continues to be the most common Agile methodology used.

The chapter then progressed to provide a history of UX (section 2.2), where the definition of UX was uncovered, and the interplay between UX and Agile was addressed. Although previous studies provided valuable insight, there were some areas that needed to be further investigated (for example, UX metrics, defining UX targets for design, responsibilities in terms of UX issues and activities etc.). A broader discussion around the view of UX dimensions, in relation to Agile software development, was discussed next. These UX dimensions were examined and compared to Agile principles, in order to identify commonalities shared for integrating Agile methods with UX.

The chapter concluded with a discussion on how UX and Agile can be integrated (section 2.3). Agile and UX seems to be the perfect match, as there are similarities such as both relying on iterative development, both encouraging user participation in the development process and both highlighting the importance of team coherence. However, related work shows strong evidence of known difficulties with Agile UX. Some of the challenges were difficulty in prioritizing UX activities, sprints being too short, difficulty in performing UX testing and the lack of documentation. It was also found that the understanding or ambiguity around the customer and the end-user's role proved to be a major difficulty in Agile and UX projects. If end-users are not included in UX and Agile activities, then requirements are based on assumptions about what the user needs.

Despite these challenges, there are opportunities to bring UX and Agile together. A process of thematic analysis was applied to achieve the good practices for the difficulties identified and was presented in a tabular comparison which highlighted the link between the philosophies of UX and Agile software development. The findings suggest that these difficulties can be overcome. However, further investigation is required to broaden the empirical basis of research in this area and suggests research questions to be asked in the South African context.

CHAPTER 3: RESEARCH DESIGN AND METHODOLOGY

3.1. Introduction

The literature review in Chapter 2 clearly indicated the importance of combining the fields of UX with Agile. There is insufficient clarity and shared direction within the field of Agile UX. Although there were a few frameworks and models introduced, the field is still young and under continuous development within the South African context. These two fields provide the underlying principles of the research methodology of this thesis and will be used to shape the design of this research. This chapter presents the research design, data collection and data analysis procedures that were deemed to be most suitable for addressing the formulated research question *"How can good UX development practices be integrated into Agile within the South African context?"*

Research design and methods can be described as different layers of how one would go about answering the research questions and will contain clear objectives which include how data will be collected and the constraints that one will inevitably have. The different layers for this research is depicted in the research 'onion' in the figure below (Saunders, Lewis and Thornhill 2009). Each of these layers will be identified and discussed in the sections that follow within the context of this research.

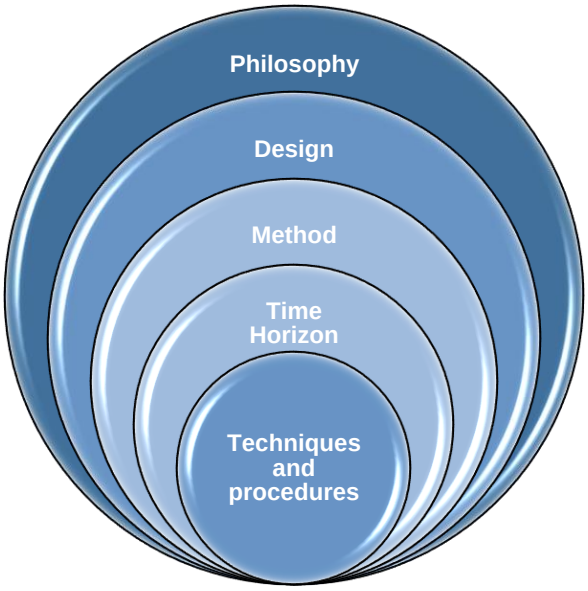


Figure 10: The Research 'Onion'
Source: Adapted from Saunders, Lewis and Thornhill (2009: 138)

3.2. Research Design

The first layer of the research 'onion' relates to the philosophy. There are a number of different research approaches available for researchers studying the various aspects of software engineering and human-computer interaction. For this study, the research approach that will be adopted needs to demonstrate how the integration of UX in Agile in the daily practices of organisations can be interpreted from the lens provided by the research approach. The synthesis of Agile UX is considered important to this thesis as it directly influences the endeavour in shaping Agile methodologies with the objective of creating better UX for the end-user. The next sub-sections delve deeper into the research approach of this study and the research paradigm.

3.2.1. Research Philosophy

According to Saunders, Lewis and Thornhill (2009) the research philosophy is defined as the development and nature of knowledge. There are two main ways of examining the research philosophy: *ontology* and *epistemology* (Saunders, Lewis and Thornhill 2009: 109). Patel (2015) explains that research can be characterised through the following:

- Ontology – What is the nature of reality?
- Epistemology – What constitutes acceptable knowledge in the field of study?
- Methodology – How do you go about finding it out?

A research paradigm, also known as a worldview, is the set of common beliefs and agreements about how a problem should be understood and addressed (Patel 2015).

A research study is usually grounded in one of four popular paradigms; namely post-positivism, constructivism, advocacy/participatory and pragmatism as summarised in the table below (Creswell 2014; George 2015)

Table 5: Research Paradigms

Postpositivism (Quantitative dominant)	Constructivism/Interpretive (Qualitative dominant)
Singular reality •Distance and impartiality •Unbiased •Deductive, theory verification, top down approach •Empirical observation and measurement	Multiple realities •Closeness •Biased •Inductive, theory generation, bottom up approach •Understanding multiple participant meaning
Advocacy/Participatory (Qualitative dominant)	**Pragmatism (Mixed method dominant)**
Political reality •Collaboration •Biased and negotiated •Participatory •Empowerment issue-oriented	Singular and multiple realities •Practicality •Biased and unbiased •Pluralistic approaches,use "what works" •Problem centred, real-world practice oriented

Source: Adapted from George (2015)

A constructivism/interpretive paradigm was applied to this study. According to Saunders, Lewis and Thornhill (2009) and Patel (2015), interpretive research is perceived as affording more in-depth investigations and is associated with a qualitative research design. Other authors such as Lee and Baskerville (2003) advocate that interpretive research is a popular variation for IT-related research. It is a suitable conduit for providing new insights emerging from the phenomena under investigation (Lee and Baskerville 2003). In terms of ontology, interpretivists accept that there is no single truth or reality (Patel 2015), as interpretivism stems from multiple participant meanings (George 2015). The epistemology of interpretive research defines knowledge as reality and needs to be interpreted from a theoretical lens through the eyes of a social construction of actors (Patel 2015). The interpretive paradigm has been summarised in the table below.

Table 6: Research Paradigm

Paradigm	Ontology	Epistemology	Theoretical Perspective	Methodology	Method
Constructivism/Interpretive	There is no single truth or reality	Reality needs to be interpreted	Interpretivism, as reality needs to be interpreted.	Ethnography, grounded theory, Heuristic enquiry, Action research	Usually Qualitative, could include: Qualitative interview, Observation, Case Study, Theme identification

Source: Adapted from Patel (2015)

This research falls within the interpretive paradigm, the purpose being to understand and elucidate the experience and theories of the participants within Agile software development and UX.

The interpretive paradigm enabled the researcher to explore the participants' perceptions through the lens of their experience working in an Agile UX environment. The table that follows motivates in more detail the research paradigm, ontology, epistemology, theoretical perspective, methodology and method that were the basis for this research study.

Table 7: Research Paradigm in Relation to This Study

		Motivation for this study
Paradigm	Constructivism/Interpretive	This paradigm allows for multiple realities. It is inductive and allows for theory generation. It allows the researcher to understand multiple participant meanings, as both Agile and UX needed to be considered during this study.
Ontology	There is no single truth or reality	Reality around the integration of UX in Agile needs to be interpreted and the best development practices for integration. There is no single reality or truth and limited studies conducted within the South African context on the topic of Agile UX.
Epistemology	Reality needs to be	The researcher needs to use methods to

	interpreted	uncover the multiple realities.
Theoretical Perspective	Interpretivism, as reality needs to be interpreted	Saunders, Lewis and Thornhill (2009) advocate that the interpretive paradigm requires the researcher to adopt an empathetic stance to the research and requires that the researcher interpret the actions of the research subjects to understand their point of view. It is also important to understand the differences between humans in their role as 'social actors' (Saunders, Lewis and Thornhill 2009). With this is mind, the researcher aims to: • Adopt an empathetic stance when conducting research to interpret the participant's point of view on integrating UX in Agile. • Understand UX and Agile team members and their roles as 'social actors' in Agile projects with the intention of identifying similarities, differences and good development practices.
Methodology	Grounded Theory	Grounded Theory allows for theory building and starts off with an initial theoretical framework, leading to a generation of theories and predictions by a series of iterations (Saunders, Lewis and Thornhill 2009: 149). With this in mind, the researcher aims to adopt this methodology as follows: • Formation of initial literature from previous studies conducted on the topic of Agile UX, being cognisant of the fact that there were no previous studies on how UX is integrated in Agile within the South African context. • Develop theory through constant analysis and observation. • Further observations are carried out to confirm the theory and identify patterns

		and themes.
Method	Qualitative	The qualitative method allows the researcher to interpret non-numerical data in the form of events and perspectives of people for a deeper understanding on Agile UX.

The next section aims to orientate the reader to the next layer of the research 'onion' and showcases, in more detail, the research approach that was applied in this study.

3.2.2. A qualitative research approach to Agile UX

The previous sub-section juxtaposed the various research philosophies available and it was concluded that the interpretive paradigm was most suitable for this study. If we look at the next layer of the research 'onion' in figure 3-1, it relates to the research design. This sub-section aims to synthesize the research approach used for this study. Before doing so, a definition of research design is provided, and the various research designs are juxtaposed with the objective of understanding which realm this study falls under.

A *research design* is described as a road map in terms of how researchers should go about responding to their research question(s) (Saunders, Lewis and Thornhill 2009). There are three research designs to choose from, namely *quantitative, qualitative* or *mixed methods.* The table below summarizes these three designs.

Table 8: Research Designs

Quantitative	Qualitative	Mixed Method
Experimental	Grounded Theory	Action Research
Between Group	Systematic	Practical
Design	Emerging	Participatory
Within Group	Constructivist	Community-Based
Designs	Ethnographic	
Quasi-Experimental	Realistic	Mixed Designs
Correlational	Critical	Triangulation
Explanatory	Case Study	Embedded
Prediction	Narrative Research	Sequential
Survey	Biography	Explanatory
Cross-Sectional	Phenomenology	Exploratory
Longitudinal	Case Study	

Source: Adapted from Saunders, Lewis and Thornhill (2009)

In the literature review it was ascertained that usability and UX have become an integral part of the philosophy of many organizations that develop user-centric products (Budwig, Jeong and Kelkar 2009; Da Silva *et al.* 2011; Humayoun *et al.* 2017; Kashfi, Nilsson and Feldt 2017). Authors such as Ferreira, Sharp and Robinson (2012) opine that the integration of UX and Agile can be defined as being dependent on the specific organisational setting where it occurs. The same authors strongly suggest the need to account for user experience and social context before choosing development methods. This argument is supported by Cohn, Sim and Lee (2009), who advocate the importance of understanding the practical settings before proposing any fundamental improvements and to also involve practitioners in these efforts.

While investigating Agile and UX principles, it became evident to the researcher that this study meandered towards a qualitative research approach. Gorman, Clayton, Shep and Clayton (2005) defines qualitative research as follows:

"A process of enquiry that draws data from the context in which events occur, in an attempt to describe these occurrences, as a means of determining the process in which events are embedded and the perspectives of those participating in the events, using induction to derive possible explanations based on observed phenomena." Gorman et al. (2005: 3)

A synthesis of the definition of qualitative research, as extrapolated from Gorman *et al.* (2005) is depicted in the table below in terms of how it was applied in this research.

Table 9: Application of Qualitative Research

Key processes in Qualitative Research	**Application in this study**
Process of enquiry for data	Interviews and questionnaires Deep understanding Rich responses Subjective data
Events that occur	Integration of UX in Agile
Describe the events that occur	Description of user requirements, user testing, user involvement, user presence
Determine the process in which events are embedded	Process of integrating UX in Agile
Perspective of those participating in events	Good practices identified Difficulties Room for improvement

The strength of using a qualitative research approach, as opined by authors such as Dittrich, Randall and Singer (2009), is that it allows for exploring and illuminating the in-situ practice of software engineering and the design of research methods is related to the research question (Seale, Gobo, Gubrium and Silverman 2004).

According to Creswell (2014), qualitative research deals with the questions of *What? Who?* and *How?* which aims at describing what is going on in the research. On the opposite end of the spectrum, quantitative research deals with the question of *Why?* which aims at exploring comparisons between groups or finding relationships between variables (Creswell 2014). To reiterate, the research question for this study is:

"**How** can good UX development practices be integrated into Agile within the South African context?"

This serves as degree of validation that this study falls in the realm of qualitative research as the research question deals with **how** UX development practices are integrated into Agile.

Fox (2010) alludes that qualitative research allows the researcher to elicit rich responses from participants in order to identify themes in the data. The human element, which is UX for this study, cannot be described using numbers neither can it be explained by manipulating, measuring or controlling variables (Fox 2010: 42). Supporting claims by John, Maurer and Tessem (2005) portray that traditionally most software research is quantitative in nature as it takes subjects out of their work context and places them into a controlled environment. However, to get a holistic understanding of research in a software project, a qualitative research design is required to understand what the important factors are and how they influence the research. For this study, the subjects are required in their work context to develop knowledge on what are the good development practices for integrating UX in Agile. It becomes even more evident in the literature that results cannot be obtained through a purely quantitative research as it requires research within a controlled environment with random sampling (Kollmann 2008; Hussain, Slany and Holzinger 2009; Fox 2010). Agile and UX rely on the interactions between UX specialists, the end-user and the Agile team. Hence the interaction cannot be described using numbers or variables nor can it be conducted within a controlled environment as the researcher will gather data from non-controlled environments (i.e. Agile software teams). A further consideration is that participants with Agile and UX experience are difficult to find. Hence, the limitation was that random sampling could not be used. The nature of this study is therefore of an open-ended inquiry rather than one where the data is measurable and observable and where the research question is specific and narrow (Seale *et al.* 2004; Creswell 2014).

Taken together, the findings from the authors mentioned above indicate that there is a growing interest in leveraging qualitative research to understand the cooperative practices of UX design and development within Agile. A point of view on these, synthesized from the literature above and from authors such as Bordin and De Angeli (2016), indicate that results from qualitative studies are more informative than those obtained through quantitative research. There is a growing interest from both the software engineering and UX communities to achieve a better understanding of Agile UX through qualitative methods. An understanding

of human and social aspects can in turn support the empirical goal of software engineering (Bordin and De Angeli 2016), which is “to improve software practices through evidence-driven research” (Sharp, Dittrich and De Souza 2016: 2). The need for evidence-driven research is particularly favoured as it calls for an appreciation of the practitioner’s viewpoint, as well as the flexibility in research design that is allowed in qualitative research (Sharp, Dittrich and De Souza 2016).

Despite the benefits and an increasing number of publications on Agile UX, there is still limited work on adopting qualitative research on the topic. Bordin and De Angeli (2016) make a strong statement that there is a need for more empirical studies on human and social factors in the Agile community and stresses that a qualitative approach is fitting as it places emphasis on communication and cooperation which is also intrinsic in the Agile methodology. There are limited academic studies focusing on Agile UX within the South African context. To the best of the researcher’s knowledge, there was no recent statistics available on Agile UX.

For the reasons outlined above, the most appropriate design for this study was the qualitative research design. The next sub-section synthesizes the next layer of the ‘research onion’ which relates to the research method.

3.3. Research Method

In the previous sections, the first two layers of the research ‘onion’ were peeled away which synthesized the research philosophy and the research design. As mentioned in the previous sections, a qualitative research design was chosen for this study. There are six common qualitative designs: *phenomenological, ethnographic, grounded theory, historical, case study* and *action research*. The next sub-section deals with the “Grounded Theory” method, which was the methodology chosen for this study.

3.3.1. Grounded Theory Approaches

3.3.1.1. Background of Grounded Theory

Grounded Theory, which was first published in 1967 by Glaser and Strauss, can be described as a methodology of developing inductive theories that are grounded in systematically gathered and analysed data (Charmaz and Belgrave 2007). Different versions of the grounded theory have been discussed and debated. The table below shows the work, in chronological order, covered by grounded theorists:

Table 10: Grounded Theory Work

Year	Author	Title of publication
1967	Glaser and Strauss	The discovery of grounded theory
1990	Strauss and Corbin	Basics of qualitative research: Grounded theory procedures and techniques
1992	Glaser	Basics of grounded theory analysis
1994	Strauss and Corbin	'Grounded theory methodology: An overview' in Handbook of qualitative research (1st Edition)
1995	Charmaz	'Grounded theory' in Rethinking methods in psychology
1998	Strauss and Corbin	Basics of qualitative research: Grounded theory procedures and techniques (2nd Edition)
2000	Charmaz	'Grounded theory: Objectivist and constructivist methods' in Handbook of qualitative research (2nd Edition)

Source: Adapted from Muhaiyuddin, Bakar and Hussin (2016)

As indicated in the table above, there are quite a few approaches for grounded theory. Studies on Agile UX research and theory building methodologies have been one of the pertinent focuses in the literature as evidenced by authors Muhaiyuddin, Bakar and Hussin (2016), Martin, Biddle and Noble (2004), Kollmann (2008), Fox (2010) and Ferreira, Sharp and Robinson (2012). Recent trends on Agile UX have confirmed the need for research approaches that aim to resolve context-specific Agile UX issues. Quite a few authors advocate that grounded theory is mostly applicable among the other research approaches available for context-specific research, as highlighted by authors such as Martin, Biddle and Noble (2004), (Kollmann 2008), Fox (2010) and Ferreira, Sharp and Robinson (2012).

Although grounded theory has been used quite extensively in Agile UX and other research, Devadas, Silong and Ismail (2011) make a strong argument that there has not been sufficient considerations of the changes and ontological segregations that are inherent in the two commonly used versions of grounded theory, which are the ***Glaserian*** and the ***Straussian*** versions. The next sub-section juxtaposes the perspectives of these two versions of grounded theory and presents the version that was adopted for this study.

3.3.1.2. Glaserian and Straussian Grounded Theory Approaches

There are numerous guidelines for conducting effective grounded theory research. The guidelines proposed by Devadas, Silong and Ismail (2011) and Muhaiyuddin, Bakar and Hussin (2016) were most appealing to this study. Devadas, Silong and Ismail (2011) compare and contrast the two approaches in their seminal paper in which they stress that Glaserian and Straussian perspectives differ in terms of the "paradigmatic dimensions, formulation of research questions, analysis procedures used, usage of literature, sampling procedures, and the procedures for validating the resultant theory" (Devadas, Silong and Ismail 2011: 348). The major differences in the Glaserian and Straussian version of grounded theory is the application of existing literature in the research process. Glaser opposed the idea of using existing literature in the research process as he was of the belief that it will negatively affect the data collection, analysis and theory development process (Muhaiyuddin, Bakar and Hussin 2016). However, evidence from previous studies suggest that the use of existing literature assists researchers in confirming if the intended study has been conducted before and also aids in refining research questions and methodological fundamentals (Muhaiyuddin, Bakar and Hussin 2016). Corbin and Strauss (1990) advocate that the data collection and analysis processes are interrelated. Hence, it is not uncommon for qualitative researchers to collect data prior to systematic analysis. Analysis is necessary from the outset of the study as it directs the next interviews and observations. In order to not miss any vital information for the study, the researcher must analyse previous data for cues and relevant issues that will be used to direct the next set of interviews (Corbin and Strauss 1990).

The following summation provides a synthesis of some of the core principles of the Glaserian and Straussian versions of grounded theory:

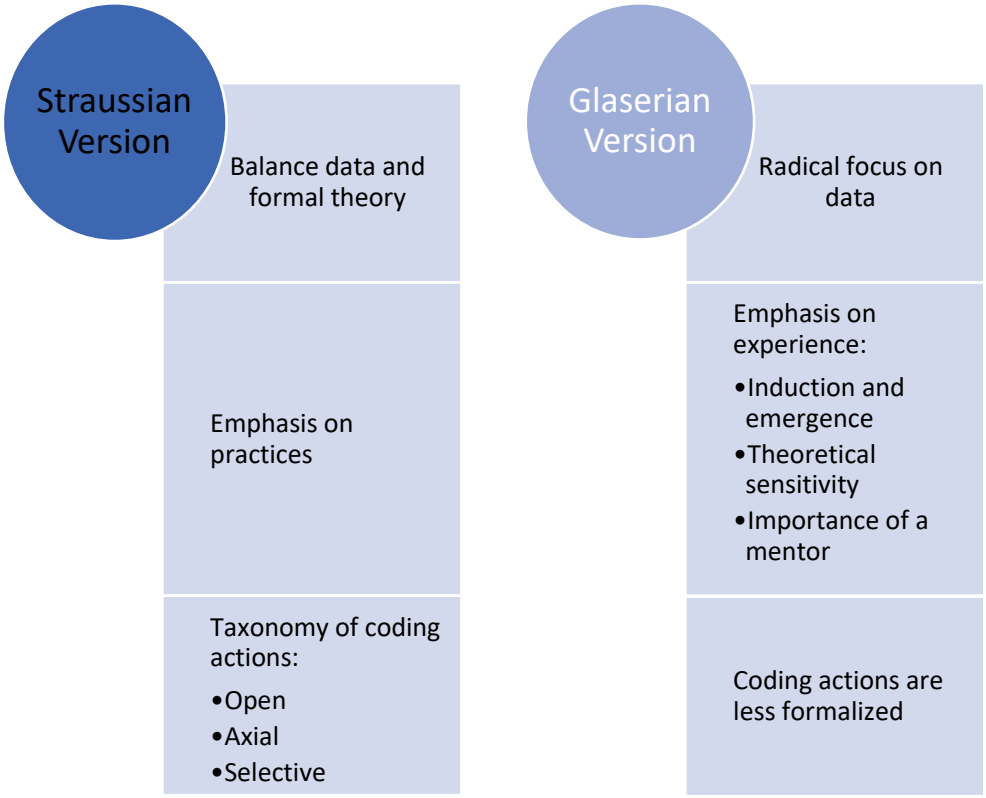


Figure 11: Straussian versus Glaserian Grounded Theory Methods
Source: Adapted from Muhaiyuddin, Bakar and Hussin (2016: 189)

The synthesis of the two commonly used versions of grounded theory above highlights the underlying philosophies of these methods. The tabular representation below has been juxtaposed using a short thematic analysis as evidenced by Devadas, Silong and Ismail (2011) and Muhaiyuddin, Bakar and Hussin (2016). The comparison highlights the rationale for adopting the Straussian version of grounded theory:

Table 11: Comparison of Glaserian and Straussian Grounded Theory Approaches

	Glaserian	**Straussian**
Paradigmatic difference	More towards post-positivism	Constructivism
Formulation of research question	No upfront research question	Some research question based on previous literature, experience
Use of literature	Oppose the use of previous literature upfront	No hard and fast rule with regards to use of literature
Sampling procedures	Theoretical sampling	Theoretical sampling
Analysis procedures	Coding and constant comparison	Open, axial and selective coding.
Procedures for validating resulting theory	• Fit • Relevance • Work • Modifiability	• Validity • Reliability • Credibility • Plausibility • Value of theory • Adequacy • Empirical grounding

Source: Adapted from Devadas, Silong and Ismail (2011) and Muhaiyuddin, Bakar and Hussin (2016)

The application of a research approach guides how the researcher makes decisions in terms of the validity, depth and breadth of the research (Devadas, Silong and Ismail 2011). It becomes even more evident to the researcher from the tabular comparison above that the Straussian version of grounded theory was the most suitable research method for this study. The Straussian version leans more towards the constructivism paradigm. As outlined in sub-section 3.2.2 the constructivism/interpretive paradigm provides a suitable conduit for providing new insights emerging from the phenomena under investigation. Moreover, the researcher is able to formulate some research questions based on the use of previous literature (as cited in Chapter Two in this thesis). Coding procedures allows the researcher to build the data into themes by using basic techniques. The Straussian version guides the researcher in organising and analysing data collection, which results in firmer coding techniques. The reliability and validity of data is apparent in the three types of coding techniques, as depicted below:

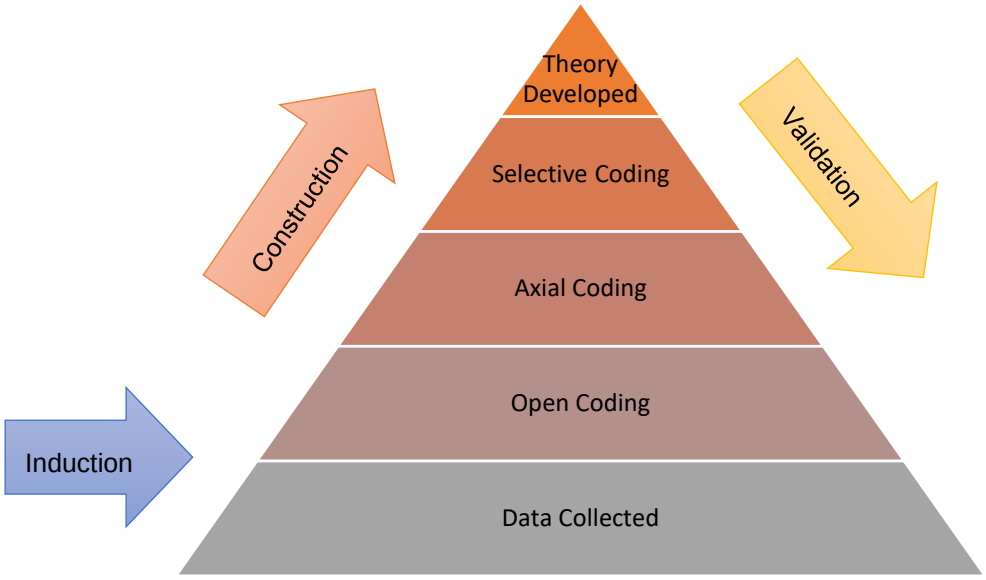


Figure 12: **Straussian Grounded Theory Method**
Source: Adapted from Corbin and Strauss (1990); (Muhaiyuddin, Bakar and Hussin 2016)

The selection of using grounded theory for this study was not arbitrary. There is a myriad of other qualitative methods. However, grounded theory has become a dominant paradigm for both the social and information systems field (Lawrence and Tar 2013). As evidenced in the previous sub-section and by studies conducted by Martin, Biddle and Noble (2004), Kollmann (2008), Fox (2010) and Ferreira, Sharp and Robinson (2012), recent trends in the literature on this method shows that it is discursive on application and philosophy. Lawrence and Tar (2013) note that grounded theory is iterative in nature and requires constant comparisons between concepts and data. The constant comparison across the data, controls the scope of emerging theories (Lawrence and Tar 2013).

Supporting this argument, Kollmann, Sharp and Blandford (2009) opine that it is a suitable method as it allows the researcher to discover patterns and themes and allows for exploration of the topic without preconceptions so that previous research can be validated or enriched by new findings. One of the objectives of this study is:

"To evaluate the existing Agile UX software development process in the South African context and confirm if approaches from previous studies applies to the South African context."

In Chapter Two, the similarities, known differences and good practices identified in previous studies were discussed. The first step of the grounded theory method is to build theory on existing literature (Saunders, Lewis and Thornhill 2009). This laid the foundation and enabled the researcher to confirm if integration practices identified in the literature applied to the South African context.

Fox (2010) advocates that grounded theory is dependent on the researcher commencing the study without preconceived theories. The study should be based on the researcher analysing the data and identifying theories or finding themes. Based on very little information being available in terms of how Agile and UX were integrated within the South African context, there were no preconceived theories going into this study. Grounded theory therefore allowed the researcher to identify patterns and themes on Agile UX through fieldwork conducted within the South African context.

These patterns and themes are the foundation in the development of the generic framework for Agile UX, which is covered in Chapter Five. The framework will be developed through the lens of the design thinking methodology. Design thinking has been considered as the chosen methodology as it enhances UX work by integrating three design approaches; design thinking, design for UX and Agile software development (Adikari, McDonald and Campbell 2013). A detailed deliberation of design thinking is covered in section 5.3.

The fundamental components of grounded theory are juxtaposed and presented below in relation to this study:

Table 12: Fundamental components of a grounded theory study

Component	Stage	Description	Application in this study
Openness	Throughout the study	Grounded theory is based on inductive analysis which requires the researcher to adopt a very open approach to the study.	The researcher was transparent with the participants in this study. The background of the study, aims and objectives were explained to all participants. Participants were also advised on the confidentiality of their responses.
Analysing immediately	Analysis and data collection	Analysis in grounded theory commences as soon as possible. Hence the researcher does not need to wait for data to be collected.	Similarities, differences and known good practices were analysed from previous studies before commencing with data collection in the South African context.
Coding and comparing	Analysis	The process of coding involves breaking down data into smaller components known as *categories* or *concepts*. The coded data is then compared to each other to understand and explain any differences.	Data collected and analysed upfront was coded in categories (viz. "Similarities", "Differences", "Good Practices"). These categories were split into concepts.
Memo-writing	Analysis	Memos are written throughout the project and can comprise of various events during the study. Memos can also include the comparison of coded data explained above.	Memos were written based on key concepts identified in the semi-structured interviews, with the purpose of tying these memos back to the coded categories and concepts.
Theoretical sampling	Sampling and data collection	Theoretical sampling involves *coding*, *comparison* and *memo-writing*. Theoretical sampling aims to identify relationships in current data sets and reveal emerging theories.	Theoretical sampling was adopted throughout the study and used in three iterations. Data from the literature of previous studies was used to identify relationships in current data

			sets within the Agile UX environment in South Africa.
Production of a substantive theory	Analysis and interpretation	The results from the study results in production of substantive theory. Substantive theory is set of concepts that are linked to one another.	The results of the study are covered in Chapter 4 of this thesis.

Source: Adapted from Sbaraini, Carter, Evans and Blinkhorn (2011: 3)

Having provided a perspective on grounded theory and the application to this study, the next sub-section looks at the sampling method.

3.4. Sampling

The idea of sampling is to choose a target population that will allow the researcher to reach conclusions regarding a broader population. The target population, sampling method and the sample size are imperative in collecting data from the correct population in order to provide accurate results for the study.

3.4.1. Target Population

According to Cooper, Schindler and Sun (2006), the target population refers to the population being identified to conduct or participate in a study. The target population for this study were key employees working in the UX and Agile development environment in South Africa. The participants were employed in medium to large companies. The researcher could not predict the size of companies as a non-probable sampling technique was used (this is covered in the next sub-section). The participants for this research included Scrum Masters, Front-end developers, Development Managers, Agile Coach, Usability engineers (and analysts), UX developers and Interaction designers. Other roles such as testers and backend developers were also considered as they frequently interact with UI/UX members.

3.4.2. Sampling Methods

The sampling method used for this study was theoretical sampling.

As per Sbaraini *et al.* (2011), theoretical sampling is a central feature of grounded theory design, and informs the coding, comparison and memo-writing process.

Theoretical sampling also allowed for comparing relationships to the theory from previous research to what is currently being researched thus highlighting gaps in existing data.

There are two categories of sampling techniques – probability and non-probability sampling. Saunders, Lewis and Thornhill (2009) defined these as follows:

Probability Sampling: The probability of each case from the population being selected is known and usually equal.

Non-probability Sampling: The probability of each case from the population being selected is not known and it is difficult to make inferences about the population.

A *non-probability purposive* sampling technique was applied to this study as the researcher could not predict or guarantee that each constituent of the population will be contained in the sample selected.

3.4.3. Sample Size

Qualitative studies typically require a smaller sample size than quantitative studies. According to Mason (2010), a number of issues can influence the sample size. However, the sample should be large enough to obtain most or all of the perceptions that will lead to attainment of saturation. This study used non-probability sampling to sample a selection of 66 key employees within the Agile and UX development environment. The sample size for the semi-structured interviews was 15. In order to adequately attain saturation, the researcher used a large enough sample. Saturation was reached after completing 10 interviews as there were no new themes being generated from the interviews. At this point, the researcher compiled the questionnaire to provide reliability and validity of the interviews and to ascertain if the results from the interviews were applicable to a larger sample size. The sample size for the online questionnaires was set as 50. Out of the 50 questionnaires distributed, 34 were successfully completed yielding a response rate of 68%.

