Lincoln University Digital Thesis

The impact of the relationship between the customer and the software development team on the outcome of a software development project

---

A thesis

submitted in partial fulfilment

of the requirements for the Degree of

Doctor of Philosophy

at

Lincoln University

by

Dipendra Ghimire

Lincoln University

2017

ABSTRACT

Abstract of a thesis submitted in partial fulfilment of the requirements for the Degree of Doctor of Philosophy

The impact of the relationship between the customer and the software development team on the outcome of a software development project

by

Dipendra Ghimire

Agile software is a collection of software development methods based on iterative and incremental development processes. In an Agile software development approach, the requirements and solutions are obtained through the collaboration and coordination of cross-functional teams with the customer. Agile software development processes aim to produce higher quality software than traditional approaches, and this is supported by both anecdotal and empirical evidence. However, there is little understanding about the relationship between the software development team and the product owner, and the impact this relationship has on project outcomes. The Teamwork model can be used to understand the relationships in the team using Agile software development.

Agile software development adopts sets of practices and roles for organizing work during a software development project. However, some challenges exist in Agile software development that impact on the relationships between development team and the product owner. This research identifies these challenges with Agile and explains their impact on the relationship using the Teamwork model.

To understand the relationship challenges, a mixed method multi-study approach was undertaken. Data collection from participants involved two instruments: an online questionnaire and a semi-structured interview. Responses from the online questionnaire were analysed using descriptive and inferential statistical methods, and thematic analysis methods were used for qualitative data analysis of the interviews. Two studies were

conducted, one with software development teams and one with product owners. Discussion of the results from both studies are presented and then compared in order to understand the relationships from both perspectives.

The findings indicated that there were challenges in communication, project requirements, project priorities and project timelines within the team as well as between the team and the product owner. Challenges in communication, project priorities, interpersonal issues and project requirements, have an impact on the completion time of the project. The Teamwork model explained some of the relationship challenges between the Agile software development team and the customer. Challenges that fall outside the original Teamwork model contributed to understanding what factors constituted effective teamwork when applying an Agile process to software development.

As a result of the findings from this current research, a revision to the Teamwork model was proposed that introduced two additional constructs: team motivation and organizational culture to the model. Team motivation impacts on the coordination, communication and team support. Organizational culture sets the overall context for the relationship between product owner and software development team. Communication within the relationship has the highest impact on project outcomes.

Keywords: Agile software development; Agile software teams; Teamwork model, teamwork, challenges in software development, team relationships

Acknowledgements

List of Figures

List of Tables

Chapter 1:   Introduction

Software development is the process of turning a need into executable computer code that satisfies that need. Early software development used the concept of "code and fix, code some more, fix-some-more" without any planning (Leffingwell, 2007; Martin, 2005). Later, in the 1960s, plan-driven methods, based on sequential activities, were introduced. The plan-driven methods used for software development include: Waterfall (Nuseibeh, 2001), Spiral (Fitzgerald, Russo, & O'Kane, 2003) and Prototyping (Paetsch, Eberlein, & Maurer, 2003a). These methods involve a number of different sequential stages that include: requirement specifications, requirement analysis, design, coding, testing and maintenance (Awad, 2005). Often bugs in software are not found until the latter stages of the development process when they are difficult to fix. Developers may be required to spend significant amounts of time fixing the errors affecting the overall productivity and efficiency of the project (Fowler, 2001).

The Waterfall model (Boehm, 2002) of software development is one of the most widely used due to its straightforward and structured nature (Larman & Basili, 2003). The waterfall approach is a predictive method for the software development process based on documentation (Boehm & Turner, 2003c). Activities included in this approach consist of requirement analysis, design and testing of code, and well-defined documentation. The defined process must be followed step-by-step (Huo et al., 2004). Although the Waterfall model, and other traditional methods, are still used in software development there are some concerns about its slow adaptation to constantly changing business requirements and a tendency to overrun the budget or fall behind the schedule (Petersen & Wohlin, 2010). Agile software development methods were proposed as an alternative to traditional methods. Agile software development projects are reported to have been more successful than traditional software development projects (Huo et al., 2004).

In an Agile software development approach, software projects are developed by coordination between the customer and the development team. Development teams are self-organizing, manage the workload and plan for project iterations to be completed within a fixed period of time. These self-organizing teams are cross-functional and must have trust,

a common understanding of the project and the ability to adapt to changes in the project (Lee & Xia, 2010; Moe, Dingsøyr, & Dybå, 2008). During the project there is direct contact between the software development team members and a customer representative, often called the product owner. Because of this, there is a relationship between the product owner and the customer. Research interest and motivation for this research is discussed in the following section.

1.1 Research interest and motivation

Software is used in many organizations to perform essential business operations; software, therefore, has become a vital business tool (Fenton & Bieman, 2014; Gollenia, 2016). The software development industry has provided many examples where there has been a failure in software development projects (Highsmith, 2013; Kaur & Sengupta, 2013). Examples include those recorded by Chow and Cao (2008), Misra et al., (2009) and Moe et al., (2010). Several factors have been identified as contributing to the failure of software development projects. Such a failure of a software development project, where huge amounts of money and other resources needed to be invested, led the researcher to consider this as a suitable area to investigate. Researchers in the software engineering have previously explored the behaviour of the Agile software development team but have not examined the relationship with the customer.

The purpose of this research is to understand the relationship between the software development team and product owner in Agile software development projects and the impact of that relationship on project outcomes.

The high level research question to be investigated is:

"What impact does the relationship between the software development team and the product owner have on Agile software project outcomes?"

To understand this relationship, the challenges in Agile software development are explored. After identification of the challenges and understanding their impact on the relationship, a revision of the Teamwork model is proposed to explain the relationships.

1.2 Thesis outline

Chapter 1: Introduction

This chapter presents the research background, motivation for this research, the contribution of the research and the structure of the thesis.

Chapter 2: Literature Review

In this chapter studies about relationships, Agile software development studies about Agile are explored. The origin and principles of Agile methods are discussed, and the roles involved in Agile software development teams are investigated. Further, the relationship between the customer representative and the software development team are studied in detail.

This chapter also includes discussion on a range of theories used in understanding relationships in teams. A number of these theories are explored individually with the aim of selecting a model that best suits this study that can be applied to answer the research questions.

Chapter 3: Research Methods

Chapter 3 outlines the methodological framework this research is based on and describes the methods used to collect the data. This chapter also presents the research process used in this study, the rationale behind the choice of the mixed method data gathering techniques used, and the justification of the method selected for this study.

Chapter 4: Software Development Team Results and Discussion

Chapter 4 presents the findings from the data collection in detail. The data analysis techniques used for this study are described. The results are studied, reviewed and explained. The results gathered are analysed and discussed in detail, the challenges tabulated, and a detailed analysis undertaken to explore the relationships. Data analysis of both the quantitative data and qualitative data, the results and a discussion are presented in this chapter.

Chapter 5: Product Owner Study - Results and Discussion

In this chapter findings from the data collection for the product owner's study are presented in detail. The analysis process and the data obtained from the respondents are presented.

Data analysis from both the quantitative and qualitative studies are presented and discussed in this chapter.

Chapter 6: Combined Discussion

In this chapter results from the software development team and product owners' studies are compared and discussed. The findings from the study are also discussed in relation to the Teamwork model in this chapter.

Chapter 7: Conclusions

This chapter provides the overview of the findings of this study. The contribution of this thesis, limitations of the study, and a discussion of the practical implications of the results from this study are discussed in this chapter.

Appendices:

The pilot study results, Human ethics approval, questionnaire and interview questions for both studies are here.

An overview of the thesis structure is presented in Figure 1-1
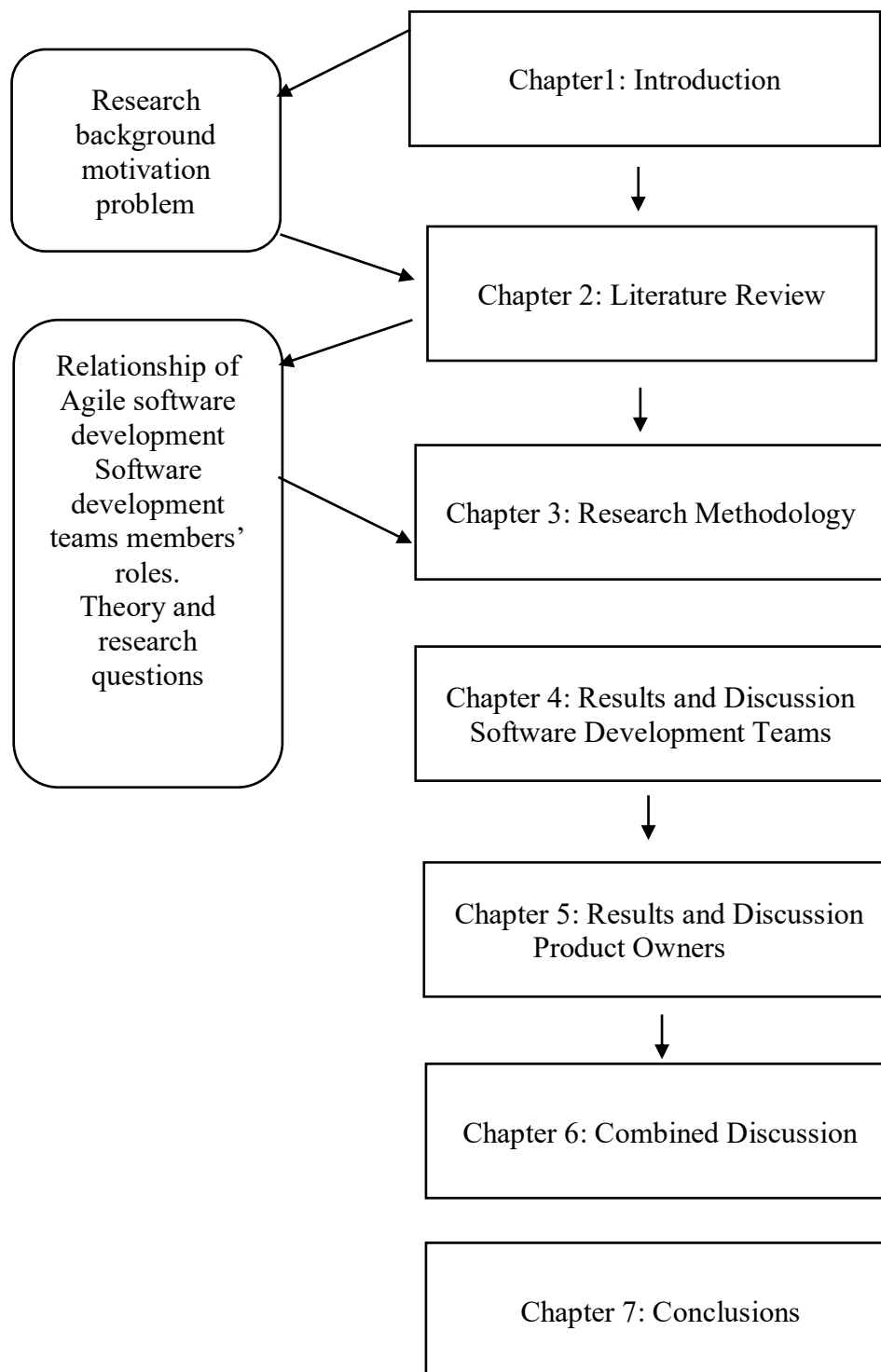
Figure 1-1 Thesis structure

1.3 Summary

This chapter gave an overview of the software development process, how changes were made to mitigate problems and introduced the rationale for the project. The next chapter follows from this one by giving more detail about the literature leading to the research questions.

Chapter 2:   Literature Review

This study is about understanding the impact of the relationships between the software development team and the owner of the projects. The purpose of this literature review is to explore the existing literature about relationships, particularly those that occur in workplaces, and about approaches to software development and the issues experienced with them. This review will focus on the relationships between the customer and the software development team. Particular attention is paid to relationships, in general, software development, software development teams, stakeholders involved in software development, and the relationships between the development team and the customer representative during the software development process.

To describe the relationships between the customer and software development team, the Teamwork model, Agency theory and Stewardship theory are reviewed as potential theories. In the first instance, relationships are described, in general, followed by software development and then Agile software development. This literature review has the following structure:

- Relationships
- Software development methods
    - Agile software development
    - Agile software development teams
- Relationships in Agile software development
- Challenges in Agile software development projects
- Teamwork
- Teamwork in Agile software development
    - Teamwork model
    - Coordination theory
    - Decision-making process
    - The theory of holographic organization
    - Agency theory
    - Stewardship theory
- Discussion
- Summary

2.1 Relationships

Relationships are ongoing interactions that occur between people (Gabarro, 1990). These interactions can be based on regular business or social commitments (Berscheid, 1999). The structure of people's lives primarily comprise his or her relationship with self, others, groups and institutions (Herrnstein & Murray, 2010). Other than family relationships, the relationships with organizations and colleagues we spend time with are probably prominent within most people's lives (Greenhaus, Collins, & Shaw, 2003). In an organisational sense, different types of relationships are formed that involve more than one person (Garbarino & Johnson, 1999) Examples of the types of relationships that occur in a typical workplace include: workplace relationships, team relationships and customer relationships (Chen & Popovich, 2003; De Dreu & Weingart, 2003; Eby, Rhodes, & Allen, 2007). These relationships are presented in   Table 2-1 and are discussed in the following sections.

Table 2-1 Relationship types

Workplace relationships	Workplace relationships are interpersonal relationships that exist and develop within an organization.
Team relationships	A team relationship is formed through interactions between team members
Customer relationships	The customer relationship is the result of interactions between the customer and people within an organisation that are providing services

2.1.1 Workplace relationships

Workplace relationships are interpersonal relationships that exist, and develop, within an organization (Guzzo & Shea, 1992). An interpersonal relationship is a close association between two or more people (Berscheid, 1999). Workplace relationships help people when performing activities, such as decision-making or sharing information (Gordon & Hartman, 2009). Sias (2005) found that higher quality workplace relationships will likely result in increased commitment by the employee to the organization. In their study, they surveyed 400 employees from a public university. The respondents were classified into different categories, such as staff, exempt staff and faculty (Sias, 2005). The results of their study

indicated that the quality of employee relationships was associated with the quality of information shared between the employees and the increased commitment of employees to the organization and satisfaction with their job.

"Business is not just doing deals; business is having great products, doing great engineering, and providing tremendous service to customers. Finally, business is a cobweb of human relationships."

Ross Perot [1]

With the quote, above, it is clear that there are different human relationships involved in undertaking business and these relationships are critical to the business (Harter, Schmidt, & Hayes, 2002). Another example of a relationship that occurs within a workplace is the relationship between teams, or subsets of workers. Team relationships are discussed in the following section.

2.1.2 Team relationships

A team is a group of people working together to achieve common objectives (Mohrman, Cohen, & Morhman Jr, 1995). A team relationship is formed through interactions between team members (Chatman & Flynn, 2001). Team relationships include the interaction between the team members in sharing of information and feedback, and recognising the needs and feelings of team members (Lovelace, Shapiro, & Weingart, 2001). Neuman, Wagner, & Christiansen, (1999) examined personality factors within a group of 323 retail assistants working in 82 teams. Their study identified a significant relationship between the team relationships and work-team performance. They suggested that a team will be more effective when the personalities of the team members are diverse and each member brings unique attributes to the team.

A team in an organization deals with the customer to provide services. During this process a relationship exists between the business and the customer. The customer relationship is explained in the following section.

2.1.3 Customer relationships

A customer is a recipient of a service provided by an organization (Mittal & Kamakura, 2001). The customer relationship is the result of interactions between the customer and people within the organisation providing the services (Harrison & Norman, 2001). Fornell

---

[1] https://www.brainyquote.com/quotes/quotes/r/rossperot159902.html

(1992) measured customer satisfaction using three indicators: expectations, perceived performance and loyalty. Their study was based on the literature related to customer satisfaction and dissatisfaction. Their study found that customer satisfaction has important implications on the performance of an organization to improve services (Fornell, 1992).

Software development happens in a particular workplace where the different people involved in a project work together. As in other workplaces, there are a number of interrelated relationships that occur (Cramton & Webber, 2005). These include the relationships between the team members themselves and also between team members and the customer. In the following section, software development, Agile software development, the different roles in Agile software development are discussed.

2.1.4   Software development

The use of software in organizations and businesses is increasing, with many organisations requiring software to be customized for their operations (Moniruzzaman & Hossain, 2013). Software for such organizations can be developed in-house or by a third-party. In the acquisition of custom software two parties exist, the customer who wants the software and the development team who creates the software (Wiegers & Beatty, 2013). The relationship between with the customer in Agile software development is discussed in section 2.6.1

2.2   Software development methods

In the early days of software development a sequence-based method was introduced, known as a plan-driven approach (Awad, 2005). One of the most common plan-driven approaches is the Waterfall model. This model works through the process of analysis, design, testing, implementation and maintenance in a sequential manner (Royce, 1970). Some argue that this model is flawed as it is impossible to complete any phase of a non-trivial project perfectly before moving on to the next phase (Awad, 2005).

Many software companies, because of the straightforward nature of the waterfall approach, use this method; however, there are some concerns about this approach (Munassar & Govardhan, 2010). These concerns include the slow adoption, constantly changing business requirements, the tendency to overrun budgets and projects failing to meet the schedule (Hannay & Benestad, 2010). Because there are a set of steps followed in the waterfall approach in order to develop projects, changes are requested during the project.

Van Lamsweerde (2000) surveyed 8000 projects developed using a waterfall approach. These projects were from 350 software companies in the US and findings revealed that one third of the projects never reached completion and half of those that did reach completion did so by achieving only partial functionality, with major cost overruns and significant time delays. When respondents were asked about the cause of such failures, 50% reported that this was due to poor detail in the project requirements, 13% of respondents reported a lack of user involvement, 12% requirement incompleteness, 11% changes in requirements, 6% unrealistic expectations and 5% unclear objectives (Van Lamsweerde, 2000).

In more recent times a new group of software development methods, known collectively as Agile Software Development, has been introduced to reduce the problems projects have in traditional software development (Highsmith & Cockburn, 2001)

2.3   Agile software development

Agile methods are based on an iterative and incremental cycle of development where solutions and requirements are developed through collaboration and coordination between cross-functional teams (Schatz & Abdelshafi, 2005). Agile software development focuses on flexibility, allowing for any changes in requirements that may occur during the software development process (Greer & Hamon, 2011).

In 2001, seven practitioners created the Agile manifesto to define the approach now known as Agile Software Development. The Agile manifesto defines four main values and twelve principles. The values are (Fowler & Highsmith, 2001):

- Individual and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following plans

As Agile places an emphasis on communication and coordination, software development using an Agile approach promotes adaptive planning, evolutionary development with continuous improvement and early completion of the deliverables (Awad, 2005; Moniruzzaman & Hossain, 2013; Turk, France, & Rumpe, 2014a). The Agile methodology is flexible and adopts the changes that may be required during the software development process (Boehm, 2002). Change to project requirements is a major concern with a waterfall

approach (Keegan & Turner, 2001). A comparison between a traditional approach and an Agile approach in relation to software development and release process is illustrated in Figure 2-1
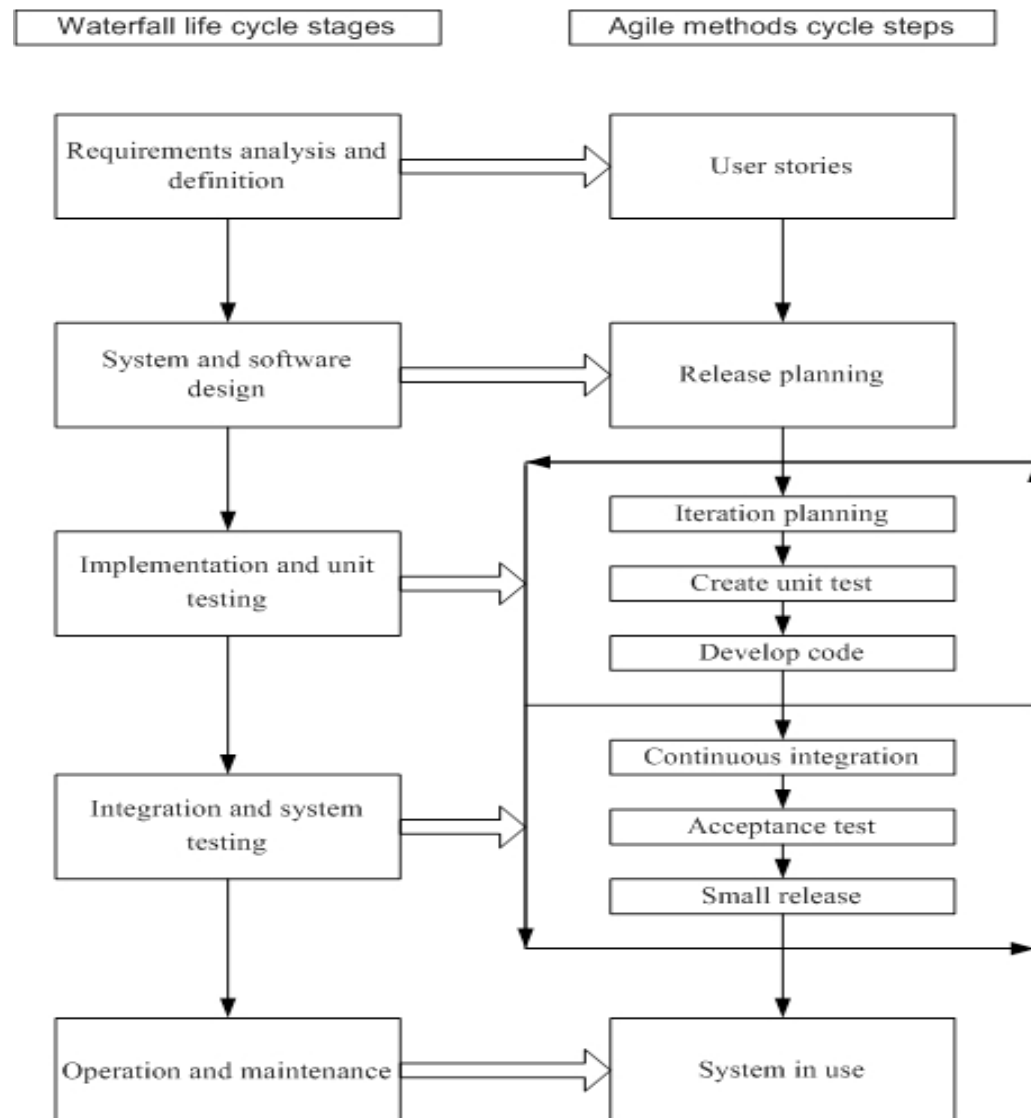
Figure 2-1 Waterfall model vs Agile (Huo et al., 2004)

Changes to project requirements can be undertaken at any stage in an Agile software development process (Dingsøyr et al., 2012a). Changes requested by the customer are likely to reflect either a change in business need or recognition of new features to be included once some of the project has been demonstrated (Tate, 2005). These changes are accommodated through discussions between the customer and the software development organization. The software development organization might agree to replace some of the features to bring in the new feature or add the new features to the list.

Agile software development approaches have the potential to provide a higher level of customer satisfaction within a shorter development cycle, with lower bug rates and quicker adoption of rapidly changing business requirements (Boehm & Turner, 2003a; Parnas, 2006). For example, Vijayasarathy & Turk (2008) found that Agile software development projects in several different parts of the world showed an increased productivity, of 80%, with compared to the traditional software development approach. They also mentioned that there was increased job satisfaction and improved predictability of schedule/costs/quality. However, they did not quantify the improvement. This means that it is difficult to know exactly how these improvements were demonstrated. (Vijayasarathy & Turk, 2008). The evaluation criteria used by Vijayasarathy & Turk (2008) were user satisfaction, delivery time, maintainability and cost. The main incentive for applying the Agile development approach is cost savings to achieve a faster development cycle while still delivering quality software (Moniruzzaman & Hossain, 2013; Wiegers & Beatty, 2013). The aim of an Agile approach is to develop software quickly in an environment of rapidly changing requirements (Cao & Ramesh, 2008). Different Agile software methods focus on different aspects of the software development life cycle, such as the practice and management of software projects (Pressman, 2005).

A range of software development approaches have had the Agile descriptor applied, these methods are shown in Table 2-2

Table 2-2 Agile approaches

Name	Reference
Extreme programming (XP)	(Beck, 2000)
Scrum	(Schwaber, 1997)
Feature-driven development (FDD)	(Palmer & Felsing, 2001)
Adaptive software development (ASD),	(Highsmith & Cockburn, 2001)
Dynamic systems development method (DSDM)	(Stapleton, 1997)

These methods (Table 2) focus on different aspects of the software development process and cover areas such as requirements, design, modelling, coding, testing, project management, project and quality. Each of these methods will be discussed in detail in the following sections.

2.3.1   Extreme programming (XP)

Extreme programming (XP), a widely used Agile method (Beck & Andres, 2004; Beck et al., 2001b), aims to enable successful software development despite constantly changing requirements. The XP process begins with user stories for capturing the requirements. A user story, in the software development context, describes a type of user, what they require and why they require it (Beck et al., 2001a). A user story is designed to be short enough to be written on a post-it note and is usually written in language a business customer would understand, essentially, user stories consist of what the user is looking for (Cohn, 2004). These stories may be the addition of functions on the current project or the requirements for a new project. User stories are collected from stakeholders and are used to describe a variety of deliverable requirements. The requirements for the project gathered from the user stories are assigned priorities in agreement with the customer and delivered in an incremental approach (Wiegers & Beatty, 2013). The development team then makes a commitment for the delivery date and the working software is delivered to the customer, typically, in intervals of 1–3 weeks. XP is said to improve the delivery of software projects by embracing communication, simplicity, feedback, respect and courage (Fitzgerald, Hartnett, & Conboy, 2006). XP avoids any activities that are not immediately required for the current implementation stage of a project (Stoica, Mircea, & Ghilic-Micu, 2013).

Another Agile software development Feature-driven Development (FDD) is explained in detail in the following section.

2.3.2   Feature-driven development

Feature-driven development (FDD) involves an object oriented approach (Beck et al., 2001b). Using this approach, objects are identified along with their attributes (Lunn et al., 2000). Objects are tangible things and include: roles, events, people, places, organization, devices and location. Operations performed by, and required for, each object are characterized and a link is established between these objects (Booch, 1986). In an object oriented approach an ordered sequence of messages between two objects can be established during the running time.

When FDD is compared to XP it seems that XP is better suited for volatile projects where the user requirements change often (Stankovic et al., 2013a). FDD helps developers to understand the strength and weakness of each model and keep those models that provide

long term value to the organization (Boehm & Turner, 2003a). A feature list is created that groups the features into sets and subject areas. After completing the grouping process a development plan is created, features are designed and then built (Boehm & Turner, 2003a).

Features are an important aspect of FDD, whereas user stories are the primary source of the requirements in Scrum, which is explained in detail below.

2.3.3   Scrum

Scrum is an iterative and incremental Agile development framework used in the development of software (Schwaber & Beedle, 2002). Scrum can be observed as a lightweight management approach rather than a full process; hence, it has gained increasing popularity for use in software development projects and has a broad applicability for managing and controlling iterative and incremental projects (Kniberg, 2010; Stankovic et al., 2013a). The Scrum process is suitable for projects that have tight timelines, changing requirements, and are critical to a business because, in Scrum, the work is divided into packets with continuous testing and documentation as the project continues (Beck et al., 2001b). The packets, known as sprints, are derived from the backlog of existing requirements. A backlog is an ordered list of the work the development team must address during the next sprint (Rising & Janoff, 2000). The sprint is a time-box (time interval) of one month or shorter, during which time, a useable and, potentially, releasable product increment is created (Schwaber & Sutherland, 2013). In Scrum, very short meetings, known as stand-ups, are conducted every day and the deliverable of the sprint is sent to the customer within the allocated time-box (Schwaber & Sutherland, 2013). The process followed in this approach can be viewed in Figure 2-2.
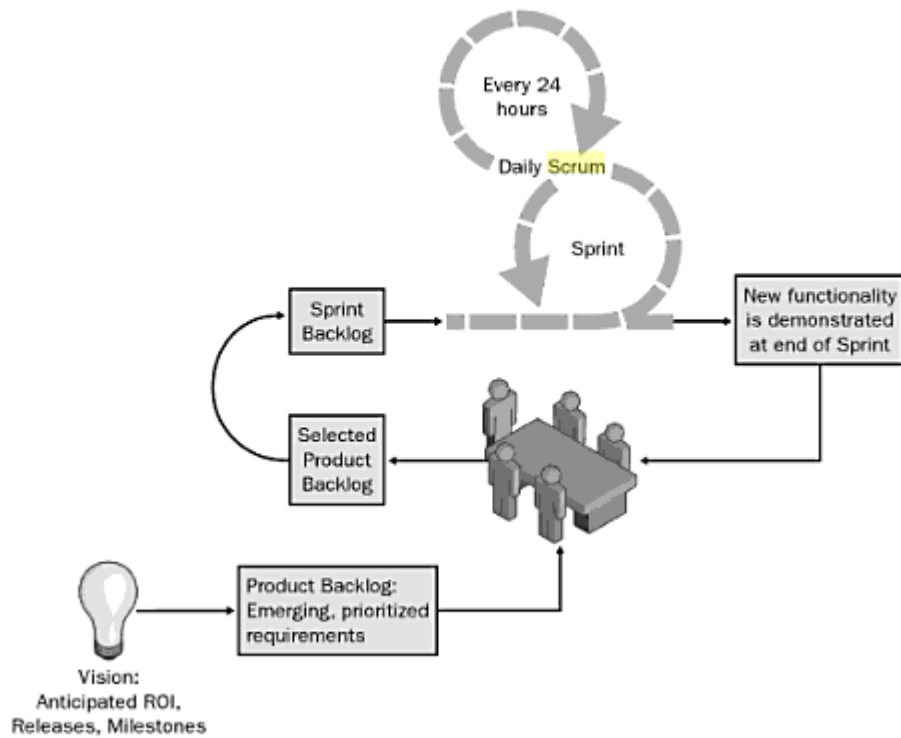
Figure 2-2 The Scrum process (Schwaber, 2004)

The Scrum team usually consists of the product owner, developers, testers and a Scrum master (Schwaber & Sutherland, 2013). Roles associated with a Scrum team are summarized in Table 2-3

Table 2-3: Summary of roles in a Scrum team

Product owner	• Creates a prioritized wish list called a product backlog • Ensures the product backlog is visible and clear to all showing what the next work for Scrum team is (Schwaber & Sutherland, 2013).
Scrum master	• The Scrum master is responsible for ensuring Scrum is understood and followed. This is done by ensuring that the team follows Scrum theory, practices and rules. • Helps in the interaction with the Scrum team to maximize the value created by the team (Schwaber & Sutherland, 2013).
Development teams	• Self-organizing • Cross-functional; have all the skills necessary to create product increments • Team members may have specialized skills and a particular area of focus, but the credit belongs to team as a whole (Schwaber & Sutherland, 2013).