3.5. Data Collection

This research relied on two methods of data collection to study the integration of UX in Agile. The first method was semi-structured interviews, which is discussed in the next section. This will be followed by the second method which was an online questionnaire.

3.5.1. The Interviews

According to Saunders, Lewis and Thornhill (2009), interviews assist in gathering valid and reliable data which assist in answering the research question(s) and the objectives of the study. Semi-structured interviews are often referred to as 'qualitative' interviews and allow the researcher to cover a list of themes. Questions can also be added or omitted based on the organisational context with regards to the particular research topic (Saunders, Lewis and Thornhill 2009: 320).

The interview comprised of five predefined questions which were broad in nature. The purpose of these questions was to elucidate the good development practices of integrating UX in Agile as well as the challenges that were faced. The five main questions are as follows:

Table 13: Interview questions and aims

Interview Question	**Aim of question**
1. Can you tell me about your job role, background in terms of employment, the type of projects you work on?	Thematic analysis around the type of roles, experience and understanding of Agile. To also help answer research sub question 3, as this relates directly to the roles that are involved in the process of integrating UX in Agile.
2. What are the processes and the activities when developing the software?	To identify processes, activities and development practices for integrating UX in Agile (Main research question). To also help identify Agile development techniques (Sub-Question 1), UX development techniques (Sub-Question 2), team members that are involved (Sub-Question 3) and problems when integrating UX in Agile (Sub-Question 4).
3. Can you describe how UX activities are integrated into the software development cycles?	To identify processes, activities and development practices for integrating UX in Agile (Main research question). To also help identify Agile development techniques (Sub-Question 1), UX development techniques (Sub-Question 2), team members that are involved (Sub-Question 3) and problems when integrating UX in Agile (Sub-Question 4).
4. How has the integration of UX in Agile	Identify problems when integrating UX in Agile

affected the software development process?	(Sub-Question 4). Also identify improvements for creating a generic framework and for recommendations for future research.
5. Are the following Agile values followed during integration of UX in the project?	Identify if the 4 keys values of Agile are followed and how they ensure that UX requirements are met (Sub-Question 1).

Each of the five questions resulted in detailed sub-questions being asked based on the response received from the interviewee. A full list of questions can be found in section 8.3 in the Appendix.

3.5.2. The Questionnaire

The second method of data collection was an online questionnaire comprising of 26 questions. According to Saunders, Lewis and Thornhill (2009), questionnaires are one of the most widely used data collection techniques as it allows each person to respond to the same set of questions and allows for collection of data from a large sample size. However, they do not work well for exploratory research with a large number of open-ended questions therefore it may be better to use them with other methods (Saunders, Lewis and Thornhill 2009).

The questionnaire was designed to be distributed to a larger population than the interviews in order to have a representative sample.

According to Kitchenham and Pfleeger (2002), some of the pre-requisites for conducting a questionnaire is to understand the objective of the study, the unit of analysis and the sample of the study. In this study, it was essential to select the appropriate participants, understand the background of the participants and align the responses to the context of the research. Thus, the questionnaire was used in a multi-method design to validate the results from the interviews and was sent to experts in the field of Agile UX after the objectives, unit of analysis and sample was clear. The target sample for the questionnaire was the same as per the interviews as well as people working in software development that have experience in these fields.

3.5.3. Research Procedure

As mentioned in the previous sub-section, the data for this study was collected using interviews and questionnaires. This section describes the research procedure that was followed when carrying out the interviews and questionnaires. The research questions not only required the analysis of Agile UX in South Africa, it also required an understanding of the principles underpinning good development practices of these methodologies. Conclusions relating to Agile UX and the philosophies and ideologies underpinning the integration cannot be made without data obtained directly from the people that are working in this environment. In order to elicit information necessary for a thorough analysis, interviews and questionnaires was conducted with Agile and UX specialists.

3.5.3.1. Research Procedure for Interviews

Techniques and guidelines proposed by Brinkmann and Kvale (2015) was adhered to when conducting the interviews. The seven-stage method of interviewing will assist the interviewer in shaping and conducting the interview process based on the knowledge of the topic from previous research, ethical considerations and anticipated results. The seven stages of an interview inquiry are depicted below (Brinkmann and Kvale 2015):

Figure 13: The Seven Stages of an Interview Inquiry
Source: Adapted from Brinkmann and Kvale (2015)

A letter was obtained from the Durban University of Technology to conduct the study and was emailed to the participants before commencing with the interviews. The letter can be found in the section 8.5 in the Appendix.

3.5.3.2. Research Procedure for Questionnaire

The researcher finalised the questionnaire with questions that would validate some of the relationships identified in the interviews. The questionnaire promoted validity and reliability of the interview data with more information from more participants. Questions was targeted at confirming if good UX development

practices identified in previous literature applies to the South African context and to confirm new practices that was identified in the interviews.

The questionnaire contained a cover letter explaining the purpose of the study and also provided assurance to the responders that all responses were treated as confidential.
The structure of the questionnaire and how it should be filled out was explained upfront. The purpose was to measure the extent at which the responders agreed with the questions on Agile UX. The questions that were developed during the interviews were broadly used in the questionnaire and were re-developed, where necessary, after the interview sessions were concluded. The questions were posed in an order that made the entire questionnaire clear and easy to understand. The sequence of questions was designed to make sure that the responders gave unambiguous answers.

A 6-point likert scale was used with majority of the questions having the response category of 'Disagree Strongly' and 'Agree Strongly' at the opposite end of the axis. The researcher opted to use a 6-point likert scale as it allowed the respondent to either choose a positive or negative option and not allowing them to 'sit on the fence' by ticking a neutral option. Previous studies conducted by Garland (1991) provides evidence that bias arising from respondents' desires to give a socially unacceptable answer or to please the interviewer can be minimised by eliminating a mid-point or neutral answer. The same author also debates that the mid-point in a scale distorts results.

Basic demographics were not obtained in the questionnaire as the questions were focused on the fields of expertise. Any conflicting answers were not explained by differences in age, gender etc. Instead the researcher based the questions on the responders' experience within the fields of UX, Agile and their combination.

The questionnaire was compiled using QuestionPro, which is a survey tool. The questionnaire was sent via email and posted in the Agile/UX groups that were used for the interviews. The questionnaire was online for two weeks.

3.6. Data Analysis

The data analysis process commenced during the process of data collection. The researcher planned to use information from each interview session in subsequent interviews to elicit more specific information. After collecting data from the interviews and online questionnaires, it was transformed into meaningful data by using transcription and coding procedures as part of the grounded theory approach. For the purposes of data management and analysis, NVivo 11 was used. This software was also used to query keywords for comparison and to code categories and themes.

The grounded theory comprises of three stages of coding activities. The goal of the coding activities is to produce a theory to explain a situation of interest Fox, Sillito and Maurer (2008).

The coding procedure and stages are defined below:

1. Open Coding:

   The first stage of coding is open coding which is the analytical process through which concepts are identified in the data (Lawrence and Tar 2013). This process involved identifying keywords in the transcribed data from the interviews and questionnaires.

2. Axial Coding:

   The next stage of coding is axial coding, which involves re-building the data by identifying relationship between categories and sub-categories identified in the open coding stage. "Axial codes typically represent categories that describe open codes" (Lawrence and Tar 2013: 32). This process involved identifying connections among the codes through comparison and analysis in terms of the responses in the interviews and questionnaires. These relationships became the core categories for the next stage of coding.

3. Selective Coding:

   The final stage of coding is selective coding which involves using the core categories in the axial coding stage to define high level concepts that form

the themes, or the big picture of the research. According to Lawrence and Tar (2013), selective coding allows for integration and refinement of the categories in the theory which account for the phenomenon currently being investigated.

The above-mentioned methods of analysis have been used in previous studies by Fox (2010) and Kollmann (2008). According to Kollmann (2008), these methods were suitable for the research as it allowed the researcher to discover patterns and themes in data instead of applying existing categories derived from related studies.

3.7. Limitations

Although there is sufficient literature from studies conducted abroad, the first limitation is the lack of UX practitioners within the South African context. UX is an evolving role and there is predominantly no UX specialist involved in projects in South Africa whilst it was evidenced in the literature in Chapter 2 that there was presence of UX specialists in projects abroad. The good news is that there has been a keen interest in UX over the past two years within the South African context, where teams are including UX in Scrum by including UX practitioners. This limitation resulted in the researcher finding it difficult to recruit participants that specialised in UX due to the lack of specialists in the field. In order to overcome this limitation, the researcher designed the questions in the interviews and the questionnaire to record the participants' responses and their level of specialization on UX and how the limitation of having UX specialists present in projects affected the end-product and end-user.

The second limitation is that the study cannot be generalised due to the small sample size that was used. The sample consisted of ten participants for the interviews, majority of which reside in Durban, Kwazulu Natal. As a result, the perceptions of the participants were based on what they perceived is happening in Durban. It is worth noting though, that all companies have branches nationally, with three companies having an international footprint. The researcher overcame this limitation by designing a questionnaire that was sent to a larger sample size that included participants nationally throughout South Africa.

3.8. Ethical Considerations

This research conformed to the ethical standards set under the auspices of the Durban University of Technology. In so doing, the participants were informed about the steps taken in the research and how their participation would be used. The participants were also informed that their participation is completely voluntary and would not affect their jobs in any way. The next section details the ethical considerations that were undertaken for this study.

3.8.1. Consent

Participants in the interviews were briefed in terms of the purpose of the study. It was clear to them that their participation was voluntary, and they can withdraw from the study at any stage. The rationale behind the study was also explained and data collection and analysis were described to them. As mentioned in the research procedure, a letter was obtained from the Durban University of Technology to conduct the study and was emailed to the participants before commencing with the interviews. Participants agreed that they understood everything clearly and gave consent to participate in the study.

Participants for the questionnaire were also made aware of the purpose of the study and were assured that all responses were treated as confidential. Instructions were given that no identifying marks must be typed on the survey as participants are meant to be anonymous.

This study deals with information relating to their experiences and problems relating to Agile UX, therefore effort was made to ensure that no identification relating to the participants and their companies will be revealed to the public. According to Saunders, Lewis and Thornhill (2009: 191) the researcher should not subject the research population to embarrassment, harm or any other disadvantage. In addition, the research needs to consider the extent at which the researcher should collect data from a population that is unaware that they are the subjects of the research and have therefore not consented (Saunders, Lewis and Thornhill 2009).

3.8.2. Risk

This research is of an exploratory nature. The interviews and questionnaires did not cause disturbances or distraction to the participants' work as the research was conducted at their convenience. The methods used in the interviews and questionnaires were designed to assist the participants to describe their experience working in an Agile UX environment. This research therefore did not modify the thinking, attitudes or behaviour of the participants. Furthermore, the questions in the interviews and questionnaires did not involve any sensitive questions that would cause any offence to the participants. According to the methods used in the study, there were no risks involved and it can be construed that the possible benefits of this study outweighs the risks.

3.8.3. Privacy

All personal information about the participants were treated as confidential and remained confidential throughout the study. No personal data appeared on the recordings of the interview sessions, as participants were briefed that they should not mention the company name or their personal details during the session. In addition, the interviewee's names and company names were replaced with pseudonyms (i.e. Participant 1, 2, 3… and Company 1,2,3…). The data gathered from the questionnaires did not include any personal information for the responders.

3.9. Conclusion

At the start of this chapter the intent was to peel through the layers of the research 'onion' in figure 3-1 and to introduce the research design and methodology that was applied in this study to answer the research question.

A discussion of the research philosophy and the research paradigm outlined the details of how the research problem should be understood and addressed. The research design for this study was a qualitative design which is congruent with the constructive/interpretive paradigm.

The grounded theory methodology was used to develop theory on how good UX development practices can be integrated in Agile. Next the sampling was

discussed, whereby key employees within the Agile and UX environments in South Africa were identified to participate in this study. All participants contributed to this theory by sharing their experiences in the Agile UX workplace and their perspectives on the similarities, good practices and difficulties with integrating these methodologies.

The sampling method selected was a non-probable purposive sampling technique due to the research being unable to predict or guarantee that each constituent of the population will be included in the sample. The first source of data collection was interviews. The researcher conducted 10 semi-structured interviews. Data gathered from the interviews helped understand the underlying beliefs and philosophies around Agile UX and confirmed if the research conducted abroad applies to the South African context.

The second source of data was a questionnaire. This data was an additional source for analysis to validate findings from the interviews. It also contributed in increasing the sample size and allowed for a more thorough analysis to support any future recommendations. The data analysis process was based on transcribing information from the interviews and questionnaires and coding activities using NVivo 11 as an analysis tool. Coding activities that are inherent in grounded theory was used to document, categorize and identify themes and patterns.

Next, the limitations of the study were examined, with the first being the lack of UX practitioners and the second being that the sample size consisted of most of the participants residing in Durban.

The chapter concludes by presenting the matter of ethical considerations in relation to consent from the participants as well as risk and privacy. Ethical considerations were addressed during all interviews and questionnaires and the researcher ensured that no sensitive questions were included. In terms of privacy, the information gathered was treated with the strictest of confidentiality and remained confidential throughout the study.

The research ‘onion’ in figure 3-1 has been updated below and summarises the various layers identified in this chapter in relation to this study.

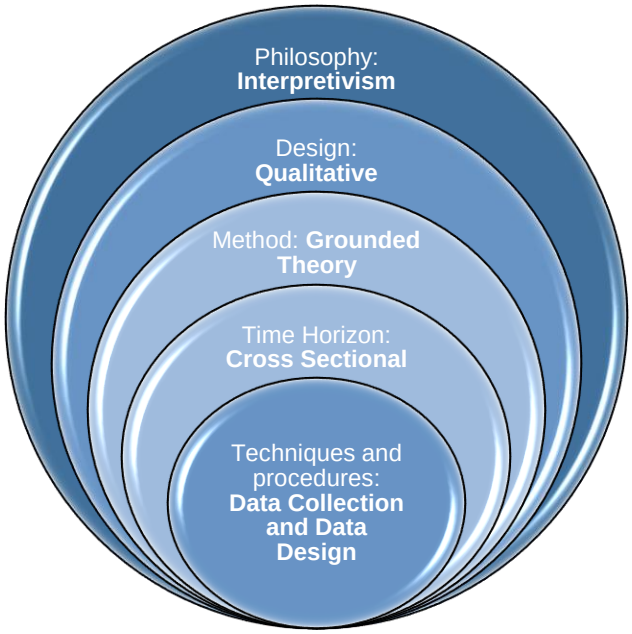


Figure 14: The Research 'Onion' in relation to this study
Source: Adapted from Saunders, Lewis and Thornhill (2009: 138)

The next chapter presents the analysis and results of this study.

CHAPTER 4: RESULTS AND DATA ANALYSIS

4.1. Introduction

The previous chapter presented a theoretical description of the research philosophy, design and methodologies and provided the data collections methods which are interviews and questionnaires.
The literature indicated that a number of findings were discovered on how Agile and UX methodologies co-exist in the software development environment. Notwithstanding, there are many known difficulties when it comes to integrating these two methodologies which has resulted in a research gap and an opportunity to conduct this research.

This chapter aims to address this gap and presents the results and data analysis in the following manner:
The first sub-section provides the results from the literature in terms of the classification of the studies, known difficulties and good practices. Initially the literature review provided insights into Agile UX and provided a view of the integration practices. These insights were taken into consideration in designing the questionnaire and semi-structured interviews. Additional findings and practices were unearthed during the fieldwork.

The next sub-section proffers the practical analysis of the data collected in this study. The first data collection method was the interviews. Grounded theory was used as the data analysis lens to structure the analysis of the research and to provide the view of the interplay between Agile and UX integration practices. Grounded theory provided an appropriate lens to describe integration practices because it is an accepted tool to assist in exposing quite a diverse range of human-computer interactions and socio-technological interactions such as people (technical users and end-users), rules, Agile methodologies, UX methodologies, business culture and business strategies. These interactions were identified in the literature and investigated through fieldwork techniques for the interviews. The knowledge areas presented were informed by the fieldwork within the South African context and this was proliferated by distributing a questionnaire as a

second data collection method. The questionnaire assisted in validating themes uncovered in grounded theory and the results were analysed and discussed within the context of the results from the interviews.

In the last sub-section, the data from both the interviews and questionnaires are categorised into themes in accordance with the interpretivist-grounded philosophical paradigm. The results from the interviews are grouped into categories; supporting evidence from the questionnaires is also categorized in the context of the interview results.

The purpose of this research was to investigate how UX development is currently being integrated into Agile software development within the South African context and to answer the following research question:
“How can good UX development practices be integrated into Agile within the South African context?”

The objectives of the study, as defined in Chapter 1, were to identify the following:
RO1: How UX developers collaborate with members of the Agile team in the software development process.

RO2: How UX is perceived by software development organisations, in order to identify the extent at which organisations have been successful in reflecting the user’s requirements in their applications or products.

RO3: To evaluate the existing Agile UX software development process in the South African context and confirm if approaches from previous studies applies to the South African context.

RO4: To propose a generic framework for integration of UX in Agile and to provide further recommendations for efficient integration of UX in Agile.

A summary of the major components of this research as discussed in the previous chapters and the sub-sections that follow are presented in the Table below:

Table 14: Summary of components of this research

Component	Description	Chapter/Section dealing with this component
Research question	Formulation of research questions and sub-questions	Chapter 1
Paradigm	Constructivism/Interpretivism	3.2.1
Research Design	Qualitative	3.2.2
Research Method	Grounded theory	3.3.1
Data Collection	• Interviews (Semi-structured) • Questionnaires	3.5.1 and 3.5.2
Data Analysis	Grounded theory – Straussian method using coding	3.3.1 and 4.3
Data Interpretation	Thematic analysis in Grounded theory	4.4

4.2. Results from Literature Review

Based on the assessment of related work covered in the literature review, it can be concluded that Agile and UX development share a lot of common principles: both being highly iterative; both encouraging regular user involvement in the development process; both relying on team coherence (Schwaber 1997; Beck 2000; Fox 2010).

These factors have led to many organizations successfully integrating UX in Agile. Notwithstanding, there are many known difficulties as evidenced by the literature, when it comes to integrating these two methodologies. There is also limited knowledge in terms of how these methodologies have been integrated within the South African context, which has resulted in a research gap and an opportunity to conduct this research. The results from the literature review, as summarized in the next section, attempts to narrow this research gap by answering the research question posed:

“How can good UX development practices be integrated into Agile within the South African context?”

4.2.1. Classification of studies from literature review

Looking back at the literature review, related work shows that Agile and UX share many common principles. Below is a tabular representation of the classification of some of the key papers that were reviewed in Chapter Two – Literature Review. A thematic analysis was used to group and classify some of the key studies presented. A full list of the classifications that were used for this study can be found section 8.2 in the appendix.

Table 15: Classification of key studies used for this study

Author	Title of publication	Nature of publication	Keywords Identified	Research Method/Artifact
Ambler (2008)	Tailoring usability into agile software development projects	Journal Paper	Interaction Design, User Story, Agile Method, Capability Maturity Model, Agile Software Development	Framework
Beyer, Holtzblatt and Baker (2004)	An agile customer-centered method: rapid contextual design	Journal Paper	Agile, User-Centred Design, Systems Design, Contextual Design	Framework
Boehm (2002)	Get ready for agile methods, with care	Journal Paper	Planning Spectrum, Developers, Customers, Architecture, Balancing Agility and Discipline	Evaluation Study
Budwig, Jeong and Kelkar (2009)	When user experience met agile: a case study	Conference Paper	Agile, Scrum, Agile Programming, User Experience, UX teams	Case Study
Chamberlain, Sharp and Maiden (2006)	Towards a framework for integrating agile development and user-centred design	Conference Paper	User-Centred Design, Agile, Usability Experts, UCD	Evaluation Study/Framework
Convertino and Frishberg (2017)	Why agile teams fail without UX research	Journal Paper	Agile, UX Research, End-user, UX Design	Evaluation Study
Federoff and Courage	Successful user experience in an	Conference Paper	Agile, waterfall, enterprise, user	Case Study

(2009)	agile enterprise environment		experience, design, research	
Ferreira, Sharp and Robinson (2012)	Agile development and user experience design integration as an on-going achievement in practice	Conference Paper	Agile development, User Experience Design, ethnographically-informed, integration, organisation	Ethnographic Field Studies
Fox (2010)	Agile Methods and User-Centered Design: How These Two Methodologies are Being Integrated in Industry	Journal Paper	User centred design, software engineering, software application, agile methods	Framework

4.2.2. Classification of studies based on known difficulties from the literature review

Related work in the literature has identified difficulties encountered when integrating UX in Agile. The table below depicts the critical information relating to the difficulties encountered, the identified process and the references that address this process.

Table 16: Classification of known difficulties for Agile UX Integration

Critical Information	Difficulty Identified	Author
Difficulties Encountered	Sprints are too short	Salah, Paige and Cairns (2015); Verdiesen (2014)
	Difficulties in prioritizing UX activities	(Miller 2005; Sy 2007; Singh 2008; Budwig, Jeong and Kelkar 2009)
	Difficulty in performing UX testing	(Kane 2003; McInerney and Maurer 2005; Chamberlain, Sharp and Maiden 2006; Kollmann 2008; Federoff and Courage 2009; Hussain, Slany and Holzinger 2009; Fox 2010)
	Customer versus User	(Ambler 2008; Kollmann 2008; Convertino and Frishberg 2017; Kieffer, Ghouti and Macq

		2017)
	Lack of documentation	(Näkki, Koskela and Pikkarainen 2011; Harbers, Detweiler and Neerincx 2015; Salah, Paige and Cairns 2015; Schön, Thomaschewski and Escalona 2017)

4.2.3. Classification of studies based on good practices from the literature review

Related work in the literature has also identified good practices for resolving the difficulties encountered when integrating UX in Agile. The table below depicts the critical information relating to the good practices, the identified process and the references that address this process.

Table 17: Classification of good practices for Agile UX Integration

Critical Information	Difficulty Identified	Good Practices Identified	Author
Good practices for resolving difficulties	Sprints are too short	Upfront design	Salah, Paige and Cairns (2015)
		UX work being desynchronised – UX work is one or two sprints ahead of development	Verdiesen (2014)
	Difficulties in prioritizing UX activities	Assigning the responsibility to the UX designer/UX practitioner	(Miller 2005; Sy 2007; Budwig, Jeong and Kelkar 2009)
		Separate UX product backlog	(Budwig, Jeong and Kelkar 2009)
		UScrum	Singh (2008)
		Separate UX team to prioritize UX activities	(Budwig, Jeong and Kelkar 2009)
	Difficulty in performing UX	Method of UX testing: • Using discount	• Fox, Sillito and

	testing	usability engineering techniques • Using the RITE method – identify and fix in the shortest possible time • Low fidelity prototyping	Maurer (2008) • Federoff and Courage (2009) • (Chamberlain, Sharp and Maiden 2006; Hussain, Slany and Holzinger 2009; Fox 2010)
		Scheduling of testing: UX testing done during Agile development tests.	Kane (2003)
		Access to users: • Planning in advance for user inclusion. • Use an existing user pool to act as development and design partner and conduct testing	• (Kollmann 2008) • (Federoff and Courage 2009)
		Shorter time to iterate design: Dedicate cycles for user feedback and incorporate into development life cycles	(McInerney and Maurer 2005)
	Customer versus User	• Separate the “chooser” from the “user”	• Convertino and Frishberg (2017) • (Ambler 2008; Kieffer, Ghouti

		• Involve real users to avoid biases • Use interaction designers to bridge gap between customer and development team	and Macq 2017) • Kollmann (2008)
	Lack of documentation	A range of artifacts can be used, with User stories being the most prevalent	(Näkki, Koskela and Pikkarainen 2011; Harbers, Detweiler and Neerincx 2015; Salah, Paige and Cairns 2015; Schön, Thomaschewski and Escalona 2017)

The objective of the literature review was to provide an overview and identify known difficulties when integrating UX into Agile and the good practices that can be applied in bridging the gap. The outcomes of the literature review and the data analysis of the results in the next sub-sections will be used to present new findings and practices and for suggesting recommendations. Some of the key inferences in terms of the challenges and recommendations that have been drawn from the literature review have influenced the design and execution of this study. The results have been analysed to create a generic framework for integrating UX into Agile, which follows in the next chapter. The results from the literature review have assisted in planning the content for the interviews and the questionnaires which form part of this study. The results can be compared and cross-verified between the two data collection methods (interviews and questionnaire) and the literature review.

4.3. Data Analysis of Interviews and Questionnaires

Data analysis using the grounded theory method is a continual process beginning with the results from the literature review. The approach to the coding process commenced with the open coding of initial concepts from the results from previous

studies and was guided further during the data collection in the interviews. The researcher used constant comparison of data to compare codes and emerging themes. As per Sbaraini *et al.* (2011) constant comparison of data allows the researcher to be more sensitive to patterns in the data if the researcher is involved from the beginning of data collection. The process followed in this sub-section is as follows:

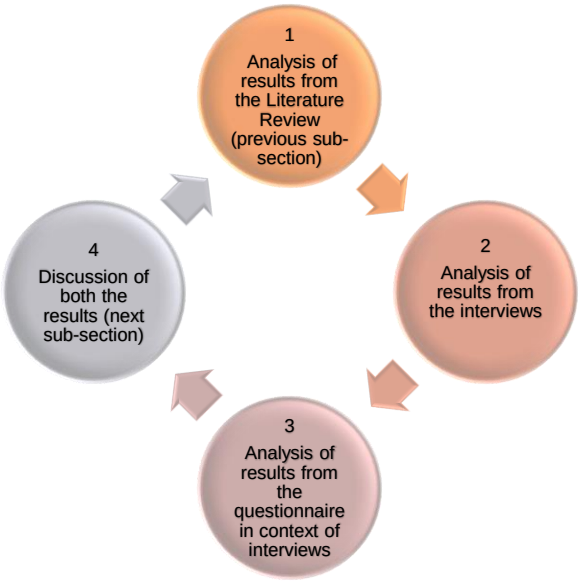


Figure 15: Data analysis process

4.3.1. Participants for the interviews

The researcher firstly conducted 10 semi-structured interviews in order to collect data. For the purpose of this study, the interviewees will be referred to as P1 (Participant 1) to P10 (Participant 10) and will appear as '*PN'* in an inline sentence and [PN] as a reference.

The table below shows the characteristics of all the participants.

Table 18: Characteristics of interview participants

Participant	Role(s)	Experience	Company Size	Duration of the Interview
P1	BA Manager	10 years	Large	59 minutes
P2	Scrum Master	4 years	Large	42 minutes
P3	Software Developer	9 years	Small	56 minutes
P4	Senior Software Development Manager	5 years	Medium	1 hour 40 minutes

P5	Agile Coach	7 years	Medium	1 hour 1 minute
P6	Quality Assurance Manager	6 years	Large	43 minutes
P7	UX Analyst	3 years	Large	55 minutes
P8	Scrum Master	6 years	Large	59 minutes
P9	Business Analyst	4 years	Large	42 minutes
P10	Back Office Team Lead	4 years	Medium	48 minutes

The semi-structured interviews were conducted either face-to-face or using Skype/Zoom. The interviewees were selected through networking with the following Agile/UX groups:

1. Durban Agile User Group

   This group is for individuals with a keen interest in Agile project and development practices and techniques. There are currently 25 active members. The group hosts monthly meeting which are centred on skills and experiences working in Agile projects (Durban Agile User Group 2018).
2. International Institute of Business Analysis – South Africa (IIBA-SA)

   This group is for individuals with a keen interest in Business Analysis, Agile Business Analysis, and Systems Analysis. There are currently 15 active members. The group hosts monthly meeting to build the South African analysis community by learning and sharing experiences with all attendees (Durban IIBA-SA 2018).
3. Interaction Design Foundation (IDF) Durban Group

   The IDF Group is for individuals who are involved in UX design and development, including aspects relating to branding, design, usability and function. There are currently 26 active members. The group is hosted on Linked In, where members share personal experiences in order to learn and connect. Their motto is “When the world becomes smaller, learning and connecting become easier.” (Interaction Design Foundation (IDF) Durban Group 2018: 1)

The interviews were conducted based on the interviewee’s availability and lasted between 42 minutes and 1 hour 40 minutes. The interview comprised of five predefined questions which were broad in nature. The purpose of these questions

was to explicate the good development practices of integrating UX in Agile as well as the challenges that were faced. Each of the five questions progressively resulted in detailed sub-questions being asked which were based on the response received from the interviewee. New questions were added to the interviews as the iterations progressed as a result of uncovering new perceptions or comments during the interviews. A full list of questions can be found in section 8.3 in the Appendix.

Interviews were carried out from July 2018 to October 2018. The data collection procedures required that initial contact be made by telephone or email before commencing. A letter was obtained from the Durban University of Technology to conduct the study and was emailed to the participants before commencing with the interviews. The letter contained the purpose of the study and provided assurance to the responders that confidentiality will be maintained during their participation in the study. The letter can be found in the section 8.5 in the Appendix. The appointments were confirmed according to the availability of the interviewees. Interviewees were briefed upfront in terms of the purpose of the research, the terms of confidentiality and were encouraged to reflect their stances both conceptually as well as anecdotally in terms of the work that they were involved in. It was also established upfront that this study will not disclose the name of the company that the interviewee was employed in or the project(s) at the company.

The interviews were recorded using a voice recording application on a mobile device and later transcribed verbatim by using a transcription tool, Express Scribe. Interviewees were made aware that the interview was being recorded and consented for this to be done. Express Scribe speeds up the transcription process considerably as it has large buttons (compared to a mobile phone) and allows the researcher to adjust the speed of the playback in order to capture the data. This tool was also used in previous studies conducted by Fox, Sillito and Maurer (2008) and Nadikattu (2016).

As per Sbaraini *et al.* (2011), grounded theory allows for discovering the phenomenon during the research process as the theory emerges from the research. It was therefore possible to add more questions or modify existing

questions as the interviews progressed. Some initial themes surfaced during the first five interviews. Hence, the researcher added clarifying questions or points to subsequent interviews in an effort to explore more on the topic. The researcher also conducted follow up interviews with two interviewees whereby the clarifying questions were posed.

Lawrence and Tar (2013) discussed the concept of saturation, where the researcher realizes that no new categories emerge from the code therefore no new theories can emerge. Once saturation is reached, the phenomenon or the theory that is being researched is said to be grounded in data (Lawrence and Tar 2013). Saturation was reached in this study after the 10th interview.
Once all the interviews were transcribed, the next step was to commence with the coding. The coding process of the grounded theory method comprises of three stages: *open coding*, *axial coding* and *selective coding*. The coding procedures are discussed in sub-section 4.3.3.

4.3.2. Participants for the questionnaires

The second method of data collection was the questionnaire. As mentioned in the previous chapter, a multi-method design was used to validate the results from the interviews. The questionnaire was sent to experts in the field of Agile UX after the interviews were concluded. The questionnaire also eliminated the possible mismatch between the sample and the target audience due to an inadequate sample size and allowed for a thorough analysis to support recommendations for the future.

The request for participants was posted in the Agile UX groups used for the interviews. The link to the questionnaire was sent via email to the participants. However, the questionnaire was not sent to any of the participants that were interviewed as the aim was to validate the results from the interviews by using new participants. The sample of the email sent to participants can be found in the section 8.6 in the Appendix. Participants were briefed upfront in terms of the purpose of the research, the terms of confidentiality and were encouraged to reflect their stances in terms of the Agile UX work that they were involved in. As

with the interviews, it was also established upfront that this study will not disclose the name of the company that the participants were employed in or the project(s) at their companies. Participants were also advised to not type any identifying marks in the questionnaire as participants are meant to be anonymous.

The questionnaire was compiled using QuestionPro, which is a survey tool (the full questionnaire can be found in the section 8.4 in the Appendix). The questionnaires were distributed on the 31st January 2019 and remained online for two weeks and within that period a total number of 34 responders filled out the survey. An overview of the responses is illustrated in the dashboard below:

Figure 16: Agile UX Questionnaire Responses

The dashboard above provides validity that only participants from South Africa responded to the questionnaire. It can also be confirmed that all responders completed the questionnaire as there were no drop outs.

The data analysis process of coding is described next.