Scrum is a commonly used method for software development and project management (Khalid, Zahra, & Khan, 2014). However, another Agile approach, Adaptive Software Development, focuses more on results and self-organization to achieve the project goal (Huda, 2011), as discussed in the following section.

2.3.4 Adaptive software development

An Adaptive Software Development (Highsmith & Cockburn, 2001) process focuses on human collaboration and self-organization. Time-boxing, where the schedule is divided into a number of separate time periods with each part having its own deliverables, deadline and budget, are used as the planning tools in this process (Beck et al., 2001b). This process offers solutions for the development of large and complex systems as it encourages incremental and iterative development, with collaboration and constant prototyping. The main properties of Adaptive Software Development (ASD) are (Beck et al., 2001b):

- Mission-driven planning

- Component-based focus

- Use of time-boxing

- Explicit consideration of risk

- Focus on collaboration for requirement gathering

- Emphasizes learning throughout the process

Adaptive Software Development is an Agile software method for building complex projects where ASD teams focus on human collaboration and team self-organization (Beck et al., 2001a). ASD encourages iterative and incremental development with constant prototyping (Highsmith, 2013).

The Dynamic System Development Method (DSDM) is similar to both XP and ASD and is used for projects with tight timeline constraints. This approach is explained below.

2.3.5 Dynamic systems development method

The dynamic systems development method (DSDM) (Stapleton, 1997) is an Agile software development approach often used for systems that have tight time constraints (Beck et al., 2001b). This method follows nine guiding principles that are listed below (Beck et al., 2001b):

- Active user involvement is imperative.

- DSDM teams must be empowered to make decisions.

- The focus is on frequent delivery of the products.

- Fit for purpose is the essential criterion for the acceptance of deliverables.

- Iterative and incremental development is necessary to provide coverage for the acceptance of deliverables.

- All changes during development are reversible.

- Requirements are base lined at a high level

- Testing is integrated throughout the life-cycle

- Follows the 'Pareto Principle' (80-20 rule) (Reh, 2005) – 80% of an application can be delivered in 20% of the time it would take to deliver 100% of the application. Only enough work should be done in each increment to facilitate movement to the next increment.

Team collaboration and interactions are the hallmarks of the Agile methods described above (Boehm & Turner, 2003a). XP is noted for pair programming and Scrum practices collaboration among the cross-functional teams (Beck et al., 2001a). A hallmark of all of the Agile development methods is the use of a cross-functional team (Hoda, Noble, & Marshall, 2013).

2.4   Agile software development teams

Agile approaches focus on the people more than the process (Cockburn & Highsmith, 2001b). This focus places an emphasis on factors such as skill, talent and communication (Cockburn & Highsmith, 2001b) rather than processes. Agile software development teams are self-organizing with cross-functional team members (Moe, Dingsøyr, & Dybå, 2010). For example, Agile software development members have different expertize to meet the common goals of the project, such as back-end programming, front-end programming and testing (Martin, 2003). Collaboration and team interactions are the basis of Agile development (Boehm & Turner, 2003a). Robinson and Sharp (2008) mentioned that Agile teams could have faith in their own abilities, show respect for their team members and maintain trust within the team.

There are a range of people involved in software development projects, including developers, testers, project leaders, and business analysts and, typically, the members of the teams are made up of a mixture of these roles. Most teams also include a customer representative or product owner, or have access to one during the development process (Coram & Bohner, 2005). Each of these roles is discussed in detail in the following sections.

2.4.1   Developer

Agile projects are largely dependent on the functioning of the software development teams (Conboy & Fitzgerald, 2010). It is preferred that software developers working in Agile teams have the necessary skill-sets to be considered cross-functional and must be willing to work as a team (Beck et al., 2001b). The cross-functional team in Agile software development refers to the team as a whole as it has all the skills they need to develop the project (Lee & Xia, 2010). Developers should be able to handle constant changes and have the necessary skill-sets to solve problems. However, they also said that some developers have characteristics that make them unwilling to share or work as a team; therefore, they may not be suitable for work that involves a great deal of collaboration. In an Agile team the role

of the software tester is integral to the project's success (Utting & Legeard, 2010). This role is discussed in the following section.

2.4.2 Software tester

Software testers the part of an Agile team who are involved in testing the software projects for quality assurance (Perry, 2007). The role of the software testers is to test the functions according to the testing criteria (Coram & Bohner, 2005). Testing criteria are created at the beginning of the project from the users' stories, in conjunction with the product owner, to test the function quality and performance, to meet the users' requirements. There are different testing procedures, such as automating test cases, test-driven development, acceptance test-driven development and manual testing for quality assurance, and the testers work closely with the development team during this process (Janzen & Saiedian, 2005; Shore, 2007; Utting & Legeard, 2010). Software testers interact with the whole team, especially the software developers and the product owner, to communicate the test results (Coram & Bohner, 2005). As with all members of an Agile development team, the software testers attend all meetings relating to the software project to understand the requirements and give feedback (Shore, 2007). Testers generally provide feedback about the testing to the other team members during the stand-ups. Testers work in the self-organizing team to ensure the quality of the product developed (Deemer et al., 2010).

2.4.3 Scrum master

A Scrum master is mainly responsible for enabling the team to remain focused on the project (Schwaber, 2004). This role includes the responsibility for coordinating between the team members and the product owner (Schatz & Abdelshafi, 2005). For example, the Scrum master is one of the team members who is in charge of solving any problems that may stop the team from working effectively (Dybå & Dingsøyr, 2008). This problem could be coordination, communication or the estimation of the work in the software development teams. If the team members are not sure about the requirements of a project the Scrum master facilitates meetings with the product owner (Paasivaara, Durasiewicz, & Lassenius, 2009). The Scrum master also maintains the product backlog for the next sprint. The Scrum master's overall role is to help the team and remind them that the goal of the project is to deliver the value to the customer.

A business analyst is required to understand the business domain and feed the business requirements of a software project to the team. The role of business analyst is explained in the following section.

2.4.4 Business analyst

A business analyst analyses the business domain, looking at how the business operates and then documents the business processes (Smith & Fingar, 2003). The role of a business analyst in an Agile software team is to produce the requirements in collaboration with the users. The business analyst also works with the business side of an organization to develop test cases for the system and create a shared understanding of the product to be built (Cohn, 2004; West et al., 2010). Business analysts perform a liaison function between the business side of an organization and the IT solution providers (Cohn, 2004). For example, the business analyst prepares the functional requirements for a software requirement specification (SRS) containing a full description of the expected behaviour of the software (Wiegers & Beatty, 2013). Software requirement specifications (SRS) are routinely used in developing, testing and quality assurance (Wiegers & Beatty, 2013).

Business analyst facilitates the process of getting information development needs from the customer/product owner. The customer/ product owner role is defined in following section.

2.4.5 Customer representative/Product owner

One of the tenets of an Agile approach is to involve the customer or a representative of the customer at all stages of the development process (Coram & Bohner, 2005). Customer representatives are people who are either from the client organisation or from the development organisation who act as customer representatives (Wiegers & Beatty, 2013). It is recommended that, for an Agile approach to be successful, where possible, a full time customer representative is present on-site who works directly with the software development team (Boehm, 2002). Customer representatives are responsible for providing the requirements and evaluating the project against the business values (Paetsch et al., 2003a). Customer representatives in Agile software development also define the scope of the project by collaborating with the software development teams (Wiegers & Beatty, 2013).

The product owner represents the customers in Agile projects using the Scrum methodology (Wiegers & Beatty, 2013). A product owner is the member of a software development team

who is acting as a customer. The product owner is responsible for the timely flow of deliverables, the overall productivity of the team and the quality of the solution (Wiegers & Beatty, 2013). A product owner can be from the client organization or internal to the software development organization (in this case they are called a proxy product owner) (Kettunen, 2009).

The different roles in Agile software development are described above. In the following section the relationships in relation to Agile software development are discussed.

2.5 Relationships in Agile software development

The different types of relationships that occur in an organization were described in section 2.4. In the following section, the relationships specific to Agile software development teams are described.

2.5.1 Customer relationships

Building a good relationship with a customer is important in building good software (Wiegers & Beatty, 2013). For example, if you have a close working relationship with the customer they might be more involved in having a discussion about eliciting the requirements (Leau et al., 2012b). Having a good working relationship involves true collaboration between the parties involved in the project, who each understand that their roles and responsibilities are crucial to a project (Highsmith, 2002). Ceschi et.al (2005) found that companies using an Agile approach tended to regulate their relationships with the customer by employing flexible contracts instead of fixed contracts with predefined functions, budgets and timeline.

Constant customer involvement is needed to develop a clear understanding of what the customer actually requires from the team (Ceschi et al., 2005). An Agile software development approach provides the opportunity for collaboration between the customer and the software development teams throughout the whole project development (Wiegers & Beatty, 2013). For example, Pikkarainen et al. (2011) mentioned that communication and collaboration between the stakeholders, testers and developers was reported to be one of the strengths of Agile software development projects. Such strength increased the understanding of the customers' needs and helped to clearly prioritize their business needs to the developers. Their study also found that collaboration with the customer increased the commitment from the development team members and the management towards the

use of an Agile approach. For example, in their study, Pikkarainen et al. (2012) viewed collaboration between the customer and development team that had been established using workshops. Those workshops were conducted to introduce organisations to an Agile approach. In addition to the workshops, a session was held to get feedback and knowledge from the stakeholders. Collaboration and coordination increased the team spirit and facilitated the team in overcoming design problems through the daily stand-up meetings. However, they said that collaboration and communication did not work as planned for all the projects because the team failed to apply the iterative approach. They said that the practice of collaboration was not adopted because the development team did not have the necessary knowledge about working within an Agile process and management was not involved in organizing the planning meetings. Customer involvement was found to be one of the reasons for a project's success since before the requirements are implemented, this needs to be understood by the developers and, in doing this, they should be able to talk to the customer (Paetsch, Eberlein, & Maurer, 2003b).

Agile software development focuses on communication to help reduce the loss of information between the customer and the development teams (Melnik & Maurer, 2004). Communication with the customer should be maximized in Agile software development through direct communication between the customer and the software development teams (Wiegers & Beatty, 2013). Feedback is important in Agile projects to help the development team to enhance the project (Lindstrom & Jeffries, 2004). Face-to-face communication, emails and telephones were some of the methods used for giving and receiving feedback. The feedback given was on the project performance and changes needed to meet the users' expectations. (Melton & Hartline, 2010). For example, an XP (extreme programming) approach requires an on-site customer with a full time presence sharing the workspace and providing quicker feedback on the project (Beck, 2000). However, Martin, Biddle & Noble (2004) found that in the three XP projects they investigated, where each had on-site customers, in each case, the customer representatives were under stress and committed to working longer than normal hours in order to complete the project. This was because the customers were required to contribute significantly more effort than the development team members. However, they also said that all the customer representatives in their study had

the full support of the development team members and direct communication helped both the development team and the customer to receive quicker feedback.

Communication with the customer can be made face-to-face or by email, telephone or through conference calls (Korkala, Abrahamsson, & Kyllönen, 2006). Orkala et al. (2006), using four case studies, found that face-to-face communication proved to be more effective and better with an on-site customer. From their study, it was found that face-to-face communication enabled quick feedback and also had the ability to transmit multiple non-verbal cues to team members. For example, email has the capability to transfer the information but the physical expression involved while transferring the information was missed (Daft, Lengel, & Trevino, 1987).

Often software development projects, both those using traditional and Agile methods, are influenced by different types of relationships between the customer and the development team (Dybå & Dingsøyr, 2008; Faraj & Sproull, 2000). These types relationships include: relationships within the team members; and the customer-team relationship. Each of these relationships is discussed in the sections that follow.

2.6 Relationship between the team members

People have been identified as one of the most important factors in Agile software development (Chow & Cao, 2008). In their study, Chow & Cao (2008) undertook a systematic review of the literature associated with failure and success factors in Agile software development research. They identified the team environment as a critical success factor in Agile software development projects. Team environment in Chow & Cao's (2008) study refers to collocation of the whole team, self-organizing teamwork and a project without multiple independent teams. In Agile software development this involved communication, coordination and mutual support. These factors contribute to the performance of the team in meeting the expectations about the quality of the outcomes, such as functionality, reliability and performance (Lindsjørn et al., 2016). Team members in Agile software development have less formal communication compared to traditional software development. Agile teams are also self-organizing teams in which the team itself makes decisions, estimates and delegates particular tasks (Turk et al., 2014a). During this process there is a focus on interactions among the team members who often are physically placed together.

Software development project can be done with the people located in the different places who work as a virtual team. Virtual teams in software development coordinate their work using issue tracking systems such as Jira and GitHub. These tools allow developers to work from anywhere and communicate asynchronously (Abrahamsson et al., 2017; Ghobadi, 2015).

In software development, the implementation of team structure is not simple and does not necessarily result in success. Because putting people to work together is not just enough. People working together apply different approaches and interpretations to their work. This can result in the conflict. This means that there are human factors involved while people are working in a team(Destefanis et al., 2016).

2.6.1 Customer-team relationships

A hallmark of an Agile approach is close customer involvement (Paetsch et al., 2003b). If the customer, internal department representative, or the product manager, does not have a good sense of the direction of the project then the development team may not get a good sense of the project's requirements (Nerur, Mahapatra, & Mangalaraj, 2005). Agile methods rely on communication and collaboration within the team and with the customer, and this requires a customer who engages with the team (Cao & Ramesh, 2008; Nerur et al., 2005). In an Agile project, the development team works closely with the customer to understand the project's requirements (Cockburn, 2006). Ceschi (2005) found that there are improved relationships between the customer and the development team during Agile software development projects than during plan-driven projects. Ceschi (2005) conducted interviews with 20 project managers from 20 companies. Ten companies were using an Agile approach. The findings showed that 60% of their respondents were satisfied with the relationship with the customer and 40% were satisfied with the delivery time of the Agile project (Ceschi et al., 2005). However, (Hoda, Noble, & Marshall, 2011) found that customer involvement was seen as the most difficult part of Agile. For example, they said that factors affecting the customer-team relationship included: distance, lack of time commitment, fixed-price contracts, and ineffective customer representation (Hoda et al., 2011). The ineffective customer representation refers to customers who are not providing the requirements and

feedback on time and appear to have a lack of understanding of the Agile processes (Hoda et al., 2011).

2.7 Contracts model

In some Agile projects contracts are used to reduce the perceived risk between the customer and the development teams (Highsmith, 2009). In contract-based Agile projects the pricing of the projects must be agreed upon between the parties before software development starts (Poppendieck & Poppendieck, 2007). Poppendiek & Poppendieck (2007) suggested two ways to understand contracts, where the purpose of contract is to either:

- Protect each party from opportunistic behaviour on the part of other party.
- Set up appropriate incentives for companies to work together in a synergistic matter.

Both these points are related to trust between the development team and the customer. When there is no significant trust between the development team and customer, one would likely try to prevent each of them from opportunistic behaviour (Turk, France, & Rumpe, 2014b).

A contract model in Agile software development is characterized by payment and the sharing of the risk between the customer and the supplier (Gopal & Sivaramakrishnan, 2008; Opelt et al., 2013). Some of the commonly used Contract models in Agile software development are Time and Materials, Fixed-price and Target-cost (Franklin, 2008; Opelt et al., 2013). These contract models are explained in the following sections

2.7.1 Fixed- price contracts

A fixed-price contract is defined as a contract in which the customer and development team (supplier) agree on the fixed price that is to be paid to the supplier for the software it provides (Franklin, 2008). In fixed-price contracts the expectations must be clearly documented and managed during the project, as is continuous communication with the customer/product owner (Poppendieck & Poppendieck, 2003b). In Fixed-price contracts the supplier is unlikely to accept changes to the requirements as this will impact on their profit (Bustard, Wilkie, & Greer, 2013). In this type of contract, the software development organization is at more risk, since they have to complete the software while not exceeding the given price (Franklin, 2008). However, in some situations, contract variations are included and, because of this, the project can accommodate changes. One such example is that the customer can add requirements to the project and remove the equivalent

requirements from the list. The risk in such a situation is that the changes in the requirements may exceed the deadline of the project and, hence, increase the cost (Poppendieck & Poppendieck, 2003b). In some projects both the customer and supplier want to share the risk uses target-cost contracts (Franklin, 2008).

2.7.2 Target-cost contracts

In Target-cost Contract models the risk is shared between the customer and the supplier, and the two parties agree on a price for a defined scope of work (Eckfeldt, Madden, & Horowitz, 2005). If the cost of completing the requirements of the project exceeds the target price, the customer and the supplier will share the extra cost (Eckfeldt et al., 2005). Sharing the risk helps this type of contract model to be more accessible than the other types of contract, as discussed previously (Poppendieck & Poppendieck, 2003a). Trust is important in all contracts; however, a greater level of trust is needed when it comes to a Time and Materials contract, because it gives the development team a possibility to take advantage (Eckfeldt et al., 2005).

The type of contract model is chosen according to the trust, the risk customer wants to take and which best suits for the customer and the supplier (Franklin, 2008).

2.7.3 Time and materials contracts

A Time and Materials contract is a contract model in which the customer pays the supplier by the hour, at a specified fixed hourly rate, and for the material used (Franklin, 2008). In this type of contract the customer is able to adjust the scope's priorities in each sprint. This type of contract is not commonly used in situations where the customer or the supplier have little knowledge of each other because of the difficulty in establishing these (Poppendieck & Poppendieck, 2003a). A Time and Materials contract allows flexibility in the design of the software increments (Franklin, 2008). However, this model is not feasible for many customers where they require agreement on the scope upfront (Franklin, 2008).

Chow & Cao (2008), when investigating the literature related to Agile software development approaches, identified several critical success and failure factors in relation to Agile software development projects. They performed a reliability and factor analysis on the identified factors and, from the results, compiled a table of 48 success factors related to Agile development projects. Of these, 11 directly relate to people and processes which may

impact on the relationship between the customer and the software development teams.
These factors are summarized in Table 2-4 and 2-5.

Table 2-4 People's success factors (Chow & Cao, 2008)

People	Failure factors	Success factors
	Lack of necessary skill-set	Team members with high competence and expertize
	Lack of project management competence	Team members with great motivation
	Lack of team work	Managers knowledgeable of the Agile process
	Resistance from groups or individuals	Managers who have a light touch or adaptive management style
	Bad customer relationship	Coherent, self-organizing teamwork
		Good customer relationships

Table 2-5 Process success factors (Chow & Cao, 2008)

Process	Failure factors	Success factors
	Defined project scope	Following Agile-oriented requirement management process
	Defined project requirements	Following Agile-oriented project management process
	Defined project planning	Following Agile-oriented configuration management process
	Lack of Agile progress tracking mechanism	Strong communication focus with daily face-to-face meetings
	Lack of customer presence	Honouring regular working schedule – no overtime
	A defined customer role	Strong customer commitment and presence Customer having full authority

After identifying the critical success and failure factors, Chow and Cao (2008) conducted a web-based survey, gathering feedback from 109 Agile projects from 25 different countries. The findings from this survey helped them further refine their list of critical success factors, resulting in a list of the most critical success factors.

These are:

- Correct delivery strategy
- A proper practice of Agile software engineering techniques
- A high calibre team.

Three other factors that could be critical for success were also identified:

- A good Agile project management process,
- An Agile-friendly team environment
- Strong customer involvement.

A systematic review of empirical studies on Agile software development projects identified 36 studies on Agile software development (Dybå & Dingsøyr, 2008). The findings from this review were divided into four groups: introduction and adoption; human and social factors; perception of Agile methods and comparative methods. Dyba and Dingsoyr (2008) reported a number of benefits from those studies, including the examination of Agile practices in both small and large organizations. These studies explored customer satisfaction with the view to providing feedback about the Agile process. They noted that the majority of studies

included in their review were with Agile teams new to the Agile approach and they suggested further investigation was needed with teams that have been using Agile for a longer period of time.

2.8 Challenges in Agile software development projects

Since their inception, Agile approaches have gone some way towards eliminating many of the problems associated with traditional software development approaches (Misra, Kumar, & Kumar, 2009). Agile approaches have become the dominant approach in software development (Begel & Nagappan, 2007b; Vidgen & Wang, 2009); however, identifying people with the correct skill sets and motivation to work in a team in the development processes is a critical issue in Agile software development (Stankovic et al., 2013b). As people move from a traditional approach, changes in management, attitude and behaviour are required (Nerur et al., 2005). As stated by Boehm and Turner (2005), Agile practices require multitasking characteristics from people. The requirements of a software development project may change at any time during the development process; therefore, team members must have wider skill sets to deal with these changes (Maruping, Venkatesh, & Agarwal, 2009). Changes to project requirements can be handled when there is collaboration between the customer and the development team (Awad, 2005). For example, Chan & Thong (2009) say that customer involvement in Agile projects is a major factor leading towards project success. A study by Hoda et al (2011), mentioned that a high level of customer collaboration is needed in order to frequently release the product and deliver business value. Their study also found that development teams complained that they did not receive enough collaboration from their customer representatives (Hoda et al., 2011). In Agile software development the team often assumes that customer can answer all the developers' questions in order to make the right decision about the requirements (Grisham & Perry, 2005). However, miscommunication between the customer and the team during the requirement gathering exercise may lead to challenges that, then, negatively impact on the success of a project (Dybå & Dingsøyr, 2008; Ghimire, Gibbs, & Charters; Ramesh, Cao, & Baskerville, 2010). For example, Dyba and Dingsøyr (2008) say that it is imperative for customer representatives to have received training in Agile methods before being involved in an Agile project. Training will help customers understand the new expectations that the development team will have of them. The iterative nature of the Agile methodology may

allow a customer the opportunity to delay making decisions about product requirements (Leau et al., 2012a). In turn, changes to requirements may be caused by contractual conflicts (Batra et al., 2010). This may be due to changes that affect the cost or the schedule of a project (Wiegers & Beatty, 2013). In some projects, a gap exists in the expectations of the customer and those of the development team in Agile software projects (Batra et al., 2010). For example, a problem may look simple to a customer but the development team knows it will be a complex change that will impact on the cost and timeline (Grisham & Perry, 2005). Ceschi et.al (2005) says that challenges exist in the customer-development team relationships. For example, one challenge they found was that some customers want faster, lower cost and higher quality software projects whereas developers are after more time and stable requirements.

2.9   Teamwork

The teamwork concept encourages the team members to listen, provide feedback, support and recognize the interests of team members for a better performance. Such characteristics are important for the team members because they promote the individual as well as team performance (Katzenbach & Smith, 2005).

There have been some studies of teams undertaken over time. A study done by Tuckman (1965) identified the different set of phases that the team goes through. The different phases are called forming, storming, norming and performing (Tuckman, 1965). Other studies have focused on the relationship between team members and say that group cohesiveness is important for the success. A study conducted by Guzzo & Dickson (1996) reviewed the literature on teamwork and found that diversity in the membership and the size of the group were related to team effectiveness. They found that team relationships were not consistent across all the groups' tasks. Their study found that the effects of the goal on the group performance were found to be positive (Guzzo & Dickson, 1996). Team performance not only depends on competence of the team itself but also on the organizational culture and context. Cohen & Bailey (1997) report the popularity of empowered teams or self-managing teams. A self-managing team promotes more satisfied employees and lower turnover (Cohen & Bailey, 1997). Other research found that self-managing teams were required for the success of innovative projects (Hoegl & Parboteeah, 2006; Takeuchi & Nonaka, 1986). Although the majority of studies have reported that self-

managing teams have positive effects on the performance of the team, some studies found it can be difficult to implement self-managing teams, who risk failure when there is not enough leadership support (Hackman, 1987).

2.9.1 Teamwork in Agile development

Teamwork in Agile software development is regarded as a black box, while the responsibilities of each team member are divided with collaboration between the team members (Hummel & Epp, 2015). In Agile software development team members are responsible for the end product. These products are developed by building a shared understanding of both the task and the teamwork (Moe et al., 2010). The project goal, project plans, project status and responsibilities must be visible to all the parties involved (Meredith & Mantel Jr, 2011; Moe et al., 2010). There are different activities involved between the team members that may impact the relationship within the team as well as between the team and management (Turk et al., 2014a). These relationships may impact the team performance and the outcome of the project.

Relationships are key in many workplaces and take several different forms (Turk et al., 2014a). These may be relationships between colleagues in a general sense or between groups of colleagues in a more formal sense. There is also a need for relationships to occur between those in a workplace and those outside with whom they deal in order to perform the work (Lee & Xia, 2010; Ryan & O'Connor, 2013).

As in other domains, relationships are of key importance in software development (Bengtsson & Kock, 2014). Software is becoming increasingly important for many organizations and businesses as many require software customized to their operations (Pressman, 2005). Software for such organizations can be developed in-house or by a third party (Ghazawneh & Henfridsson, 2013; Stol & Fitzgerald, 2014). In the process of building software two parties exist - the customer who wants the software and the development team who create the software (Highsmith, 2002). This relationship between the parties is maintained by collaboration or by contract negotiation between the customer and the development team (Cockburn & Highsmith, 2001b). In the course of any relationship, challenges can arise between the parties involved (Boehm & Turner, 2005). Relationships exist in Agile software development as there is more than one person involved in developing the projects.

Agile software development focuses on flexibility and allows for changes in requirements that may occur during the software development process (Greer & Hamon, 2011).

This research explores the challenges that can occur between the customer and the development team when an Agile software development approach is taken. After exploring the challenges between the development team and the customer, the relationship between the customer and Agile software development team is explained from a theoretical prospective. Therefore, the research questions are:

RQ1: The literature suggests that there are challenges between the software development team and the customer representative in Agile software development projects. To what extent do these challenges exist?

RQ2: What types of challenges are present in Agile software development projects?

RQ3: What is the impact of challenges on the relationship between the customer and the Agile software development team?

RQ4: Does the nature of the relationship between the customer and the development team impact on the project outcome?

This concludes the review of the Agile software development approach. Theory selection for this research is the topic of the next major sections. Theories such as Coordination theory (Malone & Crowston, 1994), the Decision-making process, the Teamwork model (Dickinson & McIntyre, 1997), the Theory of holographic organization (Morgan & Ramirez, 1984), Agency theory (Ross, 1973) and Stewardship theory (Donaldson & Davis, 1991) are examined. Once the different theories are explained and its literature examined, a mapping exercise is undertaken to discover a theory that has the potential to explain the relationships in Agile software development teams.

2.10 Teamwork model

Dickinson & McIntrye (1997) first proposed The Teamwork model. Teamwork is defined as the behaviours of team members that give rise to sharing information and coordinating activities (Moe et al., 2010). This model was developed to ensure effective individual and team performance (Brannick, Salas, & Prince, 1997). The Teamwork model comprises seven main components: communication; team orientation; team leadership; monitoring; feedback; backup and coordination, as shown in Figure 2-3.
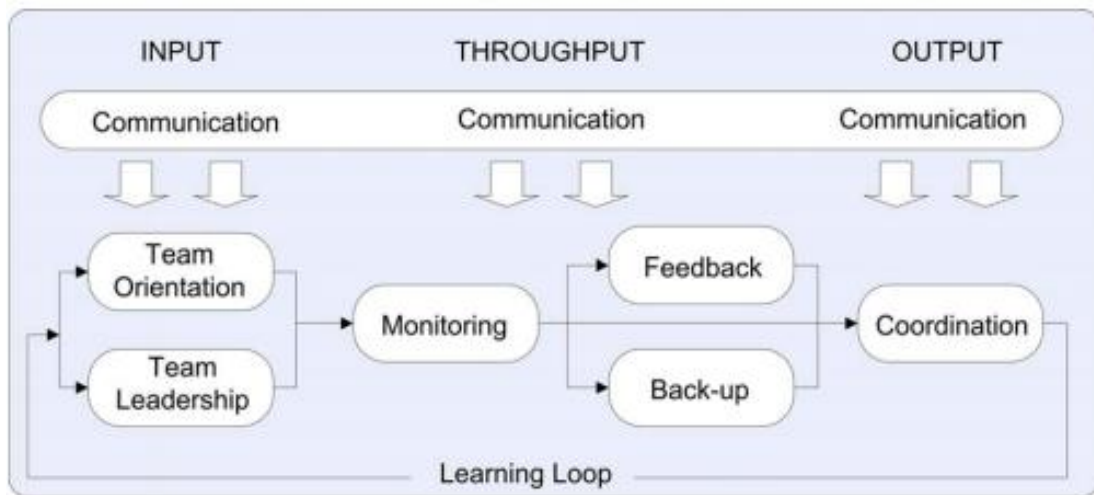
Figure 2-3: Teamwork model (Dickinson & McIntyre, 1997)

The definition and relationship of each of the component of the Dickinson & McIntyre (1997) Teamwork Model are:

Communication: Communication involves the active exchange of information between the members of the team. Communication creates the link between the other components of the Teamwork model.

Team orientation: This component includes team members' self-awareness and attitudes towards the leadership.

Team leadership: This component of a teamwork model consists of direction by the team members and implies the planning and organizing of activities to enable team members to respond.

Monitoring: This component of the Teamwork model refers to the awareness of activities and the performance of the team members. Monitoring is helpful in ensuring that team members are capable of their individual tasks and have a substantive understanding of the task of other members.

Feedback: This component refers to learning from the team member's performance, which requires receiving, giving and seeking feedback from other members working in the team.

Back up: This component refers to a team member helping other team members to complete a task.

Coordination: This component refers to the activities that will help the team member to complete the functions (Dickinson & McIntyre, 1997).

The Teamwork model focuses on the inter-relationships between the teamwork

components and identifies some of the challenges with team orientation, team leadership and coordination within the team.

Moe et al. (2010) proposed using the Teamwork model in a bid to understand teamwork in a Scrum project. The purpose of this model was to understand the nature of self-managing Agile teams and the challenges in introducing self-managing teams in work places that have previously adopted more traditional approaches to software development (Moe et al., 2010). A single-case holistic study project was designed by Moe et al. (2010) to understand teamwork. A Scrum approach was used with the focus placed on human factors that influenced teamwork (Moe et al., 2010). In this project, data were collected from direct observations, project documents and interviews conducted with the Scrum master, the product owner and developers. The Scrum team was visited once or twice a week, with 60 observations in total. Various strategies were used to analyse the data. First, the Scrum project was described in detail so an understanding of the project was gained, including specific project details. Then, aspects of team work were described using Dickinson & McIntyre's (1997) model by referring to the events in the three main phases of the project.

The results showed that bringing individual work values, such as responding constructively to other views, providing support and recognizing others' achievements, that a self-managing team required effort on the part of all team members. However, it was also noted that such changes take time to achieve (Moe et al., 2010). Dickinson & McIntyre (1997) proposed that team leadership and team orientation promoted team members capabilities to monitor the performance of other team members. However, Moe et al. (2010) found that in Scrum teams that were identical in structure, the developers focused on their own modules, typically creating their own plans and making their own decisions. They also found that team members often considered problems they experienced to be personal, such as someone not willing to share information. These findings pointed to low team orientation, resulting in team members being unaware of what others were doing and the team leader not being able to monitor team performance.

The Teamwork model can be used to explore factors related to teamwork, such as communication; team orientation; team leadership; monitoring; feedback; backup and coordination. Apart from these components, Dickinson and McIntyre's (1997) Teamwork Model also considers the teamwork process as a learning loop characterizing teams as

adaptable and dynamically changing over time, which is relevant to the Agile software development methodology.

2.11 Coordination theory (Malone & Crowston, 1994)

The coordination theory, proposed by Malone and Crowston (1990), is based on organizational theory and has been applied in fields such as economics (Simon, 1981) and computer science (Cytron, 1987). Coordination theory is constructed on the principle that activities can be coordinated between actors (Malone & Crowston, 1990), where actors may be the stakeholders, managers or staff working under the management. This is a descriptive-based theory that is used to understand the particular activities in organizational settings.

Strode et al (2012) applied coordination theory in the context of software development. In their study they investigated coordination activities in an organization using teamwork. From their study, they suggested that consideration should be given to activities, such as Agile practises at project, iteration and incorporating *ad hoc* activities in addition to the coordination strategy with the current coordination theory.

A positivist multi-case study research approach was selected by Strode et al (2012) to understand the effectiveness of coordination strategy on software development projects. Positivist research is centred on empirical testing of theories to discover the general principles overarching both the natural and the social world (Lee, 1991). They explain the effectiveness of the coordination strategy by collecting empirical data from a single software development project. In their study, Strode et al (2012) used three cases. In the selected project, the participants were co-located and the primary data were collected using semi-structured interviews. Data were analysed by preparing a full description of each case. The description of each case included details of the organization, the project, the technology used, the team, the development method and any problems related to the coordination in the project identified. Two questions were addressed: The first being how coordination is achieved using an Agile software development approach; the second, being what is the relationship between coordination strategy and project coordination effectiveness in the context of Agile software development. The results showed that coordination was achieved through three coordination mechanisms: synchronization; structure and boundary spanning. A coordination mechanism refers to activities such as

forward planning, coordination roles and contracts. Synchronisation referred to the activities performed by the team members to promote a common understanding of the task, process or expertize of the team members. Boundary spanning is the activities the team or individual perform to elicit information from some unit or organization.

According to Coordination theory, a common problem occurs with the coordination of activities. This theory has been commonly used to look at communication between actors, such as establishing a common language, selecting receivers and delivering outcomes (Lu et al., 2011). Coordination theory is useful to identify goals, allocate resources, coordinate activities and communicate (Malone & Crowston, 1990). This theory helps to examine activities, such as understanding the task, the coordination roles and structure of the team.

2.12 Decision-making process

The process of decision-making was initially observed as making rational selections on the basis of making choices (Dillon, 1998). Decision-making has been used as a theoretical tool to see "how real people think and behave" (Bell, Raiffa, & Tversky, 1988). Depending on the procedural foundation, in the decision-making process there are a number of models and theories that are descriptive, normative or prescriptive (Dillon, 1998). Descriptive decision-making focuses on a particular choice made for a given situation (Dillon, 1998). Normative decision-making describes how decision makers should think and take action based on the chosen rationale. The prescriptive decision-making model describes what decision makers should do and can do. The prescriptive model is based on the combination of normative theory and descriptive theory (Bell et al., 1988).

There are several different frameworks by which decision-making can be modelled, these include the Descriptive Decision theory adopted by Durey et al (2012) to understand decisions that an Agile software development team makes. Durey et al (2012) observed each iteration of an Agile project.

A focus group consisting of 43 participants from 36 different companies and six case studies was used by Drury et al (2012) to help identify the decisions Agile teams needed to make, and the obstacles that threatened these decisions. Different decisions made across the four periods: iteration planning; iteration execution; iteration review; and iteration retrospective, were examined. They found that Agile teams focused more on tactical decisions than on strategic decisions. They described tactical decisions as day-to-day

activities that maintained efficient and smooth operations (Chandler, 1997) and strategic decisions as the long-term health of the organization (Chandler, 1997).

In their study, Drury et al (2012) found that team members in Agile teams were unwilling to commit to a decision and there was a lack of ownership for decisions. Agile team members found obstacles in making decisions because of a lack of information and a lack of participation and interaction within the team. They said that information was not collected in Agile teams because members were inclined to rely on other team members to make decisions. When a team member did not make a decision then other team members have to step in, meaning this process may be hindered by the complexity of the problem being solved, which can make the process more difficult. They also found that the collaborative nature of the Agile software development process was seen as preventing the experts from making the required decisions. However, they made no mention of the influence of customer involvement in the decision-making process or, indeed, if the customer was involved in the decision-making processes.

2.13 The theory of holographic organization

The theory of holographic organization was proposed by Greath and Rafael, in 1984 (Morgan & Ramirez, 1984). In a metaphorical sense, a holographic organization is drawn from the premise that each part of a hologram contains a different piece of information to be used to build the organization as a whole (Nerur & Balijepally, 2007). This means that any part of an organization or team can access the whole organization or team (Moe et al., 2010). Nerur & Balijepally (2007) say that the nature of the Agile philosophy lends itself to the formation of holistic teams.

The very foundation of the Agile philosophy is that teams should contain multi-functional members, who can fill roles when members are missing, self-organize and have knowledge which overlaps (Nerur & Balijepally, 2007).

Nerur and Balijepally (2007) discussed the relationship between Agile approaches and the theory of holographic organization. Agile software development methods facilitate the formation of holistic teams through the concept of interchangeable roles (Nerur & Balijepally, 2007) This helps to build both generalizable and specialized teams that can self-organize in response to new requirements. However, there was no evidence presented in their study exploring this theory in practice.

2.14 Agency theory

In an agency relationship a principal is a person, such as a shareholder, who hires agents for a job. The agents perform the job on behalf of the principal (Eisenhardt, 1989). In this type of relationship, the principal will try to maximize the control of agents, while the agent works towards maximizing individual rewards and reducing principal control. Agency theory was developed in the 1960s and 1970s (Eisenhardt, 1989; Hannafey & Vitulano, 2013) and has been used in disciplines, such as economics, finance and information systems, to explain the relationship between the principal and the agents (Mahaney & Lederer, 2003; Wilding et al., 2012). Agency theory is used to examine components, such as goal conflict, shrinking, information sharing, task programmability and monitoring, to determine the project's success.

The principal-agent perspective has been extensively examined, for information systems development projects and in IT outsourcing projects (Bahli & Rivard, 2003). It is noted by (Aubert, Patry, & Rivard, 2005; Koh, Ang, & Straub, 2004; Mahaney & Lederer, 2003; Rai, Maruping, & Venkatesh, 2009), that Agency theory can be applied to understand failure as well as improvements in the projects.

To understand the application of agency theory in information system projects Mahaney and Lederer (2003) interviewed information system project managers to try and understand how developers were compensated, monitored and motivated. The answers to these questions were explained in two ways; first, in relation to Agency theory and, then, in relation to the project. The authors concluded that the reason for the failure of a project was due to contracts that were not outcome-based, developers not having enough commitment to achieve the goal, a lack of monitoring and a failure to manage goal conflicts. To improve project success they suggest there needs to be awareness of the role of incentives, monitoring and goal conflict in the relationship between the principal and the agent. Although Agency theory explains some possible causes for IT project failure, researchers have suggested that there might be other causes, such as turnover of key personel; changing technology, changing business environments and poorly defined or misunderstood requirements (Mahaney & Lederer, 2003). Recent applications of Agency theory can be found in the area of e-services to increase the efficiency and benefit given to

citizens from these services. The services include access to the information and explaining the relationship between government and citizen (Axelsson, Melin, & Lindgren, 2013). A study by Axelsson et al. (2013) identified several conflicts of interest between the stakeholders and the e-services users. For example, they identified a lack of information and involvement of users in the development process of the e-services.

Agency theory has also been used to examine issues around supply chain management integration and collaboration, and for enhancing operational performance (Natour, Kiridena, & Gibson, 2011). For example, it has been proposed that Agency theory could be applied to supply chain management to increase partners' cooperation, commitment to their relationship, and to build trust and alignment of the goals of an organization (Natour et al., 2011).

Although Agency theory has been applied in several different areas of research, researchers in psychology and sociology have suggested it has some theoretical limitations (Hirsch, Michaels, & Friedman, 1987). Perrow (1986), in particular, says that an assumption made in Agency theory about individual motivation creating the difference in principal-agent interests may not apply for all managers as there are different perceptions among managers when approaching the same task. Agency theory provides a useful way of explaining a relationship where the interest of a principal and an agent can be encourage towards achieving the organization's goal through trust, monitoring and direct control (Cuevas-Rodríguez, Gomez-Mejia, & Wiseman, 2012).

Agency theory has been used in looking at the relationships between teams and customers, relationships within the team. The Agency theoretical framework aids the study of variables, such as transparency, control, visibility, and performance of software development teams. Using this framework, it is difficult to look at human behaviour, such as trust, collaboration between the customer and the teams (Cuevas-Rodríguez et al., 2012).

2.15 Stewardship theory

Stewardship theory was introduced by Donaldson and Davis (1993) to define relationships based upon behavioural factors, such as trust, and the involvement of team members (Davis, Schoorman, & Donaldson, 1997). In Stewardship theory stewards are people who want to do a good job and whose behaviour has higher utility than individual and self-serving behaviours (Davis et al., 1997). Stewardship theory assumes that stewards and

managers (principals) aspire to a higher purpose in their jobs rather than simply self-serving individual economics (Schillemans, 2013). Stewards are motivated by a higher-level need to act well for their firms and are committed to making a success of a project (Miller & Breton-Miller, 2006). The steward and the manager coordinate in order to achieve the organization's goal (Muth & Donaldson, 1998). The focus of Stewardship theory is on internal rewards that are not easily quantified. These rewards include opportunities for growth and achievement (Davis et al., 1997).

Stewardship theory has been used in different contexts, such as e-governance and cloud computing (Pym & Sadler, 2010; Tan, Pan, & Lim, 2005). Stewardship theory is used to examine factors such as the behaviour of people, motivation, power, commitment and management philosophy. A study by Tan et al. (2005) on e-governance identified differences in interests among the stakeholders. They found a lack of transparency with the stakeholder being in a control-oriented mind set. A control-oriented mind set is a type of approach where stakeholders control the team rather than coordinating with the team about the work to be performed. Authority Singapore bought an e-filing system to make e-government robust (Tan et al., 2005). Tan et al (2005) discussed and formulated guidelines for e-governance with a stewardship theory focus on the collaboration rather than control approaches. A study using Stewardship theory on cloud computing by Pym & Sadler (2010) talks about the significance of Stewardship theory on the relationship between user and the resources managed on behalf of users. They have presented the model-based framework for the analysis and management information stewardship in cloud computing. Suggestions from their study include: identifying the stakeholders in the ecosystem, identifying the performances of each stakeholder, and designing how the performance can be aligned with the supporting technology.

The components of Stewardship theory are the stewards and the principals. Using this theoretical framework, we can look at the factors that impact on the relationships between the customer and software development teams, such as control, trust, human behaviour, transparency and the team members' involvements.

2.16 Discussion

Each of the theories discussed above has been applied in different contexts to look at teamwork and team behaviour. Some of the important factors from this previous research

have an impact on the working relationship between the customer/product owner in software development teams. These factors include: communication, team leadership, team culture, team member competences and characteristics, team diversity, commitment, feedback, team member behaviour, trust, information sharing, accountability, empowerment, developer skill, performance management, coordination, social interaction and motivation (Chow & Cao, 2008; Cockburn & Highsmith, 2001b; Fuggetta & Di Nitto, 2014; Lalsing, Kishnah, & Pudaruth, 2012; Lin, 2015; Misra et al., 2009). The matrix shown in Table 2-6 summarizes the factors that each theory purports to take into account.

Table 2-6 Matrix of important factors of the relationship and the theory

Theory/ Factors	Communication	Team leadership	Team culture	team diversity	Commitment	Feedback	Skill	Behaviour	Monitoring	coordination	Motivation	Attitude	Personality
Coordination theory	X		X							X			
Decision making process			X	X				X					
Teamwork model	X	X				X		X	X	X			
Agency theory			X			X			X	X			
Stewardship theory			X			X		X		X			

Grounded theory was also considered early in the study but not selected because of the large body of existing theory available to inform the study. This includes prior research on the Teamwork model which provides a conceptual framework to underpin the research. Since the aim of this project was to understand the relationship between the Agile development team and the customer it has a similar strategy as that proposed by Moe et al.,(2010) in their development of the Teamwork model, so model is thought to be significant for our study. Hence, Research Question 5 **(RQ5)** in this research.

RQ5: Can the Teamwork model adequately explain the nature of the relationship between the customer and the development team in an Agile development context?

2.17 Summary

This chapter has provided the background to the proposed study of relationships within Agile software development teams. There were three major areas this review addressed:

first, relationships; secondly, Agile software development and, in the final section, the theories used to examine the relationships were reviewed.

The first section began with the brief about the relationships. The types of relationship in the workplace followed by team relationships. A good relationship is important for any business to perform well. There are different relationships that can occur in software development (Cramton & Webber, 2005).

The second section related to software development teams. Research about Agile teams was reviewed and the roles of different team members in Agile teams were highlighted. The literature focusing on the customer involvement in Agile software development was reviewed along with the challenges between the customer and the Agile software development team. The relationship within the Agile teams and between the customer and the team was also reviewed. Agile software development, one of the commonly used software development approaches, involved relationships between individuals, such as workplace relationship, team relationship and customer relationship (Cockburn & Highsmith, 2001b; Dybå & Dingsøyr, 2008; Paetsch, Eberlein, & Maurer, 2003c). There are different instances in Agile software development where relationships play an important part. Some of these instances are: requirements collection, customer collaboration, and communication with in the team and the customer (Cockburn & Highsmith, 2001b; Dybå & Dingsøyr, 2008; Kitto, Chesters, & Grbich, 2008). To understand and explain the relationship in the final section of this chapter the potential theories that can be used to explain the relationship were reviewed.

The research methods proposed for this study are described in the next chapter.

Chapter 3: Methods

3.1 Introduction

The high level question that this research addresses is studying the impact of the relationship between the software development team and the customer representative on project outcomes. The research sub-questions being addressed are outlined in section 1.1

This chapter presents the research methods adopted in this study in order to address the research questions. The chapter starts with a discussion of the research methods chosen and provides a justification for this approach. The chapter also describes the instrument validation approach and data collection methods, including the sampling techniques used.

3.2 Overview

The purpose of research is often to test a theory or to use a theory to explain the results of research by providing a suitable foundation on which to base the explanation (Voss, Tsikriktsis, & Frohlich, 2002). Research studies about relationships between groups of people may be affected by differing conditions, both situational and cognitive (Hoegl & Gemuenden, 2001). To do this, there are number of paradigms used, such as positivist, interpretivist, transformative and pragmatic. A pragmatic (Mackenzie & Knipe, 2006) paradigm is used in this research to answer the research questions. The pragmatic paradigm contains tools from both positivist and interpretivist paradigms and helps in the knowledge generation process in research, which involves hypothesis generation and testing (Golafshani, 2003; Guba & Lincoln, 1994; Mackenzie & Knipe, 2006).

The research questions, as presented in Chapter 1 are:

RQ1: The literature suggests that there are challenges between the software development teams and the customer representative in Agile software development projects. To what extent do these challenges exist?

RQ2: What types of challenges are present in Agile software development projects?

RQ3: What is the impact of challenges on the relationship between the customer and Agile software development teams?

RQ4: Does the nature of the relationship between the customer and the development team impact on the project outcome?

RQ5: Can the Teamwork model adequately explain the nature of the relationship between the customer and the development team in an Agile development context?

To answer the research questions an exploration of the views and experience of participants working in a software development environment is required. Therefore, it is necessary to both explore and explain, rather than only explore or explain, the experiences of participants working in software development companies. It is important for this research to find out from the development team members why they believe challenges occur.

In order to obtain detailed and in depth answers for the research questions a mixed method approach is proposed. In a mixed method approach both quantitative and qualitative data are collected, based on the research questions (Creswell, 2011).

A mixed method approach is used as a strategy whereby the strength of a questionnaire can be enhanced by understanding the information in detail through semi-structured interviews (Creswell, 2011). This study consists of a questionnaire that aims to gain an understanding of the Agile processes used, and the challenges that occurred during Agile software development projects, combined with a semi-structured interview. A questionnaire was chosen as it provides a quick, inexpensive and accurate means of accessing information (Groves et al., 2011). Mixed-methods can be used to improve the power of the analysis by combining qualitative and quantitative techniques (Johnson & Onwuegbuzie, 2004a). For example, mixed methods help in developing context-specific instruments and provide a complete and broad understanding of the research problems. (Johnson & Onwuegbuzie, 2004b)

Data collection from participants took two approaches:
 I. Online questionnaire (quantitative data and qualitative data)
 II. Semi-structured interview (qualitative data)

The questionnaire for this study consisted of questions, including the respondent's role, time they have been working in software development and the size of the organization. The questionnaire also included questions to gather information about the Agile practices used in software development projects.

In order to obtain a greater understanding of a participant's experiences in an Agile team environment a follow up interview was used. Semi-structured interviews were considered

the appropriate method to gain an in-depth understanding of Agile projects (Sarantakos, 2012). An in-depth understanding of participants' work in Agile teams is a requirement of this study. Semi-structured interviews give the interviewer the chance to simplify the questions, based on the context of the interview, including the experiences of the respondent, which help make communication easier (Cohen & Crabtree, 2006). Semi-structured interviews also allow the interviewer to further explore a topic if a given answer does not sufficiently address the question (Cohen & Crabtree, 2006). Therefore, a semi-structured interview was preferred over a structured interview.

Quantitative methods are used to explore the information and a qualitative approach is used to explain challenges in detail. In quantitative data collection, data are collected that can be expressed numerically (Creswell, 2013). Because the data are in a numeric form descriptive and inferential statistical tests can be performed on the data. Descriptive statistics use the mean, median and standard deviation to describe what the data shows. Inferential statistics, such as Pearson's product-moment correlation coefficient, are used to explore the relationship between different sets of data, based on the co-occurrence of data (Lee Rodgers & Nicewander, 1988).

Qualitative data provide details about human behaviour or personal characteristics and includes information about the participants' experiences, needs, routines, desires, and uses cases and other detailed information, essential for the research that quantitative studies cannot provide (Neuman, 2005). Examples of qualitative data include open-ended questionnaires, unstructured interviews and unstructured observations (DiCicco-Bloom & Crabtree, 2006). Questionnaires are used as research instruments, in which a series of questions are asked for the purpose of gathering information from the respondent. An interview is a conversation between the interviewer and the respondents that is used to gather information from the respondents (Witzel, 2000).

A qualitative method was used to gain more detailed opinions and insights from a subset of the people who had completed the questionnaire. In this study, the qualitative data helped to explain further the different challenges people faced during software development. A qualitative study is also essential for defining, theorising, explaining, exploring and mapping fundamental analyses to understand the relationships (Shank & Brown, 2013). The

qualitative data in this study were gathered through the semi-structured interviews. Therefore, to meet the objectives of this study, a mixed method approach is chosen.

The following section outlines the research design, the validation process and the data collection.

3.3 Research design

This study was designed to understand the relationship between the software development teams and the product owner/customer. Hence, this study was designed as a multi-study project. Part 1 of the research involved participants employed in software development roles in which Agile approaches were taken. The second part of the study involved participants who have worked in the product owner's role.

A matrix of potential theories was presented in section 2.16. Creating a matrix helped in viewing which theoretical framework could be used to look at the most common factors that contributed to forming the working relationships in Agile software development teams, including the product owner. The Teamwork model was chosen for this study because this model's framework can be used to look at more components that have an impact in the relationship between the software development teams and the product owner.

A questionnaire was constructed using the teamwork model framework to inform the development of questions. It included background information, such as their role, time they have been working in software development and the size of an organization. The questionnaire also included questions to gather information about the challenges and factors that contributed to the relationships in Agile software development.

For the first stage in the data collection approach a questionnaire was developed. Content validation and a pilot study of the questionnaire was conducted, both are explained in the following sections. The second stage of Part 1 of this study involved semi-structured interviews with a subset of people who had completed the online questionnaire. The steps involved in research design are shown in Figure 3-1 and explained in detail in the following section.
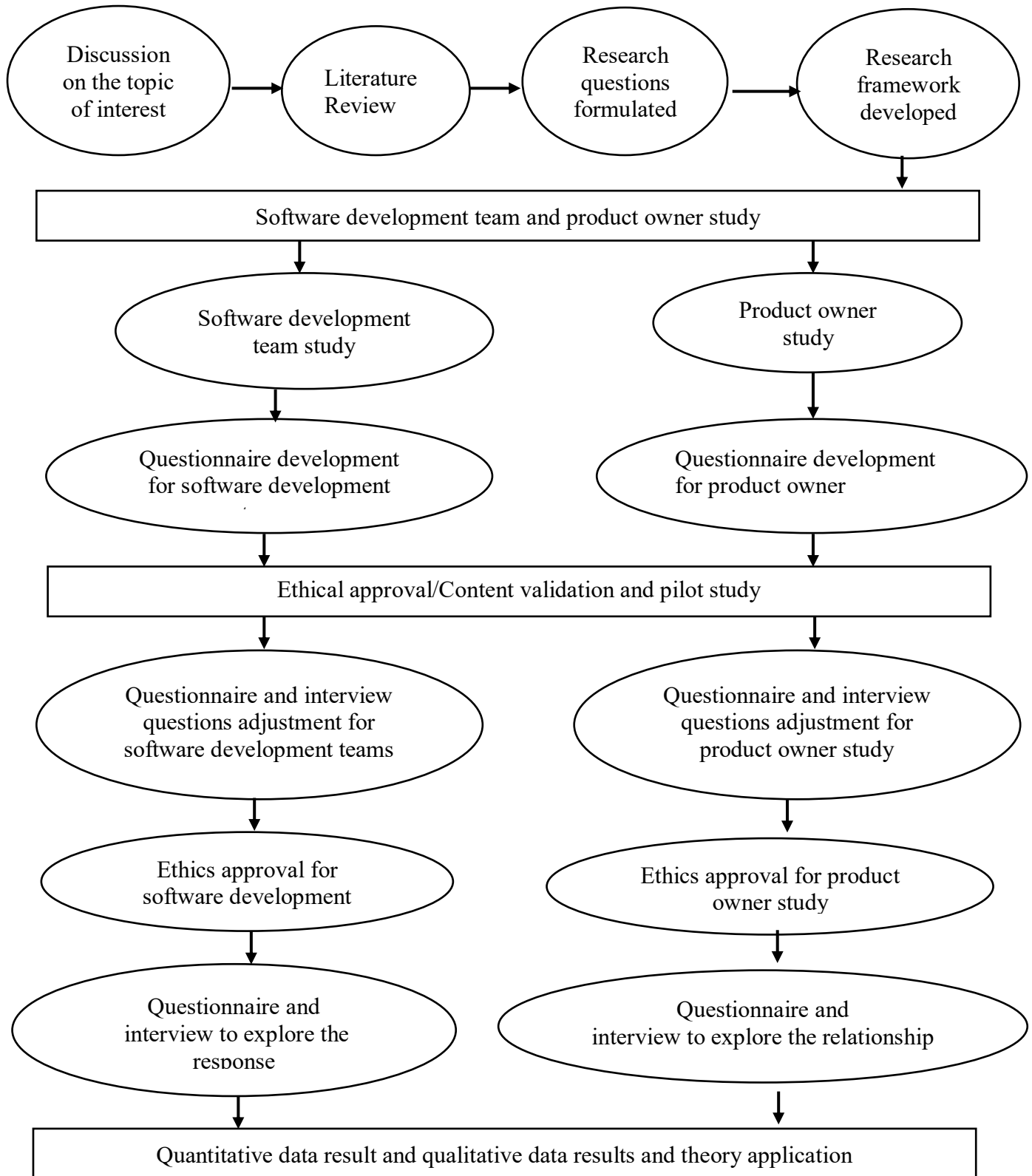
Figure 3-1 Research design overview

This research includes two studies – a study of a software development teams and a study of product owners. A separate instrument was developed for each study to understand the views of the Agile software development teams and the customer representative/product owner, to explore the challenges and to explain the relationships in Agile software development projects. The instrument development, validation and data collection process for each study are explained in the following sections.

3.4   Instrument development

Before collecting data, a research instrument was designed. For this, a list of questions focusing on the research were developed. These research instruments were written after considering the topic of the research, the literature review, and in accordance with the theoretical framework chosen for this study, as mentioned in Chapter 2. The initial questionnaire incorporated 60 items and comprized five sections:

- Background data of the software development teams/product owner

- Type of organization

- Agile practices used

- Project information

- Difficulties faced during the project development

In order to validate the research instrument developed following steps were taken.

3.5   Instrument validation

Instrument validation is carried out to find out whether the instrument measures what it is intended to measure. The instrument validity in this research was to see whether the questionnaire developed for the research was appropriate for the targeted respondents. There are different instrument validation processes, such as pre-tests or pilot studies, content validity, construct validity and reliability (Boudreau, Gefen, & Straub, 2001). The validation process used for this research are discussed below.

3.6   Content validation

Content validation concerns the extent to which the questionnaire developed for this study represents the content to be measured (Haynes, Richard, & Kubany, 1995). The process of validating content allows a researcher to find out whether the measure includes questions in all related sectors in order to test the problem (Polit, Beck, & Owen, 2007). Content validation can be done through the verdict of people with experience in a particular field,

also called a panel of experts (Lawshe, 1975). In this study, panels were formed using people with experience in software development. Typically, people chosen for content validation were professional people who have published or worked in a related study area (Haynes et al., 1995). In this study, content validation experts were the people who have published in software development or been involved in developing software. For this process, content experts were located through different networks, including university graduates of computing and software development and experts from the Agile Professional Network[2] in Christchurch. After identifying potential candidates, an email request for participation was sent. In total, four experts were used for content validation. The experts worked through the questionnaire and provided feedback.

A content validity approach was undertaken in this study as one of the methods for quantifying the data from the expert panel. One of the widely used content validity methods for quantifying data from expert panels is the content validity index (CVI) (Polit et al., 2007). The content validity index is used to calculate item-level relevance as assessed by a panel of subject matter experts (Polit et al., 2007). Lynn (1986) advised a minimum of three experts but the best number would be somewhere between three and ten (Lynn, 1986). The item ratings are typically on a 4-point ordinal scale. Lynn suggested that 3- or 5-point rating scales might be used, but she preferred using a 4-point scale to avoid having a neutral point. For each item, the CVI was computed as the number of experts giving a rating of either 3 or 4 divided by the total number of experts. For example, an item that was rated as quite or highly relevant by 3 out of 4 judges would have CVI of 0.75.

One concern with this approach was that CVI focused on the agreement and relevance without considering whether the instrument consisted of items that actually measured what they were intended to measure (Polit et al., 2007). Another concern with this approach was that there was no consideration made for a chance agreement. In order to determine the chance agreement Polit et al. (2007) modified the CVI with the kappa coefficient (k*). This kappa coefficient was to ensure that item included in the instrument remained true to what it was developed for (Polit et al., 2007). The CVI with chance agreement was used for the content validation (see Appendix A).