4.3.3. Data Analysis using Grounded Theory

The computer-aided qualitative data management software, NVivo Version 11 (QSR International Pty Ltd 2017) was used to assist with the data management and analysis. Supporting results from the questionnaire are also presented in the form of charts.

4.3.3.1. Open Coding

As discussed in the previous chapter, the process of open coding involves analytical steps through which concepts are identified in the data (Lawrence and Tar 2013). This process involved identifying keywords in the transcribed data. Coding was undertaken as it was collected and allowed the researcher to start defining and categorizing the data.

Coding was initially done manually. The manual, line-by-line coding process assisted in constant comparison of previous interviews and creating links for memo writing and theorising. According to Charmaz and Belgrave (2007), the line-by-line process forces the researcher to look at the responses in a different light and also helps by taking the familiar, routine and mundane and make it unfamiliar and new. The manual process also enabled the researcher to get closer to the data and add or remove questions as the interviews progressed. For example, when P4 was asked how long typical iterations were, her answer was:

"Three weeks. We started out using two-week iterations and following strict scrum guidelines. However, from experience and due to our development architecture being highly complex (6 agile teams), we found that the amount of work done in the two-week sprint was so small that all we were doing was having sprint planning meetings and reviews without delivering enough to do a review" [P4].

The researcher found this quite interesting as this response related to the *"Sprints are too short"* difficulty that was addressed in the literature review. This led to the researcher asking future participants if sprints were too short and enough work was done in the sprint. This question proved valuable as it provided context in terms of the collaboration between team members and delivering a shippable product at the end of each sprint.

Codes and categories were sorted as per interview questions through the different iterations and conceptualisations. The codes were captured on NVivo. NVivo has been used by qualitative researchers to automate the process of sorting, matching and analysing data (Davidson and Jacobs 2008). In addition, the software can make connections in the data and visually represent the links in the data. For this study, NVivo allowed the researcher to highlight key words and tag them as codes

for analysis. In the case of new codes being generated, they were entered into the application and tagged to the relevant categories. The figure below shows the interface and some of the codes that were generated during the first iteration and second iteration.

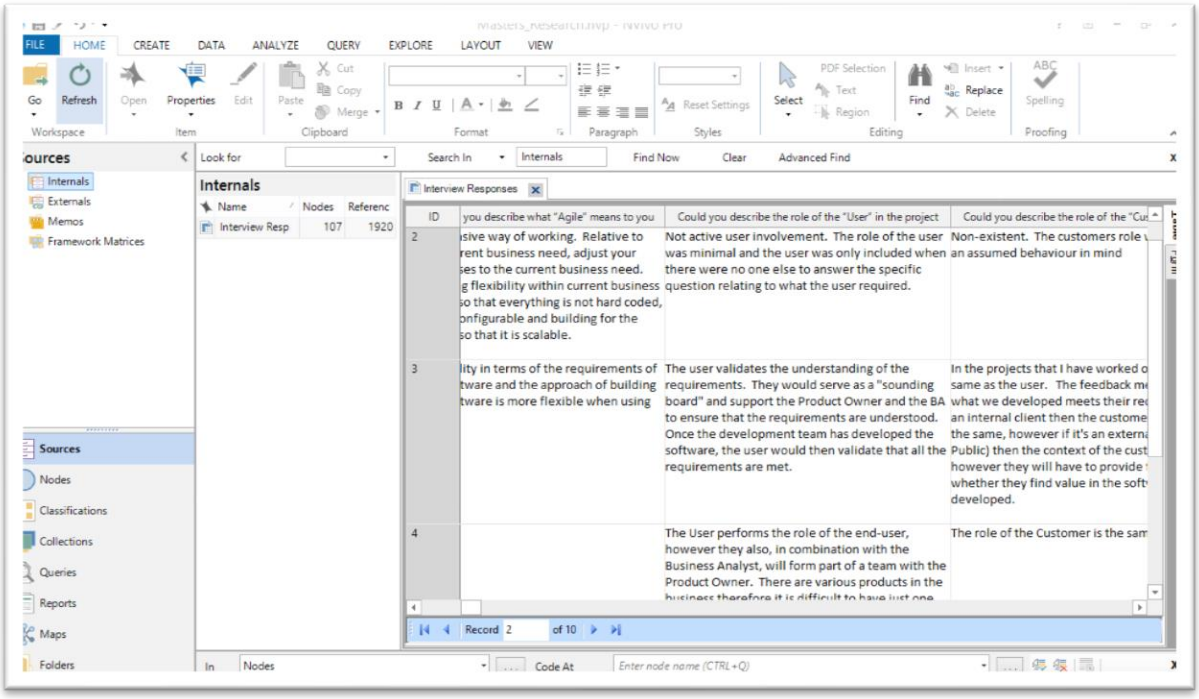


Figure 17: A view of NVivo open coding screen

NVivo allowed the researcher to create a Tree Map to identify and compare themes in the data. A view of the Tree Map is depicted below:

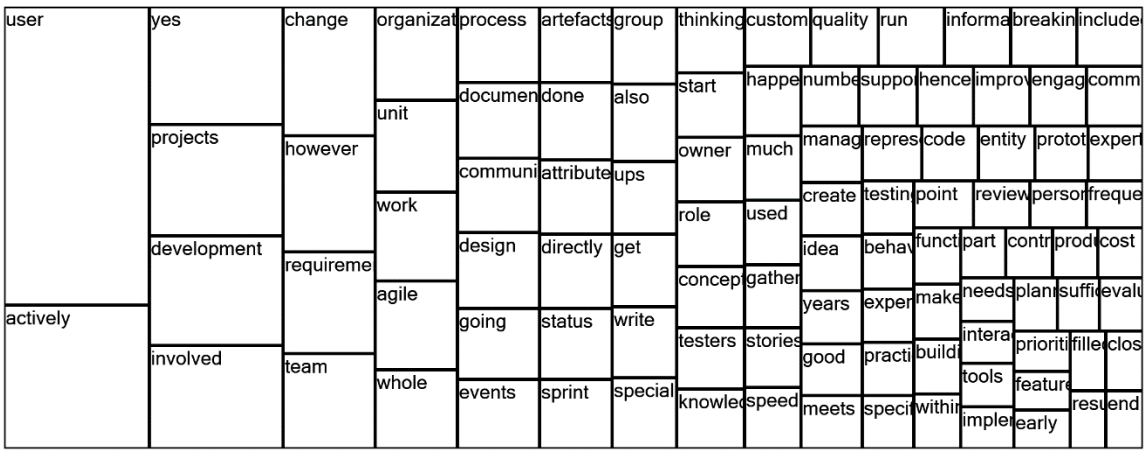


Figure 18: Tree Map from NVivo

After the interviews were concluded, the researcher launched the questionnaire. Data collected from the questionnaire formed part of the third iteration of codes and categories. This iterative process recognises the comparison of data and the emergence of themes that takes place between the participant and researcher which is inherent in the constructivist grounded theory method.

In total, 642 codes were created during the open coding stage. A sample of the open codes is shown below. The full list of codes can be found in section 8.7 in the Appendix.

Table 19: Open Codes

Questions from interviews	Open Code
UX specialist's involvement during development	No UX specialist involved
	BA fills in for UX specialist
	In the UK, the UX specialist was involved
	External clients – Yes, Internal clients – No.
	40% of projects – Yes, the balance – No.
	"Users, this is all we have time to develop, so use it or you have nothing else to use."
	Yes – for internal project, upfront with user research. External projects – very rare to have UX specialists.
	UX specialist tend to work very traditionally, instead of being Lean or Agile UX
	Too much analysis upfront
	Mostly for external projects – employ consultants/contractors
	40% of projects have UX specialists
	Yes – based abroad hence difficulty to schedule UX work.
	Yes – form part of the Experience Design Team
	UX has been included for the last 2 years
	No – defined by the user and the development team
	Yes – for the last 1 and half years
First phase of project	Requirements gathering
	Project prioritization
	Project kick-off
	Project lift off

	Sprint zero
	Concept phase
	Product vision
	Briefing phase
	Idea phase
What happens upfront?	Prioritize features
	Two sprints ahead for detail/requirements
	Solution design
	Initial engagement
	Requirements gathering
	User stories
	Epics
	High level requirements gathering
	Discussions between product owner, customer and execs followed by user requirements
	Breaking down the idea or product vision into requirements

4.3.3.2. Axial Coding

The next stage of coding is axial coding. Lawrence and Tar (2013) describe axial coding as the stage that open codes are grouped based on their categories and sub-categories. Relationships are identified in the data to form categories.

The researcher adopted an approach of highlighting relationships or categories in the data through constant analysis of the data in the interviews and questionnaires. The codes were captured and analysed using NVivo. This was an iterative process which involved comparing open codes and their similarities, as well as the frequency of use. NVivo automated the process as it has a word count feature. The researcher amended certain phrases if they had the same meaning to prevent duplication of codes. After two iterations of the open codes, the following categories were created:

- Roles and responsibilities – who is doing what?
- Participants understanding of Agile
- An Agile mind-set
- Getting involved, customer collaboration
- Sprint zero - opportunity to conduct user research
- Upfront design

- Losing the big picture
- Common Agile Methodologies used
- Common tools used - identify tools to integrate UX and Agile
- Good UX design methods
- Creating a UX vision
- Ambiguity of the user and customer role
- Existing and evolving development practices

An example of the coding for the abovementioned categories is shown in the table below. The full list of codes can be found in section 8.8 in the Appendix.

Table 20: Open codes that have been transformed into Categories during Axial Coding

Axial Code	**Open Codes**
Roles and responsibilities	BA Manager
	Scrum Master
	Project Manager
	Software Developer
	Senior Software Development Manager
	Agile Coach
	Quality Assurance Manager
	UX Analyst
	Business Analyst
	Back Office Team Lead
	Working experience
	Multiple projects
	Water-Scrum-Fall
	Agile
	Hybrid
Participants understanding of Agile	Framework
	A way of working to get better software
	Getting a better product out
	Getting products quicker to the user
	Making changes easier
	Creates team spirit
	Creates team morale
	Creates a self-organised team
	Safe environment for teams to communicate
	Shared knowledge

	Collaboration
	Responsive way of working
	Flexibility within current business needs
	Scalability
	Comply in terms of the Agile Manifesto
	“Agile is something we are, Scrum is something we do.”
	Adjust your direction without costing a fortune
	“Turning a dime for a dime.”
	Breaking down bigger projects into manageable chunks
	Estimating so that project remains on track
	Methodology with various frameworks
	Realise value sooner rather than later
	Taking feedback and using it to steer the project forward
	Iterative
	Learning
	Short, sharp deployment in the quickest possible time

It is worth noting that some of the codes appeared in more than one category. For example, the responses to the questions “Could you describe the role of the “User” in the project?” appeared in the categories “Ambiguity of the user and customer role” and “Roles and responsibilities”. The categories were grouped into themes during selective coding, which is discussed next.

4.3.3.3. Selective Coding

Selective coding is a process which involved grouping the core categories from the axial coding stage to create high level concepts that form the themes, or the big picture of the research. This process is more directive, selective and conceptual than that of the initial open coding and axial coding process. Selective coding was used to synthesise larger segments of data and identify the most significant and frequent codes (Charmaz and Belgrave 2007).

The researcher adopted a process of moving through the data collected in the interviews and questionnaires and compared the participants’ experiences, actions and their interpretations of what was happening in their organizations. For example, in relation to the code “fitting the user into Agile”, the researcher looked at all the sources of data to see how each participant talked about the user’s

involvement in the projects. The researcher looked at the reasons for the user not being included; the effects of the user not being involved; the ways that organizations have successfully included the user from the beginning of the project (for those that have included them); the supportive strategies involved to actively respond to user feedback. The researcher went on to develop the first selective code '*Agile UX Integration Practices'* as a category. A category is a theme that is interpreted based on the ideas, events or processes explicated in the data by making sense of what the participants said (Charmaz and Belgrave 2007: 91). This iterative process of comparing data collected to the codes identified, resulted in two themes being developed. The themes are shown in the table below and are discussed in detail in the next section.

Table 21: Axial codes that have been transformed into Themes during Selective Coding

Theme	**Axial Coding**
Agile UX Integration Practices	Creating a UX vision
	Getting involved
	Including users throughout the project
	Losing the big picture
	Existing and evolving development practices
	An Agile mind-set
	Identify artifacts to integrate UX and Agile
	Identify tools to integrate UX and Agile
	Opportunity to conduct user research
	Common Agile Methodologies uses
	Common artifacts used
	Common tools used
	Good UX design methods
	Sprint zero - opportunity to conduct user research
	Upfront design
UX Vision	Ambiguity of the user and customer role
	Customer collaboration
	Fitting the User into Agile
	Participants understanding of Agile
	Roles and responsibilities

A graphical representation of the some of the keywords identified in NVivo during the selective coding phase is depicted in the figure below:

Figure 19: Word Cloud from Selective Coding on NVivo

4.4. Results from interviews and questionnaires

This section describes the results, main themes and concepts that emerged from the analysis of the interviews and questionnaires. The relevant quotations from the participants from the interviews are presented and, where applicable, a graphical representation of data in the form of charts from the questionnaires is presented.

As explained in the previous chapter, the data collected in the interviews and questionnaires were analysed using the grounded theory method. After the literature review (Chapter Two) and upon completion of the fieldwork, the knowledge areas were distilled by using coding, categorising and constant comparison of data into the following set:

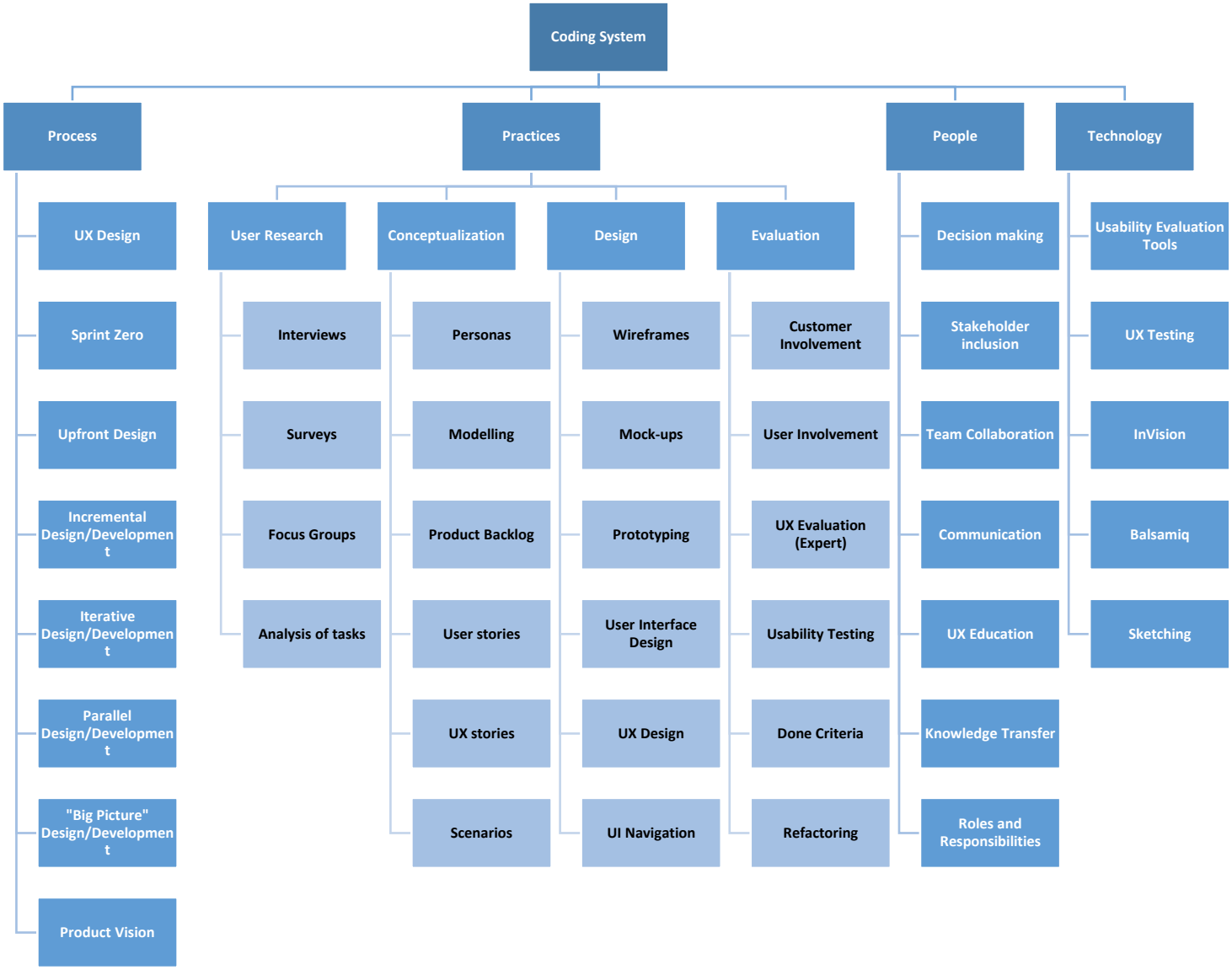


Figure 20: Coding System for Themes

The list was not designed to represent an exhaustive set of Agile UX knowledge areas. However, the researcher's best efforts have resulted in this list of knowledge areas and these will be expanded into the main themes in the next sub-sections. The presentation of the results will be showcased as follows:

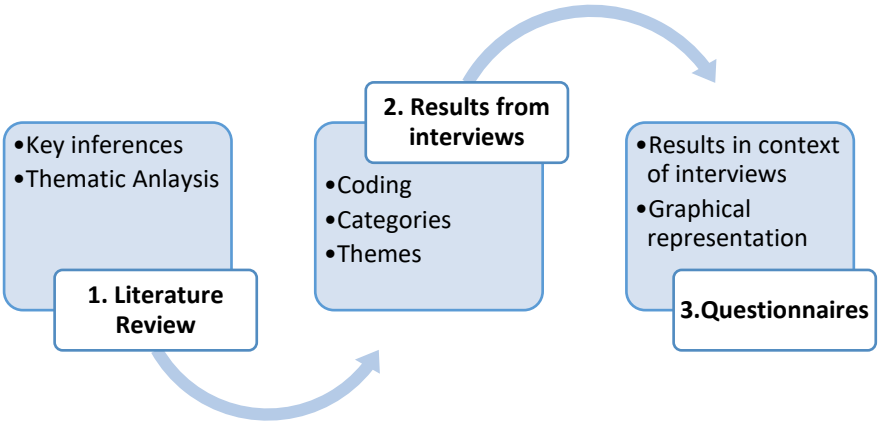


Figure 21: Presentation of results

4.4.1. Theme 1: Agile UX Integration Practices

Amongst Agile team members, there has been an increasing interest in the integration of UX in Agile software development. The first theme that emerged related to ***Agile UX integration***.

The following categories and responses were uncovered in the interviews and questionnaires in relation to the following research sub-questions:

1. "What Agile development techniques are best suited to ensure UX requirements are met?"
3. "What are the roles of team members that are directly involved in the process of integrating UX in Agile?"
4. "What are the problems when integrating UX in Agile?"

4.4.1.1. Participants understanding of Agile

Firstly, it was pertinent to understand what "Agile" meant to the participants. The question *"Could you describe what "Agile" means to you?"* was posed to the participants in order to yield results in terms of the meaning of Agile to the participants.

As explained in Chapter 2, **Agile is a framework, a way of thinking and a way of working**. The participants explicitly explained this interpretation and what it meant to them:

"A framework. A way of working in order to get better software products. I am an Agilist. I do see benefits in running projects in an Agile manner, getting a better product out, getting it quicker to the users and making changes easier, from a team perspective Agile creates a team spirit, team morale, and created a self-organised team." [P1]

"First and foremost a set of values, the Agile Manifesto allows you to comply with these values. Agile is something we are, Scrum is something we do." [P4]

"The word Agile in itself is too heavy. Agile is a methodology, and within this methodology there are various frameworks that support this methodology, example Scrum, Kanban, XP, Lean. It's an adaptation of the methodology, a way of working." [P8]

Agile has many benefits as **opposed to the Waterfall methodology**, as stated explicitly by P1:
"From a team perspective Agile creates a team spirit, team morale, and created a self-organised team. Saw it first-hand when a developer had difficulty communicating in a Waterfall project. However, she flourished working in Agile, as it was a safe environment to communicate and voice her opinion. Shared knowledge, collaboration." [P1]

Based on the responses received from the participants, they had a fairly good knowledge on the principles of Agile and four main values which were outlined in the literature review.

4.4.1.2. Adapting to change

Majority of the participants had more than three years' experience with Agile and a good understanding of the methodology and the ideas behind it. The participants

also pointed out that in order for Agile to work the team must commit to the idea that they must be **flexible and adapt to change** as it affects their day to day work:

"Responsive way of working. Relative to the current business need, adjust your processes to the current business need. Creating flexibility within current business needs so that everything is not hard coded, more configurable and building for the future so that it is scalable." [P2]

"Flexibility in terms of the requirements of the software and the approach of building the software is more flexible when using Agile." [P3]

"Agile to me is really a way of being responsive to circumstances. It is about the ability to adjust your direction quite quickly without it costing you a fortune. Turning on a dime for a dime." [P5]

"Fast to react to change, realising value sooner rather than later and taking feedback and using it to steer the project forward." [P9]

4.4.1.3. Methodology used

Almost all the participants had a fairly good knowledge of what Agile meant to them and how they participate in the process of working in an Agile context. Looking closely at the data as per the table below it can be concluded that Scrum is the most widely used methodology, with all companies using it. Only one company used Kanban as well as Scrum and according to P4, Kanban was used specifically for the production team. P4 elaborated that Kanban was used merely to visualise and improve the flow of work rather than defining processes, roles, artifacts and ceremonies which are synonymous with Scrum.

Table 22: Participants Overview

	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**
Project Type	Retail, Mobile	Retail, Mobile	Web	Banking	I.T.	Retail, Mobile	Payroll	Retail, Mobile	Gaming, Banking	Web, Banking
Agile Method	Scrum	Scrum	Scrum	Scrum	Scrum, Kanban	Scrum	Scrum	Scrum	Scrum	Scrum
Organisation Size	Large	Large	Small	Medium	Medium	Large	Large	Large	Large	Medium

The chosen quotes illustrate the participants' understanding of why Scrum and the fundamentals of Agile work for them:

"We are absolutely sold on using Agile. Scrum and Kanban speak to some of the processes and artifacts that allow the teams the flexibility to work in an Agile manner." [P4]

"It allows for breaking up of large projects into smaller, manageable chunks. Testing is also easier when you are working with a smaller component of a bigger system." [P5]

"More deliverables in a shorter timeframe. What makes it even more successful is if it is implemented across the business (not only in software development, but in HR, Finance etc.)." [P8]

"It results in more deliverables and if it is implemented correctly, it is a successful development process." [P9]

Some participants pointed out that Agile is not suitable for every project, and they also indicated that Agile was not implemented correctly:

"The implementation of most Agile approaches is flawed." [P5]

"Everyone tries to adopt the Agile approach. However, it needs to be implemented correctly to work. UX also needs to be lean to work within an Agile team." [P7]

"Agile has its merits. However, it is dependent on the type of projects and the type of organization that it is implemented in." [P10]

4.4.1.4. Roles and responsibilities

Most of the companies in this study had highly cross-functional teams with multidisciplinary skills. As mentioned by P9: *"We have cross functional team who*

are aware of what everyone is doing which assists in creating a better product." [P9]

There is an appreciation of work conducted by UX specialists and developers. However, a significant result from the study, as depicted in the table below, is that several companies still did not have a UX person/role in their team, hence there is limited usability evaluation and user research. Only 30% of the organizations from the interviews said that they used a UX person in their projects.

Table 23: UX Role per Company for interviews

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Project Type	Retail, Mobile	Retail, Mobile	Web	Banking	I.T.	Retail, Mobile	Payroll	Retail, Mobile	Gaming, Banking	Web, Banking
UX role present	No	No	Yes	No	Yes	No	No	No	Yes	No

The results from the interviews, guided by the literature, confirmed the above theme and problems. The questionnaire was finalised using this theme in order to investigate results from a wider group of participants. The results from the questionnaire confirmed and validated results from the interviews that there was a lack of a UX person/role in their team. 38% of the responders indicated that there was no UX person present in the projects that they worked on.

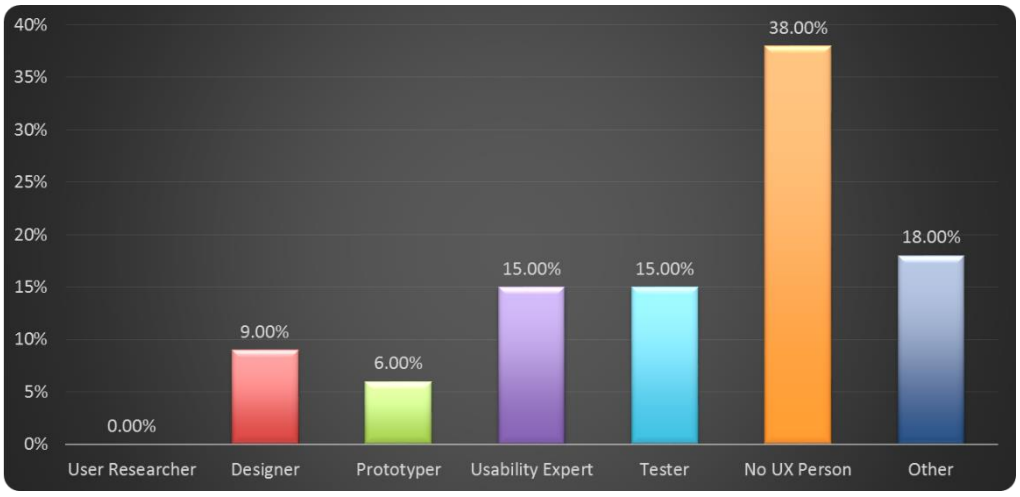


Figure 22: UX Roles

The following responses were uncovered in the interviews in relation to RQ3 "What are the roles of team members that are directly involved in the process of integrating UX in Agile?"

The participants' understanding of Agile is the basis of identifying the roles and responsibilities that the UX person/s will take on. Most of the participants mentioned that the UX design was produced upfront and handed over for development. However, Agile emphasises *customer collaboration over contract negotiation.* The next section looks at the pertinent roles and responsibilities that were identified in the research.

Individuals and Interactions

The roles that the participants have taken on in their respective teams and the responsibilities that they have carried out, often led to roles being blended. For example, P5 sums up her role as follows:

"I am an Agile Coach. Teams using Scrum or Kanban. Help the teams work in an Agile way. I work with Scrum Masters, Product Owners, Software Developers, Testers, Business Analysts, UX Analyst, UI Design Experts." [P5]

P7 explains that her role was to identify and focus on information visualization. She also explained that there is minimal input from the user:

"I am a UX Analyst. My UX role focuses on information visualization around Business Intelligence such as dashboards, user interfaces, interactions etc. The user is not involved enough in the project and they are involved at the end of the project." [P7]

Taking the above into consideration, it can be ascertained that UX design should be a communicative role. Typically, this is documented in Waterfall projects. However, Agile advocates *working software over comprehensive documentation.* Interaction between the user and the UX team reduces documentation. Participants in this study advised that instead of user specifications, tools such as screen mock ups, sketches, wireframes or prototypes were used to support the interaction between the user and the UX team. The following quote illustrates the interaction between the user and the development team:

"Users and the agile team are regularly meeting and looking at what has been developed to ensure that what is being developed meets users' requirements. On average the users meet with the team every three to four days. Teams have

learnt that meeting more often with users ensures that development remains on track." [P4]

The study showed that there is a need for more interactions between the user and the Agile UX team. P6 advised that *"external users' needs are not included in the specifications. However, internal users' needs are included."* P3 advocated that *"as a developer I am not sure whether it is the actual end-user's needs that are included in the specifications or a representative of the user as I do not have sight of this."*

The trend in more than one company was that users were included too late in the project. P7 claimed that the users were included *"only at the end when it is too late"* and P9 also concurred that users were included too late. He claimed that *"indirectly users are involved via feedback. However, this is too late."*

End-user Involvement

The study also showed that 80% of the participants in the interviews felt that there was not enough end-user involvement. P3 claimed that there is *"not even enough user feedback. Whenever a sprint has been completed and demoed, a user representative would be present at the demo and not the end-user. The end-user gets involved very late."*

P5 claims that "there is never enough end-user involvement. From the initial requirements phase, there is not enough end-user involvement. There has certainly been an improvement in the last ten years. However, it is a very slow process."

As depicted below, results from the questionnaire shows that 68% of responders strongly agreed that in order to create the best User Experience, input and involvement is required from as many different users as possible.

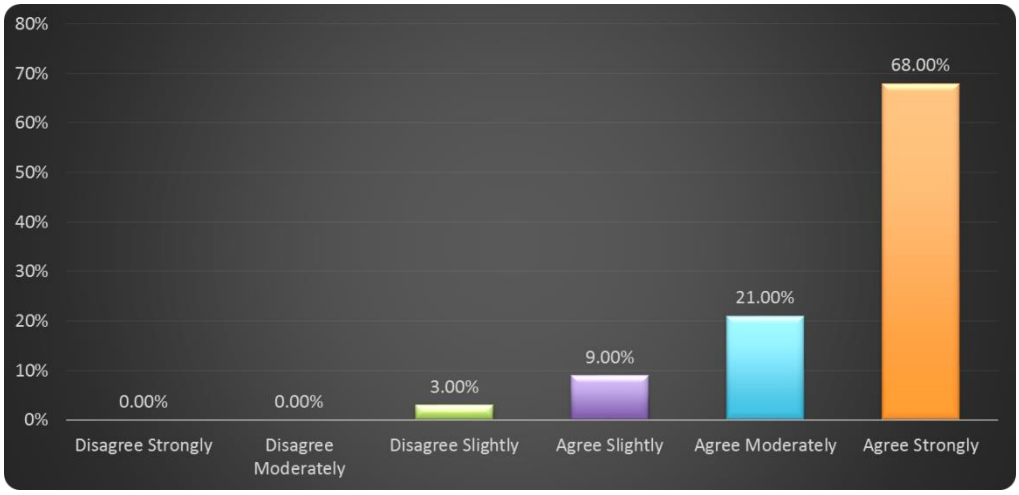


Figure 23: User involvement in relation to UX

4.4.1.5. Difficulties faced when integrating UX in Agile

An excerpt of the codes, as per table below, was extracted from the interview responses from the participants (full list of codes can be found in section 8.7 in the Appendix). The category chosen was "*Difficulties faced when integrating UX in Agile*" as the recurring codes were focused around the challenges that Agile and UX practitioners experienced when integrating UX in Agile.

Table 24: Codes for difficulties faced when integrating UX in Agile

Category	Interview Question	Codes
Difficulties faced when integrating UX in Agile	Is there any resistance from other team member with UX persons	From product owner occasionally
		If UX persons requirements are not technically feasible
		Unrealistic expectation
		Occasionally
		"They want a rocket ship when all they need is a bicycle."
		Developers
		UX person inhibits their creativity
		Testers
		Do not understand what UX is
		Educate the organisation
		Specialised skill
		Good suggestions from the UX persons
		Discipline in creating standards around UX
		Do not understand UX
	Are there issues up front with integrating UX processes?	Lack of understanding
		Resistance from upper management
		More accepting

		Buy in from business
		Filled in by someone else in the team
		Lack of resources
		Very few UX specialists in the organization
		Lack of resources

Accessibility to users

P6 states that "…the difficulty lies with accessibility - users are not always accessible." P9 is part of an international company and claimed that "accessibility to users is difficult, due to the global footprint of the organization."

When asked the question "*Do you use real users in your UCD testing*?" 40% of the participants in the interviews advised that they did not use real users. In most cases a representative of the user is involved in the testing. As stated by P5 "*there is usually a representative or proxy who fills in for the user which is the Product Owner. It is fraught with danger. I would say 10 to 15% of external projects engage early with users."*

The lack of accessibility to the users has a detrimental and costly effect on projects, as pointed out by P7 "the users get involved too late in the process which requires us to go back and do some costly development. Through feedback from the user, we discover that what we have developed is not up to standards or not what the user requires which requires us to go back and re-develop."