To compute k*, the probability of chance agreement was calculated first. The equation used

---

[2] http://www.meetup.com/Christchurch-Agile-Professionals-Network/

to calculate probability chance agreement is:

PC = ((N!/ A!(N-A)!).5N

where N= number of experts and A= number of experts agreeing on the relevance.

The equation used to calculate the chance agreement is:

K*= (I-CVI-Pc)/1-Pc

Evaluation for each item calculation k* value is presented in

Table 3-1.

Table 3-1 k* value evaluation (Polit et al., 2007)

k* value for an item	Evaluation
>0.74	Excellent
0.60 to 0.74	Good
0.40 to 0.60	Fair (considered for revision)
<0.4	Consider for deletion

A K* value was calculated for each item. (See the appendix A for the complete K* table). Out of 60 items, 54 were evaluated as excellent and six were evaluated as good. There was no item evaluated for deletion or revision. After completing the content validation process the pilot study was undertaken. This process is explained in the following section

With the completion of content validation, a pilot study was conducted to see if the instrument developed clear for the respondent.

3.7   Pilot Study

A pilot study is a research process that pre-tests selected research methods on a smaller sample (Zikmund et al., 2012). Conducting a pilot study is an important step because, when the researcher conducts a trial run of the selected method, the researcher will become aware of any faults and can make any appropriate changes (Arksey & Knight, 1999). A pilot study questionnaire was developed after content validation. The questionnaire was piloted with people involved in software development. The online questionnaire and the interview questions were tested with the pilot study participants to assess whether the questions were clear, understandable or ambiguous. Any questions that created confusion were rewritten for the main study. Pilot studies help to judge how long it takes to run an interview and also help in estimating the time required for a participant to complete a

questionnaire (Arksey & Knight, 1999).

3.7.1   Data collection from the pilot study

The purpose of the pilot study was to test the instruments developed; the process adopted to conduct the pilot study is discussed below.

Potential participants were identified from organizations whose core business was software development or, as part of their business, they developed software for in-house use. These organizations were classified into three categories: contract/bespoke, in-house development and product development. Contract/bespoke software development organizations develop software for a specific organization or user. Whereas, in-house development organizations use in-house teams to develop software for company use. Product development companies design, create and market new products or services to the customer.

A mixed method approach was used to collect the data from the participants. A questionnaire and the interview questions were developed. Quantitative data were collected through an online questionnaire and qualitative data were collected from semi-structured interviews.

Participants whose email addresses were obtained from a number of different sources, including profession networks the software professionals belong to. The contacts, once established, were located and classified in the categories of contracts/bespoke, in-house development and product development. A random selection was made from each group and an invitation email was sent to participate. If no response was received within a two-week period a reminder invitation was sent. This process was continued until ten questionnaires had been completed. In total, 30 emails were sent and 15 people replied. From those who replied five people did not fit the selection criteria so they were excluded from the study. They didn't meet the criteria as they were not directly involved in software development even though they were from a software development organization. The response rate for the pilot study was 33 per cent. (10/30). Pilot study data were collected over a two-month period. The Qualtrics platform was used for delivering the online questionnaire. The data from the questionnaire were imported into Microsoft Excel directly from Qualtrics. Recorded interviews were transcribed verbatim in Microsoft Word and stored on the networked file system of the university that was allocated to the researcher.

The pilot questionnaire included demographic and experience questions to explore the extent to which participants have experienced difficulties within the team or with a customer representative. There were three participants each from contract/bespoke development companies and the in-house development companies and four participants from product development companies.

All participants who completed the questionnaire were personally contacted to take part in a semi-structured interview (See the appendix A for the interview questions). The time and location, their consent to an audio recording, and the expected duration of the interview, were also specified in an email send during the initial contact. One pilot respondent was unable to complete the interview within the required time frame due to other commitments. Nine interviews were conducted with participants using an audio recorder to capture the interview. Interviews were transcribed and a thematic methods were used for data analysis. The data obtained from the interviews helped to revise the interview questions for clarity and ensure that required information was obtained.

3.7.2   Findings

The ten participants had a range of experience, from one year to 17 years. The average experience of the pilot participants was 13 years. The participants were employed in a variety of roles, including: Scrum master, developer, tester, Business analyst. The following Table 3-2 shows the number of participants by job title.

Table 3-2 Pilot study respondents

Job Title	Number of Participants
Operations manager	1
Software development manager	1
Developer team lead	1
Sr. Business analyst	1
Lead software designer	1
Developers	3
Technical project lead	1
Project and business solution managers	1

Of the Agile approaches solicited from participants the most used was Scrum (50%). Scrum was also used in combination with other approaches. The following table, Table 3-3, shows the approach used in the project. Some of the approaches were used in combination with others.

Table 3-3 Pilot study approaches used

Use of different approaches	Total	% (percentage)
Scrum	5	50%
Scrum and eXtreme programming (XP)	1	10%
Scrum and Feature Driven Development (FDD)	2	20%
Scrum and Kanban	2	20%
Scrum and Lean software development	1	10%
Scrum and Scrum-ban	2	20%
Test Driven Development (TDD)	3	30%
Behaviour Driven Development (BDD)	2	20%
Scrum and Waterfall development	1	10%
Waterfall	1	10%
Prototyping development	1	20%
Scrum and Prototyping development	1	10%
Code and fix	1	20%

The questionnaire consisted of 38 questions. On average, the participants answered 85% of the questions. Some of the questions were not answered as expected, this could be because the question was either not relevant to their role or not clear (Van Teijlingen & Hundley, 2002). A question was included at the end of the questionnaire asking the participant whether they found the questions easy to answer and relevant to their roles. Eighty per cent of participants answered that the questionnaire was understandable. Twenty per cent of the participants provided some valuable feedback about the relevance of questions according to their roles. The feedback was considered when developing the instrument for the study with the software development teams as well as the product owner study.

3.8   Pilot study conclusions

Feedback from the pilot study was useful in restructuring the online questionnaire and the interview questions. It was also used to determine whether the instrument was able to get the information required to address the research question. The pilot study also helped estimate the time required to complete each part of the survey.

3.9 Data collection from software development teams

Primary data were gathered by means of an online questionnaire from the people currently involved in software development around New Zealand. Respondents completed the questionnaire for software development team study in 18 minutes on average, with a range of 13 to 26 minutes. Responses were collected over a six-month period between August 2015 and January 2016. (See Appendix F). Semi-structured interviews were conducted with a stratified random sample of respondents who had completed the online questionnaire, to gain a more in-depth understanding the relationships between the customer and Agile software development teams. (See Appendix G).

3.9.1 Sample selection and conducting the online questionnaire

The total numbers of software development companies in New Zealand is estimated to be 1120. This number is calculated from business demographics in an IT sector report obtained from Statistics New Zealand[3], the Canterbury technology and innovation report from Flux[4] and the National ICT Innovation Institute[5]. The number of participants to survey for this study was chosen to be 285. This number was obtained from the calculation of sample size from the population of software development companies in New Zealand using Cochran's sample size formula for categorical data:

$n = (t)^2 * (p)(q)) / (d)^2$

^{w}here t value at 95% is 1.96

Alpha level (level of risk) = .05,

(p)(q) = estimate of variance = .25 and

Acceptable margin of error (d) = 0.25 (Cochran, 2007).

$$n_0 = \frac{(1.96)^2 * (.5)(.5)}{(0.05)^2}$$

Therefore, for a population of 1120, the required sample size is 384. However, since this sample size exceeded 5% of the population (1120 *0.05=56), Cochran's (1977) correction formula was used to calculate the final sample size. These calculations are as follows:

$$n1 = n0/(1 + \frac{n0}{population})$$

---

[3] http://www.mbie.govt.nz/info-services
[4] http://fluxnz.com/
[5] http://www.nzi3.com

= 384/(1+ 384/1120)

= 384 (1+ 0.343)

=384/1.343

=285

Participants were recruited to form a representative sample of software development organisations across New Zealand. Participants were recruited using one of these approaches:

- via email,

- requesting to forward invitation to the people,

- email address obtained from a business card,

- through the Agile Professional Network,

- through the Test Professional Network [6]

- a company's contact email address.

An online questionnaire was used to collect the data from people currently involved in software development. Participants were selected based the non-probabilistic sampling techniques of purposive and 'Snowball' sampling. These techniques were chosen to select participants from the research areas (Bryman, 2012) because of the nature of the research.

Purposive sampling, also known as judgmental or selective sampling, is a type of a non-probability sampling. The main object of this type of non-probability sampling is the requirement to rely on the judgment of the researcher (Kothari, 2004). A key to purposive sampling is not to randomly select a sample from a population in order to make generalizations about that sample but to select for a specific purpose (Patton, 2005). Purposive sampling relies on participants who are able to provide rich information about the research area. Therefore, the purposive sampling method was chosen for this study because it allowed selection of a sample based on the purpose of the study - research (Creswell, 2011; Kitto et al., 2008).

In addition to purposive sampling, an element of snowball sampling was also used. Snowball sampling is a non-probability sampling technique where data are collected from known respondents and they, in turn, are asked to provide information to locate other respondents

---

[6] http://www.meetup.com/Christchurch-Test-Professionals-Network/

(Noy, 2008). Snowball sampling was used in this study to reach to the hidden population of the people involved in Agile software development (Atkinson & Flint, 2001). The population of the product owners working in Agile software development was hidden. The number of participants needed for this study was unknown and estimated to be more than the number of organizations that followed the Agile software development approach, which was estimated to be more than 1120. Where possible, random sampling is recommended in carrying out research as randomisation reduces biases and results can be generalized (Topp, Barker, & Degenhardt, 2004). However, random sampling may not always be feasible (Bernard, 2011). This may occur because not everybody in the population is willing to participate (Kothari, 2004). The purposive sampling method can still provide the reliable and robust data needed for research despite of a perceived bias involved in sampling (Bernard, 2011). For this study, participants were chosen from several different software development organizations. The selection criterion for the respondents was the people who were involved in Agile software development teams. Participants who were working in software development organizations but not currently involved in software development were not included in this research. A filter question was used in the online questionnaire to select participants currently working in Agile software development. This had been specified in the recruitment email about the roles that will be included in the study and who will need to opt-in to take part.

3.9.2   Sample selection and conducting the interviews

Face-to-face interviews with semi-structured questions were undertaken with a subset of the participants who were randomly selected after completing the questionnaire and agreeing to supply more in-depth information about their roles in the software development process. (See Appendix C for the interview question for software development teams).

Participants were classified into three different groups (bespoke/contract, in-house, product development), according to the activity of company they worked for. Participants from each group were selected randomly. The selected participants were sent an invitation email to take part in an interview. Interviews were conducted with respondents from Christchurch, Auckland and Wellington

For this study and the pilot study, data collection involved soliciting information from people about aspects of their employment. In qualitative and quantitative research one criticism is that the nature of the research allows close contact with the people and this may lead to ethical problems (Sarantakos, 2012). The questionnaire and the interview used for this research were directed to people; hence, ethics consideration was considered at an early phase of the research. Ethical consideration was also taken, to prevent inconvenience to the participants and to avoid any circumstance that may put the participant taking part in the research at risk. For example, the data would be only used for research purposes, individual identification will not be mentioned in the results or any publication.

Under the Lincoln University Human Ethics policy, all research involving people requires approval from the Lincoln University Human Ethics Committee prior to the research beginning. During this process an application form, a copy of the questionnaire, a copy of the interview questions, the research information sheet and the invitation email used to recruit the participants were submitted to seek the permission before starting to collect data.

Separate applications were made for the following:

- Content validation and pilot study

- Study involving the software development teams and

The Approval to proceed with research was received from LUHEC for the pilot study (P-2012-02) on 25th June 2014.

Approval for the project: The impact of the relationship between a customer and a software development team in the outcome of software development projects (Application No: 2015-09) was given on 27 March 2015. (See Appendix for the HEC approval letter).

3.10 Product owner study

The main aim of this research was to understand the relationship between the software development team and the product owner. Hence, a separate study with the product owners was conducted in addition to the software development team study.

A research instrument was designed for the product owner study before conducting the data collections that had a list of questions focused on the research questions. The research topic, literature review and theoretical framework were considered when writing the

research questions. The initial questionnaire incorporated 15 questions. These questions consisted of items related to the product owner:

- Type of an organization
- Project information
- Difficulties faced during the project development

3.10.1 Product owner study instrument validation

Both the instruments for the software development teams and the product owner study were validated using content validation, which is explained in section 3.6. After content validation some of the items related to the product owners' study were also modified according to the feedback given by the experts. The pilot study was conducted after completing the content validation (see Appendix A). The questionnaire for the product owners and the software development teams were tested with pilot study participants to assess whether the questions are clear, understandable or ambiguous. Any questions that created confusion were rewritten. Interview questions for the product owner were written to get the information in detail.

Separate applications were made for human ethics approval for the study involving the product owners. Approval for the study involving customer representatives (Application No: 2015-43) was given on 27th October 2015. (See Appendix D for the HEC approval letter).

3.11 Data collection from customer representatives/product owners

Primary data were gathered by means of an online questionnaire from the people currently involved as a customer representative/product owner in software development around New Zealand (Appendix G). Semi-structured interviews were conducted with the people who had completed the online questionnaire (Appendix G).

Participants were recruited from different organizations across New Zealand who were contacted via an email circulated to different software development groups, such as the Agile professional network. Agile Auckland [7]and Agile Welly. The Agile Professional Network is a group of people working in software development using the Agile approach or who were interested in the Agile approach and were based in Christchurch. Agile Auckland is a group of people working in software development using the Agile approach or who were

---

[7] http://www.meetup.com/Agile-Auckland/

interested in the Agile approach and were based in Auckland, and Agile Welly [8]is a professional Wellington-based Agile group. Event organizers from each of these groups were contacted and asked to circulate the invitation email to their members. The organizers were also requested to allow me some time in their events to talk about my research and recruit participants.

Face-to-face interviews were undertaken with all of the participants who completed the questionnaire. Purposive sampling, as explained in section 3.9.1, was used to recruit participants.

It was difficult to determine the population size of the customer representatives/product owners. In New Zealand, there are approximately 1120 organizations whose core business is software development or that have some involvement in developing software projects. In an organization that uses Agile practices, teams were formed where at least one member took the role of a customer representative. There may be more than one team in an organization or a customer representative may be working with more than one team. In this case, it is difficult to establish number of projects that required a customer representative (product owner). Therefore, the total population of product owners/customer representatives was unable to be determined due to the lack of data on the number of projects undertaken. However, the respondents for the product owner study were aimed to include those from the different types of organizations, such as bespoke, in-house and product development.

3.12 Summary

In this chapter, the research methods have been discussed by demonstrating an understanding of quantitative and qualitative research. This chapter also presented why the mixed method was preferred for this research. The research methods and instruments used were discussed with justification for the choices made for this research. In this research two different studies were conducted, one with the software development teams and the other with the product owners. The next chapter presents the results from the quantitative data and include the results and discussion from the qualitative data for both studies.

---

[8] http://www.meetup.com/AgileWelly/

Chapter 4:   Software Development Team study Result and Discussion

4.1   Introduction

This chapter presents and discusses the findings from the software development team study conducted using a mixed methods approach, as previously described in section 3.2. Quantitative data was gathered using an online questionnaire and analysed using Microsoft Excel and SPSS. This process is described in section 3.9. Qualitative data were gathered from semi-structured interviews, then transcribed and analysed using a thematic approach.

This chapter is divided into three sections. The first section is devoted to presenting and discussing the findings from the questionnaire followed by the findings from the interview in the second section. The chapter concludes with a summary.

4.2   Questionnaire

The first phase of this study used an online questionnaire to collect data from respondents. The data collection process, data analysis and results are described in the following sections. The questionnaire (see Appendix F) was administered using the online questionnaire tool Qualtrics[9]. Respondents accessed the questionnaire through a link provided in a recruitment email.

Target respondents were those who worked on software projects using Agile software development methods. The following avenues were used to recruit appropriate respondents:

- Agile professional groups via electronic communication
- Agile professional groups via their in-person meetings
- Direct contact via employers
- Personal contacts and their networks

In general, there were not any well-defined means to identify the people involved in Agile software development who made up the sample potential participants.

It has been observed by other researchers that calculating the response rate for online questionnaires can be difficult (Sheehan, 2001; Van Selm & Jankowski, 2006; Wright, 2005). For example, a study was conducted by Zhang (2000) with the participants using an online

---

[9] https://www.qualtrics.com/

survey and post mail. Their study found that online questionnaires had challenges in their presentation and the interpretation of questionnaire results, such as representativeness, validity and response rates (Zhang, 2000). They also mentioned that in some case it was very difficult to calculate the response rate when the questionnaire was distributed through multiple related mailing lists or news groups. Similar difficulties were observed for this study and, because these difficulties, a response rate could not be calculated.

Seventy-nine respondents completed the questionnaire and, of these, 73 were software development team members and six were product owners. The data from these 73 software development team members was used for the software development team study and the other six were contacted for the product owners' study. Seventy-three responses were considered to be appropriate for this study as a sample of these respondents were contacted again to obtain further information to explore the data in detail. Hill (1998) said that when the researcher knows that the sample should be as large as possible, but does not have access to the large number of people in such cases the researchers needed to think about all the time, space and energy spent. Because of this, the researcher had to settle for a fewer respondents than expected (Hill, 1998). A higher participation rate was not achieved because of the time constraints for this research. Most time was spent in locating the respondents who were involved in Agile software development.

Data from the questionnaire were downloaded and imported into SPSS and Microsoft Excel to perform statistical analysis, such as mean, standard deviation and correlations.

The data imported from Qualtrics had option numbers rather than the actual values for the categorical data. These were converted to category labels in Microsoft Excel rather than numerical values to aid analysis and interpretation.

In the following section responses from the questionnaire are analysed.

4.2.1   Organizational profile

The first section of the questionnaire collected contextual data, such as organisational type, size of an organization, project type, and organizational size in terms of the number of employees involved in the software development.

The software development organisations were separated into three categories:

- In-house development

- Product development

- Bespoke (contract) development.

Respondents were given the option to select "other" and write in the category, if their organization did not fall into one of the three categories. Table 4-1, presents this categorization.

Table 4-1 Organizational categorization

Company type	Number	Percentage
Product development	38	50%
In-house	24	32%
Contract	11	14%
Other	3	3%

The largest group of the respondents, as shown in Table 4-1, categorised their organization as "product development"; these made up 50% (38) of the respondents. Thirty-two per cent (24) were categorized as in house and 14% (11) as a bespoke development. This suggested that there were some projects included in this study who regarded themselves as using Agile approach but used different approaches that were not common Agile practices.

To understand the nature of the projects, data related to the project contracts were collected, and summarized in Table 4-2. Respondents selected from seven options describing the contractual relationship between their organization and the client for the most recent project they had undertaken.

Table 4-2 Project Contract Models

Contract models	Respondent	Percentage
I don't know	19	26%
There was no contract	19	26%
Fixed price contract	9	12%
Time and material contracts	8	11%
I am not able to say	8	11%
Other	6	8%
Target cost	4	5%

Of the 73 respondents, 26% (19) reported that they did not have a contract for project development and 26% (19) of respondents did not know the details of the contracts. Of those with a contract the most common was a fixed price contract, which was 12% (9) of the respondents. Data related to the project type was collected to understand the type of projects the participants were involved in and also to understand the relationship between the development team and the product owner in the context of the different types of projects.

Data were collected about the size of the organization where the respondents were employed. Table 4-3 presents the data related to the size of organisation.

Table 4-3 Organization size by number of employees in software development roles

Number of people	Number of organizations	Percentage (%)
Fewer than 5	12	16.44%
5 - 10	7	9.59%
11 - 20	3	4.11%
21- 40	11	15.07%
41 - 100	25	34.25%
101 - 500	3	4.11%
501 - 1000	7	9.59%
More than 1000	5	6.85%

Table 4-3 indicates that a largest group of questionnaire respondents were organizations with between 41-100 (34%) employees. Twelve respondents, 15.8% of the total response, represented small organisations with fewer than five employees. This information suggested that the respondents for this study represented the diversity of the industry. This diversity in the industry was important for this research in collecting wide information for this current study.

4.2.2   Employment information

Table 4-4 presents the employment role of each of the respondents. Of the 73 respondents, the largest group were developers, at 42 (57%). The smallest role represented was the integrator (Jacobson et al., 1999); this role was responsible for bringing together the different software components and had 1 (1%) respondent. Respondents were able to

select more than one option. Having the broad range of roles helped in getting information about the project from different members of the software development team. This information also showed that the respondents who took part in this study had different roles.

Table 4-4 Software Development Team Study Respondents' roles

Roles	Number of respondents	Percentage (%)
Developer	42	57.53
Tester	25	34.25
Scrum master	15	20.55
Business analyst	15	20.55
Team leader	13	17.81
Product owner	11	15.07
Project manager	9	12.33
Line manager	7	9.59
Coach	6	8.22
Client representative	6	8.22
Agile mentor	6	8.22
Technical writer	5	6.85
Others	3	4.11
UI/UX	2	2.74
Trainer	2	2.74
Integrator	1	1.37

On average, each respondent had two roles within a project. The combination of Developer and Scrum Master was found to be the highest overlap. The following table (Table 4-5) presents the overlaps where the role combination was reported by more than 5 respondents.

Table 4-5 Number of roles of each team member

Roles	Numbers
Developers and Testers	7
Developers and Scrum Masters	8
Testers and Scrum Master	5
Product owner and Developer	5
BA and Tester	5

A total of 22 (30%) study respondents had been working in software development for more than 20 years and seven (9.59%) of the respondents had worked in software development for one to three years. Table 4-6 presents more details of the years of experience for the respondents.

Table 4-6 Respondents years of experience

Years of experience	Number of respondents	Percentage (%)
1 to 3 years	7	9.59%
4 to 6 years	11	15.07%
7 to 9 years	8	10.96%
10 to 12 years	7	13.70%
13 to 15 years	6	8.22%
16 to 19 years	9	12.33%
20 years or more	22	30.14%

Information from shows that the respondents for this study had a range of experience. This variation in experiences of the respondents helped to get a wider range of information about the Agile projects. The data collected from respondents with different experience helped to understand whether there was a difference in challenges in relation to their experience, as the respondents could be working for more than one project. This was important because the experience of the respondents working in Agile software development projects could provide details of information about the challenges they faced in Agile software development projects. The following section discusses the approaches and practices used by the respondents on their projects.

4.2.3 Agile approaches used

The Agile approaches used by respondents are summarized in Figure 4-1. In the questionnaire, respondents were able to select more than one approach they used during the project development. Of the 73 respondents in the software development team study the majority (59) of respondents reported that the Agile development method they used was Scrum.

The approaches reported in the other category were:

- At least we use something between Scrum and Kanban
- "Agile" with small a
- Aspects of Agile but not the formal Agile process
- Waterfall
- Clipper

- No clear method used

- Fail- fix

- V-model
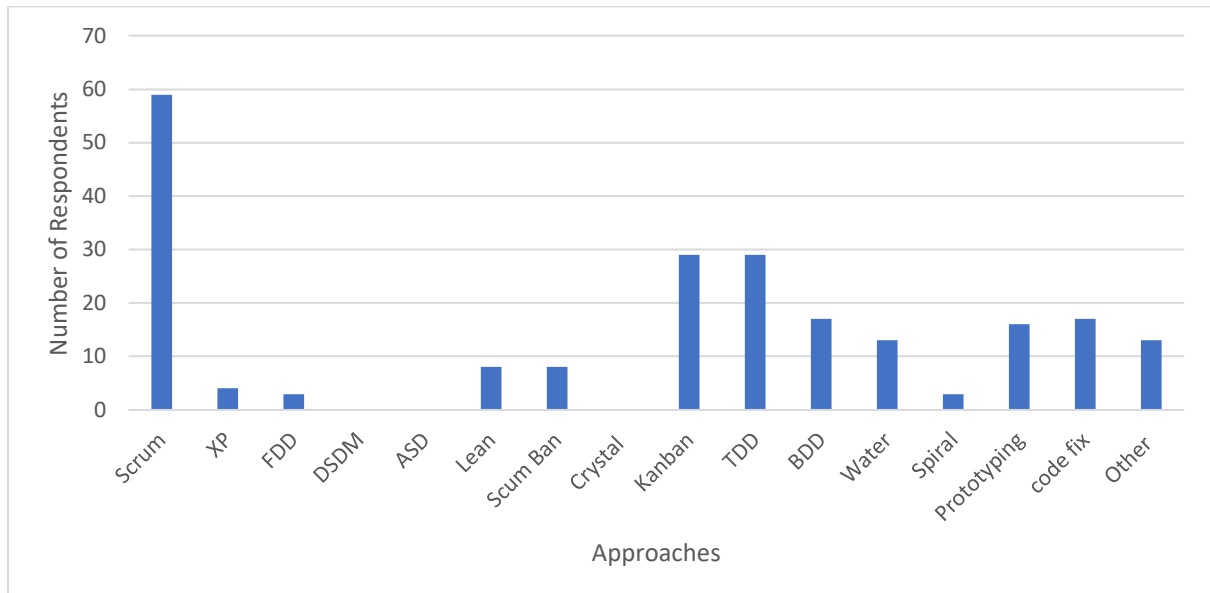
Figure 4-1 Approaches used for the project

The data from the respondents suggested that there were a range of approaches used for the projects. This wide range of respondents provided information about the context of the different approaches used. This research is about the relationships in Agile teams; hence, the wide range of approaches used provided rich information for understanding the relationships in these different contexts. This information also gave an indication of the common approaches that were used in Agile software development.

4.2.4 Practices used for projects

Respondents were asked to select the Agile practices they used in projects and they were able to select multiple practices from a list. Ninety per cent (66) of the respondents reported that they used stand-ups. The least used practices were personas used by 18% (13) and the largest number of practices used in a project was 20. The mean number of practices used in a project was M = 11, with standard deviation 4.719. Figure 4-2 summarizes the practices used in projects.
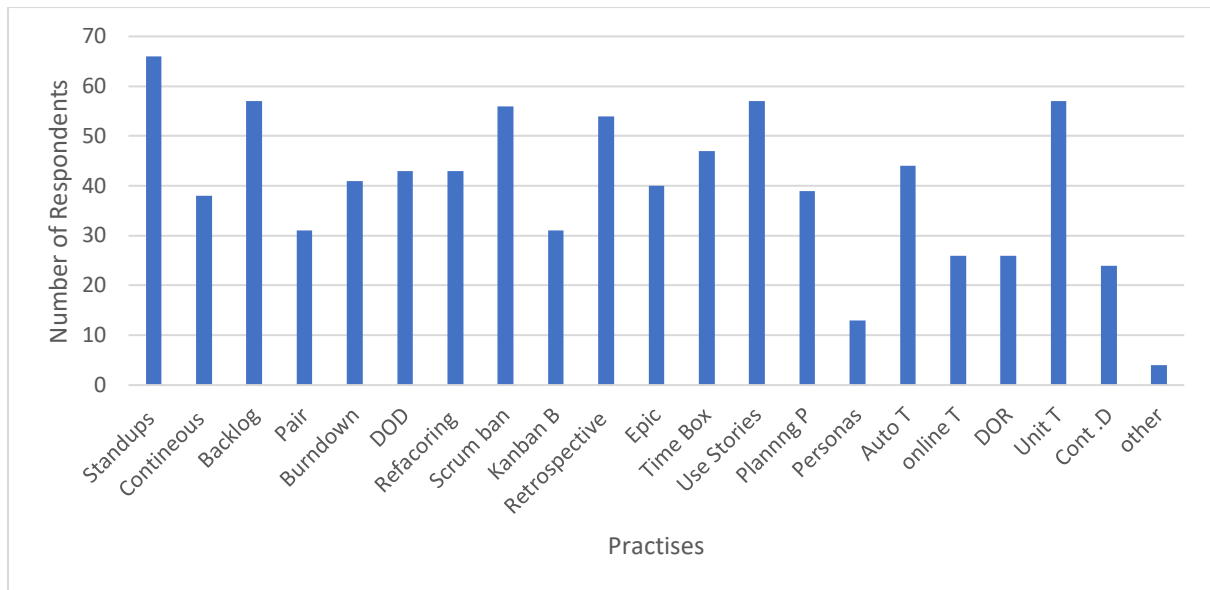
Figure 4-2 Practices used in projects

The data from the respondents suggested that the number of practices used in Agile software development varied with the project. This may be because of the way each organization followed different Agile approaches and the practices used in the projects.

The number of combination of the different Agile practices used in the projects were calculated, and shown in Table 4-7. In total, there were twenty practices listed on the question. Only two respondents reported that they used all twenty practices. One respondent reported that they used only stand-ups.

Table 4-7 Number of practices used in a project

Number of practices	Number of respondents
20	2
19	2
18	1
17	5
16	5
15	9
14	6
13	2
12	3
11	9
10	6
9	3
8	3
7	3
6	5
5	2
4	3
3	1
2	2
1	1

The information received from the respondents suggested that all projects included in this study used at least one Agile practice. The average number of practices used in the projects was eleven. Thirty-five per cent (48%) of the 73 respondents reported that they used eleven or more practices. The information suggested that the number of practices used on a project varied with the project. These data from the respondent suggested that even though all respondents considered themselves as using an Agile approach to develop the software the number of practices used were different. This data will help in understanding if there was a relationship between the number of approaches and the outcome of the projects.

4.3 Team challenges

Information related to the challenges within the team and between the customer and the product owner was collected. These challenges in software development are the difficulties faced during the development of the projects. In some research the challenges were listed

as success and failure factors (Chow & Cao, 2008; Misra et al., 2009; Nerur et al., 2005). The list of challenges for this study were created from literature relating to Agile software development. The analysis of these responses are presented in the following sections.

4.3.1 Challenges within the team

Respondents were asked to select how many times they had experienced challenges within their development team during a typical sprint, from the list provided. This indicated that the members working Agile software development had difficulties in sharing ideas despite the fact that Agile software development promoted collaboration and coordination between the team members.

Table 4-8 presents the respondents' responses. More than half, 37 (51%), of respondents reported that they faced one to three challenges in communicating during each typical sprint. Three respondents reported that they faced challenges more than nine times. Seventeen respondents reported that they did not face any challenges in communication. This suggested that not all projects have challenges during their development. This information was important for this study as it helped to determine the challenges that occurred most often in Agile software development projects. This also gave an indication of the number of challenges faced during the development of the projects.

Interpersonal challenges were reported as occurring one to three times in a sprint by 39 respondents. Twenty respondents reported they did not face any interpersonal challenges during a typical sprint. Nearly half of the respondents (34) reported that they faced difficulties in sharing ideas within the team one to three times in a typical sprint. Difficulties in dividing the work were reported by 30 respondents one to three times. Five respondents reported they faced difficulties in sharing ideas more than nine times during a typical sprint. This indicated that the members working in Agile software development had difficulties in sharing ideas despite that fact that Agile software development promoted collaboration and coordination between the team members.

Table 4-8 Challenges within the team

Difficulties	0 times	1-3 times	4-6 times	7-9 times	More than 9 times
Difficulties in communicating within the team	17	37	13	3	3
Interpersonal challenges within the team	20	39	7	4	2
Difficulties in sharing of ideas within the team	24	34	8	4	3
Problems with distribution of the work within the team	24	30	12	2	5

Information from the respondents on the challenges presented above suggested that there were challenges within the teams during project development. These challenges included communication, interpersonal issues, sharing of ideas and distribution of work. Misra et al. (2009) reported that communication and personal characteristics were important factors for successful Agile projects. However, their study did not examine how often these challenges occurred within a typical sprint. According to the Dickinson and McIntyre model, communication provided the links to all the activities occurring within the team during the work. Information obtained from the respondents suggested that more than 50% of them faced communication challenges during software development. According to the Teamwork model, communication bridged all the activities within the teamwork. When effective communication was not achieved then there was a chance of having other challenges on the projects. Information from the respondents suggested that when there were challenges in communication this challenge not only impacted on passing information between the members but also may impact on other team activities, such as coordination, collaboration, feedback and back up behaviour.

4.3.2 Challenges between the development team and the product owner

To meet the aim of examining the relationships between a team and the product owner, data on the relationship between the team and the product owner was collected. Forty-three (43) respondents reported that they have had one to three disagreements with the

customer about project priorities in a typical sprint. Three respondents reported challenges in communication with the customer as many as seven to nine times during a sprint. **Table 4-9** presents the responses about the challenges faced with the customer.

Table 4-9 Challenges with the customer

Difficulties	0 times	1-3 times	4-6 times	7-9 times
Disagreement with the customer about project priorities	23	34	6	1
Disagreement with customer about project requirements	18	43	4	1
Disagreement with the customer about the timeframe of the project	31	25	7	3
Interpersonal challenges between the team member(s) and the customer	37	26	4	0
Challenge in communicating with the customer	26	30	7	3

The information, above, suggested that there were challenges that existed between the customer and the Agile software development team. More than half of the respondents reported that they had at least one to three challenges about the project priorities, project requirements, timeframe for the project, interpersonal challenges and communication. A study by Hoda et al (2011) reported challenges in communication with the customer because of their availability during a project. Because of such challenges there were problems in clarifying the requirements, securing feedback and a resulting loss of business productivity. Chow & Cao (2008) also reported that defining a project scope as one of the challenges in Agile software development projects. For this study, it was important to gather information on the number of challenges in a typical sprint to understand the impact on the relationships between the software development team and the customer. This information helped in determining the relationship between such challenges and the project outcomes, such as completion of project on time and within the budget.

The first research question **(RQ1)** for this study is:

The literature suggests that there are challenges between the software development teams and the customer representative in Agile software development projects. To what extent do these challenges exist?

Quantitative data obtained from the software development team study confirmed that challenges existed, as suggested by the literature (Chow & Cao, 2008; Hoda et al., 2011) Some examples from the literature were challenges in the processes, people and the organization. Similarly, in this study more than half 37 (51%) of the respondents reported that one to three challenges existed about the project priorities, project requirements, the timeframe of the project, interpersonal challenges and communication. Sections 4.3.1 and 4.3.2 also confirmed that there were some projects where the frequency of challenges were more than nine times during a typical sprint, which attempted to address Research Question 1 **(RQ1)** after the indication from the results that challenges existed in Agile software development. The second aim of this research was to identify what kind of challenges existed in Agile software development. Hence, the second research question for this research is:

RQ2: *What types of challenges are present in Agile software development projects?*

Sections 4.3.1 and 4.3.2 presented above, with the data received from the respondents, suggested that common challenges in Agile software development were project priorities, project requirements, the timeframe of the project, interpersonal challenges and communication. A previous study undertaken by Chow & Cao (2008) and Hoda et al. (2011) mentioned challenges, such as availability, process and people. They also mentioned that these challenges impacted on the project's success. This study also confirmed that such challenges existed in Agile software development. In the following section the impact of these challenges on the projects is presented and discussed.

4.4   Impact of challenges on projects

Information from section 4.3 confirmed that challenges do exist within the team and between the customer and the Agile software development team. To determine the linear dependencies between pairs of study variables Pearson's Product Moment Correlation Coefficient was calculated. To quantify the strength of a relationship the correlation coefficient (r) is calculated and can have a value of between +1 and -1. Correlation

coefficient values greater than zero indicate there is a positive association between the two variables (Taylor, 1990). The closer to +1 or -1 r is, indicates the strength of relationship is (Jeong et al., 2001).

The correlation coefficient at a low p-value (such as 0.01) is taken as evidence to identify the weak, moderate and strong correlation. (Page 90-91). r value was calculated to identify the parson correlation coefficient.

$0<r/r/<.3$ weak correlation

$.3</r/<.7$ moderate correlation

$/r/>0.7$ Strong correlation

The correlation coefficient was calculated between:

- The project completion time and difficulties in communication within the team
- Disagreement with the customer about the project priorities
- Disagreement with the customer about the timeframe of the project
- Interpersonal challenges between the team member(s) and customer.

The correlation coefficients between each of these variables are presented in Table 4-10.

Table 4-10 Correlation coefficients between availability and timeframe

	Difficulties in communicating within the team	Disagreement between the team member(s) and customer regarding the project priorities	Disagreement between the team member(s) and customer regarding the project requirements	Disagreement between the team member(s) and customer regarding the timeframe of the project	Interpersonal challenges between the team member(s) and customer
Project completed on time Pearson Correlation Sig (2-tailed)	.572** .000	.329** .004	.324** .005	.394** .001	.394** .001
N	73	73	73	73	73

The correlation coefficient is used to test for linear relationships between the variables. The correlation between project completion time and difficulties in communicating within the team returned r = 0. 572. This indicated than an increase in difficulty in communicating within the team was likely to mean an increase in the project completion time. If there were fewer difficulties in communication within the team it tended to decrease the project completion time. Pikkarainen et al. (2008) noted that the introduction of Agile practices improved the communication. Their study also found that challenges existed in communication in Agile practices. For example, open space practices in the office resulted in communication challenges. These challenges were found to have an impact on the projects. Pikkarainen et al. (2008) also reported that the development team criticized the team members required to document the communication occurring in an office. Such types of communication were also found to have an impact on the project requirements and planning. However, they did not quantify the impact of those challenges on the project (Pikkarainen et al., 2008). This information helped to determine the impact of such challenges on the relationships between the team and the product owner. and their impact on the project's outcome.

The data suggested that there was a positive and linear relationship between the project completion time and disagreements with the customer about project priorities (r=329). This suggested that if there were an increase in disagreements with the product owner about priorities then this was likely to increase the project completion time. This was important to this study as it showed the impacts on the relationship between the disagreements between project priorities and project outcomes.

The data also suggested that there was a positive and linear relationship between the project completion time and disagreements with the customer about the project's requirements (r=0.324). This meant that if there were a disagreement with the customer about project requirements in a typical sprint it tended to increase the project completion time.

The linear dependencies between the project completion time and interpersonal challenges between the team member(s) and the customer were positive (r=.394). This suggested that when the interpersonal challenges between the software development team members increased, the project completion time may also increase. Balijepally et al. (2006) found that

interpersonal challenges have a negative effect of team performance (Balijepally, Mahapatra, & Nerur, 2006). Interpersonal challenges were found to have a negative impact on effective communication within the team (Leau et al., 2012b). They also mentioned in their study that the development team should have good interpersonal relationships for effective team work. This suggested that the interpersonal challenges did have impact on communication and effects on the performance of the team. This information was important for the current study in understanding the impact of such challenge on the project success.

A correlation coefficient was calculated between the product owner availability and disagreement between the development team members and the customer about the timeframe of the project. The correlation coefficient between these variables are presented on **Table 4-11**

Table 4-11 the correlation coefficient between the availability and timeframe

Variable	Disagreement with the customer regarding the timeframe of the project
Product owner availability Pearson Correlation	-.348[**]
Sig (2-tailed)	.003
N	73

Hoda et al. (2011) found that customer involvement in Agile software development in projects had some challenges, such as prioritizing requirements due to whether they were available or not (Hoda et al., 2011). However, they did not mention about interpersonal challenges or difficulties with the requirements. This information was useful for this study in understanding the impact of these challenges on the relationship between the software development team and the product owner.

The data suggested that there was a negative and linear relationship between the product owner availability and disagreement with the customer about the timeframe of the project.

(r=-0.348). This suggested that if there were an increase in the availability of the project owner there tended to be a decrease in disagreements between the development team members and the customer about the timeframe of the project. Hochmuller (2011) reported that the availability of the product owner (customer) throughout the development process was vital for maintaining communication and reducing the risk of projects in which errors and misguided development were only detected in the latter stages of the projects (Hochmüller, 2011). This suggested that there was an impact of the availability of the product owner on communication and the completion time of the project. This was important for this current study as it showed that the availability of the product owner impacted on the relationship between the team and the product owner.

Agile software development accommodates changes in requirements during the project development process (Abrahamsson, Salo, Ronkainen, & Warsta, 2017). This is considered as one of the benefits of agile software development. However, change in requirements from the customer is also considered one of the main reason for increase of projects completion time and budget (Butt & Jamal, 2017). The completion time for a project is often referred to a time estimated by the team. It is the time to complete the project in the current conditions. When there is a change in requirements the project completion time is adjusted accordingly. To accommodate this change, in Agile software development, the work is divided in units. The time required to complete each unit of work is estimated based on the experience of the team. Once the velocity (work completed per sprint) of the team is known its helps the team to estimate the project completion time. Since Agile is not an estimation methodology itself, the estimate of time is there to guide the team in completing the project(Ahimbisibwe, Cavana, & Daellenbach, 2015; Bai, Li, Pei, Li, & Ye, 2018).

In this research, the project completion time refers to the time, the team members and product owner believe the project would be completed in.

There could be confounding factors that impact project success including: Technical Competency, Training and Learning, decision time, tools and technique applied in the development process.

The challenges in Agile software development could be the symptoms of these factors. In larger projects there is more functionality and therefore more opportunity for disagreements. In these projects, there can be more disagreement in activities such as knowledge sharing and improvement, customer collaboration, release planning and architecture, inter-team coordination (Jensen, 2017).

The statistical analysis presented in section 4.4 addresses the fourth research question:

RQ4 *Does the nature of the relationship between the customer and the development team impact on the project outcome?*

The challenges in communication within the team, challenges between the product owner and the development team about priorities, requirements, the timeline of the projects and interpersonal challenges, were found to have significant impacts on the project completion time. Some of these challenges, such as defining the project scope, project requirements, lack of team work and the lack of customer relationships, have been reported in previous research (Chow & Cao, 2008; Misra et al., 2009). This research confirmed that these challenges impacted on project completion. This information showed the association between the relationships and the project outcome.

To understand the relationships between variables, such as difficulties in communication, sharing of ideas, distribution of work with the project time and the project budget, the Pearson's Product Moment Correlation Coefficient was calculated between these variables. Those correlation coefficients for the variables with non-significant (n.s) linear relationship are presented in Table 4-12. The non-significant linear relationship means that the parson correlation calculated provide little or no evidence of the relationship between these variables.

Table 4-12 Non-significant correlations between different variables

Variables	Variables	Pearson Correlation Sig. (2-tailed) and N= 73
Difficulties in communicating within the team	Project/Product time	.055 n.s
	Project budget	-.190 n.s
	Project contract	.086 n.s
Difficulties in sharing of ideas within the team	Project/Product time	.213 n.s
	Project budget	-.208 n.s
	Project contract	.152 n.s
Interpersonal challenges within the team	Project/Product time	.089 n.s
	Project budget	.065 n.s
	Project contract	.110 n.s
Problems with distribution of the work within the team	Project/Product time	.183 n.s
	Project contract	.016
	Project priorities	.253 n.s

Disagreement in project priorities with the product owner	Project budget	.094 n.s
	Challenges in distribution of work	.253 n.s
Disagreement with customer regarding project requirements	Project budget	.079 n.s
	Project contract	.239 n.s
Product owner from same organization	Project budget	-.136 n.s
	Project contract	-.007
	Difficulties in communicating within the team	-.059 n.s
	Interpersonal challenges within the team	-.269 n.s
	Problems with distribution of the work within the team	-.184 n.s
Product owner availability	Project timeframe	-.099n.s
	Project contract	-.171 n.s
	Difficulties in sharing of ideas within the team	-.199 n.s
	Interpersonal challenges within the team	-.092 n.s
	Problems with distribution of the work within the team	-.181 n.s
	Interpersonal challenges between the team member(s) and customer	-.146 n.s

In a previous study on Agile software development, interpersonal challenges were found to have an impact on a project's outcome (Liu et al., 2011). However, this study could not confirm a significant relationship between interpersonal challenges and the project outcome. This was interesting, because it was a common assumption that interpersonal challenges could impact on the relationships within the team and that these could impact on both the coordination and the project. This could mean that these challenges did impact the team environment but, because of the Agile process and practices, the development process moved smoothly so there was no impact on the timeline of the project.

The results also showed that there was a non-significant relationship between the challenges in communication and the timeline of the project. This was interesting, as the relationship was expected to be significant between the challenges in communication within the team and the timeline of the projects. However, this could not confirm that when there was a problem in communication within the team it would have an impact the project's success.

Product owner availability was found to have a non-significant relationship on the timeline of the projects. The result was not expected. The availability of the product owner was found to have an impact on the communication between the team and the product owner; hence, it could impact the project outcome. However, this result showed that the product availability may not impact on the timeline of the project.

4.5   Team member views on the type of relationship
A questionnaire was included in this study where respondents were asked to select the kind of relationships that they thought would work best for successful projects.

Table 4-13 presents the options listed from this study and the responses of the respondents in percentages. This list was created based on literature on Agile software development to understand the types of relationships that respondents thought would be best for project development. Respondents were able to select more than one option. The mean number of options selected by a participant was M=5 (S.D 1.74). Ninety-seven per cent (71) of the respondents selected that the development team and product owner should collaborate to meet the project goal. Only 14% (10) of the respondents indicated their belief that the team members were to be guided by the contracts.

Table 4-13 Relationship on the projects

Option listed	%	Count
The team members in a software development collaborate with the customer/product owner to meet the project's goal.	97.26%	71
There is complete transparency in the interactions between team members and the customer/product owner during the project development process.	90.41%	66
The motivation to succeed drives team members to complete software development projects.	67.12%	51
The team members have trust in the decisions that the customer/product owner may make.	65.75%	50
The customer/product owner trusts the team to complete the project according to the agreed contract.	63.01%	46
The main goal of both the team and the customer is to complete the project within the agreed timeframe.	34.25%	25
The team members in a software development team are guided by the instruction given by the customer/product owner with the focus on finishing the project on time.	31.51%	23
The main goal of both the team and the customer is to finish the project within the agreed budget.	15.07%	11
Team members and the customer are guided by the project contract.	13.70%	10

Collaboration was reported as most the important factor for project success by 97% (71) of the respondents. More than 50% (46) of respondents reported that transparency, motivation and trust were important for the project's success. The factors identified in this study were not found to have been reported as significant factors for project success from previous studies. For example, Chow & Cao (2008) investigated the literature relating to Agile development and identified 48 critical success factors among which motivation was one of them. After the identification of the factors they deemed critical, Chow & Cao (2008) conducted a web-based survey, gathering feedback from 109 Agile projects from 25

different countries. The findings from this survey helped them further refine their list of critical success factors, resulting in a more defined list, as shown below:

- Correct delivery strategy.
- A proper practice of Agile software engineering techniques.
- A high calibre team.

Misra et al. (2009) identified the following success factors for Agile software development after surveying 150 people involved in software development. The factors they identified were:

- Customer centric issues
- Decision time
- Corporate culture
- Control
- Personal characteristics
- Social culture
- Training and learning

The success factors reported by the respondents from this study can be visualized, as shown in Figure 4-3
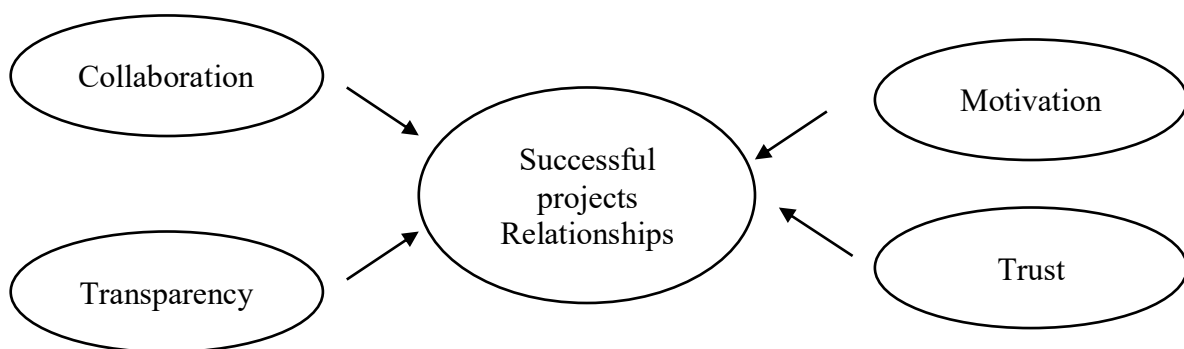
Figure 4-3 Development team views on type of relationship for Successful projects

The data collection process, and a description of the participants and companies were discussed above. We also presented the results from the questionnaire data. The questionnaire results confirmed that even though all the organizations referred to

themselves as following an Agile process, the number of Agile practices was found to be vary from one to twenty. Respondents reported that there were some challenges within the team as well as between the team and the customer. Some of the challenges, such as communication, projects requirements, were found to have an impact on project completion. This suggested that such challenges may impact on the relationship between the team and the product owner. Such challenges could also impact on the team collaboration and coordination that could also impact the teamwork in Agile software development projects.

To understand the relationship in greater detail interviews were conducted with a subset of the respondents. Qualitative data were gathered from semi-structured interviews. The findings from the interviews are presented in the following section along with the results and a discussion.

4.6   Interviews

Semi-structured interviews were conducted with twenty-two respondents, who had completed the online questionnaire, to explore the information in detail. In this research, the term saturation was used to indicate that if recent interviews provide the same information given by earlier respondents and there is no need to interview more people (Marshall, 1996). There were no new themes discovered from the participants 21 and 22 with compared to previous interviews. Hence with the 22 participants the saturation point was reached.

The interview selection process is described in the Methods chapter, section 3.9.2.

Table 4-14 shows the respondents' job roles. The majority (9, 41%) of the respondents were software developers.

Table 4-14 Respondents' roles

Job role	Number of respondents	Percentage
Software developer	9	41%
Software engineer	4	18%
Test engineer	3	23%
Business analyst	3	9%
Solution architect	2	9%
Project manager	1	5%
Team lead	3	23%
Scrum master	1	5%

In Agile software development, multiple roles people are involved in completing projects. Information from

Table 4-14 shows that the respondents who took part in this study had a number of different employment roles. This helped in getting information from a wider range of people with Agile software development teams. Agile teams consisted of people with different roles and the wide range of roles of the respondents provided information from different perspectives. This was important for this study as this study was about understanding the relationships in Agile software development. During this relationship there was involvement from different members. The information from these different members helped to understand the relationships in detail.

Based on their answer to Q 13 (Appendix B) respondents were categorised by the type of organization they work for. The categories were: In-house development, Product development and Bespoke/Contract.

Table 4-15 shows the distribution of respondents across these categories. A sample size of The respondents to interview were selected from these categories. A sample of 41%, 24% and 27% was selected from each categories for in-depth interview. (See Appendix C for interview questions). The distribution of respondents suggest that most organisation are developing product or in-house focused. These organisation are therefore more strongly represented in study from the interview questions.

Table 4-15 Respondent and project type

Project type	Number of respondents to questionnaire	Number of interviews	Percentage interviewed
In-house	24	10	41%
Product	38	9	24%
Bespoke	11	3	27%

Response were analysed using a thematic analysis method (Braun & Clarke, 2006). The process of identifying the themes is discussed in the next section.

4.7   Themes

Themes are the patterns across the entire set of data that can be used to answer research questions and address hypotheses (Braun & Clarke, 2006). To determine themes, data from the interviews were transcribed and stored in Microsoft Word. The responses from the respondents were read through to give an understanding of the information. Possible codes were identified through reading the responses. After the initial coding of the data, a complete list of codes was created and restructured to group similar codes. The codes were revisited and new themes were identified. These similar codes were then compared, reviewed and combined with the initial codes to form comprehensive themes.

In quantitative research, researchers can apply statistical methods to establish validation (Strauss & Corbin, 1990; Zelkowitz & Wallace, 1998). However, in qualitative research, methodological strategies are designed to ensure the 'trustworthiness' of the findings (Noble & Smith, 2015). The researcher has adopted different validity options that were considered to be appropriate using terms, such as quality, rigour and trustworthiness (Davies & Dodd, 2002; Seale, 1999; Stenbacka, 2001). To maintain the validity of the data a researcher needed to remain true to the respondents by ensuring the information given by them was well presented. For this reason, a number of strategies were used to ensure rigour in this study. These strategies included:

- Interviews were recorded and transcribed verbatim.
- Audio recordings were re-listened to check emerging themes and remain true to the respondents and the coding process used for this study is described.

- In order to ensure the correctness of the data, transcriptions were rechecked using a professional transcriber.

After the codes were consolidated, a list of themes were identified and are discussed in the following section.

4.8   Major themes identified

After the coding process eight major themes were identified. These themes are presented in Table 4-16 and explained in the following sub-sections

Table 4-16 Number of respondents' comments for each factor

Factors	Definition	Number of respondents' comments on each theme.
Communication	Communication is the process of transferring information between individuals who are involved in a project.	22
Feedback	Feedback refers to providing information about the performance and acceptance of the suggestion given on the project's work.	13
Team support	Team support involves being available and willing to assist other members in the team. Development team members, including the product owner, worked together to solve problems with the scope of the projects.	15
Organization culture	A self-managing team with a culture of shared responsibility. Successful Agile development teams require product ownership both from within the team and within the organization that understands and supports the iterative nature of the Agile process.	11
Team engagement	Team member's activities during the projects. The activities of each of the team members are aligned in a way such that the targets associated with a project goal are achieved.	10
Product owner engagement	Product owner role on the different activities of the project from the starting of the project to the deployment.	14
Confidence	Belief in the team to perform the given task from the product owner.	8
Team motivation	Motivation in the context of this study refer to the development team members desire to complete the task	9

4.8.1 Communication

Communication refers to the sending and receiving of information (Cockburn & Highsmith, 2001b). Communication can be achieved using different pathways such as face-to-face and email (Biehl et al., 2007). Respondents for this study reported that there were different communication paths used in the projects. The communication paths reported by the respondents in their projects were: communication within the team, communication between the customer and the product owner, communication between the team and the product owner and communication between the team and management.

Communication between the customer and the team mainly occurred during a project to clarify project requirements or to access additional information about the requirements. For this, there were discussions between the team members to gain more understanding of the requirements during the initial stage of the projects. The findings from this study showed that respondent-identified communication was one of the major factors during project development. The communication involved talk about the project and improvements that can be made to the project. Development teams were involved in reviewing each other's code, where communication within the team in such activities, helped the team member be aware of what the other team members were doing on the projects.

One respondent said:

> *"Ah, the product owner has been very good in listening to the concerns, um, and very good at interacting with the team members."* Respondent 3

And another said:

> *"There is also we've got three scrum teams, um, is knowing what the other scrum team are doing and trying not to get the crossover of working in each other's code we're very careful about that um, so the biggest way to help each other I think is the communication."* Respondent 1

And another said:

> *"The lead tester is not just responsible for the testing but he is also responsible for the communications with the business, and the client and management."*
> Respondent 9

The location of the product owner was important for communication. Ten respondents in this study reported that most of the product owners were physically located besides the team and were easily accessible. A number of communication methods were described by the respondents. These included: email, telephone and face-to-face communication. A study carried out by Pikkarainen et al.(2008) found that there was a conflict in communication when the stakeholder used written documents as communication while the development team communicated face-to-face (Pikkarainen et al., 2008). However, seven respondents in this study reported that face-to-face communication was a common practice for communication in the projects. (Henttonen & Blomqvist, 2005) found in their study that face-to-face communication presented the best way to maintain good productivity in software development companies. Respondents from this study also reported that they used direct communication with the product owner.

One of the respondent reported:

> "*We could go directly to the product owner and basically ask what we wanted to ask*." Respondent 15

And other reported:

> "*The product owner just sits with us so we talk to him directly.*" Respondent 12

Another respondent said:

> "*The product owner is in the same office although sometimes he does travel to other places and in that case we communicate by email or phone but usually he is in the same office.*" Respondent 17

All respondents reported that communication was an important factor in projects. A study by (Anderson, 2003), on Agile methods, found that improvement in communication within the Agile teams, with the customer and the business units helped in the faster development of a project. Not all projects achieved effective communication in this study. Respondents for this study reported that they had some difficulties in conveying the information about a different approach to be taken during the projects. Some of the difficulties were due to language differences, location of the team members and the availability of the person responsible to provide information. Miscommunication was also reported as one of the issues that caused the difficulties in projects (Hossain, Babar, & Paik, 2009). One respondent in this study stated that a language barrier was seen as one of the difficulties in the team.

This issue caused difficulties when explaining how the system should work and in clarifying the requirements between the team and the customer. The miscommunication when explaining may be caused because of the language barrier or then were problems in explaining even though there was a common language used in communication.

A respondent reported:

> *"Sometimes there is a miscommunication. The product owner involved in the project has an 'I don't care attitude'. Sometimes when he wants me to do testing but does not tell me the details whether he wants me to do the full regression pack or only a small part of the function. Because of this we have been delayed."* Respondent 1

And another reported:

> *"You will understand, it's a lot of the language barrier, a lot of the cultural barrier, so explaining to people how insurance works in New Zealand was difficult*. Respondent 19

One respondent talked about having a different product owner for different projects. He stated that the product owner in his previous project was not available to communicate with or to clarify any issues. This was because of the product owner was involved in many projects and did not have the necessary time to put into each one. However, the situation had changed in the current project and the current project owner was available. This change resulted in an improved communication between the product owner and the development team. This suggested that the availability of the product owner was important for effective communication. This also suggested that communication between the team and the product owner can be improved when the product owner was available to the team during project development This result also suggested that when the product owner was available there was better communication than when the product owner was not available, as in previous projects.

A respondent reported:

> *"Previously the product owner was not available for any clarification. But in the current project the product owner is available for the communication"* Respondent 17

During project development, communication was mainly between the people involved in the project. Communication was established in different ways. A daily stand-up was found to be the most widely used way for communication in the team. Three respondents reported a language barrier as one of the challenges in communication. With respect to communication with the product owner, the communication method did have some challenges, which led to miscommunications. Six respondents reported on product owner availability and the level of involvement led to challenges in communication. Teams had to wait for clarification on some of the requirements because of the non-availability of the product owner. Communication was an important success factor in Agile software development. Communication helped with collaborative practices and processes during software development. A study by Pikkarainen et al. (2008) identified situations where multiple stakeholders were involved that may impact on effective communication and the effect on the communication impact on the projects (Pikkarainen et al., 2008). It has been claimed that differences in understanding the projects among the people involved in software development may lead to project failure (Boehm & Turner, 2003b). Researchers have mentioned that most of these problems may be caused because of a lack of communication between the people involved (Coram & Bohner, 2005; Svensson & Höst, 2005).

In the Dickinson and McIntyre Teamwork Model (Dickinson & McIntyre, 1997) communication was the one of the components that linked all the other teamwork process together. It was reported in this study that direct communication with the product owner was the most used mechanism for communication. Other mechanisms used for the communication included stand-ups, phone calls and email. In Agile software development, team members, including product owners, should communicate with each other; however, because of time zone differences, language barriers and availability of the product owner communication can be difficult. Communication was improved when the product owner was available and present at the same place (Mishra, Mishra, & Ostrovska, 2012). This suggested that the product owner availability impacts on effective communication.

Figure 4-4 shows the coding process for the communication theme.

Discussion in the team is often quite clear among the members (R15)

The product owner and the team leads communicate directly with customers (R10)

Product owner is very good at interacting with the team members (R3)

We could go directly to the product owner and basically ask what we want to ask (R7)

The lead tester is not just responsible for the testing but he is also responsible for the communications with the business, the client and management (R9)

We also use social media for communicating with customers (R7)

He turns up to our daily stand-ups. We actually let him speak (R1)

The product owner just sits with us so we talk to him directly (R12)

International communication; international participation has made it difficult (R3)

Previously the product owner was not available for any clarification (R17)

Language barrier issue was observed with vendors (R19)

There is a little bit of interaction; there are some dominant team members who offer their good advice frequently (R3)

Interaction pathways

Communication medium

Difficulties in communication

Communication

Figure 4-4 Coding process for the communication theme

4.8.2   Feedback

Feedback involves giving and receiving information between team members about their performance as well on the progress of the project.