In some cases, the only time the user sees the product is during demos or during testing:

"The first time the user saw the application was during testing or when the product was being demoed to them." [P2]

Only one company had access to their users, whereby the users meet with the development team every three to four days:

"We are fortunate as the users are in the same building therefore they are accessible. Users and the Agile team are regularly meeting and looking at what has been developed to ensure that what is being developed meets users' requirements…" [P4]

Slows down Agile process

In response to the question "*Is there enough end-user involvement?*" it was ascertained that quite a few participants felt that involving the user, slows down the Agile process.

As per P6 "in most instances too much of user involvement complicates and slows down the process. Users themselves don't know what they want. The input that we receive from them are sufficient."

P7 is also in agreement and states that "business feels that involving users slows the entire process down and time is money. However, we lose money when we have to re-develop as the software does not meet users' needs."

P10 explains that "time is an important factor; the user base is quite wide therefore a lot of time will be wasted by involving too many users."

P4 advocates that too much of user involvement also slows down development: "Having too much of user involvement can also be detrimental to the project as too much user involvement slows down development and the Agile process in general."

Difficulty in prioritizing UX activities

The codes are per table below were extracted from the responses from the participants. The appropriate sub-category chosen was "Difficulty in prioritizing UX activities" which was also one of the known difficulties inferred from the literature.

Table 25: Open codes for difficulties in prioritizing UX activities

Category	Interview Question	Codes
Difficulty in prioritizing UX activities	How are UX requirements handled in a project? Are there any artifacts or deliverables that are required (e.g. UI specifications)	No UI specifications
		More branding guidelines
		Personas
		Representative of the user
		No clearly defined UX activity
		Assumption of what the user would need
		Most of the work done upfront
		UI design and the interaction
		External users' needs are not included
		Based on assumption
		SME's

The issue of prioritizing UX activities was common amongst majority of the participants. As P7 stated, “we have some specifications documentations, which have style guides and certain standards that developers have to work with. However, these are not UX requirements documents.” The same sentiments were shared by P1, P2, P5 and P8. P6 who is part of the testing team could not comment as he is not part of the process.

Another issue picked up was that 40% of the participants advised that the UX activities did not follow an iterative process. The design was done upfront, or in some cases the UX activities formed part of the development process:
“It's not a formal activity and forms part of the development process, and it is not optimal.” [P4]

Looking closely at the results, 70% of the participants accounted for upfront UX work; hence the UX requirements get passed to development in the first sprint. The idea of upfront design has been identified as one of the good practices in the literature by Salah, Paige and Cairns (2015) and is covered in the second theme of this study under the category “UX design techniques”. The caveat with upfront design is that the design focuses more on the visuals than the functionality. P3 states that “from experience, upfront UX work focuses more on the visuals rather than the functionality. As long as the screen looks pretty, that’s enough.” This sentiment is shared by P10 who explained that “you could have a perfect UI. However, if the coding or API is poorly designed then it will not address the whole user experience.”

4.4.1.6. Customer versus User

The differentiation and ambiguity between the role of the customer and the user was mentioned in the literature review. In this study, the involvement of the user in the UX design ranged from an *informative* role (the user provided user requirements) to a *consultative* role (the user was consulted to comment on predefined solutions) to a *participative* role (the user was actively involved in the design of the solution).

The codes, as per table below, were extracted from the responses from the participants. The appropriate category chosen was “Customer versus User” which was congruent with what was identified in previous studies in the literature.

Table 26: Open Codes for Customer versus User

Category	Interview Question	Codes
Ambiguity of the user and customer role	Could you describe the role of the “User” in the project?	No active user involvement
		Minimal
		“As needed” basis
		Validates the understanding of requirements
		“Sounding board”
		Supportive role to Product Owner and the BA
		Validate software that has been developed
		End-user
		Product Owner team
		Upfront involvement for requirements elicitation
		Testing before sprint review
		Sprint review
		Feedback on mock-ups
		Late involvement
		Less engagement with users in external clients
		Alpha testing and feedback
		Involved at the end of the project
		Not enough user research
		User is the product owner
		Non-existent
		Proxy as a representative of the user (e.g. BA or Product Owner)
		Uses the end-product
		Finds value in what is created for them
Ambiguity of the user and customer role	Could you describe the role of the “Customer” in the project?	Sponsor
		Same as the user
		Non-existent
		Designed with an assumed behaviour in mind

		Feedback mechanism
		Internal clients – same as the user. External client – varies
		External Client – same as user
		Proxy to the user

As confirmed in the literature, the customer's role varied based on the nature of the project (Martin, Biddle and Noble 2004). Chamberlain, Sharp and Maiden (2006) opine that design and collaboration does not include user involvement. Hence, there is uncertainty as to whether Agile focuses on the user.
This study provides evidence that there is a disparity in terms of who the "user" is and who the "customer" is.

Role of the User

The initial question posed to the participants was "*Could you describe the role of the 'User' in the project*" which garnered the following responses:
P4 conceded that "the user performs the role of the end-user. However, they also, in combination with the Business Analyst, will form part of a team with the Product Owner."
A few participants revealed that the user was used for validation rather that active involvement in the project:
"The user validates the understanding of the requirements. They would serve as a "sounding board" and support the Product Owner and the BA to ensure that the requirements are understood." [P3]

A few participants commented that the role of the user was minimal, or in certain cases, non-existent:
"Has always been a representative of the end-user…a senior person who represents the end-user…in most cases the Product Owner." [P1]
"No active user involvement. The role of the user was minimal, and the user was only included when there was no one else to answer the specific question relating to what the user required." [P2]

“The user is not involved enough in the project, and they are involved at the end of the project. We don’t do enough in terms of empathizing with what the user requires, and we don’t do enough user research.” [P7]
“Non-existent. We use a proxy as a representative of the user.” [P9]

Only two participants confirmed that the user is actively involved in the project:
“Typically, our users are actively involved. They are involved upfront in the requirements from backlog refinement and sprint planning, so they can provide input. They will also be brought in before sprint review to play around with the functionality. They are also involved in the sprint review.” [P5]
“The user is the product owner and they work closely with the team.” [P8]

Role of the Customer

The follow up question posed to the participants was “*Could you describe the role of the ‘Customer’ in the project.*” Majority of the participants affirmed that the customer role was the same as the user (90% of the responses). Only one of the participants, P10, revealed that “the Customer is the sponsor for the project.”

The overlap and ambiguity of the *customer* and *user* role was identified in Chapter Two and has been confirmed during this study. As evidenced by authors such as Martin, Biddle and Noble (2004) in the literature review, the customer has often been described as the person who pays for the product being developed and the user is the person that uses the system on a daily basis. However, one must be cognisant of the fact that in today’s world of mobile computing and e-commerce, the customer and the user can be the same person, and in most cases, it is the same person.

Another area of concern is the non-existence of the user in the project or designing with an assumed behaviour of the user in mind.
P10 endorses that “the end-user is the person that ultimately uses the end product and finds value in what is created for them.” Besides being involved in user requirements and user research, the user should sit with the team, write user stories, prioritise user stories and test the software as it is being delivered to ensure that the users’ requirements are met.

4.4.1.7. Role of documentation

The attitude with regards to documentation varied amongst participants. In general, the interviews showed that a series of attempts to support light weight documentation resulted in contrasting outcomes from participants. The following quotes illustrate this:

“Some sort of documentation is required for scope and sign off. Light weight documents are better. However, by the end of the project you will end up with a lot of documentation.” [P1]

“Most of the teams that I work on have sufficient documentation. We decide on the type of the project and what documentation is required to ensure that we cater accordingly.” [P5]

As exemplified by the question asked to participants: “*Please elaborate on how the lack of documentation has affected the project”*, the lack of documentation has resulted in issues with maintenance of the software.

“Working software always takes precedence over comprehensive documentation. Maintenance costs are high due to the lack of documentation. New developers who were not involved in the project do not understand the code and the functionality of the software, which ultimately results in delays.” [P3]

“Developers add comments in the tickets, based on changes to original specification. However, no one goes back and updates the original documentation. This results in issues with maintenance and complicates future development as documentation does not match the end product. The developer would have to physically go into the code/UI to see what was developed.” [P4]

P7 also addressed the issue of misalignment between what is documented and what is developed:

“It's not really the lack of documentation, but rather the lack of specifications. The specifications are based on assumptions of what the user requires. There is misalignment between what is being developed and what is delivered.” [P7]

4.4.2. Theme 2: UX Vision

The second theme that emerged from the research conducted in this study related to the UX vision.

There has been an increasing interest in understanding the end-user and their needs in order to design a good UX amongst Agile and UX team members. The UX vision can be conceptualized as the experience that the user should have when using the software product that the Agile and UX team have developed.

The following responses were uncovered in the interviews and questionnaires in an attempt to answer research sub-question 2: "*What are good development practices for creating good UX?"*

4.4.2.1. Fitting UX into Agile

It is important for the members of an Agile team to adopt a UX mind-set and vice versa for UX practitioners, as this will enable them to understand how and where UX can fit in. The participants were aware of the challenges. However, none of them were negative about integrating UX into Agile. Almost all the participants mentioned that Agile was a successful approach and it was flexible enough to fit UX into the iterations, thus making UX more Agile in itself. The chosen quotes below illustrate the participants' perception on UX fitting into Agile:

"The designer being involved made it effective in the one project and made a difference and an impact." [P3]

"Comparison between teams that do have a UX specialist and those that do not shows a vast difference in the user experience of the end product, hence UX is effective." [P5]

"From the testing perspective, it does make it easier as there are fewer defects and speeds up the process of testing and releasing." [P6]

"From a user perspective it is something tangible that they review and provide feedback on before the product is deliverable. A wireframe, a piece of paper and

pen is sometimes useful as the user gets to see what will be delivered to them." [P10]

As depicted below, results from the questionnaire also confirms that UX insights should be encouraged in order to improve the quality of the software. 65% of the responders have indicated that they strongly agree that UX insights during development vastly improve the quality of the software.

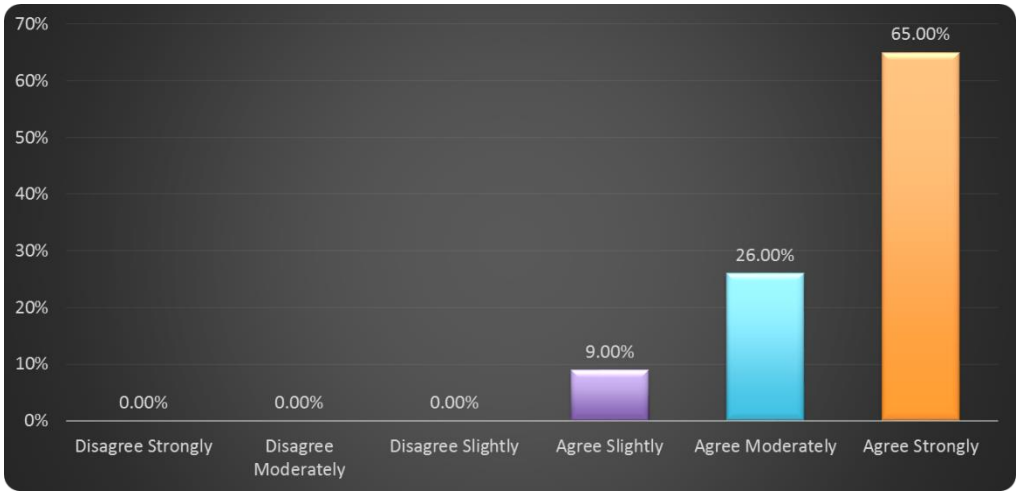


Figure 24: UX insights during development

A few participants pointed out that integrating UX into the Agile process enhanced the quality and usability of the product and increased the satisfaction of the users. Some also mentioned that the integration added value to the team and the entire Agile process:

"Agile and UX fit well together as it results in better user experience when it is given the priority it requires." [P2]

"The whole purpose of Agile development is to get software into the hands of the users are quickly as possible. The whole purpose of UX itself is to ensure that what is delivered to the user is something that can be used. Agile also ensures that waste is eliminated early." [P3]

"The UX specialist can form part of the team and the conversation where the user, the developer, the BA can gather requirements and it will be a much-refined process." [P4]

“There will be bugs and fixes, Agile allows you to stagger the processes and detect feedback earlier and it is cheaper to fix bugs and defects if it is early in the process.” [P6]

“They both follow similar cycle. With Scrum, you design, you develop, you test and evaluate. Similarly, in UX you design, you develop the prototype, you test and you evaluate. These two cycles work so well. Hence, if they are integrated together it results in a better product and better User Experience.” [P7]

“By integrating UX and Agile, you get to understand what the user requires, empathize with the user, do paper sketching prototypes and make sure the product meets the users’ requirements.” [P9]

As depicted below, results from the questionnaire also confirm that close collaboration between UX and Agile team members assist in defining clear requirements.

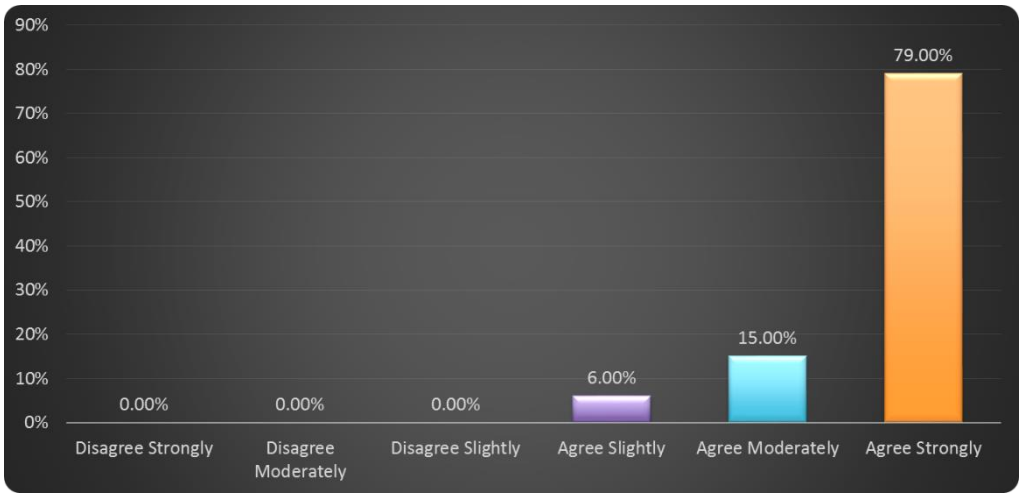


Figure 25: Close Collaboration between UX and Agile team members

4.4.2.2. UX design techniques, tools and methods

In order for UX to be integrated successfully into Agile, there needs to be a compromise between teams. Some of the UX design techniques discovered in this study are described next.

Upfront design (Sprint Zero)

As outlined in previous studies, the lack of time during sprints have been addressed by upfront design (Verdiesen 2014; Salah, Paige and Cairns 2015). Majority of the participants affirmed that UX designs were done upfront. Agile practitioners need to be cognisant of the fact that a portion of resources need to be allocated upfront for UX activities. The question posed to the participants was "*how much of time do you spend on UX before development begins?"* and 80% of the responses were that the first sprint is used to finalize UX work. As per P1 "*we spend at least one sprint (ramp up sprint or sprint zero) on UX work, depending on the product."* Some of the participants argue that one sprint is insufficient to conduct in-depth UX work. P7 states that *"from my experience the first phase should be user research. We have a team called the Voice of the Customer. However, this team is not directly working with the customer and usually comprises of a customer representative."*

It is worth noting that in order for upfront designs to work within Agile, UX work needs to be lean instead of being done traditionally (Waterfall). As advocated by P5, *"what I have found is that UX specialist tend to work very traditionally, instead of being Lean or Agile UX. They want to do all the analysis upfront; they want to do the design and testing, high fidelity prototype instead of low fidelity prototype. The traditional way of thinking needs to be adjusted in order for UX to work with Agile."*

The length of the sprint, which in most cases was two weeks, can be too short especially if there is a lack of UX resources and if users are not easily accessible. For both P9 and P10, lack of UX specialists posed as a problem:

"There is a lack of resources as there are very few UX specialists in the organization." [P9]

"We are in a smaller team therefore lack of resources can be an issue. There are also very few UX specialists in the organization." [P10]

All participants agreed that in order for sprint zero to work successfully, the target users, business goals and team roles needs to be clear before sprint zero. They also agreed that sprint zero was an opportunity to conduct user research and to create the product vision.

Some of the pertinent codes discovered in the axial phase of coding are presented in the table below:

Table 27: Axial Code for Upfront Design

Category	Axial Code	Open Codes
Upfront Design (Sprint Zero)	Sprint zero - opportunity to conduct user research, getting involved, customer collaboration	Requirements gathering
		Project prioritization
		Project kick-off
		Project lift off
		Sprint zero
		Concept phase
		Product vision
		Briefing phase
		Idea phase

<u>One sprint ahead</u>

Similar to approaches identified in the literature review, e.g. Verdiesen (2014), a few participants highlighted that the UX work was done a sprint ahead of the current development sprint. Working a sprint ahead of development reduced the integration of UX features into the current sprint as participants advised that the challenge their teams faced was due to frequently changing requirement.

As depicted below, results from the questionnaire show that majority of the responders (67%) agree moderately that the UX team should work at least one sprint ahead of the development sprint as the developers are aware of design progression and feedback can be given early:

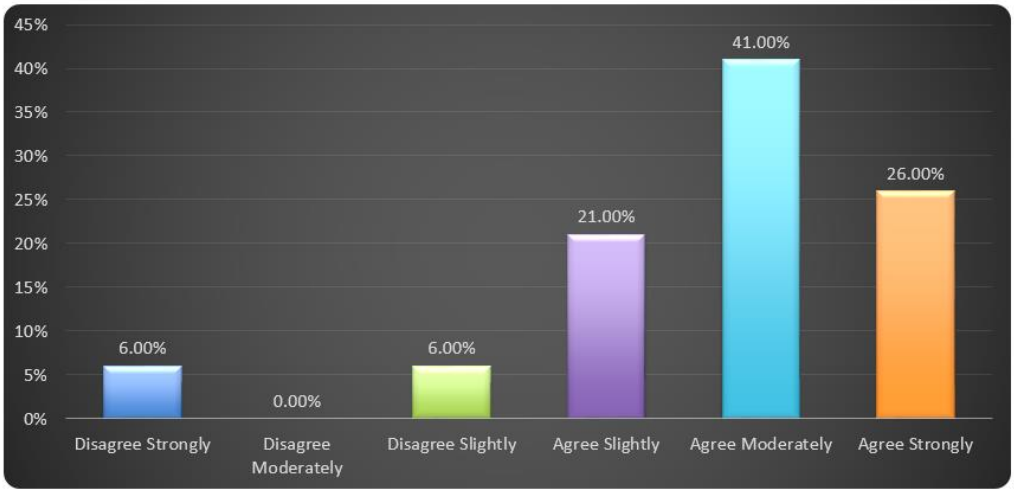


Figure 26: Working one sprint ahead

Two participants (P4 and P9) advised that the UX work was done two sprints ahead of the current development sprint. The reason for working two sprints, as

per P9 was due to “a lack of resources as there are very few UX specialists in the organization.” P9 goes on to explain that the lack of resources makes it difficult to conduct user research in one sprint therefore they worked two sprints ahead.

As depicted below, results from the questionnaire shows an upward trend, with majority of the responders (26%) agreeing strongly that the UX team should work at least two sprints ahead of the development:

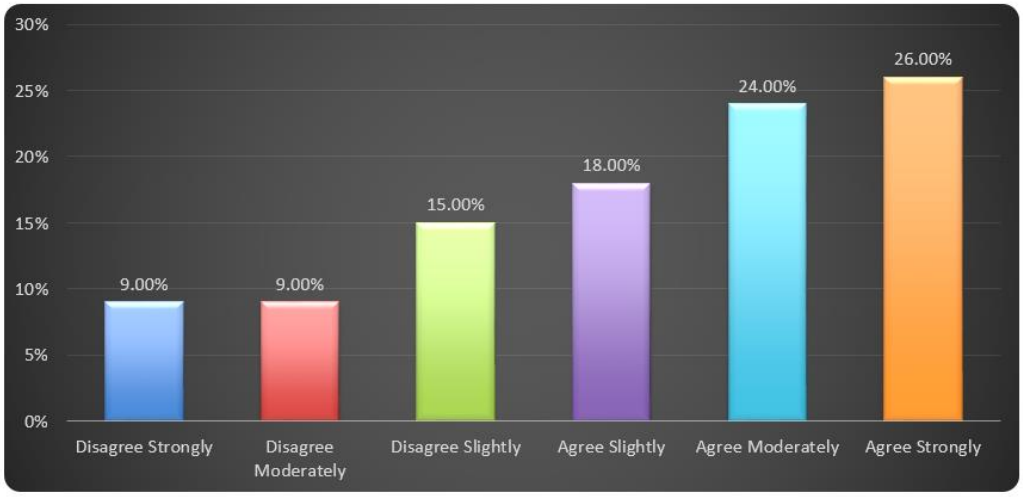


Figure 27: Working two sprints ahead

Contemporary UX Tools and Methods Used

The codes as per table below were extracted from the responses from the participants. The appropriate category chosen was “Contemporary Tools and Methods Used”.

Table 28: UX Tools and Methods Used

Category	Axial Code	Open Codes
Contemporary Tools and Methods Used	Good UX design methods	Native design application used
		Screen mock-ups
		Low fidelity prototypes (rarely due to cost)
		Sketches
		User research
		User validation
		Journey maps
		Interviews
		Wireframes
		Balsamiq (low fidelity)
		Justinmind
		Invision (high fidelity)
		Zeplin

Screen mock-ups and wireframes were the most common design methods used. The mock-ups and wireframes were predominantly designed with assumed user behaviour in mind, without actually consulting with the user to conduct user research. As per P7, "*we are not doing enough user research like interviews with users; we are not using enough brainstorming tools with our users."*

P1 also pointed out that there was no UX specialist involved. Prototypes were rarely used - as per P4 "prototypes are done where the user cannot understand what the end product would look like. Prototypes are used very rarely due to the cost factor to the business." P5 advocates the use of low fidelity prototypes as high-fidelity prototypes slows down the Agile process:

"[We use] sketching, as we consider this as a low fidelity prototype. We don't use clickable prototypes as these results in high fidelity prototyping and slows down the Agile process." [P5]

In terms of the tools used, most of the participants were unaware of any UX tools that could be used. Only one participant (P7), who was in a mature Agile and UX

team, commented that they use Balsamiq for low fidelity prototypes and either Justinmind or InVision for high fidelity prototypes.

In terms of tools that are used to integrate UX into Agile development, P7 and P9 advised that they used InVision. P7 also commented that their team, based in the United States, used Sketch and Zeplin:
“Our team in the US uses Sketch and another product called Zeplin. Zeplin allows the sketches that are designed for the UI to be converted into code and this can be copied into the software which eliminates coding the UI.” [P7]

4.4.2.3. Creating a UX vision

User Research

In order for teams to establish and inculcate a UX vision, it is important for the participants to understand the users. The upfront user research phase is important:

“The inclusion of a UX person in the research phase of the project is key. Currently the person that does the initial UX research does a handover to the UX person in the Agile team. The UX person should be involved from the research phase to ensure that UX requirements are met.” [P5]

“We are not doing enough user research, interviews with users; we are not using enough brainstorming tools with our users. There needs to be more involvement of users from the user research phase to development as well as testing.” [P7]

UX Education

The biggest challenge experienced with the UX vision and the business as a whole, is communication. The best UX vision will be useless if it is not articulated with the business.
Quite a few participants explained that the business did not fully understand the role of UX in a project. The first step in implementing UX is to get buy in from business:

“[Getting] buy in from business for UX activities would help. Whenever I mention the word "UX" they immediately think "User Interface". However, there is so much more to UX than the interface. It's about the view, the product, quality assurance, and it's about what the user sees and how they work with it.” [P7]

Business is also under the notion that the UX role can be filled in by another member of the team like the Business Analyst or the Developer:
“One of the cons [of employing a UX specialist] is the cost factor as many businesses feel that the UX role can be filled in by another member of the team.” [P9]
“They do not understand the importance of this role and generally feel that it can be filled in by someone else in the team (e.g. BA, developer).” [P8]
“UX is seen as a skill that anyone can do, for example the Business Analyst can do it, or the developer can do it. However, it is a specialised skill.” [P6]

If the whole team shares an understanding of what UX is, it makes it easier to live and breathe the UX vision. One of the ways to visually articulate the UX vision, like in P7’s company, is to host a UX Education series with the business:
“As part of my role I run a UX Education series. Every month I run something different on UX and I create a video on it and then a survey goes out to the organisation (approximately 52 participants) with questions relating to the video.” [P7]
P7 endorses the UX Education series, as it educates the organization on what the difference between “UX” and “UI” is, as well as topics on “Usability”, “User Research” and “Design Thinking”.

4.4.2.4. Strategies to include UX

This section analyses the strategies that emerged when trying to include UX and the end-user into the design and the development. Such strategies are described as follows and are based on the categories that they were coded in.

Including UX specialists

As per the literature review, the idea of including a UX person in all sprints is not a new one, e.g. Schwaber (1997), Travis (2011). Unfortunately, within the South

African context, the idea of including a UX specialist throughout the project is an emerging skill. P5 advocates that *"I have worked with UX specialists over the past 6 or 7 years. However, the role has become a loaded term and quite relevant in the past 4 years."*

Looking at results from the questionnaire also shows that combining the fields of UX and Agile software development is an emerging skill. As depicted below, 41% of the responders have 1 to 3 years of experience in combining UX and Agile:

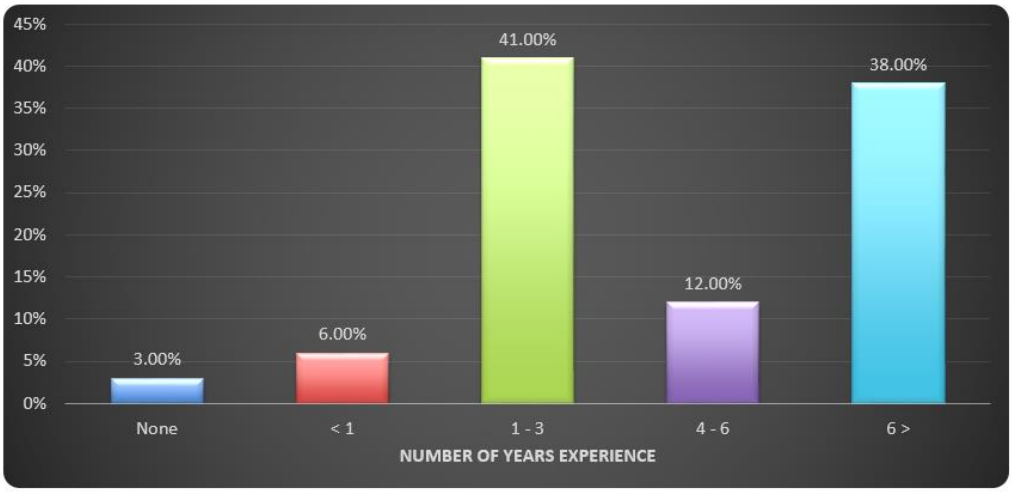


Figure 28: Number of years' experience in combining UX and Agile

P1 who worked in the United Kingdom for a period of two years was able to compare her experience abroad to the South African context:
"In South Africa, sometimes the Business Analyst's goes for UX training and they would have to create the UX elements for the project. In the UK, a UX specialist was part of the Agile team for designing of the UX." [P1]

The chosen quotes below illustrate the participants' responses on the lack of UX specialists within the South African context:
"If it was an external facing user, then a UX specialist was brought into the project. However, all internal projects did not have a UX specialist. It was a brand-new initiative to bring in the UX specialist for the external facing projects." [P2]
"At most about 40% of the projects I worked on had a UX specialist in the team." [P3]

“It is lacking in the business. Our end-users/customers that use the systems are all internal to the company hence it has allowed us to become a bit autocratic in terms of how the software is developed. Users, this is all we have time to develop, so use it or you have nothing else to use.” [P4]

“I would say roughly 40% of projects that I work on include a UX specialist.” [P6]

It was evident during this study that frequent interaction between the UX specialist and developers was seen as being more effective for the project as opposed to a handoff of specifications to the development team upfront. P5 advocates that direct interaction is much more effective in terms of the speed and the quality of a project:

“In most of the teams, the UX specialists are included in the Agile team hence direct interactions. However, for a few of the external clients, the UX teams are a separate team and we have found this to be less effective. There is a handoff of specifications and it does not work as well as when the UX specialist is within the team. If you want speed and quality, then the UX specialist should be in the team.” [P5]

The results from the questionnaire are congruent with the above. As depicted below, 65% of the responders strongly agreed that having a UX expert as part of the Agile team improved the quality of the software.

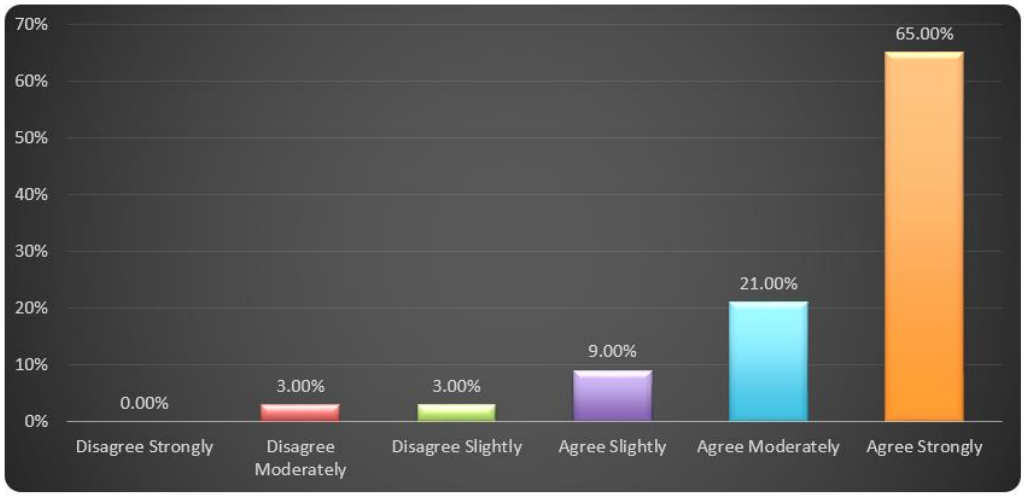


Figure 29: Including a UX expert improves the quality of software

Increasing the iteration time

As described in the previous theme, all companies in this research adopted the scrum methodology. The scrum development process uses the term **sprint** for its iterations. Majority of the companies used two-week sprints, with the exception of one company which used three-week sprints. When questioned on the reasoning behind using three-week sprints, P4 advised that they initially used two-week sprints and followed strict scrum guidelines. However, from experience, and due to the development architecture being highly complex (they have six Agile teams), they found that the amount of work done in two-week sprints was too small.

"All that we were doing was having sprint planning meetings and reviews without delivering enough to do a review. It took us two years to adjust and try something different. We tried four weeks, and then reduced it to three weeks. One of the scrum teams still uses the two-week iteration as it works well with them." [P4]

Including the user throughout

To bring UX and Agile together, the focus should be on inclusion of end-user for *user research, demoes* and *user acceptance testing (UAT).* As exemplified by the questioned asked to participants: "*Is there enough end-user involvement?" 80% of the responses received was "no".* P3 explained that not even user feedback was received. Whenever a sprint was completed or demoed, a user representative was present and not the end-user.

P7 explains that "business feels that involving users slows the entire process down and time is money." P10 is of the same opinion: "Time is an important factor; the user base is quite wide therefore a lot of time will be wasted by involving too many users."

P8 advocates that a creation of a user group with super users can be beneficial. A specialist in each area can be involved in the user group.

P4 works in a mature Agile team and advised that user involvement over the past four years have improved. Previously, software would be released to production and there would be comebacks as certain elements were not tested, or bugs were not picked up during UAT. In the current environment, users are identifying errors before UAT due to them being more involved in the development process. By the

time the software reaches UAT, the user would have had sight of it so many times therefore it speeds up the UAT process and there is less design wastage.

P4 confirms that in the beginning of the implementation of Scrum, users only had sight of what was developed at the end of the sprint. The team felt that the wait was too long and too much was developed therefore it was decided that users would be involved more frequently.