Feedback related to the project can be obtained from different sources, such as from within the team, from the customer, from the management or from other business people who are involved in the projects. Respondents for this study reported that they normally received feedback from the customer. Respondents also reported that when a team member received positive feedback team performance was found to improve. During project development, the Agile team, the product owner, the customer and management were all involved in giving and receiving feedback. It was reported that the development team felt good when they received positive feedback from the customer. Feedback during the project improving the team performance was reported on in the study by Guzzo & Dickson (1996).

One of the respondent for this said:

> "*I quite like it when the customer turns around to us and says good job guys and that's perfect, or that new form and process you delivered is, you know, the users love it, 'cos we get pretty good feedback from the customer. In a lot of cases I think customers generally give more negative feedback than positive feedback.*"
> Respondent 20

And another said:

> "*Customers ring up and they say look you have done this and it's not what I wanted or it's not how I imagine at all or not how it should work or I changed my mind.*"
> *Respondent 11*

Team members, including the product owner, used different media to provide feedback, these included: the telephone, daily stand-ups and email. Respondents reported that there were difficulties during the feedback process. The difficulties were when the product owner initially accepted the feedback but then changed their mind. Because of this, there was some rework to be done on the project.

One respondent said.

"I think he listens most of the time but there are instances where he says ok that makes sense and then, at a later stage, he says, 'Oh actually no, I thought about that and that doesn't make sense.' " Respondent R17

Another respondent stated:

"Product owner say look you have done this and it's not what I wanted or it's not how I imagine at all or not how it should work or I changed my mind." Respondent 11

Despite respondents reporting that the product owner was open to listening to the feedback and suggestions on the project from the development teams, it was mentioned that the product owner looked at the experience of the team member when accepting suggestion given on the project.

One respondent said:

"For us for the new development, like for sometimes for new people or not experienced people if we share with him he doesn't believe us, but if senior people more than senior people like four or five years after we spend there and if they said that to them then he will instantly believe but for us it is very hard to say to the things there." Respondent 2

In summary, the team members were encouraged to hear feedback from the customer and the management. Dingsøyr et al. (2012) reported that when the customers were actively involved in the development process, giving feedback can lead to more satisfying outcomes Generally, team members were involved in making suggestions and offering feedback about the project to the product owner. In most cases, the product owner accepted the team's suggestions. "Agile software development: It's about feedback and change." (Williams & Cockburn, 2003) reported the importance of the feedback in adapting Agile software development.

The product owner in Agile software development was involved in prioritization. In this study, there were instances where the product owner changed their mind later. One concern that some of the development teams had was around the product owner's acceptance of suggestions offered by development team members. In some instances respondents felt that a product owners appeared to be more accepting of suggestions when

they came from senior team members rather than more junior members. This perception meant that some of the more junior staff were reluctant to suggest any changes for a particular product or project. This information suggested that, in some projects, the experience of the team member working on a project may have an impact on the relationship with the product owner.

In the Dickinson and McIntyre Teamwork Model (Dickinson & McIntyre, 1997) feedback was one of the intermediate processes for effective team work. Feedback referred to providing information about performance. Seeking feedback referred to requesting and receiving guidance, and accepting positive and negative information about performance. In this study, three respondents reported that the product owner was involved in giving positive feedback on the performance. This suggested that there were some instance in projects where the product owner was involved in giving feedback on the performance to the team. However, this was not the case in all the projects. This may be because the product owner was not involved in monitoring the team members' performance to give the feedback and this could mean that the product owners needed to be aware of the team's performance through monitoring. As mentioned in the Teamwork model, when a member was involved in monitoring another's performance then there will be more chances for giving and receiving feedback about effective teamwork. In the context of this study, feedback differs from communication, in that communication was the process of transferring any information related to the projects whereas feedback was the process of giving or seeking information about the performance of the members. Feedback was one of the constructs of the Teamwork model. This construct was important for effective teamwork when members were working in a group. shows coding process for the feedback theme.

Product owner gives feedback on the team performance (R17)

I quite like it when the customer turns around to us and says good job guys and that's perfect, or that the new form and process you delivered is, you know, the users love it, 'cos we get pretty good feedback from the customer. In a lot of cases I think customers generally give more negative feedback than positive feedback." Respondent 20

They ring up and they say look you have done this and it's not what I wanted or it's not how I imagine at all or not how it should work or I changed my mind (R11)

We have the meeting with the manager, you know, every month and he is saying like, ah, he will always point my, like ah, limitations, like a good things and like ah, which area I have had to develop on (R2)

I quite like it when the customer turns around to us and says good job guys and that's perfect (R20)

They ring up and they say look you have done this and this is not what I wanted (R11)

I appreciate that email you sent me, what do you actually mean, what you need to know (R19)

I think what we found at the council is quite often the product owners don't have a great understanding of the whole process to comment on the projects (R14)

I think he listens most of the time but there are instances where he says ok that makes sense and then at a later stage he says oh actually no I thought about that and that doesn't make sense (R17)

For us for the new development, like, for sometimes for new people or not experienced people, if we share with him he doesn't believe to us, but if senior people more than senior people like four or five years after we spend there and if they said that to them then he will instantly believe but for us it is very hard to say to the things there." (R2)

Involvement in the feedback

Comments on performance

Medium used in commenting

Difficulties in giving feedback

Acceptance of suggestions

Feedback

Figure 4-5 Coding process for the feedback theme

101

4.8.3   Team support

Agile software development projects are developed by cross-functional and self-managing teams (Moe et al., 2008). Team members involved in projects provide support for each other during the development process. In the terms of this study, team support referred to the willingness of team members to support each other when completing a task. Respondents for this study reported that team support in the project was demonstrated through shared leadership, backup behaviour, and a helping attitude towards the other team members. This study suggested that team support was demonstrated by a team member explaining to other team members what was needed from the team and then willingly listening to the concerns of the team.

One respondent reported:

> "I think that most people come to work every day with the hope of doing a good job and generally if someone has knowledge that someone else needs they are delighted to share that." Respondent 9

One respondent reported that there was a great deal of support and direction given to the team members during the project he was involved in. In that case, there were team members involved in explaining to other team members how work in the project was carried out. However, not all team members have a leadership role while working on the project.

One respondent reported:

> "If someone is senior he mostly spends time helping others. If someone's a junior or a graduate, he will mostly ask for help." Respondent 12

Another reported:

> "Team members are very good and so often as well as doing all the work, also helping all the others out too and get them to up to come up top speed so." Respondent 21

Backup behaviour was reported within team support where a development team was involved in giving assistance, seeking assistance and completing the task as a team. Agile software development relied on self-organizing teams. Development team members in this study reported some evidence of backup behaviour in the form of assistance and coaching

between team members. Development team members said that they were happy to assist other members of their teams if there was a problem on the project they were working on.

One of the respondent said:

> *"Most of the time like, I am constantly helping other people, people are helping me if I have a question, so they are always approachable, and also we also have shared stand-up so that's also a point to share any ideas to solve this problem."* Respondent 10

Generally, respondents reported a culture of helping each other. It was mentioned in the Agile manifesto that team members in Agile teams were expected to help other members in the team to further their knowledge by learning from those who tried to help (Fowler & Highsmith, 2001). However, some respondents noted that there were some people who did not like to ask for help as well as those who were unwilling to accept help from their team even though they were having difficulties. The difficulties may be either that the person did not like to accept help or was not aware about the help he can get while working in Agile teams. A study by Conboy et al.(2010) reported that team members should be aware that they can get help in Agile teams to complete tasks as well as improve their skills (Conboy et al., 2010). However, there was nothing mentioned about what could be done to encourage team members to seek for help if they were having any difficulties.

One respondent said:

> *"I try to help people as often as possible but so there are people who just don't ask for help even when they need it, so sometimes that's difficult."* Respondent 13

Another said:

> *"Helping others, If someone one junior mostly asks for help if it's [if ''ts] if someone needs help like a blocking issue he might tell them during stand-up."* Respondent 11

Within some software developments teams, team support worked well with relatively few issues to deal with. In most of the teams the senior members were found to be helping the junior members.

Collaboration was reported within the teams. In terms of this study collaboration referred to people who were involved in Agile software development working together to complete the projects.

One respondent reported that the team members discussed with each other about making decisions on the project. Team members also go through code of other members to know the progress made.

One respondent said:

> "We have a lots of chats in the office because we work in the same space. We have online channels where we chat. We also have a code reviews so as we work through features we will read each other codes to see how we going." Respondent 6

Team members were reported as being involved in the project from the beginning of the project, along with the product owner. Four from twenty-two respondents reported that the development team was included from the beginning so as to have more ideas on how the job can be completed.

One of respondent said:

> "We've got such a small team we would encourage the development team to sit in on meetings because the ones who know how to do the job are the developers and they get the inside information on the domain knowledge." Respondent 20

The team work in Agile software development was dependent on the type of work the team was doing. If the team was working on the same features then the team will need more coordination between team members. However, one respondent reported that the team made the decision about whether to work on the same features or work on different features. Team members discussed with each other during the requirements refinement, process, stand-ups etc. as a part of Agile process.

One respondent said:

> "Depending on the nature of the work sometimes all the developers working on the same things and sometimes we all working on the different features, [it] just depends on what makes more sense." Respondent 10

Respondents reported that team members and the product owner were involved in providing feedback. One respondent also reported a case where a team member was thinking from the solution point of view, rather than coordination, with other team members. When the person thought from the solution point of view while other members of the team were working to understand the problem there were difficulties in coordinating

the team as a whole. A study by Begel & Nagappan (2007) reported that when using an Agile software development approach there was an improvement in coordination (Begel & Nagappan, 2007a). However, in this study, one respondent mentioned that there were problems in coordination because of the nature of that person. Dickinson and McIntyre (1997) argued that, when all the team members' worked together it provided the platform for team coordination. The response from this respondent suggested that they had a problem in coordinating because of the team member's view on the project was different.

One respondent said:

> "*We had a problem with one person in a team. I think the problem was that he was very technical person and he was always thinking in terms of the solution not in terms of [the] problem. Team members were finding hard to coordinate with that person.*" Respondent 10

Agile software development project relies on self–managing teams Self-managing in the context of software development referred to teams who were responsible for managing and monitoring their own performance when completing the task (Moe, Dingsøyr, & Dybå, 2009).

One of the respondents for this current study said:

> "*I found that as the scrum master one of the challenging most challenging aspect is to try and help the team to self-manage effectively. Um, if you know, they would tend to get a very good strong sort of sense of team work and collaboration, particularly around the planning and sort of be prepared to help each other out but, as I say, the teams got really, sort of, got into that we're all going to do this together.*" Respondent 21

This particular respondent mentioned that it was challenging to try to help the team to become self-managing. A lot of the time there was disappointment in the team when the team had not performed well in terms of completing the work and those disappointments came when the team were not collaborating together. This suggested that when team members did not like the help they received it may impact on the collaboration in the software development team.

Team members in Agile software development teams were aware of the importance of the collaboration (Cockburn & Highsmith, 2001a). It was sometimes difficult to engage people to work together because of their natures. It was reported that only some people in the team were willing to work on their own.

> *"I would just say some people are naturally more individualist or have team members who are more individualist and didn't take, found it difficult to really sort of, um, let others in on what they were doing then share that then they tended, even though there were others in the team who wanted to collaborate, they were tending to sort of forced into being individuals and each having a story."* Respondent 22

In the Agile software development process software was built with collaboration and coordination between the team members, customers and the business people. This concept was followed by most of the Agile teams, as reported by the respondents in this study. However, there were some cases where it was difficult to encourage people to work together. This may occur because some people see their contributions as being more individual than team orientated. Not everyone was suited to working in the collaborative model that Agile encouraged. This could be because the members were not used to working in a group or who did not follow the proper Agile process during project development.

Figure 4-6 shows the coding process for the Team support theme.

I think that most people come to work every day with the hope of doing a good job and generally if someone has knowledge that someone else needs they are delighted to share that.

Someone is senior he mostly spends time helping others. If someone's a junior or a graduate, he will mostly ask for help. Respondent 12

Most of the time like, I am constantly helping other people, people are helping me if I have a question, so they are always approachable, and also we also have shared stand-up so that's also a point to share any ideas to solve this problem. Respondent 10

Helping others, If someone one junior mostly asks for help if it's, if it' if someone needs help like a blocking issue he might tell them during stand-up. Respondent 13

I try to help people as often as possible but so there are people who just don't ask for help even when they need it so sometimes that's difficult. Respondent 11

We had a problem with one person in a team. I think the problem was that he was very technical person and he was always thinking in terms of the solution not in terms of [the] problem. Team members were finding hard to coordinate with that person. Respondent 10

We have a lots of chats in the office because we work in the same space. We have online channels where we chat. We also have a code reviews so as we work through features we will read each other codes to see how we going. Respondent 6

Team leadership

Backup behaviour

Team help

Collaboration

Team support

Figure 4-6 Coding process for team support

4.8.4   Organisational culture

Organisational culture in Agile software development encouraged self-managing teams with the culture of shared responsibility (Dybå & Dingsøyr, 2008). Organisational culture enabled people to organize their work in a way that gave the development team a common ownership and a shared responsibility for the project (Dingsøyr et al., 2012a). Respondents for this study reported that team members generally took ownership of the work they had not completed.

> *"It's not like there will be blame thrown around the people. No one will feel guilty if they don't finish* something *within a day. They will just come to a next day and say. Ok I did this this this but it turned out to be different. I need to do this which I didn't account for. We will adjust and add more post notes tasks and resize the story."*
> Respondent 12

In Agile projects, the software development team members have different responsibilities. When a team member took accountability for a particular task, then there was a better chance of a good product; however, the general direction of the project was the responsibility of the product owner. West et al.(2010) reported that the product owner was responsible for answering questions, defining the general product direction and prioritizing the work (West et al., 2010).

> *"Taking a sense of responsibility, like if you are making a change, then you should feel partly responsible for it, like be able to champion it amm communication is big."*
> Respondent 6

A respondent for this study reported that there were some cases where one person was given multiple roles. Situations where one person had more than one role may cause some difficulties, especially in instances where that person had resigned from the organisation.

> *"I have [I have] seen that, I have had experience with that actually. I think two different people who left the company because I think it's, yeah like at, when I first started working I had two different people that were in the same situation so one person was hired [he was hired] I believe as a business analyst and then he was given sort of the product owner role as working so he was doing both and was doing it across two teams and I think it was too much for him so and he resigned and the*

same thing for another person as well who was hired after he also had same thing, so yeah, I think on, from their perspective it's, I think it's quite difficult, um, from my perspective." Respondent 17

This information given by the respondent suggested that there was a problem in defining the roles of members working in the team. In this particular occasion the individual was hired in the role of a business analyst but was given a different role later in the project. This could impact on the performance of the individual as the individual may not have the knowledge of working in these different roles.

Respondent 13 also reported some instances in an organization where the product owner was not involved in the project from the beginning. The product owner also did not have experience of working on the kind of product the company was developing. Agile software development process promoted team work (Highsmith & Cockburn, 2001). However, one respondent reported there were some organisations where there was only one individual in a team following the Agile process. The respondent in this project mentioned that he had a team but within that team his role was largely working on his own and sometimes he was asked to do manual testing. This suggested that the correct Agile process was not followed in that organization. This information also suggested that there were organizations who did not follow the completed Agile approach as there was no team working in the project.

"It's a bit of an odd model because the scrum model would ordinarily suggest that everybody in the team helps everyone with everything that they're doing but I am largely by myself in the work that I do because there aren't really many other people who understand what I do." Respondent 13

Despite nine respondents who said that the product owner and the senior management were available most of the time; however, one respondent reported that when the development team needed to discuss feedback often took longer than was practical. The respondent reported that the longer time may have occurred because of availability of the person realizing (or not) the importance of their answers to the development team.

"When it comes to talking to people in a business, particularly senior people, you may have to wait days before you can get an answer to a simple question." Respondent 9

As well as being cross-functional, Agile teams often consisted of members from different cultures. One respondent in this study said that, at times, cultural issues were evident within their teams. Culture can have significant effect on how people interpret certain situation and reaction to such situation (Holmstrom et al., 2006).

> *"There's cultural issues. Some people are sensitive so you've got to be careful about that but generally if you approach anyone at work with a smile and a positive attitude, you know, [if] you're are serious about what you are doing you will get a good response and, if not, that's a management issue."* Respondent 9

Most of the respondents reported that there was shared responsibility and ownership with the people involved in the projects. A study by Jakobsen & Johnson (2008) found that a product owner could also have multiple roles, such as software architect or a user experience manager (Jakobsen & Johnson, 2008). However, they did not mention the challenges with the product owner from having multiple roles. In this study, two respondents reported that there were some issues with the multiple roles given to the product owner in some organizations. Even though Agile software development support team worked together there were some instance sin an organization where one member had to work on his own. There were also some cultural issues within the team, but this was only reported by one respondent.

Organizational culture was not included as one of the components in the Dickinson and McIntyre's Teamwork Model (Dickinson & McIntyre, 1997) However, previous research undertaken by Brown and Starkey (1994) found that organizational culture had effects on communication (Brown & Starkey, 1994). Organizational culture was also reported to have an impact on factors, such as communication and collaboration in Agile software development (Cockburn & Highsmith, 2001b).

Figure 4-7 shows the coding process for the organization culture theme

Ownership here, referred to the duty of the team member to either complete a development task, to repair a defect, or even to improve the work. Responsibility referred to the quality of the completed task.

It's not like there will be blame thrown around the people. No one will feel guilty if they don't finish something within a day. They will just come to a next day and say. Ok I did this this this but it turned out to be different. I need to do this which I didn't account for. We will adjust and add more post notes tasks and resize the story. Respondent 12

Taking a sense of responsibility like if you are making a change then you should feel partly responsible for it, like be able to champion it amm communication is big. Respondent 6

Multiple roles in the project R (17)

It's a bit of an odd model because the scrum model would ordinarily suggest that everybody in the team helps everyone with everything that they're doing but I am largely by myself in the work that I do because there aren't really many other people who understand what I do. Respondent 13

When it comes to talking to people in a business, particularly senior people, you may have to wait days before you can get an answer to a simple question, having said that we can look in other directions for those kinds of answer." Respondent 19

Passing of knowledge from other parts of the business occurs in this organization and this company has a culture of helping each other.

Ownership

Responsibility

Difficulties within an organization

Culture

Organisational culture

Figure 4-7 Coding process for organizational culture

111

4.8.5   Team engagement

Team engagement referred to understanding the role of the team members in an organization and being able to take part in different activities during project development to meet the objectives of the project. Team member engagement was essential in Agile software development to deliver the best results for the project (Sohaib & Khan, 2010). Generally, in software development projects using an Agile approach, all team members were involved throughout the project. Only two respondents in this study reported that all team members were involved in all aspects of the project, from the requirement gathering stage in the project, despite Agile software development encouraging collaboration and coordination between the team. Team members were mainly involved from refining the requirements and breaking down the user stories during the project development process. The involvement of team members on the project from such activities could have created good interactions between the team embers. A study by Sureshchandra & Shrinivasavadhani (2008) also found that Agile teams were involved in breaking down the user stories to detail work and plan iteration accordingly (Sureshchandra & Shrinivasavadhani, 2008).

One of the respondent for this study said:

> *"Team breaks down the user stories."* Respondent 1

And another said:

> *"For the last project that we were involved with, the refinement was, um, there is a backlog, backlog of issues and the refinement was with the teams the development teams on a on a weekly basis so come into a refinement meeting that would usually last two to three hours, once a week and talk through the backlog and any urgent requirements."* Respondent 3

The above information from that respondent suggested that the involvement of team members during the meeting gave them a chance to discuss the project, to define the requirements, the urgent work to be done, and to talk about issues. Such activities could also help the communication about the project between the team members. Once project requirements were refined the team members divided the work based on the skill of each team member. In most projects the roles of the team member were defined by the product owner.

One respondent said:

> *"Most people already have their roles fairly well defined from a project manager, product manager, business analyst and developer etc. Within the developers it usually comes down to who has the particular skill set."* Respondent 4

The information above suggested that the teams were self-organising teams in which the team made the decisions, the estimates and then delegated particular tasks.

In the Agile process, if the customer made changes to the requirements team members were prepared for the change request as the Agile approach welcomed changes in requirements (Beck et al., 2001b). Likewise, teams in this study recognized that when working on Agile projects there was a chance of changes in the requirements.

One respondent said:

> *"We are prepared for change in scope as we are working on Agile environment."* Respondent 14

Agile software development supported collaboration and coordination between cross-functional teams. However, sometimes team members behaved more like individuals than members of a team. One respondent reported their experience of what happened in this type of situation. One respondent also reported that they did not have cross functional teams involved at present. However, he reported that they were aiming to have such a kind of team. This suggested that this particular organization was in the process of adapting more practices of Agile software development.

 It was found by Olsson et al.(2012), that cross-functional teams will help in removing the barrier in collaboration and communication issues (Olsson, Alahyari, & Bosch, 2012).

One of the respondent for this study said:

> *"One example we had one of the senior, um yeah, more senior developers, um, making decisions in isolation of the rest of the team and not letting rest of the team know so that had, the impact that caused was a fair bit of rework, um, by the rest of the team."* Respondent 5

And another said:

"We are aiming to build up the competencies within our teams and so that means their skill sets so at the moment we don't have testers who can do development and we don't have developers that can do testing. "Respondent 15

In most projects, team member engagement in the project started at the very beginning, where requirements were broken down into user stories. However, there were only a few projects where all team members were involved in refining the requirements. During the project development team members were prepared for the change. One of the respondents reported that management had tried to add some work to the team without discussing it with the product owner. Such activities may create confusion on the development team and could create conflicts between the management and the product owner. As in the Agile process, the requirements and changes came through the product owner to the team. However, the development team followed the Agile process by referring management to the product owner. But the management were not prepared to talk with the product owner. Team members in the projects were involved in dividing the work. Helping each other to complete the projects. There were some instances where team members made decisions on their own. When the team members made decisions on their own then it could impact the whole team. Lindsjørn et al., (2016) stated that Agile teams were self-organising teams in which the team made the decisions, estimates and delegates the particular tasks. If the team members worked on their own then that could impact on the teamwork.

Dickinson and McIntyre's Teamwork Model (Dickinson & McIntyre, 1997) did not contain team engagement as a separate component in their model. However, the team were found to be engaged in activities, such as stand-ups, retrospectives in Agile software development (Drury, Conboy, & Power, 2012). The involvement of the team member may have an impact on the communication and collaboration and feedback component of the team work model. Figure 4-8 shows the coding process for team engagement.

Team breaks down the user stories (R1)

For the last project that we were involved with, the refinement was, um, there is a backlog, backlog of issues and the refinement was with the teams the development teams on a on a weekly basis so come into a refinement meeting that would usually last two to 3 hour, once a week and talk through the backlog and any urgent requirements. Respondent 3

Our product owner team leader is generally involved in he runs our stand-ups and stuff so he is quite heavily involved in the process deciding what needs to get done (R13)

One example we had one of the senior, um yeah, more senior developers, um, making decisions in isolation of the rest of the team and not letting rest of the team know so that had, the impact that caused was a fair bit of rework, um, by the rest of the team. Respondent 5

It's really hard to get product owners to actually understand and even sometimes getting them to come along to meeting and to be involved (R14)

Multiple teams work together (R12)

We have [a] tribe structure so there are three team in one tribe and they work together

We are aiming to build up the competencies within our teams and so that means their skill sets so, at the moment, we don't have testers who can do development and we don't have developers that can do testing Respondent 15

Team members' activities

Involvement

Number of teams

Knowledge

Team engagement

Figure 4-8 Coding process for Team engagement

4.8.6   Product owner engagement

The product owners in Agile software development have different roles depending on the project or the organization (Dybå, Dingsøyr, & Moe, 2014). These roles included developing and maintaining the product backlog and prioritizing the requirements. In general, software development teams involved in this study said that the product owner was involved from the beginning of a project. The product owner was found to be involved in prioritizing the requirements, offering direction to the team and bridging communication between the customer and the development team.

One respondent said that their product owner was not a traditional product owner. In Agile software development traditional product owners became involved in the activities, such as, collecting users' needs, describing the requirements, deciding on the release date and content, responsible for project success, prioritisation and accepts or rejects work's results (Beck et al., 2001a).

One respondent who was working as a test analyst said:

> *"He just says, basically, this is the direction you need to go in and I define how I get in that direction. It's not really a traditional product owner role I don't think."*
> Respondent 13

It was reported by one respondent that some product owners were highly involved in the project while others were less involved. When the product owner was highly involved in the project there could be effective communication and feedback on the projects. The product director in their project was in a senior role and he was looking after whole projects, so he made the high level decisions and prioritizations.

One respondent said:

> *"The product director is worried about the whole product. The product owners are worried about from a what's going to go into [it], so we've got that sort of levels and the key thing is that they are looking for what features are required and try[ing]*
>
> *to get a product backlog together starting on that.*" Respondent 1

Another respondent said:

> *"Our product owner is generally involved in our stand-ups and stuff so he is quite heavily involved in the process deciding what needs to get done."* Respondent 11

The product owner, generally, was the contact point for the development team when the team wanted any clarifications about the project. The availability of the product owner was important for Agile projects. However, it was reported by six respondents that the unavailability of the product owner was one of the issues they had in their projects. Hoda et al., (2011) said that the unavailability of the product owner could result in challenges in communication, project priorities, prioritization and clarification of requirements.

One respondent said:

"So it's really hard to get product owners to actually understand and even sometimes getting them to come along to meetings and to be involved in the process, because they don't have really have. It's really hard getting a true sense of ownership." Respondent 14

Another said:

"I've worked at another organizations where the product owner is not available and he might be only available once a month, um, at best for a short amount of time." Respondent 5

And another said:

"The product owner not being available. It's very, very common." Respondent." 17

And another said:

"The product owner in our tribe as I've mentioned is mostly in regard to prioritization and, like having a big, higher level overview in his head of features so that's pretty much where his involvement ends." Respondent 12

Seven of the twenty-two respondents reported that the product owner they had in the project did not have explicit knowledge about the Agile process. This was reported by the respondents when the respondents were asked about the knowledge of the product owner working in Agile software development. One of the respondent reported that the product owner learnt about the process as they worked on the projects

"Our product owner, in particular, had no knowledge of working in Agile before coming into the project. The development manager, ah, decided that we would suddenly go to Agile and then told the product owner right you're doing an Agile

process now and that made a[ll] very, very difficult so there really was no knowledge,
so it was all learning as we go." Respondent 3

Another said:

"Had product owner that does not know what he was thinking, we delivered this big
feature that nobody wanted and there would be no uptake." Respondent 13

And another said:

"It's really hard to get product owners to actually understand and even sometimes
getting them to come along to meetings and to be involved in the process, because
they don't have really have … it's really hard getting a true sense of ownership."
Respondent 14

One respondent in this study also reported that the product owner involved in the project
did not know about the Agile process as they were used to with the traditional approach.

"Well, in the beginning, it was more that person did not know the role of product
owner, um, they were used to that waterfall methodology and their being quite far
away from actual development and so we had to educate the product owner to let
him be more involved with day to day running of the project and we had to educate
him about how user stories work." Respondent 15

Product owners in most of the projects spoken about by respondents in this study were
involved throughout projects from the refinement of the stories, to prioritization, to the
delivery of the projects. Data collected in this study showed project owners were involved in
all processes, including stand-ups, retrospectives and communication with the team and the
customer. Seven (32%) respondents reported that their product owner in the projects they
were involved in was new to the Agile process and six (27%) respondents reported that
there was a problem with the availability of the product owner. There were some instances
where the product owners' involvement was low. For example, these product owners were
not attending the meetings. It was reported by Power (2010) that the product owner
availability and participation during the project was important to deliver a quality product
(Power, 2010). The participation of the product owner in such activities will help the team to
get clarification on the requirements and prioritization. The other important point a
respondent mentioned was about the product owners knowledge of working in Agile teams.

When the product owner did not have knowledge of working in an Agile team then the product owner may not be aware of the agile practices (Santos, Goldman, & De Souza, 2015). The product owner may find difficulties in using the appropriate practices during software development. Practices used in Agile projects included stand-ups and retrospectives so the product owner's role in Agile projects were important in Agile software development projects. If the product owner lacked such knowledge then it could create difficulties for the team working on the software development projects. The impact of this could be that the team may not be able to get the right directions and this may have negative impact on the teamwork as well as the outcome of the projects.

Another challenge the development team reported in this study with the product owner was having a product owner who has multiple roles in the projects and, because of this, product owners were not fully engaged on the project. Having such roles could make difficulties for the member in making a decision about the project as these members needed to think from both perspectives. Hoda et al. (2011) also found that the development teams did not receive the level of customer involvement required by Agile methods. This created problems in clarifying requirements and getting the feedback from the product owner or for prioritization, as these activities were some of the responsibilities of a product owner (Bass, 2015). Their study also reported that the availability of the proxy product owner helped in coordinating with the team members. From this previous research (Hoda et al., 2011), it was suggested that the engagement of the product owner impacted on communication, one of the components of Dickinson and McIntyre's Teamwork Model (Dickinson & McIntyre, 1997). When the product owner had high involvement in the project then it could impact the teamwork in Agile software development.

Figure 4-9 shows the coding process for the product owner's engagement theme.

The product owner is actually responsible for releasing [I mean kind of] business requirement collection and also finalizing those stuffs and validating the business requirements (R8)

He just sees, basically, this is the direction we need to go and I define how I get in that direction (R13)

Our product owner team leader is generally involved in [that] he runs our stand-ups and stuff so he is quite heavily involved in the process deciding what needs to get done (R13)

The product owner is quite keen on getting those releases out of the door with all the different features. So, he is quite committed to that (R17)

It's really hard to get product owners to actually understand and even sometimes getting them to come along to meeting and to be involved (R14)

Client is involved throughout the project (R8)

Product owners are highly involved (R6)

They used to look for even multiple teams before (R8) (R21)

Our product owner, in particular, had no knowledge of working in agile before coming into the project. The development manager, ah, decided that we would suddenly go to agile and then told the product owner, right you're doing an agile process now and that made a very, very difficult so there really was no knowledge so it was all learning as we go. Respondent 3

Product owner is very good being involved with agile projects. (R6)

Product owner role in the project

Involvement level

Agile process Knowledge
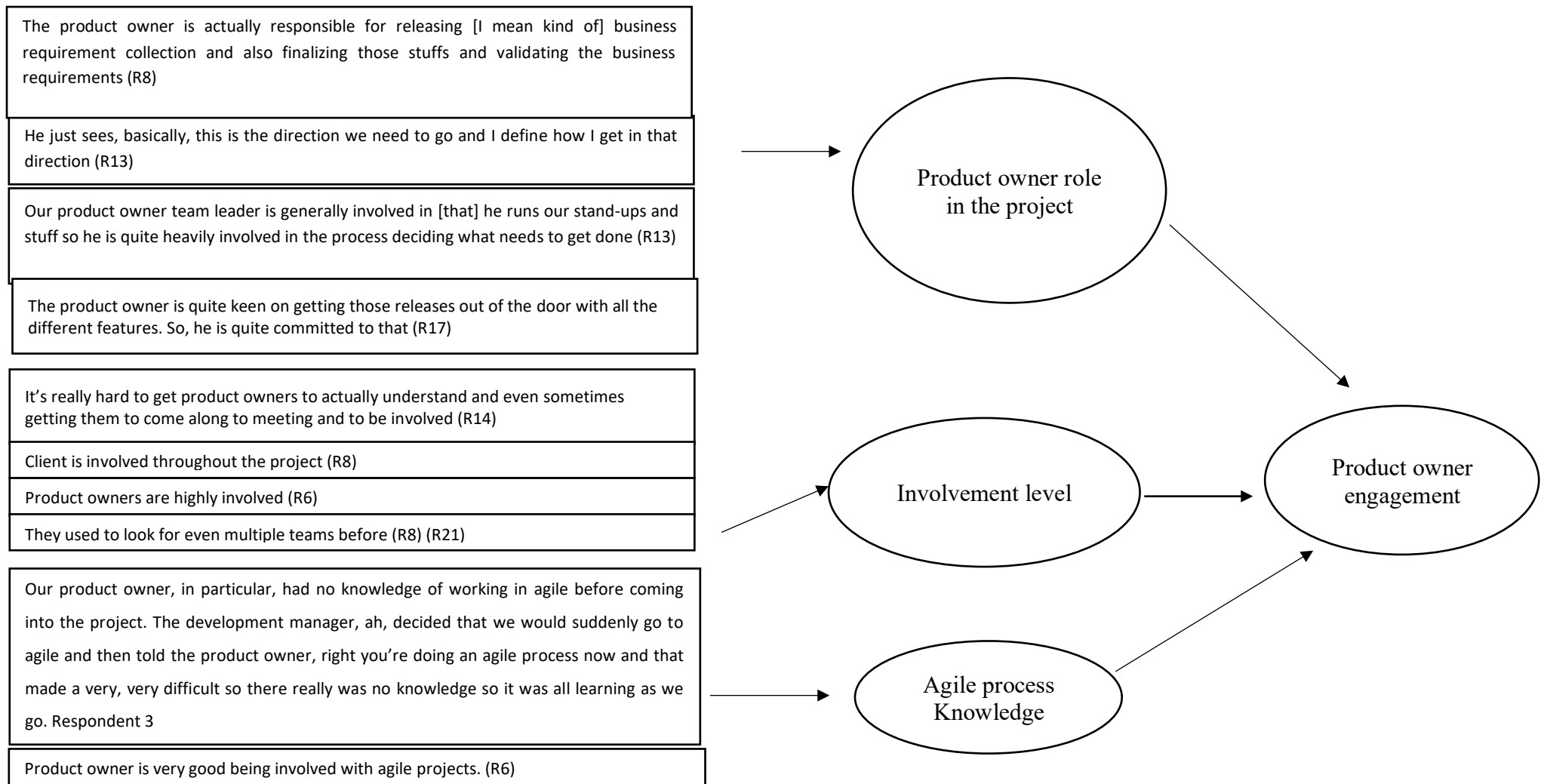
Product owner engagement

Figure 4-9 Coding process for product owner engagement

120

4.8.7 Confidence

In this section trust and confidence are two terms used to explain the information obtained from the respondents. Trust referred to the belief of the product owner on the ability of the team members to complete the task in the right way. Trust also referred to the belief in the honesty and fairness of the team members and dependence on the character, strength and ability of the particular team member in completing the task. Confidence in the context of this study referred to the belief of the product owner about the competence of the team member in completing the task. In this section, both trust and confidence were used as the respondent used both terms during the information gathering. The findings from this study suggested that, generally, product owners had confidence in the teams to do their jobs. McHugh (2012) said that a product owner must trust the team (McHugh, Conboy, & Lang, 2012). This will help the product owner to establish open and frequent communication with the team, as well within the team, during the sprint/iteration retrospective. However, some respondents for this study did not believe all the team members had the same level of confidence from the product owner.

One respondent said:

> *"You can explain, OK, its two days to do this work roughly, two to three days, and that allows us to be faster, in designing and implementing this other, the features down the road and he will listen to that, and he'll weigh it all up and go, ok do it."*
> Respondent 1

Another one said:

> *"With the product owner, the product owner trusts the team, he does not care how it's done. He cares if it's done*." Respondent 6

Some of the challenges with trust in the team with Agile practices were reported in a previous study (McHugh et al., 2012; Misra et al., 2009; Moe et al., 2010). These challenges included team pressure, tension between the product owner and the team, and the team members underestimating the task (McHugh et al., 2012). However, the level of trust in the team depending on experience was not reported. This current study respondents reported that trust on the team was dependent on the experience of the team members working In the project. Product owners were found to have trust in the experienced team members who had been working on the project for a substantial period of time.

One respondent said:

"Our product manager trusts the people who have been working with the company for a longer time. These people obviously have more experience working in the projects." Respondent 2

Seven respondents reported that product owners had confidence in the team. Two of the respondents reported that the product owner was involved in monitoring and observing the work of the team rather than contributing in a more practical sense. This could mean that the product owner either did not have confidence on the team or may like to know about the work team members were doing in detail to give direction to the team.

"He does not contribute much; .they are there to monitor, I should change that, they do contribute by monitoring us, so any particular show stoppers, we might have a show stoppers which, I don't know, might be semi-serious but the business might turn around and say. 'No, no, no that impacts say correspondence so this is huge'." Respondent 9

Another respondent said:

"The product owner is involved in observing the work. He asks people to help each other when they have problems." Respondent 3

Trust was an important factor in Agile software development. Having the trust in the team motivated the team members to do a better job. One Agile principle was that the project should be built around a motivated individual, who should be trusted (Cockburn & Highsmith, 2001a). There were times where the product owner had confidence in the senior people in the team. The junior people in the team have to build confidence with the product owner through their performance. This could mean that the product owner may be confident about team members once they had seen the performance of the team. McHugh (2012) found that trust in Agile teams increased communication and feedback. And the communication and feedback were two of the component in teamwork (McHugh et al., 2012). When there was trust on the team then the members may be willing to share the information within each other. Teams could be more self-organizing. Such activities will impact on the coordination, team backup and feedback within the team and could a have positive impact on the teamwork.

Figure 4-10 shows the coding process for the confidence theme.

You can explain, OK, its two days to do this work [in]roughly, two to three days, and that

allows us to be faster, in designing and implementing this other, the features down the

road and he will listen to that, and he'll weigh it all up and go, ok do it." Respondent 1

Work division is based on skill (R6)

Documentation was more about what was needed and user stories did not contain
anything about how we do it cause was up to the team (R15)

Oh, with the product owner, the product owner does not care how it's done (R6)

The product owner is involved in observing the work. He asks people to help each other

when they have problems." Respondent 3

Our product manager trusts the people who have been working with the company for a

longer time. These people obviously have more experience working in the projects.

Respondent 2

Depends upon who is asking. Some people, some developers, they can say that's not

right. They will be asked how we fix it and then let's just do that other one, will depend

on how they trust their developers (R10)

Belief on the teams
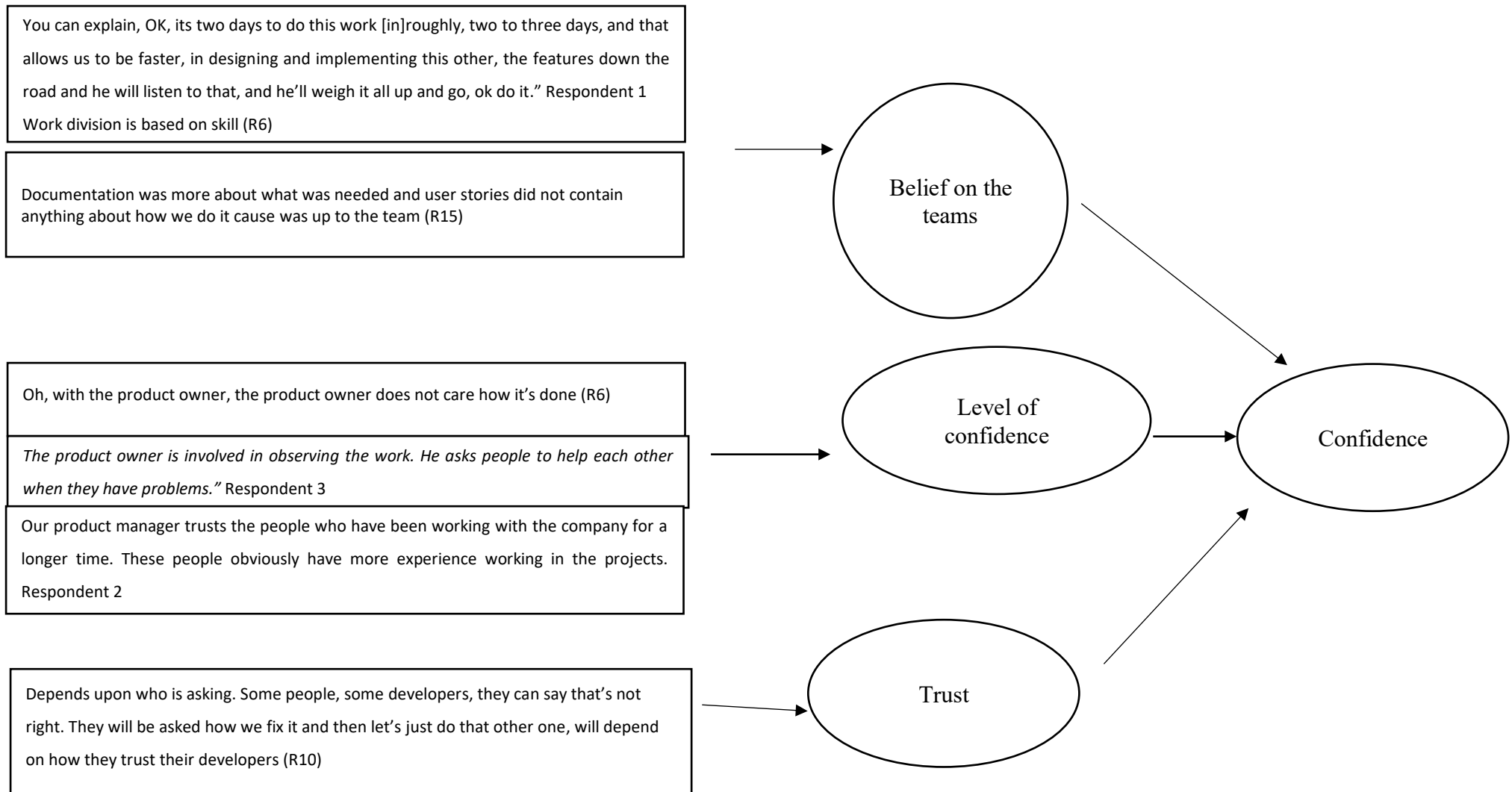
Level of confidence

Trust

Confidence

Figure 4-10 Coding process for the confidence theme

4.8.8   Team motivation

Motivation is one of the important properties that humans need in order to achieve goals with good quality results (McHugh, Conboy, & Lang, 2011). Team motivation included: initiation, direction, intensity and behaviour towards the work. Team motivation comes from the organisational context, the specific job being undertaken and the profession (DeChurch & Mesmer-Magnus, 2010; Salas, Cooke, & Rosen, 2008; Whitworth & Biddle, 2007).

In Agile software development there were different members in a team. Each of these members were motivated by different factors. Respondents for this current study reported that development team members were motivated by different reasons, such as project success, team success and working in a group. Despite the respondents reporting that the teams were motivated to obtain good results, two respondents also reported challenges in keeping team members motivated. This suggested that motivation was important for Agile teams for effective teamwork. However, motivation was still reported to be one of the challenge in Agile software development.

Four respondents in this study reported that team members in Agile teams were motivated with the salary they have been getting in return for the work they were doing. Team members were also motivated by professional pride they get when they receive positive feedback. They were also motivated by the professional opportunities that involvement in the Agile process provided.

One respondent said:

> *"So as long as they are comfortable with what they are been paid generally then they tend to draw motivation from success in the project that they are doing and I think the Agile approach really helps to constantly reinforce that if they're succeeding."* Respondent 21

Other one said:

> *"I take professional pride in what I do. You know, at the end of the day, I quite like it when the customer turns around to us and says good job guys and that's perfect, or that new form and process you delivered is, you know, the users love it, 'cos we get pretty good feedback from the customer."* Respondent 20

And another one said:

"When I joined there, I was like the junior developer and now I become like intermediate and somebody like who are system analyst and after the system analyst they will want to go to the BA you know like after, that's why and when our when they change their role their money will be also good there, so it is a motivation for money and some increment in their level." Respondent 2

The response above from the respondent suggested that team members were motivated to work in a group. But by doing a good job a team member can make progress. This will help them to further their career along with the better pay.

In Agile software development, the software was developed with self-motivating teams where success drove motivation.

One respondent reported:

"Team success would definitely be a motivation factor, yeah, with individual success and team success." Respondent 19

Team members became motivated not only with the teams' success but also by the opportunity to be part of a successful project. Team success here referred to an effective team and being able to complete the task. When the team delivered the project to the customer, and the customer was satisfied with the project, that created value in the business. This could also motivate the team to work in the projects. There were other motivational factors that the respondents reported.

One respondent said:

"I have a motivation, me personally, I have a motivation to work in the groups that I do because I am motivated to help those people, to help them get their job done." Respondent 13

And another said:

"What motivates people to work together I guess it's easier solving problems that way. Ah, we discourage people from, you know, when someone stuck and spends like two days of googling and pulling his hair out trying to figure out the solution." Respondent 12

Even though the Agile software development process has been given as the platform for team member motivation (Fowler & Highsmith, 2001), not all respondents in this study

reported a high level of team motivation. In cases where a team member may report a concern to management, and then get no response about this concern, their motivation can be negatively affected. This suggested that product owners listening to team concerns and be able to be involved with them could motivate teams to do good work during the project development.

> *"Another thing is the team voice a concern to the top management and then if the management does not listen to that concern the team just carries on you know, they are not motivated, they just continue doing the work, yeah."* Respondent 17

Eleven respondents reported that team members were motivated with the success of the team and the project's success. Team members were also motivated because they felt they were being sufficiently rewarded by their salary for their involvement in projects. Another important factor that motivated people was the organisations' social culture. In organisations where there were regular staff events, motivation levels were higher than in organisations that did not have a social culture. Two of the twenty-two participants reported that some of the management activities. such as not listening to team concerns, were impacting on team members' motivation. Chow & Cao (2008) found that team motivation to be one of the success factors in Agile software development. It was also found from a study by Ceschi et al. (2005) that motivation was the most important quality to being able to work in groups. They also mentioned that continuous training and regular communication improved teamwork ability and motivation (Ceschi et al., 2005). Figure 4-11shows the coding process for the team motivation. This suggested that team motivation was important in Agile teams for effective teamwork. When the teams were motivated to do the work they will be able to deliver a good result.
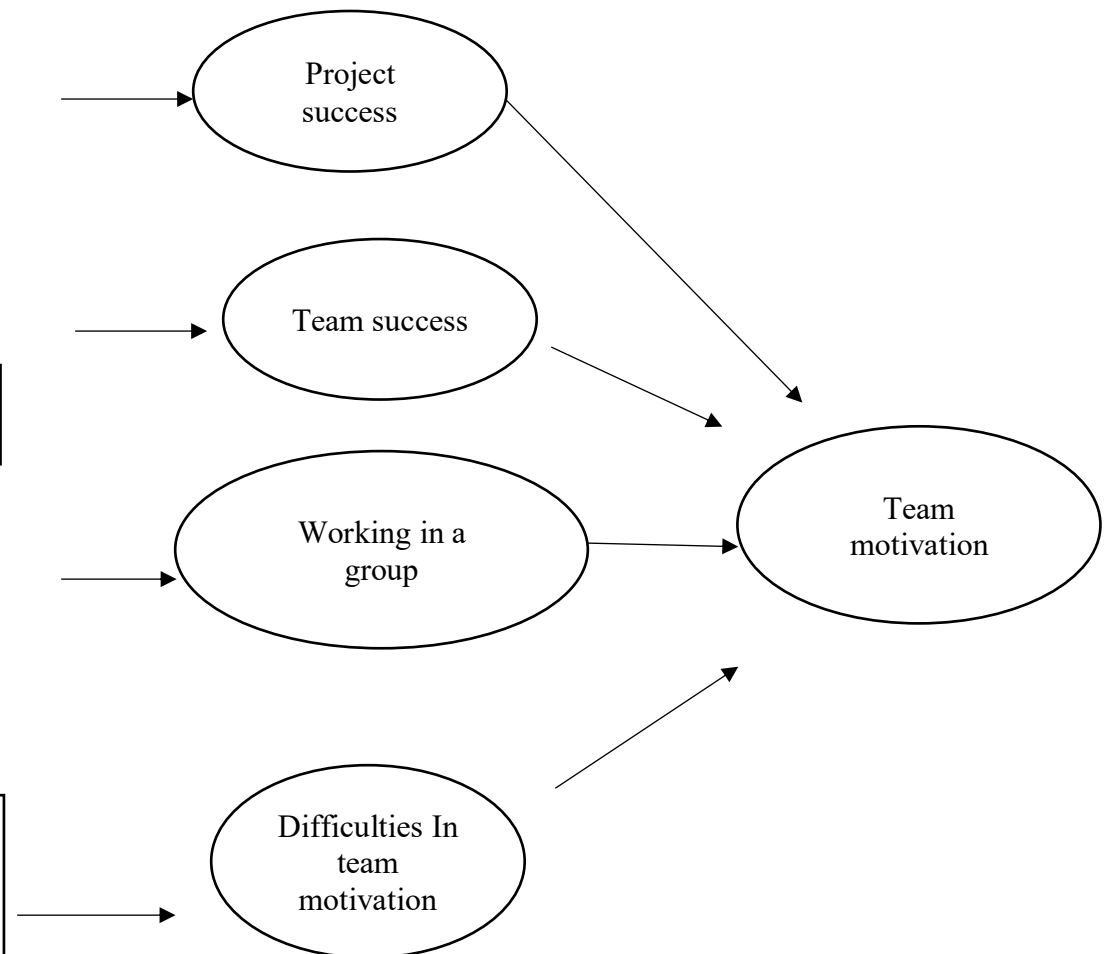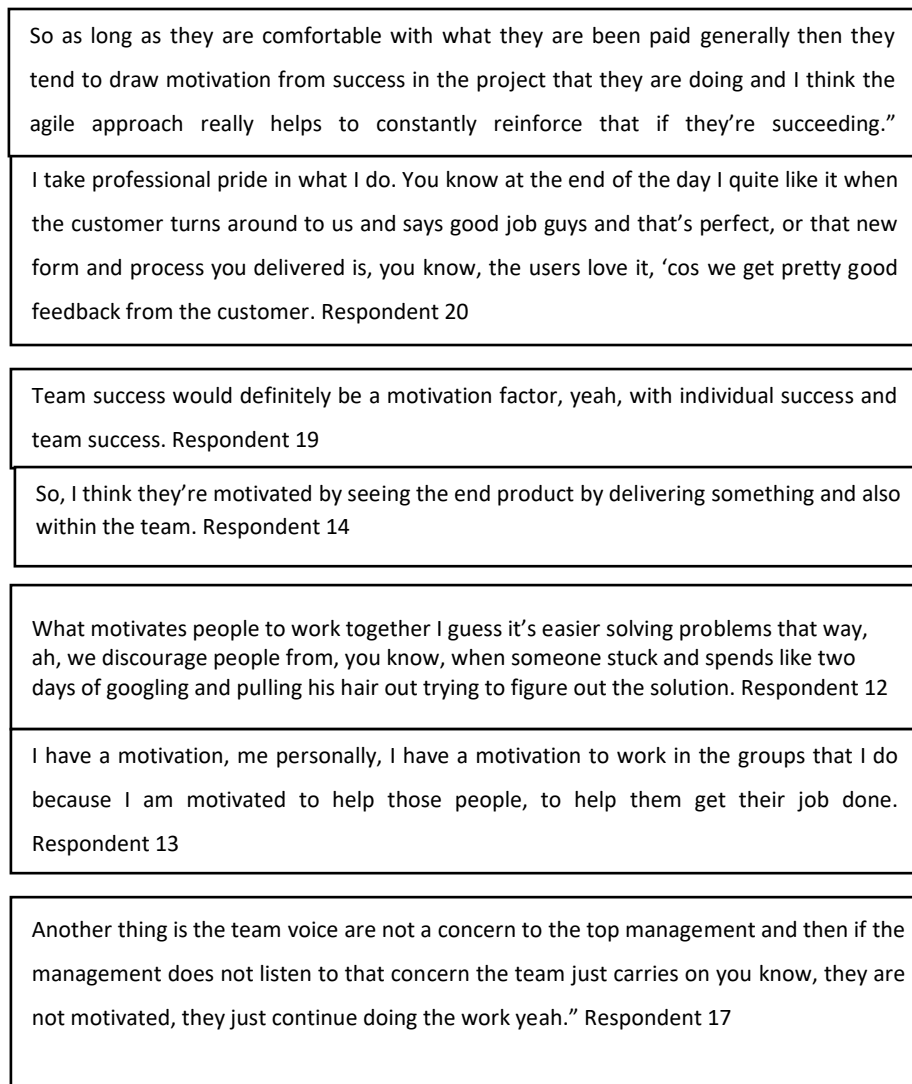
So as long as they are comfortable with what they are been paid generally then they tend to draw motivation from success in the project that they are doing and I think the agile approach really helps to constantly reinforce that if they're succeeding."

I take professional pride in what I do. You know at the end of the day I quite like it when the customer turns around to us and says good job guys and that's perfect, or that new form and process you delivered is, you know, the users love it, 'cos we get pretty good feedback from the customer. Respondent 20

Team success would definitely be a motivation factor, yeah, with individual success and team success. Respondent 19

So, I think they're motivated by seeing the end product by delivering something and also within the team. Respondent 14

What motivates people to work together I guess it's easier solving problems that way, ah, we discourage people from, you know, when someone stuck and spends like two days of googling and pulling his hair out trying to figure out the solution. Respondent 12

I have a motivation, me personally, I have a motivation to work in the groups that I do because I am motivated to help those people, to help them get their job done. Respondent 13

Another thing is the team voice are not a concern to the top management and then if the management does not listen to that concern the team just carries on you know, they are not motivated, they just continue doing the work yeah." Respondent 17

Project success

Team success

Working in a group

Team motivation

Difficulties In team motivation

Figure 4-11 Coding process for team motivation

4.9 Summary

The respondents who took part in this study reported about projects they were involved as developers, testers or business analysts. From the beginning, the majority of the respondents reported that the team members were mainly involved in Agile process activities, such as stand-ups and retrospectives. This involvement of the team members during project development was different in some projects. For example, in some projects the team members were not involved in setting the priorities.

During project development, there were exchanges of information between the development team and the product owner. Communication was mostly reported to be good in most of the projects. Despite this, there were some cases where there were challenges in communication because of language and time zone differences. Team members collaborated to perform tasks. Team members were found to be given feedback from the product owner. There was only one case where team members received feedback directly from the customer.

Team members were found to be motivated by different factors, such as social interactions, seeing value delivered to the customer and working in a group to complete the task.

Most organizations were following the Agile culture. However, some respondents reported that they worked on their own. Five respondents mentioned that their organizations had implemented the Agile software development approach recently and expects that they are better in following the Agile approach as they kept had previously develop projects using this approach.

The aim of this research was to understand the relationship between the software development and the product owner. Information from the software development team is presented in this chapter. In the next chapter, the results and discussion from the product owner's study are presented.

Chapter 5: Product Owners' Study Results and Discussion

5.1   Introduction

The purpose of this study was to understand the relationship between an Agile software development team and their product owner. A mixed method approach (questionnaire and interview) was used to collect the data from product owners, as described in the Methods chapter. The data collection process, and the results and discussion for this study are presented in the sections that follow.

5.2   Data collection

The questionnaire (shown in Appendix G) was designed to gather data from the product owners. The questionnaire was administered using the online questionnaire tool Qualtrics. Respondents responded to the questionnaire by clicking the link provided in their email or from a message forwarded by their colleagues. Completing the questionnaire took 10 to 15 minutes. on average.

Purposive and snowballing sampling was used to select respondents for this study. An invitation was sent to the respondents who were currently involved in software development roles, such as a customer representatives or product owners. If the respondents were not able to participate, they were asked to provide the details of a colleague who could participate from their organization. Respondents for this study were recruited from professional groups dedicated to Agile processes as well as the researcher's personal contacts and their networks.

A total of nine product owners from eight different organizations took part in this study. Of the nine respondents, one respondent had already completed the questionnaire for the software development team study so, in total, eight respondents were included in the data analysis.

To obtain the information in detail semi-structured interviews were conducted with each of the respondents who completed the online questionnaire for the product owner study. A total of seven interviews were conducted as one of the respondent became unavailable.

The product owners who took part in this study were from different organizations. Of the eight product owners, four were from product development organizations, two from in-

house developments and two from bespoke development companies. All the participants were proxy product owners[10] and were employees of the software development company. In the following section, responses from the questionnaire are analysed and discussed. In the second section the interview data are analysed and discussed.

5.3 Quantitative data

The first phase of this study used an online questionnaire to collect data from respondents. The data collection process, data analysis and result are described in the following sections.

5.3.1 Respondent experience

Respondents were asked how long they had been involved in software development. Table 5-1 presents more details of the years of experience for the respondents. However, there was no information about how many years' experience they had in using an Agile approach during software development. This information gives an overview of the experience of the product owner respondents.

Table 5-1 Respondents' experience

1- 5 years	0
5- 10years	4
10-15 years	2
More than 15 years	2

5.3.2 Challenges between the product owner and the development team

To gather information on the challenges product owners faced with the development team, respondents were asked categorize a number of typical challenges by placing an occurrence factor on each. Sprint in the Agile context is a period where certain tasks are completed. A typical sprint in Agile software development was two to four weeks (Beck et al., 2001a). Table 5-2 presents the responses from the product owners about the challenges they were faced with during the software development.

---

[10] A proxy product owner is a person who acts as a customer and is responsible for managing product backlog and ensuring the value of the work.

Table 5-2 Challenges between the product owner and the development team in a typical sprint

Difficulties	0 times	1-3 times	4-6 times	7-9 times	More than 9	Don't know
Disagreement with the development team about project priorities	1	4	1	1	0	1
Disagreement with the development team about project requirements	0	3	1	2	1	1
Disagreement with the development team about the timeframe of the project	3	1	1	1	1	1
Interpersonal challenges between the team member(s) and customer	2	4	1	1	0	0
Challenge in communicating with the development team	2	1	3	1	1	0

There were five categories of challenges listed on the questionnaire. The responses from the respondents for each category is discussed, below. This information indicated the types of challenges and their occurrence during a sprint.

5.3.2.1 Project priorities

Of the eight respondents, four reported that they have had disagreements one to three times with the development team about the project's priorities in a typical sprint. Only one respondent had no challenges in relation to the project's priorities. Of the eight respondents. one respondent had four to six challenges, and one had seven to nine times. This suggested that the level of challenges in regard to the project priorities between the product owner and the development team was higher in some projects than in other challenges. This could mean that the challenges may have occurred because of level of involvement of the product owner, as well as the team members, during the refinement of the requirements. Chow & Cao (2008) found that challenges related to defining the project scope were one of the failure factors in Agile software development. They said that the challenge was related to the process of defining the project scope. This study adds more details to the findings of Chow & Cao (2008) in that there were challenges that existed between the software development teams and the product owners in defining the scope of the project's processes. This may lead to impact on the outcome of the project.

5.3.2.2 Project requirements

Two respondents reported disagreements about project requirements with the team as many as seven to nine times during a sprint. Three respondents reported that they had one to three disagreements, one respondent each reported that they had a disagreement four to six times and more than nine times. There was one respondent who did not know about the challenges. These findings might suggest that challenges in project requirements do exist and may impact on the project outcome. This impact may be as drastic as causing cancellation of a project or more minor in that a slight time delay was encountered.