"On average the users meet with the team every three to four days. Teams have learnt that meeting more often with users ensures that development remains on track." [P4]

The attitude of the participants confirmed that the involvement and the engagement of end-users with the development teams effectively enhanced the outcome of the projects.

Results from the questionnaire also indicate that an increase in end-user involvement in the project improves the testing of software. As depicted below, 56% of the responders strongly agreed on increasing end-user involvement in order to improve testing of the software:

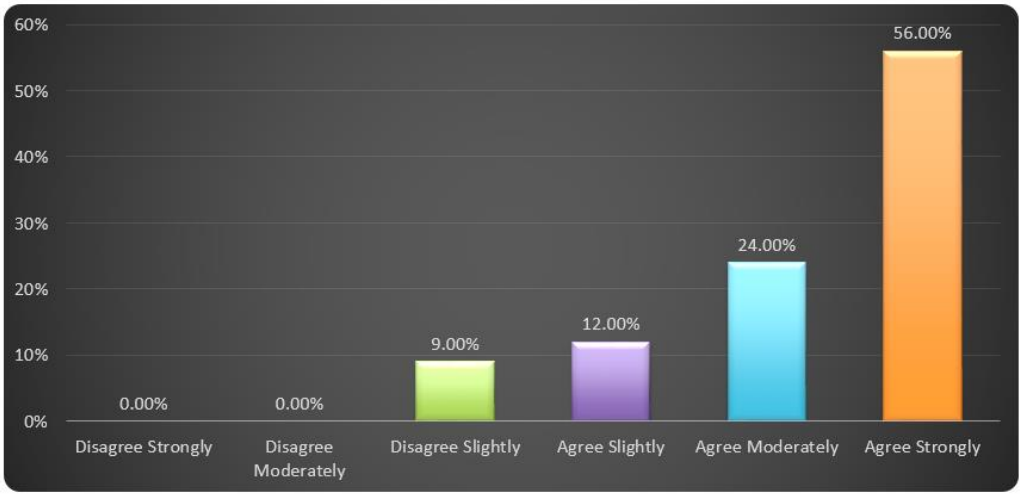


Figure 30: Increase end-user involvement for testing

4.5. Conclusion

This chapter has presented the classification of studies from the literature review which provided insight in terms of the types of studies conducted, the research methods used and the artifacts that were collected from the studies. The classification of studies in terms of the known difficulties and good practices followed. Next, the data analysis of the interviews and questionnaires was discussed which presented the iterative process of coding data using open, axial and selective coding methods. The results from the interviews and questionnaires followed – two themes emerged from the rich data collected in the study.

The first theme, ***Agile UX Integration Practices***, presented the participants understanding of Agile, their willingness to adapt to change and presented the Agile methodologies used. Scrum was the most commonly used methodology, with only one company using Kanban. The roles and responsibilities followed, where it was concluded that there is a lack of a UX person in majority of the companies in the study. UX roles were also blended, which resulted in Agile being demanding and stressful, due to the lack of UX resources. Most importantly, the participants highlighted that UX design should be a communicative role with active end-user involvement, as a thorough involvement of the end-user is the basis for good UX design. The difficulties faced when integrating UX in Agile was discussed next. *Accessibility to users*, *UX slowing down the Agile process* and *prioritising UX activities* were highlighted as difficulties faced when integrating UX in Agile. Next, the ambiguity of the customer versus user role was discussed. The involvement of the user in UX design ranged from an *informative* role to a *consultative* role to a *participative* role. The challenges around the ambiguity of the *customer* and *user* role have been confirmed within the South African context with 90% of the participants advising that the customer and user was the same person. The final category in the first theme related to the role of documentation. Participants advocate the use of lightweight, lean documentation for Agile UX projects. The trade-off with using lightweight documentation is that it results in issues with maintenance of the software. Participants also advised that there is a misalignment with what is documented and what is developed as the specifications are based on assumptions of what the user requires.

The second theme, ***UX Vision***, discussed how UX can be fitted into Agile. Participants saw the advantage of adopting a UX mind-set and endorsed close collaboration between UX and Agile team members as it assisted in defining clear requirements. The design techniques, tools and methods were discussed next. Upfront design (sprint zero) and one sprint ahead were common design techniques used. Screen mock-ups and wireframes were the most commonly used design methods. UX tools such as Balsamiq was used for low fidelity prototypes and either Justinmind or InVision were used for high fidelity prototypes. Sketch and Zeplin were also used to integrate UX and Agile. Most importantly, it was highlighted by participants that there is a need for user research as this was lacking in all companies. One of the biggest challenges experienced with the UX vision and the business as a whole was communication. The recommendation is to firstly get buy in from business, then visually articulate the UX vision to business.

At the start of this chapter, the aim was to present the results and the data analysis of this study. The figure below is a representation of the two themes and the research sub-questions in relation to each theme.

Theme 1 Agile UX Integration Practices	**Research Sub-Question 1**	Participants understanding of Agile
		Adapting to change
		Methodology used
		Role of documentation
	Research Sub-Question 3	Roles & Responsibilities
		Customer versus User
	Research Sub-Question 4	Accessibility to users
		Slows down Agile process
		Difficulty in prioritizing UX activities
Theme 2 UX Vision	**Research Sub-Question 2**	Fitting UX into Agile
		UX design techniques, tools and methods
		Creating a UX vision
		Strategies to include UX

Figure 31: Summary of themes and research sub-questions

In the next section, the generic framework for integrating Agile in UX is presented. The framework has been created based on results from the interviews and questionnaires.

CHAPTER 5: A GENERIC FRAMEWORK FOR INTEGRATING UX INTO AGILE

5.1. Introduction

During this study, a number of findings were discovered in terms on how UX and Agile methodologies co-exist in the software development environment. Referring to the previous sub-section, the set of two themes constituted contribution towards answering the research questions of this study. This section presents the findings in the following manner: First, it describes the objectives of creating a framework for integrating UX into Agile. Second, it synthesises the use of design thinking as a systemic design approach in the development of the framework. The process of using design thinking and the inputs and outputs for the framework are presented. Next, the generic framework for integrating UX into Agile is showcased. Finally, the framework is evaluated by leveraging feedback from a focus group who are experts in the field of Agile and UX. The framework, which formed part of a paper, was also presented at the Information Communication Technology and Society Conference (ICTAS) on the 6th March 2019 in Durban, South Africa where attendees from South Africa as well as abroad were able to provide feedback on the paper and the framework (see Pillay and Wing 2019). The framework was therefore validated in two ways; the focus group and the conference. The acceptance of the framework at the internationally recognised IS conference validates that the framework is competently built. The dissertation progress is depicted below:

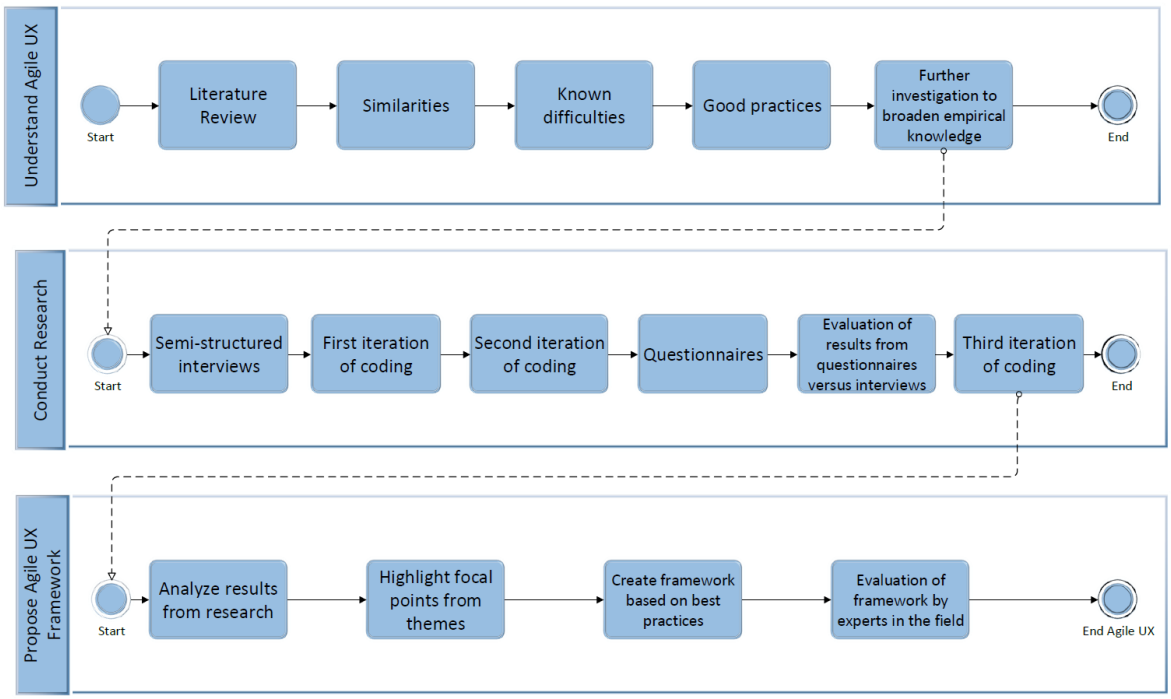


Figure 32: Dissertation progress

5.2. Objectives of a Framework for Agile UX

According to Kheong and Jayaratna (2009), a framework is a way of understanding a problem and the organizational context of where the problem exists. Application of a framework assists in evaluation of methodologies, structure, steps, forms etc. Moreover, it provides a structured approach to draw conclusions or lessons from critical reflection and evaluation of the whole experience (Kheong and Jayaratna 2009). A framework is designed with a problem scenario in mind and is often depicted as a graphic that shows the attributes and characteristics of the problem in question.

Previous work on frameworks for Agile UX strongly suggests utilizing approaches where UX work is separated into their own cycles or tracks (Schwaber 1997; Fox 2010). However, authors such as Ferreira, Sharp and Robinson (2012) advocate that cross-functional teams inclusive of UX processes are more efficient than those that operate in separate UX cycles. This study strongly supports the latter approach as one of the most common approaches on how to improve UX work in Agile in this study was to include UX work in the Agile development team. Participants from this study identified separate Agile and UX teams as a hindering issue in Agile UX work. Problems related to weakened communication, difficulty in

timing sprints, accessibility to users and differences in work practices. This study strongly suggests that ensuring good UX development practices necessitates a level of teamwork between the Agile and UX disciplines.

In the last two decades, there has been a surge in the use of *'design thinking'* in the UX design profession and it is considered as a paradigm that deals with problems in many disciplines (Adikari, McDonald and Campbell 2013). A widely cited definition by Dunne and Martin (2006) describes design thinking as follows:

"Design thinking is the way designers think: the mental processes they use to design objects, services or systems, as distinct from the end result of elegant and useful products. Design thinking results from the nature of design work: a project-based work flow around 'wicked' problems." (Dunne and Martin 2006: 517)

The emphasis of the above definition stresses three pertinent points:

1. Design thinking is a paradigm to assist in the creation of a new or enhanced object, service or system that meets the users' needs with added value.
2. Design thinking is a paradigm for designing. The core focus is the design; hence designers design solutions with a creative mind-set towards discovering 'elegant and useful products'.
3. A 'wicked' problem exists. Dunne and Martin (2006) describe 'wicked' problems as a class of social or system problems where the information is confusing, where there are stakeholders with conflicting views and where the ramifications in the whole system are thoroughly confusing.

It has been established in the Literature Review (see Chapter 2) and the Results and Data Analysis (see Chapter 4) that there are a number of problems associated with the integration of UX in Agile. The objectives for the framework are designed to address the issues identified.

The objectives for this framework are as follows:

- The framework should be applicable to any Agile methodologies. The growing interest in Scrum shown by software development practitioners and academics as well as Scrum being the methodology adopted in all companies that participated in this study warrants the necessity of depicting Scrum in the proposed framework. However, it is worth noting that the conclusions, recommendations and best practices might be

applied to any Agile methodology and are not limited to the Scrum methodology.

- The framework should promote collaboration between users and stakeholders. Adikari, McDonald and Campbell (2013) opine that systems, users and stakeholders require specific focus to gain a deeper understanding and view of the whole systems landscape. This assists in identifying system issues, constraints, system goals and requirements.
- The framework needs to provide clear guidance in terms of who the participants are and what their roles are. One of the problems identified in this study (as evidenced in Chapter 4) was the ambiguity of the customer and user role in Agile UX projects. Results also indicated that UX roles were blended into the Agile team.
- The requirements for the Agile UX project are not only limited to technical IT requirements but also considerations of interactions between technology and people using the technology (i.e. the end-user). In the Literature Review, the interplay between UX and Agile was addressed. Kuusinen *et al.* (2016) and a few other authors consider four dimensions that are enabling factors for evolution of performance management systems: *process, people, systems* and *culture.* One of the problems identified in this study (see Chapter 4) is that focus is placed on processes and systems whilst people and culture are overlooked.

Thus, the Agile UX framework is designed with these objectives in mind.

The context that is assumed to exist is as follows:

- A wicked problem exists around the integration of UX in Agile therefore the requirements need to be constructed through a technological and social lens.
- The problem includes aspects related to building software using the Agile methodology.
- The problem includes aspects related to UX and end-users.
- Multiple stakeholders exist, some of which may be affected due to a change in the business practice.

Having provided the objectives of a framework and the context to this study, the next sub-section looks at the application of Design Thinking in the development of the framework.

5.3. Design Thinking in the Development of the Framework for Agile UX

5.3.1. Consideration of using Design Thinking for this research

Design research has positioned information technology artifacts at the forefront of information systems disciplines and *design thinking* has received increasing scholarly attention in the past two decades for dealing with problems in many disciplines (Dunne and Martin 2006; Adikari, McDonald and Campbell 2013). Design thinking was first coined by Arnold (1962) and can be described as an approach to creatively solve problems. Authors such as Dunne and Martin (2006) and Adikari, McDonald and Campbell (2013) have conducted systematic comparisons in their seminal papers of the most salient characteristics of this design approach. According to Adikari, McDonald and Campbell (2013), design thinking enhances the current UX by integrating three design approaches – design thinking, designing for UX and Agile software development. Design thinking results in the creation of a framework that includes functionality of the system and aspects of UX based on shared understanding gained from all stakeholders in the context of their roles (Adikari, McDonald and Campbell 2013).

There are a host of studies that critically compare design thinking research and have used it to develop frameworks that include aspects of UX and Agile (Dunne and Martin 2006; Innes 2011; Adikari, McDonald and Campbell 2013; Fauquex, Goyal, Evequoz and Bocchi 2015; e Sá 2016). Authors such as Dunne and Martin (2006) opine that the design thinking paradigm encourages a deep understanding of the users and also recognises the value in the contribution of all stakeholders in a project. Adding to this argument, authors such as e Sá (2016) advocate the call for creation of multidisciplinary teams and early incorporation of users' feedback to address the existence of wicked problems. Based on the results from a critical review of the literature, e Sá (2016) advocates the use of design thinking to guide the design of software solutions with the purpose of it being effective and efficient

as possible. “Designers are expected to understand not only the technical aspects of a given problem, but also a wide range of social, economic and psychological impacts of the design decisions proposed.” (e Sá 2016: 6)

For this study, the use of design thinking was justified as it allowed the researcher to design an artifact to solve the wicked problem of integrating UX in Agile. Supporting literature by a myriad of authors suggests that UX and design thinking are two faces of the same coin (Innes 2011; Adikari, McDonald and Campbell 2013; Fauquex *et al.* 2015; e Sá 2016). Design thinking is essentially a problem-solving paradigm that allows the researcher to create a framework based on the following principles, as outlined by e Sá (2016):

- On-going cycle of generating ideas (Abduction)
- Predicting consequences (Deduction)
- Testing
- Generalising (Induction)

Following the process of design thinking, the framework for Agile UX will then be built to a specific area with a wicked problem, which in this case is the integration of UX in Agile that includes a clearer understanding of end-users’ requirements. An on-going cycle of generating ideas (abduction) will be used to facilitate the emergence of understanding users’ requirements and understanding the problem. This phase will benefit from the existing body of knowledge from the literature (Chapter 2) and any other good practices that were identified in this study (Chapter 4). The second phase will be used to predict consequences (deduction) on the best practices of integrating UX in Agile in the framework. This phase has an added element of evaluation whereby the learning’s from the first phase is used to predict consequences. The third phase stresses testing; during this phase the framework will be presented in a focus group to experts in the Agile UX field. An issue that has been neglected in many Agile UX studies is the evaluation of frameworks. Despite the researcher’s efforts, it was found that there was a lack of academic literature on evaluation of Agile UX frameworks. The focus group will not include any of the participants that were used in the interviews and questionnaires to avoid any biases. The last phase involves generalising (induction) and learning from the evaluation of the framework. In addition to

understanding the end-users' needs to design a fluent integration process, the framework needs to consider the significance for the improvement of the Agile UX integration. Issues such as technical limitations, cost for the different design possibilities, development quality and means to fulfil conflicting users' needs are to be considered.

Having provided the consideration of using design thinking for this study, the next sub-section looks at the process of applying design thinking in the development of the framework.

5.3.2. Process of applying Design Thinking Research

The design thinking methodology (e Sá 2016: 93) has five major steps as depicted in the figure below:

Figure 33: Design Thinking Steps

Source: Adapted from e Sá (2016: 93)

Although the five steps are depicted in a linear fashion, they can occur in any order and can be repeated. The cycle ends when there is sufficient evidence and no new learning's. The researcher synthesized the views from well cited papers by Adikari, McDonald and Campbell (2013), Dunne and Martin (2006), Fauquex *et al.* (2015) and e Sá (2016), and these are presented below. A process of thematic analysis was applied to achieve the results presented. There are many variations on the application of design thinking in the development of a framework. The researcher continually organized and reorganized similar themes on design thinking in the literature until a relevant set of concepts were generated. The five steps in the design thinking methodology and the application of these steps in this study are illustrated in the table below:

Table 29: Design Thinking and Application in this Study

Design Thinking Steps	Application in the development of the framework
1. Empathy	The 'Empathy' step involves emotional concerns of all users and has been acknowledged as an important part of the design. As highlighted by Adikari, McDonald and Campbell (2013), empathy informs designers to create designs matching what the user needs and wants.
2. Define	The 'Define' step seeks to characterise the user and what is required by the user by means of personas, role objectives and challenges experienced.
3. Ideate	The 'Ideate' step ensures that there are many ideas or alternatives as possible. Ideas are shared and are also prioritized.
4. Prototype	The 'Prototype' step advocates building simple prototypes based on the most promising ideas generated. The designer must be able to iterate quickly and make changes to the prototype.
5. Feedback	The 'Feedback' step is a review of the users' reactions to the prototype. This step is used to understand the impediments, what works and what does not work and the roles that each stakeholder needs to play.

In the current study, the theory supporting the development of the framework from the 'Empathy', 'Define' and 'Ideate' steps have been examined and presented in the previous chapter. The framework for integrating UX in Agile is therefore drawn together using design thinking with consideration of the existing theory inferred from the literature and from the research conducted in this study. This therefore completes the steps one, two and three in design thinking which is then followed by the building of the prototype (step four) and the feedback of the framework (step five).

The next sub-section proffers the inputs and outputs that will be used to build the framework (step four of design thinking).

5.3.3. Inputs and Outputs for the Agile UX framework

The 'Fishbone' diagram below presents the various contributors that form part of the input for the framework. The diagram was adopted to a similar approach used

by Kuusinen and Väänänen-Vainio-Mattila (2012). The objectives are shown on the left of the diagram and include the scope, quality and cost as performance metrics. As per Kuusinen and Väänänen-Vainio-Mattila (2012) and the results of this study (see Chapter 4), the *scope*, *quality* and *cost* are factors that contribute the most towards the UX.

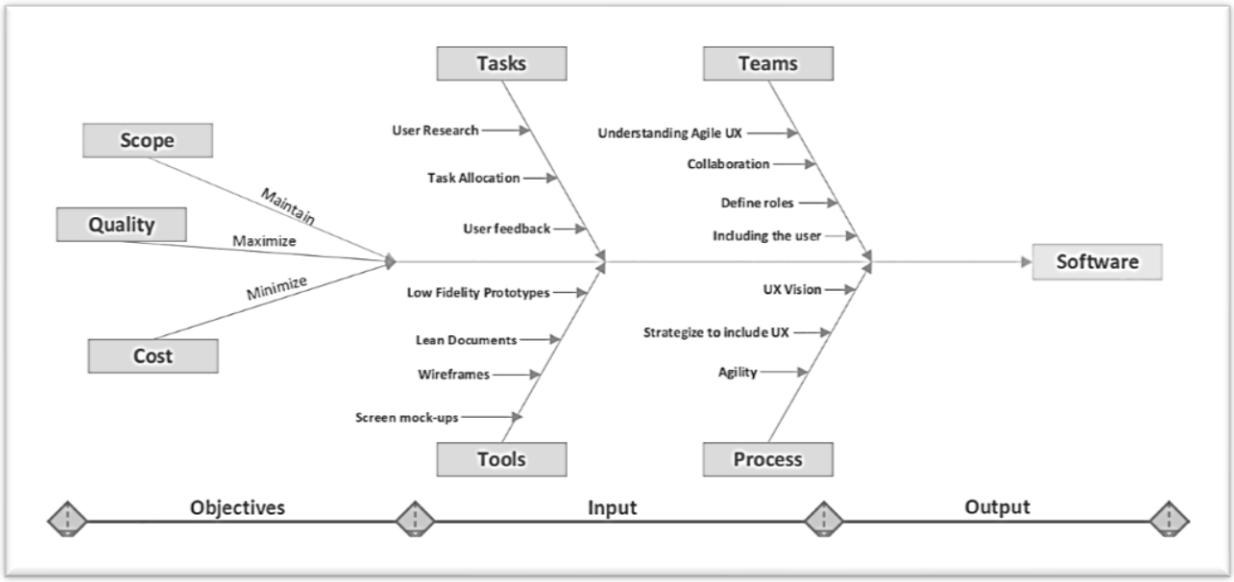


Figure 34: Objectives, Inputs and Outputs for Framework
Source: Adapted from Kuusinen and Väänänen-Vainio-Mattila (2012: 69)

The '*tasks' component* in the diagram relate to practices that have been identified in this study such as user research, task allocations to the various team members and incorporating user feedback into Scrum cycles.

The '*teams'* component relate to the teams understanding of Agile and UX, close collaboration between Agile and UX team members, defining roles (e.g. Customer versus User) and including the user throughout the project.

The '*tools'* component relate to the pertinent tools that were outlined during this study that facilitate UX integration into Agile.

The final component relates to '*process'* and speaks to the strategies to include UX into Agile. Moreover, the process needs to adopt agility in order to facilitate the quick feedback from the user. The element of the UX vision is also important in order to educate various stakeholders to adopt a UX mind-set.

Having provided the inputs and outputs for the framework of this study, the next sub-section showcases the framework. This is followed by the last step of design thinking, which relates to 'Feedback'.

5.4. The Agile UX Model

The framework for this study has been named "The Agile UX Model". Keeping in mind the strategies currently being used by organizations in South Africa, this model aims to facilitate more robust integration between Agile and UX by merging best practices from two fundamental methodologies – Agile and UX. In addition, it addresses the problems that are encountered when integrating UX in Agile. Finally, it aims to maximise the quality of the software whilst minimising costs. The Agile UX model is presented below:

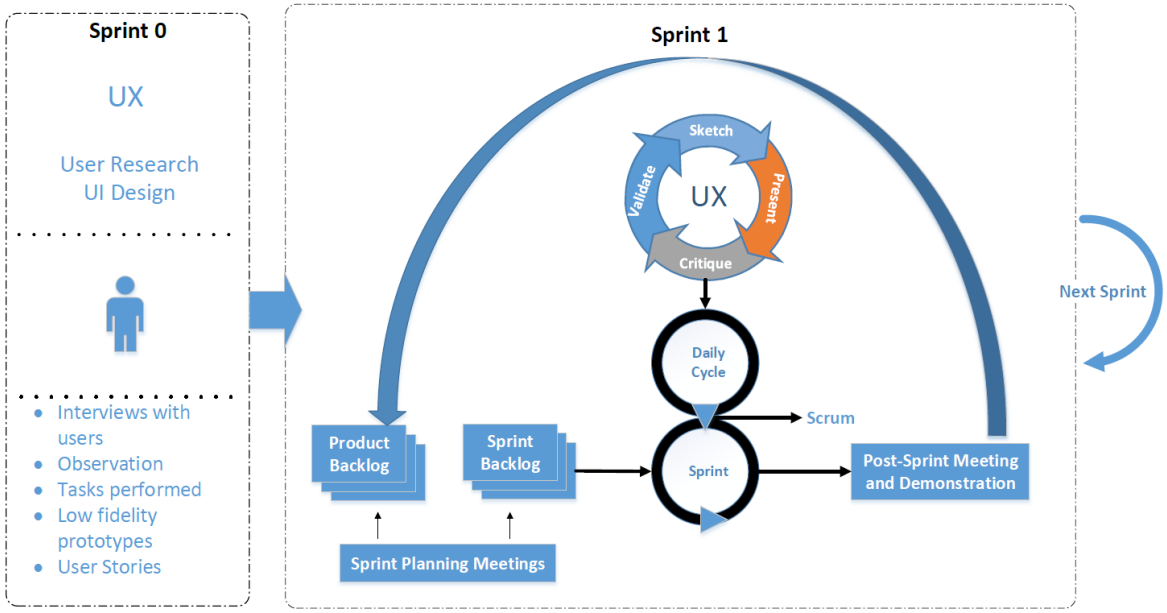


Figure 35: The Agile UX Model

The model consists of the following main components:

- Upfront UX Work (Sprint 0)

  An element of upfront UX work is required, which includes user research and UI design, and this is depicted as Sprint 0 in the figure above. This only happens once in the lifecycle of the project. This pre-sprint phase was common to the majority of the participants in this research as well as in the

literature and most of the participants remarked that this stage of the project lasted for two weeks.

- Traditional Scrum Ceremonies (Sprint 1)

  After the pre-sprint phase is completed, it is passed to the Agile development team to initiate the first official sprint of the project, which follows all the ceremonies of Scrum (i.e. product backlog, sprint backlog, daily cycles, sprint, post sprint meeting and demo etc.)

- Lean UX Cycle (Sprint 1)

  In addition to the traditional Scrum ceremonies, a lean version of UX has been embedded in the sprint. In order for UX to work, the traditional way of thinking needs to be adjusted and a Lean or Agile UX mind-set needs to be adopted. The UX cycle in Sprint 1 comprises of the following tasks:

  - *Validate* – allows for the Agile UX team to check and validate the UX work that was carried out in Sprint 0. As the project progresses, this phase is important to ensure that the team does not lose the "big picture".
  - *Sketch* – allows for the UX practitioners to "sketch" the UX requirements for the sprint. Feedback from the validation process will be factored into this process.
  - *Present* – the sketches are presented to the broader team for feedback.
  - *Critique* – the team critique the sketches, and this is re-factored into the sketches.

The Sketch, Present and Critique processes are iterative as it allows for UX work to also be done for the next sprint. This allows for the pattern of working at least one sprint ahead of the current development. However, this is factored into the current sprint.

The model follows the Scrum framework as it was the most widely used Agile method in all organizations. The Scrum framework resembles Highsmith's diagram for Scrum (Highsmith 2002). Even though the labels of "Product backlog", "Sprint backlog", "Sprint planning meetings", "Post-sprint meetings" and "Scrum" is derived from the Scrum methodology (Schwaber 1997), this does not imply that the framework is only applicable to the Scrum methodology. Rather, its definition implies that the features of this framework is a representation of the

various stakeholders needs, prioritising requirements and goals over the lifecycle of the project and ultimately maintaining the overarching design vision.

The key features of the proposed model are to tackle issues relating to user research and to align all team members with the UX vision of the project. The additional UX layer on top of the existing sprint layer, ensures that the team works in parallel and there is less design wastage.

Having presented the Agile UX model, the next sub-section synthesizes the final step of design thinking which relates to “Feedback”.

5.5. Evaluating the Agile UX Model

In this study, the theory ingrained artifact was the Agile UX model as presented in the previous sub-section. The framework was based on design thinking principles for Agile UX. During the iterations of this study, the researcher challenged the participants’ existing ideas and assumptions on integration of UX into Agile with the intention of creating and improving the design of the framework. In design thinking, this process is referred to as framing and reframing. While, framing refers to the cognitive process of capturing, analysing and creating knowledge, reframing is the process of exploring the same knowledge from multiple perspectives to create new knowledge (Adikari, McDonald and Campbell 2013: 9).

As mentioned in section 5.1.2 the ‘Feedback’ step is a review of the users’ reactions to the prototype and is used to understand the impediments, what works, what does not work and the roles that each stakeholder needs to play. Following the principles of design thinking, the next step is the evaluation of the framework and reflection on whether the framework will facilitate improvement in the integration of UX into Agile. It has been established for this framework that the context is as follows:

- A wicked problem exists around the integration of UX in Agile therefore the requirements need to be constructed through a technological and social lens.
- The problem includes aspects related to building software using the Agile methodology.
- The problem includes aspects related to UX and end-users.

- Multiple stakeholders exist, some of which may be affected due to a change in the business practice.

This evaluation of the framework relied on the focus group method, which is discussed in the next section. The framework will be evaluated to ensure that it meets the objectives that were outlined in section 5.2.

5.5.1. Evaluation of framework using a focus group

In recent years, there has been emphasis on research methods to improve validity, reliability and generalizability of research results (Kontio, Lehtola and Bragge 2004; Al Thunibat, Zin and Sahari 2011). There is also an interest in a broader range of empirical methods in order for the appropriate method to be selected and utilized for the research problem. The focus group method is described by Al Thunibat, Zin and Sahari (2011) as a qualitative method designed to elicit opinions, beliefs and perceptions in order to gain insight on a particular topic. The same authors opine that focus groups can be used to conduct evaluation, planning and needs assessments. Moreover, focus groups combine elements of interviewing and observation and provides the opportunity to probe the participants' cognitive and emotional responses (Massey 2011). For this study, it was the intention of the researcher to conduct an evaluation of the proposed framework in order to promote validity, reliability and generalizability of the research result. As evidenced in the literature, the focus group method has been widely used across the software engineering and user experience communities (Kontio, Lehtola and Bragge 2004; Heikkinen, Olsson and Väänänen-Vainio-Mattila 2009; Massey 2011; Abrahamsson *et al.* 2017). There are several well cited papers that provide detailed guidelines on how to plan and run focus groups. Based on a short thematic analysis on focus group research (Kontio, Lehtola and Bragge 2004; Heikkinen, Olsson and Väänänen-Vainio-Mattila 2009; Massey 2011; Abrahamsson *et al.* 2017), the steps illustrated in the figure below were followed in the evaluation of the Agile UX model.

Figure 36: Steps in Focus Group Research

1. Define research problem:

   The research problem was the identification of obstacles in bridging the integration of UX in Agile. The outcomes of the literature review together with the results of the interviews and questionnaires have been analysed and aggregated to create a generic framework for Agile UX.

2. Select participants:

   Two main criteria were used in the selection of participants for the focus group. First, they must be employed within the UX and Agile development environment in South Africa. Second, if they participated in either the interview or the questionnaire they will not be able to be included in the focus group. To avoid bias and skewing of results, the researcher did not use participants from the same sample used for the interviews and questionnaires. This also improves the validity, reliability and generalizability of research results as the participants' opinions represented the wicked problem that was experienced in their environment.

   A non-probability purposive sampling technique was applied that corresponded to the interviews and questionnaires. Participants were key employees within the UX and Agile development environment in South Africa which corresponded with the above criteria.

3. Plan focus group session:

   The researcher contacted the participants directly telephonically or via email to participate in the study. Participants used for focus group session included Scrum Masters, Front-end developers, UX Analysts, UX developers, Interaction designers and Testers. The total number of participants was eleven.

4. Conduct focus group session:

   The session commenced with an overview of the objectives of the study and the process that the participants should follow in the session. The researcher acted as the chair and as with the interviews and questionnaires, the researcher informed the participants that the session will be audio recorded for the sole purpose of transcribing the session and documenting the points raised. The session was recorded using a voice

recording application on a mobile device. They were also aware that all responses were treated as confidential and will remain confidential throughout this study. The Agile UX model was presented to the participants and special emphasis was given to them to ensure that their opinions should represent a real situation from their organizational perspective. A presentation of the each of the methods in the model was showcased including a few practical examples of how the model could be used. The participants were asked to spend a few minutes to deliberate on the following high-level questions:

1. Does the model address the problems experienced when integrating UX in Agile?
2. Are there any recommendations for improving the model?

Thereafter, a semi-structured discussion took place where participants were asked to voice their opinions on the questions above as well as any other input they had. The session lasted for approximately one and a half hours. The next sub-section synthesizes the results from the focus group session which is the final step of design thinking (i.e. “Feedback”).

5.5.2. Analysis of results from focus group

The data collected from the focus group session was transcribed verbatim by using the transcription tool, Express Scribe. The “feedback” step of design thinking, as described by e Sá (2016), requires the researcher to learn from the users’ reactions to the solution that is being proposed (i.e. the artifact in the form of a prototype, framework, experience blueprints etc.). The analysis and results from the focus group were as follows:

1. The process of wicked problem formulation and proposing a solution is an iterative process rather than a sequential one and requires continual refinement. The Agile UX model is iterative by nature, using the principles of Scrum and can be used by any Agile methodology.
2. It is important to understand the UX design ethos. Most of the participants advocate that it is critical to understand the end-users’ requirements and data, but it is even more critical to include the end-user during the design process. The Agile UX model has the additional UX layer on top of the

existing sprint layer which ensures that the end-user is involved in UX activities and collaborates closely with the Agile team. The consensus in the session was that the more the user becomes involved in the project, the closer the team is to end-product.

3. Cross-functional teams ensure that the work is done collaboratively. It was evident in the interviews and questionnaires that there were mixed roles in most of the Agile UX teams. One of the pertinent questions raised by one of the participants was "where does the Business Analyst role sit in Agile UX." The researcher's response was that there is no Business Analyst role in the Agile methodology or UX. The Business Analyst role has been blended into teams. There is also a lack of UX specialists in the South African context and this role is filled by other members in the team (in some instances the Business Analyst or the developer). It is thus important for this person to be skilled or trained in UX in order for UX requirements to be met (refer to Theme 2- UX Vision in Chapter 4).
4. The user is the *end-user* and **not** the *customer.* The findings from the literature review and the results of this study clearly indicated the customer and user role is ambiguous and often one and the same in projects. There is a call by all participants that the end-user role be clearly defined upfront as the whole design process will centre around the end-user's needs and expectations.
5. The feedback from the session was mostly positive and all participants agreed that the Agile UX model improves the integration of UX in Agile as it facilitates close collaboration between the teams and it ensures that users' needs and expectations are included from sprint zero (through activities such as user research).
6. The focus group study resulted in recognition and categorization of good practices, pertinent issues, risks and theoretical evidence that supported results from the interviews and questionnaires.

The framework has achieved the following aims:

- It is applicable to any Agile methodology. The features of the framework are a representation of the various stakeholders needs, prioritising

requirements and goals over the lifecycle of the project and ultimately maintaining the overarching design vision.

- The framework promotes collaboration of the users and stakeholders by catering for upfront UX work (Sprint 0) and the Lean UX cycle.
- The framework provides clear guidance in terms of roles. The user role is also clearly defined as being the end-user in the project and not the customer.
- The framework promotes close collaboration between the various teams thus enabling the four factors of performance management systems: process, people, systems and culture. The upfront UX work ensures user research is catered for. Moreover, the Lean UX cycle promotes close collaboration between the UX and Agile teams.

From a theoretical standpoint, the framework has also demonstrated the following evaluation criteria, as defined by Gregor (2006) in her paper on theory types for information systems research:

- Relevance – the Agile UX model was relevant to this study and addressed the wicked problem.
- Completeness – no changes were made to the Agile UX model. The participants acknowledged that the framework provided a more complete understanding of the situation of concern.
- Consistency – the results from the literature, study and the focus group were consistent. The various methods used to analyse results were also consist in order to promote validity and reliability.
- Comprehensive – to avoid bias and skewing of results, the researcher did not use participants from the same sample used for the interviews and questionnaires. The perspectives of different stakeholders were used to ensure that emergent understanding was comprehensive.

5.6. Conclusion

The integration of UX into Agile and the development of software in line with the users' needs have emerged as a major problem in the literature and the results of this study. As a response to these concerns, the framework for integrating UX into

Agile is justified as a systemic approach based on the principles of design thinking which was applied as a design methodology for development for this framework.

By carefully analysing their principles and stages, the development of the framework followed the steps of design thinking (as shown in figure 5-2). The design artifact in this study is the Agile UX model. The model was built using evidence inferred from the literature (Chapter 2) and results from this study (Chapter 4). A description of the inputs and outputs for the model was presented. The scope, quality and cost were the factors that contributed the most towards UX. Tasks, teams, tools and processes were the key drivers and served as inputs to the model. The output was software that meets the users' needs and expectations.

A presentation of the Agile UX model revealed that it aims to facilitate more robust integration and also aims to address the pertinent problems that were encountered when integrating UX in Agile. Moreover, it aims to maximise the quality of the software by including the user whilst minimising costs. The proposed model recommends upfront lean UX work to be carried out in Sprint 0 which requires active involvement from the user in order to create the UX vision for the project. Sprint 1 has an additional UX layer on top of the traditional sprint layer which allows for the team to work in parallel and to also ensure that the team members are aligned in terms of what needs to be done in the sprint.

The framework was evaluated using a focus group. The focus group offered the researcher the opportunity to obtain insights from the participants and to also promote validity, reliability and generalizability of the research results and the framework. The attitudes, beliefs and opinions of the participants provided a greater understanding of the study area and will also assist in providing conclusions and recommendations.

The next chapter revisits the research questions and attempts to answer these questions. It also identifies limitations of the study and provides recommendations of possible areas for future research.

CHAPTER 6: DISCUSSION, CONCLUSIONS AND RECOMMENDATIONS

6.1. Introduction

In this chapter the findings of this study are presented in relation to the aim, which was to investigate how UX development is currently integrated into Agile software development. In order to strategize the aim, the first three objectives sought to uncover how UX developers collaborate with members of the Agile team, how UX is perceived by software development organisations and to evaluate the existing Agile UX software development process in the South African context. The findings from these three objectives contributed towards the final objective of this study; the development of a generic framework for the integration of UX in Agile. In the quest to achieve the aim and objectives, this study adopted an interpretive paradigm, using grounded theory as a research method for theory formulation by using interviews and a questionnaire as data collection methods. The data was analysed through the lens of the Straussian version of grounded theory by applying coding methods which helped shape the data into two main themes that emerged, being 'Agile UX Integration Practices' and 'UX Vision'. These two themes became the foundation on which the generic framework was developed. The framework was developed using design principles that are inherent in the design thinking methodology.

This chapter provides an overview of this research; the research questions are revisited, and results are presented; the research contributions are then described. The chapter concludes by providing the limitations of the study and suggestions for future research.

6.2. Revisiting the Research Objectives and Research Questions

6.2.1. Attainment of Research Objectives

The objective of this research, as defined in Chapter One was as follows:

RO1: To study how UX developers collaborate with members of the Agile team in the software development process.

RO2: To understand how UX is perceived by software development organisations, in order to identify the extent at which organisations have been successful in reflecting the users' requirements in their applications or products.

RO3: To evaluate the existing Agile UX software development process in the South African context and confirm if approaches from previous studies applies to the South African context.

RO4: To propose a generic framework for integration of UX in Agile and to provide further recommendations for efficient integration of UX in Agile.

To the researcher's best knowledge, this is the first comprehensive study within the South African context to elucidate the integration practices of UX in Agile. The representation of these knowledge areas and activities could not be investigated in isolation. This study's foundation lies in the representation of knowledge areas from the literature.

Objective 1 and 2 were unearthed in Chapter Two. More specifically, sub-section 2.2.3 and 2.2.4 sought to address the interplay between UX and software development using the Agile framework and to provide a view of UX dimensions in Agile. The mediating relationships between Agile and UX for objectives 1 and 2 were conceptualized in Chapter Four in order to attain research objective 3 of this study.

Underpinning the practices of Agile UX was a bottom up approach as this study relied on cases that were identified in the literature in order to evaluate these practices within the South African context. A fair level of analysis and interpretation of the data from the interviews and questionnaire was unravelled in Chapter Four and these results were aligned to the previous literature. These integration practices from Chapter Four shaped the attainment of objective 4.

The Agile UX framework presented in Chapter Five was based on the principles of design thinking and posits a solution for integrating UX in Agile exploring the good UX development practices unearthed in this study. The objectives of this study have thus been attained. The results from the literature review along with the results from the interviews and questionnaire have been analysed to create a consolidated generic framework for Agile UX.

6.2.2. Revisiting the Research Questions

Chapter One described the process of formulating the research question in this thesis. The main research question sought to answer the real-world problem and examine why this problem is becoming an issue. The decision to formulate the research question using general terms (e.g. Agile) was deliberate. The researcher wanted to cast a wide net to gather as much information as possible on the disciplines of Agile and UX. The primary research question was as follows:

"How can good UX development practices be integrated into Agile within the South African context?"

The purpose of the primary research question was to gain a deeper understanding of the development practices of integrating UX into Agile within the South African context. The development techniques, roles of team members and the pertinent problems when integrating UX in Agile were discovered to be strongly connected with the primary research question. For this reason, the following sub-questions were formulated to clarify the main question:

1. "*What Agile development techniques are best suited to ensure UX requirements are met?"*
2. *"What are good development practices for creating good UX?"*

3. *“What are the roles of team members that are directly involved in the process of integrating UX in Agile?”*
4. *“What are the problems when integrating UX in Agile?”*

The above questions sought to expose the impact that UX has on Agile software development. These questions featured in Chapters Two and Four.

In Chapter Two, a discussion on the current integration practices of Agile UX followed. The analysis and interpretation from previous literature provided initial insights into the unique integration practices of Agile and UX in software development organisations. However, it must be stressed that the current literature lacked insights and emphasis on how Agile and UX is integrated within the South African context.

Chapter Four analysed and interpreted the data from this study and unravelled integration practices that were aligned to the previous literature as well as new integration practices. The results from this study indicated that the integration practices and the underlying problems for Agile UX appeared to be similar to the previous literature, both in the framework and the day to day practices. The data analysis and interpretation in Chapter Four revealed little variation in the integration practices. A summary of the research questions and the answers from the literature review in comparison to the answers from this study is juxtaposed in the figure below.

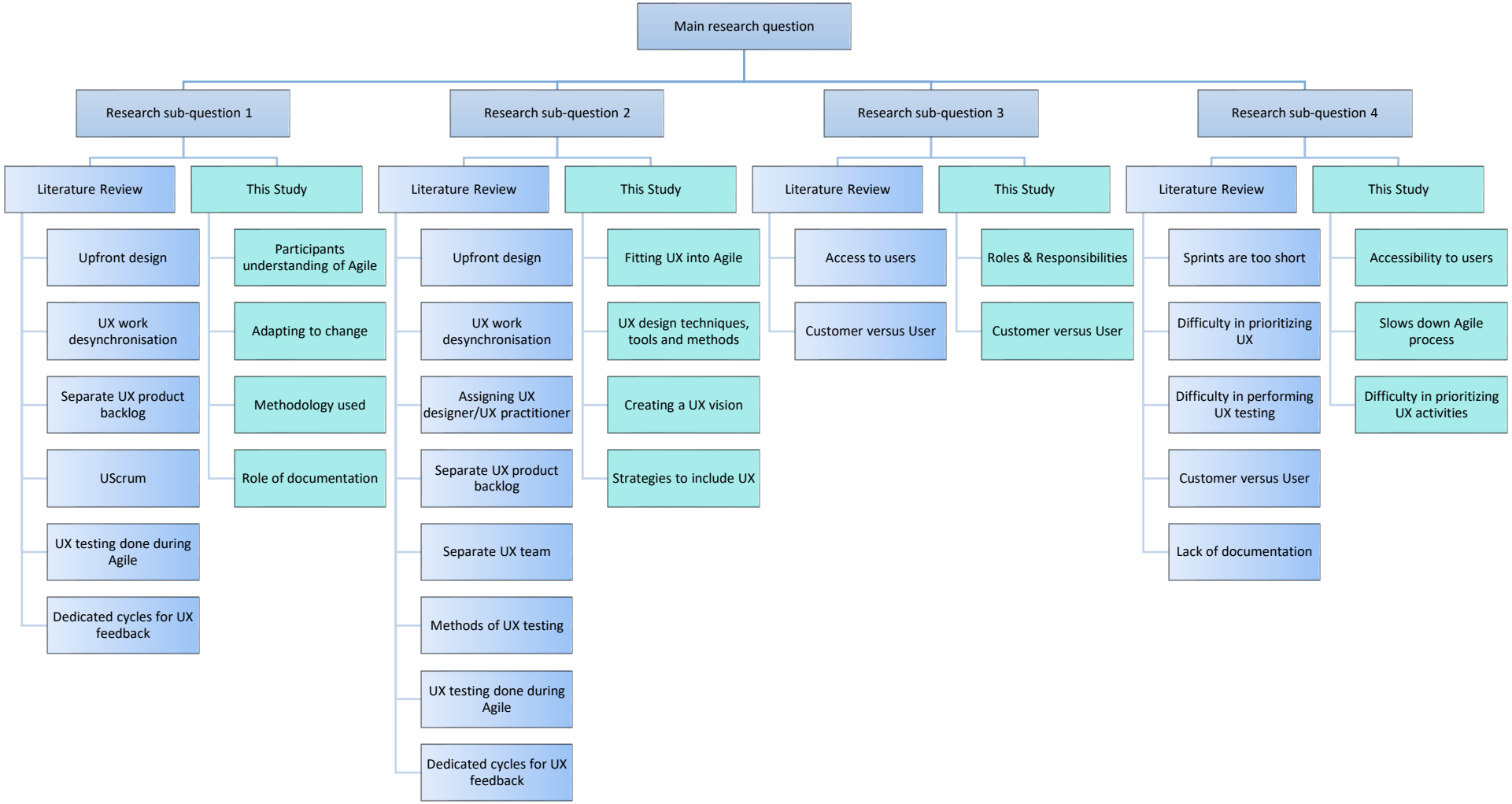


Figure 37: Comparison of research questions and answers

6.3. Limitations

This study sought to unearth the real-world problem of integrating UX into Agile software development. The situation of concern exposed conflicting interests between UX and Agile practitioners within the South African context. The first limitation of this study is the limited number of UX practitioners in South Africa. This study would have been enhanced with the inclusion of more UX practitioners. However, UX is still an evolving role and despite the researcher's best efforts, there were very few instances of UX specialists being involved in projects in South Africa. It is worth noting that there has been a keen interest in the role of UX specialists over the past two years in South Africa as confirmed by participants in this study. This will hopefully create new territories for more detailed investigations for future researchers.

The second limitation is that the study cannot be generalized as a small sample size was used. Golafshani (2003) opines that if the validity of a qualitative study can be maximised or tested then this will lead to generalizability. The same author suggests that the ability to generalize findings to a wider group and circumstances is one of the most common tests of validity. Some authors argue that generalization for qualitative research is fundamentally different to that of quantitative research (Mayring 2007). This line of criticism stems from the interpretivist's standpoint. Mayring (2007) argues that interpretivists reject generalization as results from a subjective perspective are time and context specific. Moreover, human interactions (e.g. UX) always have multiple meanings, "leading to inherent indeterminacy and prohibiting generalizations" (Mayring 2007: 3).

For this study, the sample consisted of ten participants for the interviews. Majority of the participants resided in Durban, Kwazulu Natal. Would the results be different if the participants resided across South Africa? As noted in Chapter Four (sub-section 4.3.2), the researcher used a questionnaire as a second research method for data collection. The questionnaire attempted to eliminate the possible mismatch between the sample and the target audience due to an inadequate sample size.

The third limitation was that this study was entirely dependent on the participants volunteering to be interviewed or to complete the questionnaire. There is a possibility that this limitation could have had an influence on the results of this study as it is possible that key stakeholders could not participate due to time constraints.

6.4. Recommendations for Future Research

During this study several areas of future research were uncovered. First, the study can be expanded to small companies. The selection of participants did not dictate company size. However, most of the participants were from medium to large companies. Medium to large companies consist of multiple hierarchies and many concurrent projects. Similar studies conducted by Kuusinen, Mikkonen and Pakarinen (2012) in medium to large companies revealed that the organizations were advanced in both Scrum and UX work. However, the results were inconclusive as there is a great extent of addressing organizational issues rather than methodological issues. The authors call for further research to refine Agile UX methodologies to better support UX related issues. Stickel, Ogonowski, Jakobi, Stevens, Pipek and Wulf (2016) opine that many organizations have difficulty in mapping UX development with Agile processes for many reasons ranging from hierarchical issues to financial reasons. The same authors make a strong statement that this problem seems more prevalent in small companies as constraints are much tighter. The question that arises is: what are the integration practices for UX and Agile through the lens of a small organisation? Conclusions drawn by authors such as Kuusinen, Mikkonen and Pakarinen (2012) and Stickel *et al.* (2016) suggest that there is no one-size-fits-all template for Agile UX. The diversification of Agile culture, company culture, roles, tools and channels necessitate a case-by-case study (Kuusinen, Mikkonen and Pakarinen 2012; Stickel *et al.* 2016).

Second, the generic framework for Agile UX presented in this thesis was theoretically validated by a focus group. The value of the framework could be explored in a practical setting in order to further validate and possibly refine the framework. Another direction would be to apply the framework to one of the other

Agile methodologies as Scrum was used due to it being the most commonly used Agile methodology. The purpose would be to explore if the proposed framework is effective for other Agile methodologies.

Finally, there is a shortage of research with regards to Agile and UX in the South African context. There is a dire need for statistics on company profiles, Agile methodologies adopted, good practices for integration, project success rates and reasons for success and failure. This should be done on a wider scale within the South African software landscape.

CHAPTER 7: REFERENCES

Abrahamsson, P., Salo, O., Ronkainen, J. and Warsta, J. 2017. Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*: 10-18.

Adikari, S., McDonald, C. and Campbell, J. 2013. Reframed contexts: design thinking for agile user experience design. In: Proceedings of *International Conference of Design, User Experience, and Usability*. Springer, 3-12.

Agile Manifesto. 2001. *Manifesto for Agile Software Development*. Available: http://agilemanifesto.org/ (Accessed 25 July 2018).

Akif, R. and Majeed, H. 2012. Issues and challenges in Scrum implementation. *International Journal of Scientific & Engineering Research*, 3 (8): 1-4.

Al Thunibat, A., Zin, N. A. M. and Sahari, N. 2011. Identifying user requirements of mobile government services in Malaysia using focus group method. *Journal of e-government studies and best practices*, 2011: 1-14.

Ambler, S. W. 2008. Tailoring usability into agile software development projects. *Maturing Usability*: 75-95.

Ardito, C., Buono, P., Costabile, M. F. and Lanzilotti, R. 2012. Are software companies aware of UX? In: Proceedings of *I-UxSED*. 10-13.

Arnold, J. E. 1962. Useful creative techniques. *Source Book for Creative Thinking*: 251-268.

Ballé, F. and Ballé, M. 2005. Lean development. *Business Strategy Review*, 16 (3): 17-22.

Beck, K. 2000. *Extreme programming explained: embrace change*. addison-wesley professional.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. and Jeffries, R. 2001. *Manifesto for Agile Software Development*. Available: http://agilemanifesto.org (Accessed 19 February 2017).

Bevan, N. 2009. What is the difference between the purpose of usability and user experience evaluation methods. In: Proceedings of *Proceedings of the Workshop UXEM*. 1-4.

Beyer, H., Holtzblatt, K. and Baker, L. 2004. An agile customer-centered method: rapid contextual design. *Extreme Programming and Agile Methods-XP/Agile Universe 2004*: 527-554.

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

Boehm, B. and Turner, R. 2003. *Balancing Agility and Discipline: A Guide for the Perplexed, Portable Documents*. Addison-Wesley Professional.

Bordin, S. and De Angeli, A. 2016. Focal points for a more user-centred agile development. In: Proceedings of *International Conference on Agile Software Development*. Springer, Cham, 3-15.

Brhel, M., Meth, H., Maedche, A. and Werder, K. 2015. Exploring principles of user-centered agile software development: A literature review. *Information and software technology*, 61: 163-181.

Brinkmann, S. and Kvale, S. 2015. *InterViews: Learning the craft of qualitative research interviewing*. Sage Publications.

Brosens, J. 2018. The UX entrenchment enigma: why user experience practices are not being incorporated in South African enterprises' Information Systems Development Methodologies (ISDMS). University of Pretoria.

Brown, D. D. 2013. Five agile UX myths. *Journal of usability Studies*, 8 (3): 55-60.

Budwig, M., Jeong, S. and Kelkar, K. 2009. When user experience met agile: a case study. In: Proceedings of *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM, 3075-3084.

Chamberlain, S., Sharp, H. and Maiden, N. 2006. Towards a framework for integrating agile development and user-centred design. *Extreme programming and agile processes in software engineering*: 143-153.

Charmaz, K. and Belgrave, L. L. 2007. *Grounded theory*. Sage Publications.

Charvat, J. 2003. *Project management methodologies: selecting, implementing, and supporting methodologies and processes for projects*. John Wiley & Sons.

Chiyangwa, T. B. and Mnkandla, E. 2017. Modelling the critical success factors of agile software development projects in South Africa. *South African Journal of Information Management*, 19 (1): 1-8.

Cho, J. 2008. Issues and Challenges of agile software development with SCRUM. *Issues in Information Systems*, 9 (2): 188-195.

Chow, T. and Cao, D.-B. 2008. A survey study of critical success factors in agile software projects. *Journal of systems and software*, 81 (6): 961-971.

Cockburn, A. 2002. *Agile software development*. Addison-Wesley Boston.

Cockton, G., Lárusdóttir, M., Gregory, P. and Cajander, Å. 2016. Integrating User-Centred Design in Agile Development. In: *Integrating User-Centred Design in Agile Development*. Springer, 1-46.

Cohn, M. L., Sim, S. E. and Lee, C. P. 2009. What counts as software process? Negotiating the boundary of software work through artifacts and conversation. *Computer Supported Cooperative Work (CSCW)*, 18 (5-6): 401.

Coleman, K. 2018. Designers' influence on attitude change towards user experience (UX) in a software development environment. University of Pretoria.

Convertino, G. and Frishberg, N. 2017. Why agile teams fail without UX research. *Communications of the ACM*, 60 (9): 35-37.

Cooper, D. R., Schindler, P. S. and Sun, J. 2006. *Business research methods*. McGraw-Hill Irwin New York.

Corbin, J. M. and Strauss, A. 1990. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative sociology*, 13 (1): 3-21.

Creasey, T. 2019. *Stop Confusing agile with Agile*. Available: https://empower.prosci.com/stop-confusing-agile-with-agile-thank-you?submissionGuid=54c75a9a-5cd6-4a79-b3ff-3e58ef563b9e (Accessed 2 December 2019).

Creswell, J. W. 2014. *A concise introduction to mixed methods research*. SAGE publications.

Da Silva, T. S., Martin, A., Maurer, F. and Silveira, M. 2011. User-centered design and agile methods: a systematic review. In: Proceedings of *Agile Conference (AGILE), 2011*. IEEE, 77-86.

Dahl, A. 2012. Agile/UX Integration:: how user experience-related practices and processes are integrated with Agile development processes in real-world projects. Masters Thesis, University of Oslo.

Davidson, J. and Jacobs, C. 2008. The implications of qualitative-research software for doctoral work considering the individual and institutional context. *Qualitative Research Journal*, 8 (2): 73-80.

Devadas, U. M., Silong, A. D. and Ismail, I. A. 2011. The relevance of Glaserian and Straussian grounded theory approaches in researching human resource development. *Int. J. Model. Optim*, 11: 348-352.

Dittrich, Y., Randall, D. W. and Singer, J. 2009. Software engineering as cooperative work. *Computer Supported Cooperative Work (CSCW)*, 18 (5-6): 393.

Doody, O. and Bailey, M. E. 2016. Setting a research question, aim and objective. *Nurse researcher*, 23 (4)

Dunne, D. and Martin, R. 2006. Design thinking and how it will change management education: An interview and discussion. *Academy of Management Learning & Education*, 5 (4): 512-523.

Durban Agile User Group. 2018. *Durban Agile User Group Meetup*. Available: https://www.meetup.com/Durban-Agile-User-Group-Meetup/ (Accessed 31 July 2018).

Durban IIBA-SA. 2018. *Durban IIBA-SA Business Analysis Lean Coffee*. Available: https://www.meetup.com/Durban-IIBA-SA-Business-Analysis-Lean-Coffee/ (Accessed 31 July 2018).

e Sá, P. M. 2016. Design Thinking and QFD: two faces of the same coin? In: Proceedings of *Proceedings of the 2nd ICQEM Conference*. ICQEM, 87-99.

Edeki, C. 2015. Agile Software Development Methodology. *European Journal of Mathematics and Computer Science Vol*, 2 (1)

Fauquex, M., Goyal, S., Evequoz, F. and Bocchi, Y. 2015. Creating people-aware IoT applications by combining design thinking and user-centered design methods. In: Proceedings of *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 57-62.

Federoff, M. and Courage, C. 2009. Successful user experience in an agile enterprise environment. In: Proceedings of *Symposium on Human Interface*. Springer, 233-242.

Ferreira, J., Sharp, H. and Robinson, H. 2012. Agile development and user experience design integration as an ongoing achievement in practice. In: Proceedings of *Agile Conference (AGILE), 2012*. IEEE, 11-20.

Fox, B. D. 2010. Agile Methods and User-Centered Design: How These Two Methodologies are Being Integrated in Industry. Doctoral Dissertation, University of Calgary.

Fox, D., Sillito, J. and Maurer, F. 2008. Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. In: Proceedings of *Agile 2008 Conference*. IEEE, 63-72.

Garland, R. 1991. The mid-point on a rating scale: Is it desirable. *Marketing bulletin*, 2 (1): 66-70.

George, K. 2015. The relationship between psychological ownership, work engagement and happiness. Doctoral Dissertation, University of Pretoria.

Golafshani, N. 2003. Understanding reliability and validity in qualitative research. *The qualitative report*, 8 (4): 597-606.

Gorman, G. E., Clayton, P. R., Shep, S. J. and Clayton, A. 2005. *Qualitative research for the information professional: A practical handbook*. Facet Publishing.

Gregor, S. 2006. The nature of theory in information systems. *MIS quarterly*: 611-642.

Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J. and Cajander, Å. 2003. Key principles for user-centred systems design. *Behaviour and Information Technology*, 22 (6): 397-409.

Harbers, M., Detweiler, C. and Neerincx, M. A. 2015. Embedding stakeholder values in the requirements engineering process. In: Proceedings of *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, 318-332.

Hassenzahl, M. and Tractinsky, N. 2006. User experience-a research agenda. *Behaviour & information technology*, 25 (2): 91-97.

Heikkinen, J., Olsson, T. and Väänänen-Vainio-Mattila, K. 2009. Expectations for user experience in haptic communication with mobile devices. In: Proceedings of *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 28.

Highsmith, J. A. 2002. *Agile software development ecosystems*. Addison-Wesley Professional.

Humayoun, S. R., Catarci, T., Kimani, S. and Dubinsky, Y. 2017. Managing User-Centered Design in Agile Projects.

Hussain, Z., Slany, W. and Holzinger, A. 2009. Investigating agile user-centered design in practice: A grounded theory perspective. In: Proceedings of *Symposium of the Austrian HCI and Usability Engineering Group*. Springer, 279-289.

Innes, J. 2011. Why enterprises can't innovate: helping companies learn design thinking. In: Proceedings of *International Conference of Design, User Experience, and Usability*. Springer, 442-448.

Interaction Design Foundation (IDF) Durban Group. 2018. *Interaction Design Foundation (IDF) Durban Group*. Available: https://www.linkedin.com/groups/13577628 (Accessed 3 August 2018).

ISO. 2009. 9241-210: 2009. Ergonomics of human system interaction-Part 210: Human-centered design for interactive systems (formerly known as 13407). *International Organization for Standardization (ISO). Switzerland*,

John, M., Maurer, F. and Tessem, B. 2005. Human and social factors of software engineering: workshop summary. *ACM SIGSOFT Software Engineering Notes*, 30 (4): 1-6.

Jönsson, E. 2013. Agile Development and User-Centered Design-a case study at Sony Mobile Communications AB. *LUTMDN/TMAT-5168-SE*,

Joseph, N. M. C. and Santana, M. 2016. Agile software development and IT project performance in South Africa: a positive relationship. In: Proceedings of *The 25th International Conference for Management of Technology*. 335-355.

Jurca, G., Hellmann, T. D. and Maurer, F. 2014. Integrating Agile and User-Centered Design. In: Proceedings of *Agile Conference*.

Kane, D. 2003. Finding a place for discount usability engineering in agile development: throwing down the gauntlet. In: Proceedings of *Agile Development Conference, 2003. ADC 2003. Proceedings of the*. IEEE, 40-46.

Kashfi, P., Nilsson, A. and Feldt, R. 2017. Integrating User eXperience practices into software development processes: implications of the UX characteristics. *PeerJ Computer Science*, 3: e130.

Kheong, L. S. and Jayaratna, N. 2009. *Framework for structuring learning in problem-based learning*. McGraw Hill.

Khramtchenko, S. 2004. Comparing eXtreme Programming and Feature Driven Development in academic and regulated environments. *Feature Driven Development*,

Kieffer, S., Ghouti, A. and Macq, B. 2017. The agile ux development lifecycle: Combining formative usability and agile methods.

Kitchenham, B. A. and Pfleeger, S. L. 2002. Principles of survey research part 2: designing a survey. *ACM SIGSOFT Software Engineering Notes*, 27 (1): 18-20.

Kollmann, J. 2008. Designing the user experience in an agile context. Master's dissertation (HCI and Ergonomics), UCL.

Kollmann, J., Sharp, H. and Blandford, A. 2009. The importance of identity and vision to user experience designers on agile projects. In: Proceedings of *Agile Conference, 2009. AGILE'09.* IEEE, 11-18.

Kontio, J., Lehtola, L. and Bragge, J. 2004. Using the focus group method in software engineering: obtaining practitioner and user experiences. In: Proceedings of *Proceedings. 2004 International Symposium on Empirical Software Engineering, 2004. ISESE'04.* IEEE, 271-280.

Kuusinen, K., Mikkonen, T. and Pakarinen, S. 2012. Agile User Experience Development in a Large Software Organization: Good Expertise but Limited Impact. *Human-Centered Software Engineering: 4th International Conference, HCSE 2012, Toulouse, France, October 29-31, 2012. Proceedings*: 94-111.

Kuusinen, K. and Väänänen-Vainio-Mattila, K. 2012. How to make agile UX work more efficient: management and sales perspectives. In: Proceedings of *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*. ACM, 139-148.

Kuusinen, K., Väätäjä, H., Mikkonen, T. and Väänänen, K. 2016. Towards understanding how agile teams predict user experience. In: *Integrating User-Centred Design in Agile Development*. Springer, 163-189.

Larman, C. and Basili, V. R. 2003. *Iterative and incremental development: a brief history*. Available: https://pdfs.semanticscholar.org/058f/712a7dd173dd0eb6ece7388bd9cdd6f77d67.pdf (Accessed 28 February 2018).

Lárusdóttir, M. K., Cajander, Å. and Gulliksen, J. 2012. The big picture of UX is missing in Scrum projects. In: Proceedings of *Proceedings of the International Workshop on the Interplay between User Experience (UX) Evaluation and System Development (I-UxSED)*. 49-54.

Law, E. L.-C., Abrahão, S., Vermeeren, A. and Hvannberg, E. T. 2012. Interplay between user experience evaluation and system development: state of the art. In: Proceedings of *International Workshop on the Interplay between User Experience (UX) Evaluation and System Development (I-UxSED 2012)*. 14-17.

Law, E. L.-C., Roto, V., Hassenzahl, M., Vermeeren, A. P. and Kort, J. 2009. Understanding, scoping and defining user experience: a survey approach. In: Proceedings of *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 719-728.

Law, E. L.-C. and van Schaik, P. 2012. To Measure or not to Measure UX: An Interview Study. In: Proceedings of *I-UxSED*. 58-63.

Lawrence, J. and Tar, U. 2013. The use of grounded theory technique as a practical tool for qualitative data collection and analysis. *The Electronic Journal of Business Research Methods*, 11 (1): 29-40.

Lee, A. S. and Baskerville, R. L. 2003. Generalizing generalizability in information systems research. *Information systems research*, 14 (3): 221-243.

Leffingwell, D. 2016. *User Experience – Scaled Agile Framework*. Available: http://www.userzoom.com/agile-ux-research-and-testing/integrating-ux-into-an-agile-environment/ (Accessed 12 March 2018).

Liker, J. K. and Morgan, J. M. 2006. The Toyota way in services: the case of lean product development. *Academy of management perspectives*, 20 (2): 5-20.

Lin, Y., Hilaire, V., Gaud, N. and Koukam, A. 2011. Scrum conceptualization using K-CRIO ontology. In: Proceedings of *International Symposium on Data-Driven Process Discovery and Analysis*. Springer, 189-211.

Lindell, R. 2012. The craft of programming interaction. In: Proceedings of *7th Nordic Conference on Human-Computer Interaction (NordiCHI 2012)*. 26-30.

Lodato, T. J. 2015. The work of user experience design: materiality and cultures in designing. Doctoral Dissertation, Georgia Institute of Technology.

Martin, A., Biddle, R. and Noble, J. 2004. The XP customer role in practice: Three studies. In: Proceedings of *Agile Development Conference, 2004*. IEEE, 42-54.

Mason, M. 2010. Sample size and saturation in PhD studies using qualitative interviews. In: Proceedings of *Forum qualitative Sozialforschung/Forum: qualitative social research*.

Massey, O. T. 2011. A proposed model for the analysis and interpretation of focus groups in evaluation research. *Evaluation and program planning*, 34 (1): 21-28.

Mayring, P. 2007. On generalization in qualitatively oriented research. In: Proceedings of *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research*.

McInerney, P. and Maurer, F. 2005. UCD in agile projects: dream team or odd couple? *interactions*, 12 (6): 19-23.

McKendrick, J. 2018. *More work required to understand the user in UX, survey shows*. Available: https://www.zdnet.com/article/making-user-experience-an-agile-pursuit/ (Accessed 15 June 2019).

Miller, L. 2005. Case study of customer input for a successful product. In: Proceedings of *Agile Development Conference (ADC'05)*. 24-29 July 2005. 225-234.

Mnkandla, E. and Dwolatzky, B. 2004. A survey of agile methodologies. *The transactions of the SA institute of electrical engineers*, 3: 236-247.

Mudarikwa, G. and Grace, T. D. 2018. Agile system development methodologies usage and acceptance in South African banking firms: an exploratory analysis. In: Proceedings of *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists*. ACM, 248-257.

Muhaiyuddin, N. S. M., Bakar, H. S. A. and Hussin, H. 2016. The multiple approaches of grounded theory: Justification for Straussian version. *PEOPLE: International Journal of Social Sciences*, 2 (1)

Nadikattu, S. R. 2016. Integrating User Experience (UX) Development with Agile SoftwareDevelopment Practices.: A Multiple Case Study Involving Organizations DevelopingInteractive Healthcare Technology (IHT) Applications.

Näkki, P., Koskela, K. and Pikkarainen, M. 2011. Practical model for user-driven innovation in agile software development. In: Proceedings of *Concurrent Enterprising (ICE), 2011 17th International Conference on*. IEEE, 1-8.

Nielsen Norman Group. 2019. *World Leaders in Research-Based User Experience*. Available: https://www.nngroup.com/ (Accessed 11 June 2019).

Norman, D., Miller, J. and Henderson, A. 1995. What you see, some of what's in the future, and how we go about doing it: HI at Apple Computer. In: Proceedings of *Conference companion on Human factors in computing systems*. ACM, 155.

Olsson, H. H., Alahyari, H. and Bosch, J. 2012. Climbing the" Stairway to Heaven"--A Mulitiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software. In: Proceedings of *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on*. IEEE, 392-399.

Oprins, R. J., Frijns, H. A. and Stettina, C. J. 2019. Evolution of Scrum Transcending Business Domains and the Future of Agile Project Management. In: Proceedings of *International Conference on Agile Software Development*. Springer, 244-259.

Patel, S. 2015. *The research paradigm - methodology, epistemology and ontology - explained in simple language*. Available: http://salmapatel.co.uk/academia/the-research-paradigm-methodology-epistemology-and-ontology-explained-in-simple-language (Accessed 15 July 2018).

Pillay, N. and Wing, J. 2019. Agile UX: Integrating good UX development practices in Agile. In: Proceedings of *2019 Conference on Information Communications Technology and Society (ICTAS)*. IEEE, 1-6.

Poppendieck, M. and Poppendieck, T. 2003. *Lean Software Development: An Agile Toolkit: An Agile Toolkit*. Addison-Wesley.

QSR International Pty Ltd. 2017. N-Vivo (computer software). Available: https://www.qsrinternational.com/ (Accessed 30 October 2018).

Salah, D., Paige, R. and Cairns, P. 2015. Patterns for integrating agile development processes and user centred design. In: Proceedings of *Proceedings of the 20th European Conference on Pattern Languages of Programs*. ACM, 19.

Salah, D., Paige, R. F. and Cairns, P. 2014. A systematic literature review for agile development processes and user centred design integration. In: Proceedings of *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. ACM, 5.

Saunders, M., Lewis, P. and Thornhill, A. 2009. *Research methods for business students*. Pearson education.

Sbaraini, A., Carter, S. M., Evans, R. W. and Blinkhorn, A. 2011. How to do a grounded theory study: a worked example of a study of dental practices. *BMC medical research methodology*, 11 (1): 128.

Scaled Agile Academy. 2019. *Scaled Agile Framework (SAFe)*. Available: https://www.scaledagileframework.com/ (Accessed 11 June 2019).

Schach, S. R. 2007. *Object-oriented and classical software engineering*. McGraw-Hill New York.

Schön, E.-M., Thomaschewski, J. and Escalona, M. J. 2017. Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces*, 49: 79-91.

Schwaber, K. 1997. Scrum development process. In: *Business object design and implementation*. Springer, 117-134.

Schwartz, L. 2013. Agile-User Experience Design: an Agile and User-Centered Process? In: Proceedings of *Proc. the 8th International Conference on Software Engineering Advances*. 346-351.

ScrumAlliance. 2009. *The smooth corner*. Available: http://scrummaster.no/2009/vi-er-verdens-smidigste-nasjon/ (Accessed 25 May 2019).

Seale, C., Gobo, G., Gubrium, J. F. and Silverman, D. 2004. *Qualitative research practice*. Sage.

Sebega, Y. and Mnkandla, E. 2017. Exploring Issues in Agile Requirements Engineering in the South African Software Industry. *The Electronic Journal of Information Systems in Developing Countries*, 81 (1): 1-18.

Sharp, H., Dittrich, Y. and De Souza, C. R. 2016. The role of ethnographic studies in empirical software engineering. *IEEE Transactions on Software Engineering*, 42 (8): 786-804.

Singh, M. 2008. U-SCRUM: An agile methodology for promoting usability. In: Proceedings of *Agile, 2008. AGILE'08. Conference*. IEEE, 555-560.

Sohaib, O. and Khan, K. 2010. Integrating usability engineering and agile software development: A literature review. In: Proceedings of *2010 international conference on Computer design and applications*. IEEE, V2-32-V32-38.

Spence, J. W. 2005. There has to be a better way![software development]. In: Proceedings of *Agile Development Conference (ADC'05)*. IEEE, 272-278.

Spool, J. M. 2018. *Agile Isn't Supposed To Be UX Hostile*. Available: https://medium.com/user-interface-22/agile-isnt-supposed-to-be-ux-hostile-756c92fc4d5f (Accessed 15 June 2019).

Stevens, E. 2018. *The Fascinating History of UX Design: A Definitive Timeline*. Available: https://careerfoundry.com/en/blog/ux-design/the-fascinating-history-of-ux-design-a-definitive-timeline/ (Accessed 04 March 2019).

Stickel, O., Ogonowski, C., Jakobi, T., Stevens, G., Pipek, V. and Wulf, V. 2016. User Integration in Agile Software Development Processes: Practices and Challenges in Small and Medium Sized Enterprises. In: *Integrating User-Centred Design in Agile Development*. Springer, 49-76.

Subair, S. 2014. The Evolution of Software Process Models: From the Waterfall Model to the Unified Modelling Language (UML). *International Journal of Information Technology & Systems*,

Sy, D. 2007. Adapting usability investigations for agile user-centered design. *Journal of usability Studies*, 2 (3): 112-132.

Thabane, L., Thomas, T., Ye, C. and Paul, J. 2009. Posing the research question: not so simple. *Canadian Journal of Anesthesia/Journal canadien d'anesthésie*, 56 (1): 71.

Travis, D. 2011. *ISO 13407 is dead. Long live ISO 9241-210!* Available: http://www.userfocus.co.uk/articles/iso-13407-is-dead.html (Accessed 26 September 2017).

Treder, M. 2013. *Lean UX vs. Agile UX – is there a difference?* Available: https://www.uxpin.com/studio/blog/lean-ux-vs-agile-ux-is-there-a-difference/ (Accessed 14 January 2019).

Vaismoradi, M., Turunen, H. and Bondas, T. 2013. Content analysis and thematic analysis: Implications for conducting a qualitative descriptive study. *Nursing & health sciences*, 15 (3): 398-405.

Verdiesen, B. 2014. Agile user experience. MSc dissertation, Radboud University Nijmegen, Nijmegen.

Version One. 2019. *State of Agile Report*. Available: https://www.stateofagile.com/#ufh-c-473508-state-of-agile-report (Accessed 10 June 2019).

CHAPTER 8: APPENDIX

8.1. The Agile Manifesto

The Agile manifesto was during a meeting of seventeen independent-minded software practitioners in 2001. The manifesto reads as follows:
“We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

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

That is, while there is value in the items on the right, we value the items on the left more.” (Agile Manifesto 2001: 1)

The group of practitioners also described twelve principles which support the four main values. These principles aim to help Agile teams to adhere to the four key values. As per Agile Manifesto (2001) these are the twelve principles (the numbering for the principles has been added by the author for ease of reference):

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome change requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support their need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

8.2. Classification of Previous Studies

Table 30: Classification of previous studies

Author	Title of publication	Nature of publication	Keywords Identified	Research Method Used
Ambler (2008)	Tailoring usability into agile software development projects	Journal Paper	Interaction Design, User Story, Agile Method, Capability Maturity Model, Agile Software Development	Framework
Beyer, Holtzblatt and Baker (2004)	An agile customer-centered method: rapid contextual design	Journal Paper	Agile, User-Centred Design, Systems Design, Contextual Design	Framework
Boehm (2002)	Get ready for agile methods, with care	Journal Paper	Planning Spectrum, Developers, Customers, Architecture, Balancing Agility and Discipline	Evaluation Study
Budwig, Jeong and Kelkar (2009)	When user experience met agile: a case study	Conference Paper	Agile, Scrum, Agile Programming, User Experience, UX teams	Case Study
Chamberlain,	Towards a	Conference	User-Centred Design,	Evaluation

Sharp and Maiden (2006)	framework for integrating agile development and user-centred design	Paper	Agile, Usability Experts, UCD	Study/Framework
Convertino and Frishberg (2017)	Why agile teams fail without UX research	Journal Paper	Agile, UX Research, End-user, UX Design	Evaluation Study
Federoff and Courage (2009)	Successful user experience in an agile enterprise environment	Conference Paper	Agile, waterfall, enterprise, user experience, design, research	Case Study
Ferreira, Sharp and Robinson (2012)	Agile development and user experience design integration as an on-going achievement in practice	Conference Paper	Agile development, User Experience Design, ethnographically-informed, integration, organisation	Ethnographic Field Studies
Fox (2010)	Agile Methods and User-Centered Design: How These Two Methodologies are Being Integrated in Industry	Journal Paper	User centred design, software engineering, software application, agile methods	Framework
Harbers, Detweiler and Neerincx (2015)	Embedding stakeholder values in the requirements engineering process	Conference Paper	Sensitive Design, Requirement Engineer, Requirement Elicitation, User Story, Elicitation Technique	Evaluation Study
Hussain, Slany and Holzinger (2009)	Investigating agile user-centered design in practice: A grounded theory perspective	Conference Paper	Agile Methods, Extreme Programming, Scrum, Usability, User-Centered Design	Evaluation Study
Kane (2003)	Finding a place for discount usability engineering in agile development: throwing down the gauntlet	Conference Paper	Discount Usability Engineering, Agile, Visibility, Users, Gaps	Evaluation Study/Framework

Kieffer, Ghouti and Macq (2017)	The agile ux development lifecycle: Combining formative usability and agile methods	Conference Paper	Agile UX, Cycles, Cross-functional, Sprints, Design	Case Study
Kollmann (2008)	Designing the user experience in an agile context	Conference Paper	qualitative study, interaction design	Evaluation Study
Kuusinen, Mikkonen and Pakarinen (2012)	Agile User Experience Development in a Large Software Organization: Good Expertise but Limited Impact	Conference Paper	User experience (UX), Agile development, human-centered design (HCD), human-computer interaction (HCI)	Case Study
Larman and Basili (2003)	Iterative and incremental development: a brief history	Journal Paper	Agile, Values, Principles, Methodology, Iterative, Incremental, Qualitative comparative analysis, Time boxing, Software engineer	Framework
Law et al. (2009)	Understanding, scoping and defining user experience: a survey approach	Conference Paper	Definition, User experience, Survey, ISO, Usability	Survey
Lodato (2015)	The work of user experience design: materiality and cultures in designing	PHD Dissertation	User Experience, UX, UX Design, Human-Computer Interaction, Design Science	Case Study
Martin, Biddle and Noble (2004)	The XP customer role in practice: Three studies	Conference Paper	Agile Process, XP, Customer, Sustainability	Case Study
McInerney and Maurer (2005)	UCD in agile projects: dream team or odd couple?	Journal Paper	Agile, Scrum, UCD, Customer, Domain Expert, UI Design	Case Study
Miller (2005)	Case study of customer input for a successful product	Conference Paper	Usability, Product, Agile, Customer, Focus Groups,	Case Study

			Demos, Tests	
Mnkandla and Dwolatzky (2004)	A survey of agile methodologies	Journal Paper	Agile methodologies, Traditional Methodologies, Iterative Incremental Development	Evaluation Study
Nadikattu (2016)	Integrating User Experience (UX) Development with Agile Software Development Practices	Case Study	User Experience, Agile Software Development, Usability, User-Centered Design	Case Study
Näkki, Koskela and Pikkarainen (2011)	Practical model for user-driven innovation in agile software development	Conference Paper	Programming, Technological innovation, Companies, Media, Lead, Software, Context	Empirical Case Study
Olsson, Alahyari and Bosch (2012)	Climbing the" Stairway to Heaven"--A Mulitiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software	Conference Paper	agile software development, customer collaboration, continuous integration, continuous deployment	Case Study
Salah, Paige and Cairns (2015)	A systematic literature review for agile development processes and user centred design integration	Conference Paper	Agile Software Development Processes, User Centred Design, Agile User Centred Design Integration	Evaluation Study
Schön, Thomaschewski and Escalona (2017)	Agile Requirements Engineering: A systematic literature review	Journal Paper	Systematic literature review, Agile software development, Requirements	Evaluation Study

			Engineering	
Schwartz (2013)	Agile-User Experience Design: an Agile and User-Centered Process?	Conference Paper	Agile-UX, Agile Software Techniques, Software Engineering, User-Centered Design	Evaluation Study
Singh (2008)	U-SCRUM: An agile methodology for promoting usability	Conference Paper	usability, user experience, product owner, roles, two, dual, process	Framework
Sy (2007)	Adapting usability investigations for agile user-centered design	Journal Paper	Usability method, Agile, XP, iterative development, software, case study, field study	Case Study
Verdiesen (2014)	Agile user experience	MSc Dissertation	Agile Software Development, User Experience, Customer, User	Evaluation Study

8.3. Interview Questions

Research Objectives

The objective of conducting interviews and sending questionnaires is to answer the main research question, and the sub-questions:

The research question is as follows:

“How can good UX development practices be integrated into Agile within the South African context?”

The following sub questions have led to conclusion of the main question:

1. “What Agile development techniques are best suited to ensure UX requirements are met?”
2. “What are good development practices for creating good UX?”
3. “What are the roles of team members that are directly involved in the process of integrating UX in Agile?”
4. “What are the problems when integrating UX in Agile?”

Interviews:

Outcomes:

The following is the desired outcomes from interviewing participants:

1. The participant's understanding of Agile – do they see it as a framework, process, way of working, philosophy?
2. How is UX design methods used? Do they work?
3. The level of user involvement, and if the participant is satisfied with the level of interaction with users.
4. The participant's satisfaction with the Agile – how it affects the team as a whole.
5. How does UX design methods in Agile affect the quality of the software?
6. How does UX design methods in Agile affect the user acceptance of the software?
7. The participant's work environment – does the UX team work independently or do they engage with the developer, Agile team members?
8. What does the participant recommend as changes, improvements? What are some of the good practices?

Interview Questions:

There are four main, open ended questions followed by sub-questions that will tie up to the main question.

Question 1: Can you tell me about your job role, background in terms of employment, the type of projects you work on?

- What is your role in the team?
- How many years' experience do you have working in this position?
- What type of projects are you working on e.g. Single or multiple projects, Agile or Waterfall?
- Could you describe what "Agile" means to you?
- Could you describe the role of the "User" in the project?
- Could you describe the role of the "Customer" in the project?
- Is the "User" and the "Customer" the same person in the project? If yes, at which point do they get involved in the project?

- Do you work with UX specialists when developing, for example the UI for the project? If yes, how long have you been doing this?

Question 2: What are the processes and the activities when developing the software?

- What is the first phase of project?
- What happens up front, and when do you get involved?
- Who are at the planning meetings?
- Who are the members of your team and what are their roles?
- Which members of the team are you readily interacting with?
- What Agile methodologies and tools are used for the projects?
- What user-centred design methods do you apply?
- Are there any interactions/processes between UX team and developers?
- Do you user real users in your UCD testing?
- How do you go about testing?
- Is there enough end-user involvement? If yes, why? If no, why not?
- Do you plan to increase end-user involvement in projects? Please elaborate.
- What artifacts do you use in the project (e.g. whiteboards, use cases, user stories, storyboards, personas, navigation maps). What is the purpose for these artifacts
- How satisfied are you with work carried out by yourself (and the team) on this project/s?

Question 3: Can you describe how UX activities are integrated into the software development cycles?

- What are the common processes that occur in an iteration?
- How long are typical iterations?
- How are UX requirements handled in a project? Are there any artifacts or deliverables that are required (e.g. UI specifications)
- Does the UCD activities include the specification of the users' needs?

- Is the user involved in the design of the UI?
- Is the user involved in the evaluation of the UI?
- Is the design based on explicit understanding of users?
- Is the design based on explicit understanding of tasks?
- Is the design based on explicit understanding of environment?
- Do UX activities follow an iterative process?
- In a typical iteration is the UX team working ahead of a development cycle in terms of UI testing and design?
- What effect does UX testing have on product backlog? How are delays mitigated?
- Does the UCD get passed before the end of the sprint for some sort of UI testing or verification, or is it all just done at the end?
- Are there tools that are used to integrate UX and Agile development?
- Do you feel Agile methods is a successful development process?
- Is the addition of UX to the Agile methods effective?
- When does the UCD design get passed to development?
- How much time do you spend before development begins?
- Is there a UX person(s) throughout the development lifecycle?
- How often do you communicate with the UX/Agile team?
- Does the UCD process address the whole user experience?
- Doe the team include multidisciplinary skills?

Question 4: How has the integration of UX in Agile affected the software development process?

- Is there any resistance from other team member with UX persons?
- Are there issues up front with integrating UX processes?
- Do you feel that Agile and UX fit together well? What are the pros and cons?
- Has the UX team member/s been included from the beginning of the project?
- What did you do to overcome the obstacles of bringing UX into your process?
- Did bringing in UX impact everybody in a positive perspective?

- Do you find that the development is more successful using UX and Agile together?
- Do you intend on keeping your current practices in your development process?
- Do you think the integration of UX in Agile projects could be improved? How so?

Question 5: Are the following Agile values followed during integration of UX in the project?

- Individuals and interactions over processes and tools? Describe the individuals and type of interactions that occur between Agile and UX teams.
- Working software over comprehensive documentation? Please elaborate on how the lack of documentation has affected the project.
- Customer collaboration over contract negotiation? Who is the customer and who is the user?
- Responding to change over following a plan? How is change managed in the project to ensure Agile and UX processes remain on track?

8.4. Questionnaire

Privacy Statement

I am a student at the Durban University of Technology and I am completing my dissertation as part of my Masters in Information and Communications Technology. The topic of the study is “Agile User Experience: Integrating good User Experience development practices into Agile within the South African context”.

The aim of this survey is to identify current methods of integrating User Experience (UX) in Agile. It also aims to confirm if the previous studies conducted applies within a South African context.

Please do not write any identifying marks on the survey as participants are meant to be anonymous.

All information will be kept confidential.

Any concerns can be communicated to Narendren Pillay (email: pillay.naren@gmail.com).

Thank you for your time and cooperation.

Based on your experience, answer the following questions by selecting the most appropriate answer

1. What is your role in the team?
2. How many years’ experience do you have working in this position?

3. In order to successfully integrate a software solution you need information about the environment it will be deployed in.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

4. To create the best User Experience, input is required from as many different users as possible.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

5. Developers in Agile teams focus on the details of a solution and often lose sight of the project as a whole (i.e. working on small pieces of functionality results in a loss of holistic view).

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

6. When working in an Agile team, UX insights should be encouraged during development in order to improve the quality of the software.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

7. Close collaboration between UX and Agile team members assists in defining clear requirements.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

8. Thinking about the motivation and needs of your end-users helps produce a better User Experience in the end product.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

9. Having a UX expert as part of the Agile team improves the quality of the software.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

10. Involving users early in the project minimizes design wastage, and results in improved designs and functionality of the software.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

11. Including visual design elements into sprints improves the user acceptance of the software product.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

12. Including visual design elements into sprints improves the quality of the software product.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

13. Including navigation elements into sprints improves the user acceptance of the software product.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

14. Including navigation elements into sprints improves the quality of the software product.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

15. Including User Interface elements into sprints improves the user acceptance of the software product.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

16. Including User Interface elements into sprints improves the quality of the software product.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

17. The UX team should work at least one sprint ahead of the development sprint as the developers are aware of design progression, and testing and feedback can be given early.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

18. The UX team should work at least two sprint ahead of the development sprint as the developers are aware of design progression, and testing and feedback can be given early.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

19. An increase in end-user involvement in the project will improve the testing of the software product.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

20. An increase in end-user involvement in the project will improve the user acceptance of the software product.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

21. Face to face interaction between UX and Agile team members assists in clearing up problems much faster.

Disagree Strongly	Disagree Moderately	Disagree Slightly	Agree Slightly	Agree Moderately	Agree Strongly
○	○	○	○	○	○

22. How many years of experience do you have within the field of Agile Software development

None	<1	1-3	4-6	>6
○	○	○	○	○

23. How many years of experience do you have within the field of User Experience

None <1 1-3 4-6 >6

○ ○ ○ ○ ○

24. How many years of experience do you have combining the fields of User Experience and Agile Software development

None <1 1-3 4-6 >6

○ ○ ○ ○ ○

25. Is there anything omitted from this survey that you feel should have been included?

26. Do you have any final comments about this survey or on the topic of integrating UX in Agile?

Thank you for participating in this survey. Your assistance is greatly appreciated.

For any further information, you can contact me at [pillay.naren@gmail.com]

8.5. Sample Email for the Questionnaires

Survey Invitation: Agile UX

Naren Pillay <pillay.naren@gmail.com> Jan 31, 2019, 9:18 PM

Hi,

I would appreciate your feedback in this online survey for research on the topic of *Agile UX*. The survey will take less than 5 minutes to complete and all responses will remain confidential and secure.

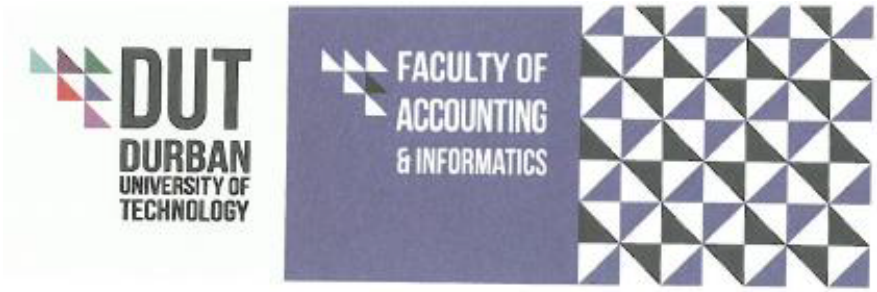
Please click on this link below to complete the survey, and please feel free to share this survey with your team/colleagues who have experience in Agile UX.

https://www.questionpro.com/t/ANc57Zb0em

Thanking you in advance for your response.

Regards,
Naren Pillay

8.6. Letter of Consent for Interviews



26 April 2017

Dear Sir/Madam,

RE: PERMISSION TO CONDUCT RESEARCH FOR MASTERS QUALIFICATION

Mr Narendren Pillay is completing his Masters qualification in Information and Communications Technology at the Durban University of Technology. He is required to conduct research on the topic of his choice that will contribute to his thesis, and also provide knowledge/confirm theories from previous research conducted.

The title of his research is as follows:

Agile User Experience: Integrating good User Experience development practices into Agile within the South African context

The concept of integrating User Experience (UX) into Agile is one of critical components in ensuring that the software meets the user's needs. The research will assist in identifying good development practices, as well as confirm if previous studies conducted applies within the South African context.

Furthermore, the study will contribute to academic knowledge and will provide a possible framework for integrating UX into Agile.

Permission is hereby requested to conduct research within your organisation. Responses are highly confidential and will be treated as such. The study will also not make direct reference to any of the projects, customers, employees etc.

Thank you in advance for your assistance and co-operation.

Dr J Wing

Supervisor

jwing@dut.ac.za

Dr D. Heukelman

Faculty Research Coordinator (Acting)

8.7. Open Codes

Questions from interviews	Open Code
Role – what is your role in the team?	BA Manager
	Scrum Master
	Project Manager
	Software Developer
	Senior Software Development Manager
	Agile Coach
	Quality Assurance Manager
	UX Analyst
	Business Analyst
	Back Office Team Lead
	Working experience
Type of projects	Multiple projects
	Water-Scrum-Fall
	Agile
	Hybrid
What “Agile” means to you?	Framework
	A way of working to get better software
	Getting a better product out
	Getting products quicker to the user
	Making changes easier
	Creates team spirit
	Creates team morale
	Creates a self-organised team
	Safe environment for teams to communicate
	Shared knowledge
	Collaboration
	Responsive way of working
	Flexibility within current business needs
	Scalability
	Comply in terms of the Agile Manifesto
	“Agile is something we are, Scrum is something we do.”
	Adjust your direction without costing a fortune
	“Turning a dime for a dime.”
	Breaking down bigger projects into manageable chunks

	Estimating so that project remains on track
	Methodology with various frameworks
	Realise value sooner rather than later
	Taking feedback and using it to steer the project forward
	Iterative
	Learning
	Short, sharp deployment in the quickest possible time
Role of the “User” in the project	No active user involvement
	Minimal
	“As needed” basis
	Validates the understanding of requirements
	“Sounding board”
	Supportive role to Product Owner and the BA
	Validate software that has been developed
	End-user
	Product Owner team
	Upfront involvement for requirements elicitation
	Testing before sprint review
	Sprint review
	Feedback on mock-ups
	Late involvement
	Less engagement with users in external clients
	Alpha testing and feedback
	Involved at the end of the project
	Not enough user research
	User is the product owner
	Non-existent
	Proxy as a representative of the user (e.g. BA or Product Owner)
	Uses the end-product
	Finds value in what is created for them
Role of the “Customer” in the project	Sponsor
	Same as the user
	Non-existent
	Designed with an assumed behaviour in mind
	Feedback mechanism
	Internal clients – same as the user. External client

	– varies
	External Client – same as user
	Proxy to the user
For projects where the user and the customer is the same person, at which point do they get involved?	Beta testing and feedback
	Project kickoff
	During the “imagining” phase
	Demo’s
	Sprint retrospect
	Requirements elicitation
	Varies – 10 to 15% of external projects engage early with users
	After initiation and before design phase
	Too late in the project
	From product vision
	Involved early as possible but no collaboration during development
UX specialists’ involvement during development	No UX specialist involved
	BA fills in for UX specialist
	In the UK, the UX specialist was involved
	External clients – Yes, Internal clients – No.
	40% of projects – Yes, the balance – No.
	"Users, this is all we have time to develop, so use it or you have nothing else to use."
	Yes – for internal project, upfront with user research. External projects – very rare to have UX specialists.
	UX specialist tend to work very traditionally, instead of being Lean or Agile UX
	Too much analysis upfront
	Mostly for external projects – employ consultants/contractors
	40% of projects have UX specialists
	Yes – based abroad hence difficulty to schedule UX work.
	Yes – form part of the Experience Design Team
	UX has been included for the last 2 years
	No – defined by the user and the development team
	Yes – for the last 1 and half years

First phase of project	Requirements gathering
	Project prioritization
	Project kick-off
	Project lift off
	Sprint zero
	Concept phase
	Product vision
	Briefing phase
	Idea phase
What happens upfront?	Prioritize features
	Two sprints ahead for detail/requirements
	Solution design
	Initial engagement
	Requirements gathering
	User stories
	Epics
	High level requirements gathering
	Discussions between product owner, customer and execs followed by user requirements
	Breaking down the idea or product vision into requirements
When do you get involved?	BA Manager - From the beginning
	Scrum Master – solution design
	Junior/Intermediate developer – after requirements gathering
	Senior developer – requirements phase
	Senior Software Development Manager – after the requirements phase
	Agile Coach – at any point of the process, internal projects – project lift-off, external projects - varies
	Quality Assurance Manager – first sprint. However, testing happens later
	UX Analyst – after requirement phase
	Scrum Master – product backlog defining and unpacking
	Business Analyst – requirements phase
	Back Office Team Lead – requirements phase after product vision has been defined
Who are at the planning meetings?	Scrum Master

	Tester
	BA
	Product Owner
	Developer
	Product Manager
	Product Owner
	Team Leaders
	System Testers
	Technical Lead
	Development Manager
	Agile Coach
	Testers
	UX Designers
	Architects
	Representatives from the business
	Test Lead
	Programme Manager
	UX Analyst
	Senior Developers
Members of the team	Agile Coach
	Architect
	Architects
	Automation Engineer
	Business Analyst
	Business Representatives
	Developers
	Development Manager
	Entry level tester
	Experience Analyst
	Functional tester
	Interaction Design
	Non-Functional Tester
	Product Owner
	Project Manager
	QA
	Scrum Master
	System Testers
	Team manager
	Technical Lead

	Test Lead
	Tester
	UX Analyst
	UX Designers
Who are you readily interacting with?	All members of the team
	Everyone
	Developer
	Testers
	Business Analyst
	Product Owner
	Scrum Master
	Product Owner
	Agile Coach
	Scrum Master
	UX Designers
	UI Designers
	Architects,
	Business Representatives,
	Technical Lead
	Portfolio Manager
	Test Lead
	Programme Manager
Agile Methodology used	Scrum (all companies)
	Kanban (for production only)
	Sprint planning
	backlog grooming
	sprint planning
	daily stand-up
	review
	demo
	retrospective
Tools used	Jira
	Asana
	Trello
	Confluence
	Microsoft word
	PDF
	GIT (source code management)
	Slack

	TFS
	Microsoft Team Foundation Services
	Streams
	Axosoft
User-Centred design methods applied?	Native design app used
	Screen mock-ups
	Low fidelity prototypes (rarely due to cost)
	Sketches
	User research
	User validation
	Journey maps
	Interviews
	Wireframes
	Balsamiq (low fidelity)
	Justinmind
	Invision (high fidelity)
Any interactions/processes between UX team and development	None
	During the testing phase
	Frequent meetings with development team (users)
	Too long wait for users to review at the end of a sprint
	Meeting more often ensures that development remains on track
	Separate UX team – handoff of specifications to development and it doesn't work well
	If you want speed and quality, the UX person should be in the same team
	Experience Design Team interacts with the developers to discuss functionality and user interface
	All sprints
	Front end development
Do you user real users in your UCD testing?	No (50%)
	Yes – very late in user acceptance testing
	Internally – yes. Externally – it varies. However, in most cases real users are not used in testing
	Yes – SME (Subject matter experts)
	No – the Product Managers or the BA usually does this

Testing process	Test team does the testing
	Outsourced test team
	UAT with clients
	No user involved
	Tester builds test script based on functional specifications then test
	A representative of the user will do the testing (e.g. Business Analyst)
	Developer does initial testing by building a test script then pass to test team
	Regression testing, stress testing, functional testing, unit testing by test team
	Core testing group “The Champions” – business wide that does the testing and sign off
	Pilot testing
	Test cases drawn up – test after development complete
	Spira – testing tool
	No user involvement in testing
	No lab testing – users are global
	Interviews with users
	Sandbox test environment
	A user researcher, the user, business analyst and testers are involved in the process
	Accessibility to users is a problem
	Testers embedded in the team
	Handover to SME’s for User Acceptance Testing
	Independent testing team – lack of visibility
	BA/Product Manager handles initial testing then regression testing commences
	Requirement traceability also done
Is there enough end-user involvement?	No (80%)
	Accessibility to users is difficult
	Global footprint therefore difficult to access users
	Only involved at the end
	Time limitation
	User base is too wide, cannot involved too many users
	User representative involved – no end-user

	involvement
	Not even user feedback received
	Yes (20%)
	Early involvement and bugs identified before UAT
	Early involvement and ability to identify enhancements or missing elements before it's too late
	Only involved in requirements and left out of the rest of the sprints
	"Too much of user involvement complicates and slows down the process"
	"Users themselves don't know what they want"
	"I think the difficulty lies with accessibility - users are not always accessible"
	"Business feels that involving users slows the entire process down and time is money. However, we lose money when we have to re-develop as the software does not meet user's needs."
Do you plan to increase end-user involvement in projects? Please elaborate.	Plans but it has not materialized
	Yes for bigger projects
	No for smaller projects
	User education from early as possible
	Show users the milestones achieve
	Feedback from users must be built into future sprints
	No – we have a high amount of user involvement.
	Too much of user involvement slows down development and the Agile process in general
	The conversation has started but no action has been taken
	Involving end-users at key stages of the project will assist
	No set plan
	User groups have been proposed
	User training has been proposed
	Focus user groups with super users
	Shaping and moulding the product by involving users
Artifacts used in the project?	Personas for UX

	Whiteboards
	User stories
	Storyboards
	Story mapping
	Prototypes
	Screen mock ups
	Story point estimations
	Sprint boards
	Entity relationship diagrams
	Process models
	Test plan
	Test cases
	Requirements document
	Functional document
	Feature list
	Streams
	Whiteboard
	Navigation maps
	Process flows
	Customer facing feature documents
How satisfied are you with the work carried out by you and the team?	Very satisfied
	Room for improvement
	Reducing the number of clicks
	Poor navigation
	Poor support
	Too many pop ups
	More discipline
	Burndown charts
	Communication
	Satisfied
	Faster
	Deliver more
	Delays with UAT
	Team ethos
	Collaborate
	Deliver
	Constraints
	UX activities
	User interface

	Quality
	Slow
	Wagile
	Shippable products
	Sprint
	Improvement
	Hybrid
	Efficient
	Value
Common processes in an iteration?	Sprint planning
	Daily stand-up
	Backlog grooming
	Sprint check-in
	Sprint demo
	User reviews
	Sprint retrospectives
	Story point estimation
	Backlog refinement
	User feedback
	Execution of testing
	Logging of defects
	Re-testing
	Regression testing
	Test report
	Minimum viable product
	Requirements
	Design
	Build
How long are typical iterations?	Two weeks
	Three weeks
	Development architecture being highly complex
	Work done in two week sprints were small
How are UX requirements handled in a project? Are there any artifacts or deliverables that are required (e.g. UI specifications)	No UI specifications
	More branding guidelines
	No
	Personas
	Representative of the user
	No clearly defined UX activity
	Assumption of what the user would need

	Most of the work done upfront
	UI design and the interaction
	External users' needs are not included
	Based on assumption
	SME's
Is the user involved in the design of the UI?	No (60%)
	Yes
	Low fidelity prototypes
	Refinements
	External users are not included
	Internal users are included
Is the user involved in the evaluation of the UI?	No (40%)
	Yes
	During testing
	Representative of the user
	External users are not included
	Internal users are included
	Only at the end when it is too late
	Indirectly they are involved via feedback
	Too late
Is the design based on explicit understanding of users?	With as good knowledge as we have
	Not typically the needs of the user
	Grey area
	Yes (50%)
	No (40%)
	N/A
Is the design based on explicit understanding of tasks?	With as good knowledge as we have
	Yes (50%)
	Grey area
	No (40%)
	N/A
Is the design based on explicit understanding of environment?	With as good knowledge as we have
	Yes (40%)
	No (30%)
	Based on assumption of the environment
	Some considerations that are made in terms of the environment
Do UX activities follow an iterative process?	No
	Yes

	Most of the UX work is done upfront
	Forms part of the development process
	Designed, reviewed, re-designed
In a typical iteration is the UX team working ahead of a development cycle in terms of UI testing and design?	No
	Yes
	Two sprints ahead
	On par with the current sprint
	A sprint ahead of the current sprint
What effect does UX testing have on product backlog? How are delays mitigated?	No UX testing
	next sprint
	Ran smoothly as the UX specialist informed the testing team
	Bottom of the list
	Least important
	Future iteration
	Ranked based on priority
	Re-prioritized
	Minimal UX processes
	Involve the user
	Defining and approving UX requirements upfront
	Within the same sprint
Does the UCD get passed before the end of the sprint for some sort of UI testing or verification, or is it all just done at the end?	At the end of the sprint
	At the end
	Iterative process
	Majority is done at the end
	Yes
	Early as possible
Are there tools that are used to integrate UX and Agile development?	Proto IO
	No
	Sketch
	Zeplin
	Invision
	Prototyping tools
Do you feel Agile methods is a successful development process?	Yes
	Works well
	We are absolutely sold on it
	The implementation of most Agile approaches is flawed
	Needs to be implemented correctly to work

	UX also needs to be lean to work within an Agile team
	More deliverables in a shorter timeframe
	What makes it even more successful is if it is implemented across the business
	Dependent on the type of projects and the type of organization that it is implemented in
	Shorter timeframe for deliverables is certainly one of the advantages
Is the addition of UX to the Agile methods effective?	Make it easier
	Fewer defects
	Speeds up the process of testing and releasing
	Made a difference
	It can be
	It varies
	If implemented correctly
	Vast difference in the user experience
	UX is effective
	You need to get "buy in" from the Agile team
	Yes
	End-product is better
	"A wireframe, a piece of paper and pen is sometimes useful as the user gets to see what will be delivered to them."
When does the UCD design get passed to development?	In the sprint
	Upfront
	As soon as the UX designs have been marked as complete
	Just-in-time
	UX team works a sprint or two ahead of the development
	Iterative process
	After the concept phase
	Initiation phase
	Sprint zero
	Sprint 2
	Sprint 1
	First sprint
How much time do you spend before	One sprint

development begins?	Sprint zero
	Ramp up sprint
	Ahead of sprint planning
	After the first sprint
	Lift-off takes two weeks
	Two weeks
Is there a UX person(s) throughout the development lifecycle?	No
	It varies
	40% of projects
	Yes
	Assumption of what the UX should look like
How often do you communicate with the UX/Agile team?	Daily (Agile)
	No UX team
	Every two weeks (UX)
	Two to three times a week (UX)
	Daily
Does the UCD process address the whole user experience?	No
	UX focuses more on the visuals rather than the functionality
	As long as the screen looks pretty, that's enough
	Yes
	There is definitely gaps
	There is not enough user involvement
	When I joined the team, it was something that was overlooked
	Based on assumptions on what the user expects
	You could have a perfect UI. However, if the coding or API is poorly designed then it will not address the whole user experience
Does the team include multidisciplinary skills?	Yes
	No
	The Experience Design Team lacks the visual designer, the interaction designer
	There is sufficient Agile members in team. However, UX is lacking
	Cross functional team who are aware of what everyone is doing
Is there any resistance from other team member with UX persons	Yes
	No

	From product owner occasionally
	If UX persons requirements are not technically feasible
	Unrealistic expectation
	Occasionally
	"They want a rocket ship when all they need is a bicycle."
	Developers
	UX person inhibits their creativity
	Testers
	Do not understand what UX is
	Educate the organisation
	Specialised skill
	Good suggestions from the UX persons
Are there issues up front with integrating UX processes?	No
	Discipline in creating standards around UX
	Do not understand UX
	Lack of understanding
	Resistance from upper management
	More accepting
	Buy in from business
	Filled in by someone else in the team
	Lack of resources
	Very few UX specialists in the organization
	Lack of resources
Do you feel that Agile and UX fit together well? What are the pros and cons?	Yes
	Allows you to stagger the processes and detect feedback earlier
	Cheaper to fix bugs and defects if it is early
	if you have too much of UX it slows down the process
	unhappy team
	They both follow similar cycle
	With Scrum, you design, you develop, you test and evaluate through each iteration
	in UX you design, you develop the prototype, you test and evaluation in each iteration
	These two cycles work so well hence if they are integrated together

	better product
	better User Experience
	people factor
	understand the role of UX in the project
	resistance from business as well as the Agile team
	adequate skills to understand the standardization of the organization
	UX requirements of the organization
	what the user requires
	empathize with the user
	do paper sketching prototypes
	meets the user's requirements
	cost
	design thinking perspective
	I do not think that Agile facilitates this
	I feel that they are not tightly coupled
	There is limited interaction with the user as they hamper the Agile delivery process
	framework to develop a product that is better for the user
	if you do UX and Agile together, UX mostly is done using in a waterfall methodology
	needs to be prioritized and feed into the current sprint
	better user experience when it is given the priority it requires
	Agile also ensures that waste is eliminated early
	The whole purpose of UX itself is to ensure that what is delivered to the user is something that can be used
	The whole purpose of Agile development is to get software into the hands of the users are quickly as possible
	The UX specialist can form part of the team and the conversation
	Not traditional UX
	Lean UX definitely fits well as it follows the same principles as Agile
Has the UX team member/s been	Yes

included from the beginning of the project?	No
	No UX specialist
What did you do to overcome the obstacles of bringing UX into your process?	Accessibility
	Availability
	Does not get the focus that it deserves
	Not clearly defined
	Not developed
	Developed based on assumption
	Developers are not skilled in UX
	It is a separate field
	User is involved often
	Educate the team on what the role of a UX person is
	Where they would get involved in the process
	Overlap in terms of what the UX person does
	No obstacles
	Educate stakeholders
	Importance of UX
	Actively involved in the projects
	Resource was shared
	UX to be brought into the process
	We are driven by the User Interface
Did bringing in UX impact everybody in a positive perspective?	Yes
	the design aspects were covered and based on good UX practices
	developer did not have to cater for this
	the developer had to go on his/her gut feeling on how the UX elements should look like
	slow down the Agile process
	difficult for the UX person and the development team to understand each other as they use different terms and the dictionaries are different
	It starts out difficult but it gets better
	Developers are certainly not people's people
	It allows for identifying issues early and fixing them early in the project
	UX slows the process down

	why should we be paying for someone that makes the software look nice
	There is a sense of pride in what the team is developing
	Everyone buys into it as it results in a better product
	Your user/customer gets something tangible to see before the product is delivered.
	The developers also have a guideline in terms of what the vision is and what needs to be developed
	UX also drives testing, as the testers are aware of what they need to test.
	UX is a form of documentation
Do you find that the development is more successful using UX and Agile together?	Yes
	Better end-user adoption
	For development where there isn't any external constraints then it is successful
	For teams where the UX is separate, then not as successful
	quality of the software
	less design wastage
	fewer sprints
	no back and forth between development and the customer
	Any customer facing software is more successful using UX and Agile together
	They are not tightly coupled
	there is no other way of getting a product that meets user's requirements without using UX and Agile together
Do you intend on keeping your current practices in your development process?	room for improvement
	UX should be brought in early
	Inform the user interfaces and the software as a whole.
	it is important for us to adapt and improve
	Evolving the current process of bringing the testing process in house
	At the moment 60% of the company understand what UX is

	"what do we want" and "what do we need"
	a need for more UX specialists within the organization
Do you think the integration of UX in Agile projects could be improved? How so?	Yes
	including a UX practitioner
	End-user upfront
	Consider a consultant to be included in not all but some of the sprints
	Over engineering the UX
	User involvement
	Bringing in a focus group
	including a UX specialist
	inclusion of a UX person
	Involvement of more users
	user research
	testing should be done in iterations
	UX is a specialised field
	Not really certain
	Sketch to demo and get feedback from the user
	Getting developers and testers involved much earlier

8.8. Axial Codes

Axial Code	**Open Codes**
Roles and responsibilities	BA Manager
	Scrum Master
	Project Manager
	Software Developer
	Senior Software Development Manager
	Agile Coach
	Quality Assurance Manager
	UX Analyst
	Business Analyst
	Back Office Team Lead
	Working experience
	Multiple projects
	Water-Scrum-Fall

	Agile
	Hybrid
Participants understanding of Agile	Framework
	A way of working to get better software
	Getting a better product out
	Getting products quicker to the user
	Making changes easier
	Creates team spirit
	Creates team morale
	Creates a self-organised team
	Safe environment for teams to communicate
	Shared knowledge
	Collaboration
	Responsive way of working
	Flexibility within current business needs
	Scalability
	Comply in terms of the Agile Manifesto
	"Agile is something we are, Scrum is something we do."
	Adjust your direction without costing a fortune
	"Turning a dime for a dime."
	Breaking down bigger projects into manageable chunks
	Estimating so that project remains on track
	Methodology with various frameworks
	Realise value sooner rather than later
	Taking feedback and using it to steer the project forward
	Iterative
	Learning
	Short, sharp deployment in the quickest possible time
Roles and responsibilities, Fitting the User into Agile, Ambiguity of the user and customer role, an Agile mindset	No active user involvement
	Minimal
	"As needed" basis
	Validates the understanding of requirements
	"Sounding board"
	Supportive role to Product Owner and the BA
	Validate software that has been developed

	End-user
	Product Owner team
	Upfront involvement for requirements elicitation
	Testing before sprint review
	Sprint review
	Feedback on mock-ups
	Late involvement
	Less engagement with users in external clients
	Alpha testing and feedback
	Involved at the end of the project
	Not enough user research
	User is the product owner
	Non-existent
	Proxy as a representative of the user (e.g. BA or Product Owner)
	Uses the end-product
	Finds value in what is created for them
Roles and responsibilities, Fitting the User into Agile, Ambiguity of the user and customer role	Sponsor
	Same as the user
	Non-existent
	Designed with an assumed behaviour in mind
	Feedback mechanism
	Internal clients – same as the user. External client – varies
	External Client – same as user
	Proxy to the user
Getting involved, customer collaboration	Beta testing and feedback
	Project kickoff
	During the “imagining” phase
	Demo’s
	Sprint retrospect
	Requirements elicitation
	Varies – 10 to 15% of external projects engage early with users
	After initiation and before design phase
	Too late in the project
	From product vision
	Involved early as possible but no collaboration during development

Getting involved, customer collaboration	No UX specialist involved
	BA fills in for UX specialist
	In the UK, the UX specialist was involved
	External clients – Yes, Internal clients – No.
	40% of projects – Yes, the balance – No.
	"Users, this is all we have time to develop, so use it or you have nothing else to use."
	Yes – for internal project, upfront with user research. External projects – very rare to have UX specialists.
	UX specialist tend to work very traditionally, instead of being Lean or Agile UX
	Too much analysis upfront
	Mostly for external projects – employ consultants/contractors
	40% of projects have UX specialists
	Yes – based abroad hence difficulty to schedule UX work.
	Yes – form part of the Experience Design Team
	UX has been included for the last 2 years
	No – defined by the user and the development team
	Yes – for the last 1 and half years
Sprint zero - opportunity to conduct user research, Getting involved, customer collaboration	Requirements gathering
	Project prioritization
	Project kick-off
	Project lift off
	Sprint zero
	Concept phase
	Product vision
	Briefing phase
	Idea phase
Upfront design, opportunity to conduct user research, Getting involved, customer collaboration, losing the big picture	Prioritize features
	Two sprints ahead for detail/requirements
	Solution design
	Initial engagement
	Requirements gathering
	User stories
	Epics

	High level requirements gathering
	Discussions between product owner, customer and execs followed by user requirements
	Breaking down the idea or product vision into requirements
Getting involved, customer collaboration, including users throughout the project	BA Manager - From the beginning
	Scrum Master – solution design
	Junior/Intermediate developer – after requirements gathering
	Senior developer – requirements phase
	Senior Software Development Manager – after the requirements phase
	Agile Coach – at any point of the process, internal projects – project lift-off, external projects - varies
	Quality Assurance Manager – first sprint. However, testing happens later
	UX Analyst – after requirement phase
	Scrum Master – product backlog defining and unpacking
	Business Analyst – requirements phase
	Back Office Team Lead – requirements phase after product vision has been defined
Getting involved, customer collaboration, including users throughout the project	Scrum Master
	Tester
	BA
	Product Owner
	Developer
	Product Manager
	Product Owner
	Team Leaders
	System Testers
	Technical Lead
	Development Manager
	Agile Coach
	Testers
	UX Designers
	Architects
	Representatives from the business
	Test Lead

	Programme Manager
	UX Analyst
	Senior Developers
Getting involved, customer collaboration, including users throughout the project	Agile Coach
	Architect
	Architects
	Automation Engineer
	Business Analyst
	Business Representatives
	Developers
	Development Manager
	Entry level tester
	Experience Analyst
	Functional tester
	Interaction Design
	Non-Functional Tester
	Product Owner
	Project Manager
	QA
	Scrum Master
	System Testers
	Team manager
	Technical Lead
	Test Lead
	Tester
	UX Analyst
	UX Designers
Getting involved, customer collaboration, including users throughout the project	All members of the team
	Everyone
	Developer
	Testers
	Business Analyst
	Product Owner
	Scrum Master
	Product Owner
	Agile Coach
	Scrum Master
	UX Designers
	UI Designers

	Architects,
	Business Representatives,
	Technical Lead
	Portfolio Manager
	Test Lead
	Programme Manager
Common Agile Methodologies used	Scrum (all companies)
	Kanban (for production only)
	Sprint planning
	backlog grooming
	sprint planning
	daily stand-up
	review
	demo
	retrospective
Common tools used, identify tools to integrate UX and Agile	Jira
	Asana
	Trello
	Confluence
	Microsoft word
	PDF
	GIT (source code management)
	Slack
	TFS
	Microsoft Team Foundation Services
	Streams
	Axosoft
Good UX design methods	Native design app used
	Screen mock-ups
	Low fidelity prototypes (rarely due to cost)
	Sketches
	User research
	User validation
	Journey maps
	Interviews
	Wireframes
	Balsamiq (low fidelity)
	Justinmind
	Invision (high fidelity)

Getting involved, customer collaboration, including users throughout the project, creating a UX vision	None
	During the testing phase
	Frequent meetings with development team (users)
	Too long wait for users to review at the end of a sprint
	Meeting more often ensures that development remains on track
	Separate UX team – handoff of specifications to development and it doesn't work well
	If you want speed and quality, the UX person should be in the same team
	Experience Design Team interacts with the developers to discuss functionality and user interface
	All sprints
	Front end development
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	No (50%)
	Yes – very late in user acceptance testing
	Internally – yes. Externally – it varies. However, in most cases real users are not used in testing
	Yes – SME (Subject matter experts)
	No – the Product Managers or the BA usually does this
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	Test team does the testing
	Outsourced test team
	UAT with clients
	No user involved
	Tester builds test script based on functional specifications then test
	A representative of the user will do the testing (e.g. Business Analyst)
	Developer does initial testing by building a test script then pass to test team
	Regression testing, stress testing, functional testing, unit testing by test team
	Core testing group "The Champions" – business wide that does the testing and sign off
	Pilot testing
	Test cases drawn up – test after development

	complete
	Spira – testing tool
	No user involvement in testing
	No lab testing – users are global
	Interviews with users
	Sandbox test environment
	A user researcher, the user, business analyst and testers are involved in the process
	Accessibility to users is a problem
	Testers embedded in the team
	Handover to SME's for User Acceptance Testing
	Independent testing team – lack of visibility
	BA/Product Manager handles initial testing then regression testing commences
	Requirement traceability also done
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	No (80%)
	Accessibility to users is difficult
	Global footprint therefore difficult to access users
	Only involved at the end
	Time limitation
	User base is too wide, cannot involved too many users
	User representative involved – no end-user involvement
	Not even user feedback received
	Yes (20%)
	Early involvement and bugs identified before UAT
	Early involvement and ability to identify enhancements or missing elements before it's too late
	Only involved in requirements and left out of the rest of the sprints
	"Too much of user involvement complicates and slows down the process"
	"Users themselves don't know what they want"
	"I think the difficulty lies with accessibility - users are not always accessible"

	"Business feels that involving users slows the entire process down and time is money. However, we lose money when we have to re-develop as the software does not meet user's needs."
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	Plans but it has not materialized
	Yes for bigger projects
	No for smaller projects
	User education from early as possible
	Show users the milestones achieve
	Feedback from users must be built into future sprints
	No – we have a high amount of user involvement.
	Too much of user involvement slows down development and the Agile process in general
	The conversation has started but no action has been taken
	Involving end-users at key stages of the project will assist
	No set plan
	User groups have been proposed
	User training has been proposed
	Focus user groups with super users
	Shaping and moulding the product by involving users
Common artifacts used, identify artifacts to integrate UX and Agile	Personas for UX
	Whiteboards
	User stories
	Storyboards
	Story mapping
	Prototypes
	Screen mock ups
	Story point estimations
	Sprint boards
	Entity relationship diagrams
	Process models
	Test plan
	Test cases
	Requirements document
	Functional document

	Feature list
	Streams
	Whiteboard
	Navigation maps
	Process flows
	Customer facing feature documents
Roles and responsibilities, Fitting the User into Agile, Ambiguity of the user and customer role, an Agile mindset	Very satisfied
	Room for improvement
	Reducing the number of clicks
	Poor navigation
	Poor support
	Too many pop ups
	More discipline
	Burndown charts
	Communication
	Satisfied
	Faster
	Deliver more
	Delays with UAT
	Team ethos
	Collaborate
	Deliver
	Constraints
	UX activities
	User interface
	Quality
	Slow
	Wagile
	Shippable products
	Sprint
	Improvement
	Hybrid
	Efficient
	Value
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	Sprint planning
	Daily stand-up
	Backlog grooming
	Sprint check-in
	Sprint demo

	User reviews
	Sprint retrospectives
	Story point estimation
	Backlog refinement
	User feedback
	Execution of testing
	Logging of defects
	Re-testing
	Regression testing
	Test report
	Minimum viable product
	Requirements
	Design
	Build
Existing and evolving development practices	Two weeks
	Three weeks
	Development architecture being highly complex
	Work done in two week sprints were small
Existing and evolving development practices	No UI specifications
	More branding guidelines
	No
	Personas
	Representative of the user
	No clearly defined UX activity
	Assumption of what the user would need
	Most of the work done upfront
	UI design and the interaction
	External users' needs are not included
	Based on assumption
	SME's
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	No (60%)
	Yes
	Low fidelity prototypes
	Refinements
	External users are not included
	Internal users are included
Getting involved, customer collaboration, including users throughout the project, creating a UX	No (40%)
	Yes
	During testing

vision	Representative of the user
	External users are not included
	Internal users are included
	Only at the end when it is too late
	Indirectly they are involved via feedback
	Too late
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	With as good knowledge as we have
	Not typically the needs of the user
	Grey area
	Yes (50%)
	No (40%)
	N/A
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	With as good knowledge as we have
	Yes (50%)
	Grey area
	No (40%)
	N/A
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	With as good knowledge as we have
	Yes (40%)
	No (30%)
	Based on assumption of the environment
	Some considerations that are made in terms of the environment
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	No
	Yes
	Most of the UX work is done upfront
	Forms part of the development process
	Designed, reviewed, re-designed
Existing and evolving development practices	No
	Yes
	Two sprints ahead
	On par with the current sprint
	A sprint ahead of the current sprint
Existing and evolving development practices	No UX testing
	next sprint
	Ran smoothly as the UX specialist informed the testing team
	Bottom of the list
	Least important

	Future iteration
	Ranked based on priority
	Re-prioritized
	Minimal UX processes
	Involve the user
	Defining and approving UX requirements upfront
	Within the same sprint
Existing and evolving development practices	At the end of the sprint
	At the end
	Iterative process
	Majority is done at the end
	Yes
	Early as possible
Common tools used, identify tools to integrate UX and Agile	Proto IO
	No
	Sketch
	Zeplin
	Invision
	Prototyping tools
Roles and responsibilities, Fitting the User into Agile, an Agile mindset	Yes
	Works well
	We are absolutely sold on it
	The implementation of most Agile approaches is flawed
	Needs to be implemented correctly to work
	UX also needs to be lean to work within an Agile team
	More deliverables in a shorter timeframe
	What makes it even more successful is if it is implemented across the business
	Dependent on the type of projects and the type of organization that it is implemented in
	Shorter timeframe for deliverables is certainly one of the advantages
Roles and responsibilities, Fitting the User into Agile, an Agile mindset	Make it easier
	Fewer defects
	Speeds up the process of testing and releasing
	Made a difference
	It can be

	It varies
	If implemented correctly
	Vast difference in the user experience
	UX is effective
	You need to get "buy in" from the Agile team
	Yes
	End-product is better
	"A wireframe, a piece of paper and pen is sometimes useful as the user gets to see what will be delivered to them."
Existing and evolving development practices	In the sprint
	Upfront
	As soon as the UX designs have been marked as complete
	Just-in-time
	UX team works a sprint or two ahead of the development
	Iterative process
	After the concept phase
	Initiation phase
	Sprint zero
	Sprint 2
	Sprint 1
	First sprint
Existing and evolving development practices	One sprint
	Sprint zero
	Ramp up sprint
	Ahead of sprint planning
	After the first sprint
	Lift-off takes two weeks
	Two weeks
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	No
	It varies
	40% of projects
	Yes
	Assumption of what the UX should look like
Getting involved, customer collaboration, including users throughout the project, creating a UX	Daily (Agile)
	No UX team
	Every two weeks (UX)

vision	Two to three times a week (UX)
	Daily
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	No
	UX focuses more on the visuals rather than the functionality
	As long as the screen looks pretty, that's enough
	Yes
	There is definitely gaps
	There is not enough user involvement
	When I joined the team, it was something that was overlooked
	Based on assumptions on what the user expects
	You could have a perfect UI. However, if the coding or API is poorly designed then it will not address the whole user experience
Roles and responsibilities, Fitting the User into Agile, an Agile mindset	Yes
	No
	The Experience Design Team lacks the visual designer, the interaction designer
	There is sufficient Agile members in team. However, UX is lacking
	Cross functional team who are aware of what everyone is doing
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	Yes
	No
	From product owner occasionally
	If UX persons requirements are not technically feasible
	Unrealistic expectation
	Occasionally
	"They want a rocket ship when all they need is a bicycle."
	Developers
	UX person inhibits their creativity
	Testers
	Do not understand what UX is
	Educate the organisation
	Specialised skill
	Good suggestions from the UX persons

Getting involved, customer collaboration, including users throughout the project, creating a UX vision	No
	Discipline in creating standards around UX
	Do not understand UX
	Lack of understanding
	Resistance from upper management
	More accepting
	Buy in from business
	Filled in by someone else in the team
	Lack of resources
	Very few UX specialists in the organization
	Lack of resources
Roles and responsibilities, Fitting the User into Agile, an Agile mindset, Getting involved, customer collaboration, including users throughout the project, creating a UX vision	Yes
	Allows you to stagger the processes and detect feedback earlier
	Cheaper to fix bugs and defects if it is early
	if you have too much of UX it slows down the process
	unhappy team
	They both follow similar cycle
	With Scrum, you design, you develop, you test and evaluate through each iteration
	in UX you design, you develop the prototype, you test and evaluation in each iteration
	These two cycles work so well hence if they are integrated together
	better product
	better User Experience
	people factor
	understand the role of UX in the project
	resistance from business as well as the Agile team
	adequate skills to understand the standardization of the organization
	UX requirements of the organization
	what the user requires
	empathize with the user
	do paper sketching prototypes
	meets the user's requirements
	cost
	design thinking perspective

	I do not think that Agile facilitates this
	I feel that they are not tightly coupled
	There is limited interaction with the user as they hamper the Agile delivery process
	framework to develop a product that is better for the user
	if you do UX and Agile together, UX mostly is done using in a waterfall methodology
	needs to be prioritized and feed into the current sprint
	better user experience when it is given the priority it requires
	Agile also ensures that waste is eliminated early
	The whole purpose of UX itself is to ensure that what is delivered to the user is something that can be used
	The whole purpose of Agile development is to get software into the hands of the users are quickly as possible
	The UX specialist can form part of the team and the conversation
	Not traditional UX
	Lean UX definitely fits well as it follows the same principles as Agile
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	Yes
	No
	No UX specialist
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	Accessibility
	Availability
	Does not get the focus that it deserves
	Not clearly defined
	Not developed
	Developed based on assumption
	Developers are not skilled in UX
	It is a separate field
	User is involved often
	Educate the team on what the role of a UX person

	is
	Where they would get involved in the process
	Overlap in terms of what the UX person does
	No obstacles
	Educate stakeholders
	Importance of UX
	Actively involved in the projects
	Resource was shared
	UX to be brought into the process
	We are driven by the User Interface
Getting involved, customer collaboration, including users throughout the project, creating a UX vision	Yes
	the design aspects were covered and based on good UX practices
	developer did not have to cater for this
	the developer had to go on his/her gut feeling on how the UX elements should look like
	slow down the Agile process
	difficult for the UX person and the development team to understand each other as they use different terms and the dictionaries are different
	It starts out difficult but it gets better
	Developers are certainly not people's people
	It allows for identifying issues early and fixing them early in the project
	UX slows the process down
	why should we be paying for someone that makes the software look nice
	There is a sense of pride in what the team is developing
	Everyone buys into it as it results in a better product
	Your user/customer gets something tangible to see before the product is delivered.
	The developers also have a guideline in terms of what the vision is and what needs to be developed
	UX also drives testing, as the testers are aware of what they need to test.
	UX is a form of documentation
Roles and responsibilities, Fitting the User into Agile, an Agile mindset,	Yes
	Better end-user adoption

Getting involved, customer collaboration, including users throughout the project, creating a UX vision	For development where there isn't any external constraints then it is successful
	For teams where the UX is separate, then not as successful
	quality of the software
	less design wastage
	fewer sprints
	no back and forth between development and the customer
	Any customer facing software is more successful using UX and Agile together
	They are not tightly coupled
	there is no other way of getting a product that meets user's requirements without using UX and Agile together
Roles and responsibilities, Fitting the User into Agile, an Agile mindset, Getting involved, customer collaboration, including users throughout the project, creating a UX vision	room for improvement
	UX should be brought in early
	Inform the user interfaces and the software as a whole.
	it is important for us to adapt and improve
	Evolving the current process of bringing the testing process in house
	At the moment 60% of the company understand what UX is
	"what do we want" and "what do we need"
	a need for more UX specialists within the organization
Roles and responsibilities, Fitting the User into Agile, an Agile mindset, getting involved, customer collaboration, including users throughout the project, creating a UX vision	Yes
	including a UX practitioner
	End-user upfront
	Consider a consultant to be included in not all but some of the sprints
	Over engineering the UX
	User involvement
	Bringing in a focus group
	including a UX specialist
	inclusion of a UX person
	Involvement of more users
	user research

	testing should be done in iterations
	UX is a specialised field
	Not really certain
	Sketch to demo and get feedback from the user
	Getting developers and testers involved much earlier