The findings from this study suggested that there were challenges in defining the requirements and the number of challenges varied in relation to the different projects. These findings were in line with those of Ceschi et al. (2005), who found that fifty per cent of their respondents reported that challenges in defining project requirements was one of the problems faced with the customer. However, they did not mention how often these

challenges occurred in the projects. The challenges in project requirements were related to the processes followed for defining the project requirements. This may mean that the challenges in project requirements did exist in Agile software development and these challenges were there due to either the software development team or that the product owner did not have had a clear understanding of the project requirements or the product owner not being able to clearly define the requirements to the software development teams.

The findings from this study suggested that challenges existed between the product owner and the software development team during the process of defining the project's requirements, which could impact on the relationship between the product owner and the software development, resulting an impact on the project outcome. The impact may be positive in that a greater understanding was achieved between stakeholders or it may be negative in that less understanding was achieved between the stakeholders.

5.3.2.3 Project timeframe

Three respondents reported that they did not have any disagreements about the project timeframe in a typical sprint. One respondent each reported that they had disagreements about the project timeframe of the project one to three times, four to six times, seven to nine times and more than nine times, respectively. There was only one respondent who was not aware about any disagreements over the timeframe of the project. This findings was interesting, because previous research (Chow & Cao, 2008; Misra et al., 2009) had identified success and failure factors related to Agile software development. The failure factors included the Agile process, people, management and technical issues. However, there was no information found in the literature that specifically identified time factors as being the root of challenges between teams and the product owner. This suggested that there needed to be clear discussion between the product owner and the development team in regard to the timeline allocated for the tasks that can be accommodated in a typical sprint. This could help both the product owner and the software development team to have a common understanding of the timeframe required for particular task during the sprint.

5.3.2.4 Interpersonal challenges

Interpersonal challenges were one of the categories of challenges listed in the questionnaire. Four respondents reported that they had interpersonal challenges one to three times during a typical sprint. Two respondents reported that they had not had any

interpersonal challenges. One respondent each reported that they had interpersonal challenges four to six times and seven to nine times. The findings in this study suggested that interpersonal challenges did exist in Agile development between the software development teams and product owners. A study by Estler et al. (2014), identified the presence of interpersonal conflicts in Agile software development. The interpersonal conflict can be due to the lack of confidence in the team. This result was obtained by conducting interviews with 18 different projects. However, they did not specifically define what interpersonal conflicts were and how often these occurred. The implications of this information suggested that interpersonal challenges existed in Agile software development and this could be because of a lack of confidence in the team by the product owner. This meant that when the team felt that the product owner did not have confidence in the team they may not be motivated to do a better job.

5.3.2.5 Communication challenges

Challenges in communication referred to the difficulties in passing the information from one team member to another member. These difficulties may be because of location, language, failure to listen, authority and inadequate knowledge. Three respondents reported that they had challenges in communication four to six times in a typical sprint. Two respondent reported that they did not had any challenges in communicating. One respondent each reported that they had challenges in communication one to three, seven to nine and more than nine times. Previous research has identified communication as important for organizational relationships as communication was a way of helping people to avoid conflicts and achieve the project's goal (Malone & Crowston, 1994; Pikkarainen et al., 2008). Mark et al.(2001) also noted that when the team has communication problems they were likely to experience coordination problems during the completion of the work (Marks, Mathieu, & Zaccaro, 2001). Communication was one of the major components of the Dickinson and McIntyre Teamwork Model (Dickinson & McIntyre, 1997), which facilitated coordination, feedback and monitoring in the team. Dickinson and McIntyre (1997) also mentioned that challenges in communication had an impact on the feedback and coordination in the teams.

Information obtained from the respondents in this study suggested that challenges existed between the product owner and the software development team during software

development projects. All the respondents had disagreements on at least one of each of the list (Table 5-2) with the development team. This ranged from zero to nine times or more. The data also suggested that these were more likely to concern the project's requirements and priorities compared with interpersonal, communication and the timeline of the projects. This may mean that, because most of the Agile team worked in a team and defined method from communication, there were fewer challenges in communication compared to the challenges in the project's requirements. This was good from the point of view that the development team and the product owner had good communication in the projects. The positive impact of this could be that the team may have a clear direction and understanding of the projects as well as the coordination between them all be better.

5.4   Interviews

To explore the information for the product owner, study semi-structured interviews were conducted. (See Appendix G). Semi-structured interviews were conducted to understand how the team worked on the software projects. What roles did the product owners and the development team have for the projects? Data obtained from the interviews were analysed using a thematic analysis method (Braun & Clarke, 2006). The data analysis process is described in the Methods (Chapter 3:)

Seven factors, product owner engagement, team involvement, communication, team support, feedback, trust and organizational culture, were identified contributing to the relationship challenges, as described in the sections that follow.

5.5   Factors contributing to the challenges

Seven factors identified from the qualitative data analysis are discussed in the following sections, the definition of each factor identified from this study and the number of individual respondent's comments on each factor are presented in Table 5-3.

Table 5-3 number of respondents' comments on each factor

Factors	Definition	Number of individual respondents' comments on each theme.
Communication	Communication is the process of transferring information between individuals who are involved in a project.	7
Team support	Team support involves being available and willing to assist other members in the team. These teams work together to solve problem within the scope of the project.	4
Feedback	Feedback refers to providing information about the performance and acceptance of the suggestions given on the project's work.	5
Product owner engagement	Product owner role for the different activities of the project, from starting the project to deployment.	5
Team engagement	Team member's activities during the projects. The activities of each of the team members are aligned in a way such that the targets associated with a project goal are achieved.	4
Confidence	A belief in the team to perform the given task from the product owner	5
Organization culture	A self-managing team with a culture of shared responsibility. Successful Agile development teams require product ownership, from both within the team and within the organization, that understands and supports the iterative nature of the Agile process.	6

5.5.1 Communication

Effective communication was a significant factor in a project's success (Hummel, Rosenkranz, & Holten, 2013) because communication helped in having a common understanding of the project and coordinating the team to complete the task. According to the Agile principles, face-to-face communication was the best form of communication (Pikkarainen et al., 2008). The product owner respondents reported that communication was effective when the product owner was sitting with the team, as the team can approach the product owner directly. This could mean that product owner's location in the team could impact on the project's success. There were several communication paths used on the projects. These included the product owner and the team; between the product owner and the customer; between the management and the product owner. Likewise, there were different communication forms used for the project, such as face-to-face communication, online tools and emails. Skype[11] was one of the online tools used to communicate where people can see each other and hear the voice of the call participants. However, Skype may not be comparable to face-to-face communication since videoconferencing changed the nature of the information processing by the participants. Participants attending the call through the video conferencing were influenced by the speaker appearing interesting and friendly rather than the quality of the speaker's arguments (Ferran & Watts, 2008). Some of the quotes related communication mentioned by the respondents are:

> *"I sit with the apps team because when they've got questions about functionality features and how they are doing stuff they can just come and talk to me."* Product owner 1

One of the product owner respondents mentioned that the Agile approach was used in their organization because the team members sit together with the product owner. However, he also mentioned that they did use the other practices such as stand-ups, retrospectives during the project development.

One respondent said:

> *"I told you we are using the Agile so just the whole team with the product owner sit together"* Product owner 2

---

[11] https://www.skype.com/en/

Another said:

> "*I manage the teams in terms of what work they were bringing in, so ordering the priority backlog so we were working on the right stuff, um, and yeah, manage their, the communication path between them, and there was an external provider as well that the client was using that we needed to coordinate with as well.*" Product owner 3

And another said:

> "*When it comes to the weekly backlog refinement phone calls that we have with our general product owners we have one to two members from each team involved in that as well.*" Product owner 7

All respondents in this study noted that effective communication was important during the project. However, some of the respondents also noted that there were some difficulties in communication during the project, including team members who did not want to, or found it difficult to, communicate. Members who did not want to communicate verbally used the project's documentation to communicate rather than speaking with others directly. This could delay the passing of information and, also, the information may not be interpreted correctly.

Naur et al. (2005) found that an Agile software development approach can reduce the amount of documentation created for the project. This suggested that if the team members totally depended on these documents for communication then there was a chance of missing important information.

One respondent said:

> "*It depends on the people within the team as well. Some people just take the scope document and they'll just deliver to that without talking to me, other people will come to me every day and talk to me about the requirement making sure they are getting it exactly right.*" Product owner 1

There was some miscommunication between the teams and the product owners. One of the respondents mentioned that they had this miscommunication problem because they didn't use formal communication methods, such as sending an email. There was an instance reported by one respondent about a development team who thought that they had

communicated with the product owner about the project. However, the product owner mentioned that he had not been told. This suggested the importance of formal communication methods and more documentation for important information, including the meeting notes. This document could work as a backup when the team needed clarification.

Hossain et al. (2009) found that maintaining documents with important information can reduce instances of miscommunication. They also mentioned that some of the development teams used tools, such as issue tracker, enterprise wikis and project management tools to maintain the documentation (Hossain, Babar, Paik, et al., 2009). These tools were used to share the document, track issues and distribute tasks across the team. Three respondents for this study mentioned that they used similar software.

One respondent said:

> *"Miscommunication, and that went both ways so obviously not having [those] overarching steps. Ah spec and scopes meant that we missed some pretty critical areas and then there was miscommunication from them so they thought that they had told me a very important piece of information about how this solution was being developed and they hadn't. And by the time I found out we were too late to make changes to it."* Product owner 3

One respondent for this study mentioned that they had one team in their organization whose product owner was located in another country so was contacted using online tools. Even though he was available most of the time there were some difficulties observed during the communication process. This was a problem that some project risked running into due to working across countries. This was in line with findings by Phalnikar et al. (2009) who found that time zone differences meant fewer hours available where both sets of participants were in the office. This can decrease the communication between the participants as one of them may not be available when needed. (Phalnikar, Deshpande, & Joshi, 2009).

Respondents said:

> *"The product owner was based out of California and he used to manage one team and that was OK because, again, even he was always online on Skype when he was travelling, he was not, but for the majority of time he was available. And also*

California, just depending on time of year between three to five hours' difference with us so it's not too bad, and if you know the timeslot of the other person you can accommodate. I mean we even have guy in UK. Sometimes talk to either in my early morning or my late evening." Respondent 6

It was noted in the Agile manifesto that face-to-face communication was the best form of communication (Alliance, 2001). Trunk et al. (2014) stated that face-to-face communication gave a chance to the team members and other business people to change the direction of the discussion, if required, during a conversation However, one product owner respondent in this study reported that it can be difficult to be sure that team members have understood all of what the product owner was communicating even though they were in the same location. This may be because of team members who came from different cultures with different language understanding. This could create communication gaps between the members in the team and could impact on their relationships.

This was similar to that found by Damian (2002) who said that cultural differences were believed to impact on the communication process in projects. The cultural difference could be differences in languages, national traditions, values and norms of behaviour (Holmstrom et al., 2006).

One respondent said:

"Communication is key. It's very hard and but it's completely mandatory but you cannot do without communication. Sometimes it's hard because we've got a lot of different cultures in the team in terms of we don't have the same model in mind so when we communicate we need to be kind of genuine and not judge people and just try to explain things as they were and be sure that the people in front of you understand it." Product owner 4

And another said:

Information from this study identified different communication types, such as informal (face-to-face discussions in-collocated teams or by telephone) and formal communication (stand-ups, group meetings or formal meetings). In this study, some difficulties were found in communication that may be caused due to language differences, differences in time zones and the availability of the product owner. Similar difficulties were reported by

(Sutherland et al., 2007). In their study this difficulty was found to have a negative impact on both the relationships between the software development teams and the product owner. This suggested that the communication challenges were present in Agile software development and could have a negative impact on the project's outcome.

According to the Dickinson and McIntyre's Teamwork Model (1997), communication was one of the important components of the Teamwork Model where all the other activities, such as coordination and feedback have an impact on the communication in the team. This meant that communication in Agile software was important for effective teamwork.

5.5.2 Team Support

Team support referred to team member's attitudes towards the other team members when completing a task. Team support also referred to team members being available to a team member who was unable to a perform task, and being willing to help a team member to correct a mistake. Respondents from the product owner study mentioned that team members in Agile teams were willing to help each other during the development of the project. Despite having a helping culture within the team there were challenges, such as the lack of cross-functional teams. This could impact on the teamwork in Agile software development.

Cross-functional teams meant that the team had all the skills needed for software development (Hole & Moe, 2008; Kahkonen, 2004). Respondents from the product owner's study also reported that team members helped other team members to solve issues. Team support in the team may have a positive impact on the coordination between the team members. A product owner could help the team when they were unable to obtain a clear understanding of the problem.

> *"The willingness is certainly there and when there are opportunities to do it the guys are very good about helping each other out, trouble shooting the different issues as they come in."* Product owner 1

One respondent said that being a product owner meant he was helping the team by giving direction. This was similar to what was found by West et al. (2010), who said that the role of the product owner was to give the team general direction. The product owner was able to guide the team to give better projects outcomes to the customer.

One respondent said:

"On the questions when they were not able to come out, we should go like this. We should go like that and try to help the development team here to focus on the outcome to the customer." Product owner 5

Collaboration was observed within the team and between the team and the customer to ensure that the product built was according to the vision of the customer (Cockburn & Highsmith, 2001a; Hoda, Noble, & Marshall, 2010). Collaboration in teams and with the customer helped in defining clear project requirements. And this could have a positive impact on the outcome of the projects. Two respondents in this study mentioned about collaboration. One respondent reported that the team members were involved in working together to perform the work and remove any difficulties.

"We do the sprint so we feed in some changes but each time we are facing impediments there is a very good collaboration within the team to help people to remove these impediments, so that's great." Product owner 4

And another said:

"Currently we are working on the project we need to wait them to finish their parts so, um, we need to come up[with] a plan[on] how can we collaborate with each other but it's the progress is quite slow." Product owner 2

In some Agile projects, some features may be built by external service providers (Batra et al., 2010). Collaboration was important when there was an involvement of such providers. A respondent reported that collaboration in such situations was slower than expected and this impacted on several different components, such as the timeline of the project, team cohesiveness and collaboration. It was negatively impacting the project's timeline as the software development organization had to wait for the eternal provider to finish the given task.

One respondent mentioned that some difficulties existed in collaborating with different external providers in some projects. This was reported by this particular respondent as they were using an external provider for some of the services in the projects. This suggested that there could be a difficulties in collaboration when the external providers were involved in

the project. The impact of this on the project was that it could delay it and, also, the completed project may not be able to meet the expectations of the customer.

Nerur et al. (2005) found that a collaborative culture within an organization should be encouraged because organisational culture influenced the behaviour and actions of the members (Nerur et al., 2005). This type of culture may also impact on the decision-making processes, relationships and planning and controlling mechanisms during project development. This finding went some way to suggesting that this type of culture was suited to an Agile approach where effective teamwork was an imperative.

Three respondents in this study reported that their teams supported each other during project development. This support was evidenced through the help they provided to team members during the completion of the project. Team support helped to facilitate knowledge sharing and helping the team members. In doing this, they passed the knowledge the team members had because of their experience and skills. Drury et al. (2012) also found that teams sharing knowledge helped team members to gain a shared understanding of the business domain and the functional requirements of the projects. Increased sharing of knowledge between team members may impact the decisions individual members made on their own tasks (Ghobadi & Mathiassen, 2016; Moe et al., 2008). Team coordination and back-up were two of the components included on the Dickinson and McIntyre Teamwork Model (1999). Team support identified from this study overlapped with the team coordination and back-up components in the Dickinson and McIntyre's Teamwork Model (1999). This suggested that team support could be important factor for effective teamwork.

5.5.3   Feedback

Feedback involved giving and receiving information among the team members about their performance (Whitworth & Biddle, 2007). Respondents in this study reported that feedback was given by the team: to the product owner, the customer to the team, management to the team and the team to the product owner. This meant that there was feedback between the members involved in Agile projects. Three respondents from the product owner study mentioned that team members gave them feedback on the project during review sessions. This helped the product owners to analyse the progress and the value of the projects, as the product owner was able to gather more information on the project about the progress,

difficulties and improvements. Feedback on the project could be negative or positive. Positive feedback may help to motivate the team for better performance. Respondents for this study did not mention that the product owner or the team were involved in giving negative feedback. This suggested that the constructive feedback from the team or the product owner could impact on the team's performance.

The feedback process in Agile software development can be between: the customer to the product owner, product owner to the team, and the team to the product owner. One respondent product owner said that the customer was involved in giving feedback to the product owner. He said that he was open to discuss with the customer any questions and concerns about the project. He, in turn, then passed on these concerns and questions to the appropriate members of the team if this was deemed to be necessary. This may mean that this product owner, in particular, had the confidence of the team that the feedback they gave was important for the discussion. Dingsøyr et al. (2012) found that the involvement of the customer in facilitating feedback can lead to more satisfying projects because the development team can receive an indication of the impact of their work on the projects and may get suggestions for improvement .

One respondent said:

> *"Our customers bring me in on a call where someone has concerns or questions or just wants to discuss how to make it work better."* Product owner 6

This information suggested that the product owner was also involved in getting feedback from the customer. This also meant there was communication between the product owner and the customer to improve the project to meet customer's need.

Feedback in Agile software development was given through different media, such as face-to-face or through email (Lindstrom & Jeffries, 2004). Two respondents in this study said that the retrospective and sprint reviews were occasions used to give feedback. It was noted that the retrospective was a useful place to get a quick feedback.

One respondent said:

> *"It will happen in the retrospective or the review so have a sprint review and the retrospective review meeting after that so that will be the usual forum for getting quick feedback on all that kind of stuff."* Product owner 1

And another one said:

"We will have sprint review every sprint so I think that is the time they give us feedback. They want to change anything so they can give us feedback during the sprint review." Product owner 2

Feedback in an Agile project was used for improving the project and to meet the customer's needs. One respondent reported that they were seeking feedback by sending a prototype to the customer. Likewise, Karlstrom & Runeson (2005) said that getting early feedback from the customer removed incorrect functionality at an early stage of the development.

One respondent said:

"We were doing a lot of early prototype development of functionality as we developed one change and delivered to them, give them prototype radio so they can give us some feedback and we tried to do that actually in person." Product owner 5

Despite the importance of feedback one respondent reported that their team did not get feedback from the customer and the end users as they did not have direct contact with the end users. This suggested that the feedback was important in Agile projects because this helped to confirm some the work done and the improvements that can be made.

"We got prioritization so everyone understands why it's clear but I am not sure that's the right prioritization right now as a product owner because I don't have the customer and end user's feedback." Product owner 4

In this study team members and customers involved in projects were reported to be actively involved in the process of facilitating feedback. The feedback was given at different meetings such as the retrospective or the sprint review. Because these were the times where all the team member were together to talk about the sprint. This was important because this will allow the team members to share information related to the projects, such as what went well and what could be improved in the next sprint. Williams and Cockburn (2003) said that in Agile software development the feedback and changes were important as they helped to eliminate errors in the functions and led to more satisfying functions. One respondent reported that they were unable to get the feedback from the end users because they were not accessible to the development team. This meant that there could be difficulties in understanding the exact requirements of the user. In the Dickinson and

McIntyre Teamwork Model (1997), feedback was an intermediate process between monitoring and coordination for ensuring communication. This will help in acknowledging the information received as well as provide additional information to improve the project . Three respondents reported that team members were involved in giving feedback to the product owner. However, one project spoken about in this study did not have any customer involvement in the feedback. This suggested that perhaps there needed to be an increase in feedback to help improve teamwork in this type of project environment. This could help in effective teamwork and may impact on the project's outcome.

5.5.4   Product owner engagement

The product owner in an Agile software development project was involved in different stages of the project from the very beginning to deployment. This role involved defining the general direction of the project, deciding on project priorities and with the responsibility for answering questions from the development teams and the customer (West et al., 2010). The product owner was involved in providing the information about the projects to the development team. This was aligned with the findings from Kettunen (2009), that the product owner should ensure that the customer expectations were addressed with the product backlog (Kettunen, 2009). This suggested that the involvement of the product owner in Agile projects was crucial for the project's success, because product owner was the person whom the team can approach when they needed any clarification about the requirements. From the information supplied by respondents in this study it appeared that prioritization was the area a product owner was likely to be most engaged with.

One respondent said:

> *"So my role in that particular project was to champion the project; I dealt with all the end users, gathered their requirements and processes, passed them on and, in some cases, documented them up, and passed them on to the development guys to create the work flows and forms, and then I arranged the demonstrations back to the stakeholders and sign off when they were to go live…and the rest of it as well."*
> Product owner 1

And another said:

> *"I provide more information, detail[ed] information for team and then the team will do the implementations; then just answer all the questions and be the bridge*

between the stakeholder and the team and then ensure the quality of the product and deliver the product to the market, Yeah." Product owner 2

The idea of having the product owner involved in Agile software development was to involve them in requirement refinements and to provide direction to the software development team (West et al., 2010). Product owners in Agile projects were responsible for the delivery of a project (Beck et al., 2001a; Pikkarainen et al., 2008). Two respondents in this study reported they had full authority on the project. When full authority was given to the product owner, this may impact on the communication and decision making process in the projects. These findings align with those from Sutherland et al.(2008), who said that for a project to be successful the product owner should be given clear responsibility and a defined authority (Sutherland, Jakobsen, & Johnson, 2008).

One of the respondents said:

"The product owner should have the full control of the product features functionality, everything." Product owner 6

Two respondents said that in their experience a product owner can be involved in multiple roles within the team. Both of these respondents reported that having multiple roles can be difficult to manage. One respondent mentioned that the product owner was not able to access the end user of the project because of which it was hard to capture the users' needs.

"It's how we can access to real user of the application. So first you need to understand that I am working in [a] contact where the customer. So this guy who buys the application, pays for this application, is not the guy who is using the application. So the end user and the customer are different and usually you deal with the customers, you don't deal with the end users. But if the end user is not using your application they will not buy your application, so it's really hard to access these end users." Product owner 4

And another said:

"I was taking in the requirements from the clients. I run a kind of dual role where I am a product owner and I am also a business analyst." Product owner 3

All the product owners who took part in this study were from software development organizations, meaning that they were all internal customers (proxy product owners) responsible for ensuring the quality of the products. The results from this study could have been different if the product owner respondents for this study had been real customers. This was because the different organizational culture the customer belonged to was different from that of the software development organization. To deliver quality products, product owners focused on prioritization, release planning activities and deployment of the project to the customer (Power, 2010). There were some projects where the product owner had more than one role. In such cases, it was observed to be difficult for the product owner to be fully engaged in the projects. The impact of this on the project was that the product owner found it hard to find a balance between the two roles. When a proxy product owner was representing the customer, they needed to understand the needs of the customer. In one case, even though the product owners were fully engaged in the projects, they were unable to access the ends user of the project to deliver their exact needs because of the nature of the projects. This project was developed for an organization which bought the product and sold it to their customers, who were the end user of the projects. The implication of this was that the project delivered may not meet the expectations of the customer.

Of the seven respondents, three were the product owners for the team they belonged to as well as a team member. Researchers have mentioned that the involvement of the product owner was important throughout the Agile projects (Paasivaara, Heikkilä, & Lassenius, 2012; Power, 2010). However, this study has indicated that the some of the product owners were found not to be fully engaged in some projects because of their role and availability within the projects.

West et al.(2010) found that in Agile software development a product owner was responsible for the general direction of the project.

The level of engagement of the product owner may impact on the communication with the team as the team needed to wait for the product owner be available or the team needed more time to explain the situation to the product owner. Because communication was one of the components of the Dickinson and McIntyre Teamwork Model (1997) that linked team activities. If there were a challenge with the level of engagement with product owners this

may have an impact on general communication within a team. This may mean that the engagement of the product owner may have an impact on the project outcome. The findings, above, suggested that product owner engagement could be added to the Dickinson and McIntyre Teamwork Model (1997) for better teamwork in the Agile teams because the product's owner engagement may have impact on the communication, feedback and backup behaviour constructs of the teamwork model.

5.5.5   Team engagement

Team member engagement in Agile projects was vital (Hoda et al., 2010). Because, in Agile software development, the software was developed by teams. In two of the projects in this study a team member was acting as a product owner so was undertaking multiple roles. According to the Agile Guidelines "Agile team members are expected to show continuous attention to technical excellence, self-organizing, reflect on how the team can become more effective and build projects around team as motivated individuals. Product owners in Agile teams are expected give team a direction and support they need during the project" (Alliance, 2001).

Four respondents in this study reported about team members' involvement in story grooming sessions where they discussed among the other members, including the product owner, and decided about the solutions. During this, they talked about the requirements and broke down the requirements. This meant there were some project where team members were involved from the beginning of the project. However, two respondents also reported that not all team members were involved on the project during prioritization. This could be either these project did not follow proper Agile approach or they did not want to involve the development team during the requirements refinement process. One respondent also reported that team members were not good in estimating the time required for a task. And another product owner reported that prioritization was the job of the of the product side where the team did not have any say.

One respondent said:

> *"Probably the most difficult one is some team members consistently undersize around the task they work which is a real problem 'cos it leads to the blowing out of the, um, timeframes and deliverables, which soon affects the stakeholder, the customer."* Product owner 1

Another said:

> "No, the team cannot have any say on the priorities because I think [it[is the right of the product side." Product owner 2

And another said:

> "No, not all team members right now is involved in the prioritization. And I think it's a very good idea. I am quite sure that everyone is not open to this conversation yet." Product owner 4

One respondent reported that the whole team became involved in the grooming, expecting the team may come up with the better solution. A study undertaken by Williams (2012) mentioned that the involvement of team members, such as testers in story grooming, gave an opportunity for reducing the defect on the product backlog.

> "We get the whole team involved in the story grooming because you get five different perspectives and, so yes. And also in the story planning period if someone comes up with a better solution, again, we leave the technical solution to the team so they can decide the best technical way to approach it." Product owner 7

During project development, Agile teams get involved in different activities, such as iteration planning, iteration execution, iteration review and retrospective. The level of involvement of teams in such activities was found to have a positive impact on the decision making processes during the projects (Drury et al., 2012). Two respondents in this study reported that their development team did not get involved in activities, such as project prioritization. This suggested that when these team members were not involved in the activities, such as prioritization, that could impact on the decision-making process in the projects.

The product owners also stated that those activities were not within the team's role in projects. However, one respondent said that the team's involvement in the project during the grooming has helped by giving them the opportunity to get a view of the project from a different perspective. This particular respondent gave an example where product owner was asking them to work on a feature that was estimated to take three points. One of the development team members explained how the same work can be done with ½ a point. This suggested that there can be some valuable input from the development team members if

they were included in the story grooming session. This suggested that involvement of the team member from the beginning of the project could have a positive impact on the estimation of the tasks. The Dickinson and McIntyre Teamwork Model (1997) includes communication as one of the major element for effective teamwork. The findings from this study suggested that the team engagement impacted on communication in Agile teams. When the team members were engaged, then there can be effective communication, coordination between the team members can be improved and this can provide support to the other team members working in the projects. Hence, team involvement could be another potential component of the Dickinson and McIntyre Teamwork Model (1997).

5.5.6 Confidence

In this section two terms were used by the respondents. Trust referred to the ability of the team members that the task will be done in the right way. Trust also referred to belief in the honesty and fairness of the team member and this depended on the character, strength and ability of the team member when completing the task (McHugh et al., 2012). Confidence in the context of this study referred to belief in the competence of the team member in completing the task. In this section, both trust and confidence are used, as the respondent used both terms during information gathering.

Not all product owners reported that they had confidence in their team. However, most of the respondents reported that they had confidence in the team. Agile software development projects were developed with trusted team members.

One respondent said:

> *"I really trust them on their estimate and all the rest of it as well. So when we are sizing things, I do take part in the sizing. It's just as one of six people and so I go, I am just one of the team and they appreciate that I am just one of the people within the staff."* Product owner 1

One product owner mentioned that when they knew the team and had a background of the software development there were fewer problems. A study conducted by Ramesh et al. (2006) mentioned that there was increased trust when the team consisted of team members who have previous experience of working with each other (Ramesh et al., 2006). These findings indicated that when the product owner was more familiar with the team then there was an increase in the level of trust on the team.

One of the respondent said:

> *"I have no problems working with the development team. So, 'cos it was largely a software project and, because I come from that background, I come from that team [and] I know them they know me, I think there is a good trust there. Occasionally they say I've got it wrong and some get a bit angry with me and that is how it is. But, overall, I think it went very well in terms of relationship with the development team."*
> Product owner 5

Trust required the product owner to believe that their team members had the knowledge and competence to complete the task. The respondents in this study reported that the team members involved in the project had different understandings of the project and different development methods because they had a spent long time on the project. A study by Cockburn & Highsmith (2001) stated that Agile software development projects should be built around a motivated individual who should be trusted. However, in this project this was not found to be true.

One respondent said:

> *"First thing is, I don't have confidence with the development team from my perspective. I think the project is not that big but that totally depends on different teams, they have [a]different understanding."* Product owner 2

Another respondent mentioned that team members' needed to have enough information so that the team can make the right decision, as this built the trust of the team. The development team had access to all the information that was needed for the project. With such information team members can make decisions about the projects. A study conducted by McHugh et al. (2012) mentioned that an increase in transparency, communication, sharing of knowledge and feedback increased the trust during the project development (McHugh et al., 2012). This was also reported by one of the respondents in this study who said:

> *"I need to trust people to take the right decision but for them to take the right decision they need to have enough information, and so in order to have enough information we need visibility on this information."* Product owner 4

In Agile software development agility depends on trusting individuals to apply their knowledge in effective ways. One of the respondent reported that he trusted the team and

did not apply any quality control on the project. At the later stage of the project he realized that the results were not as he had expected.

> *"I think I went to the other extreme where I trusted the team too much, I didn't implement enough quality control or enough testing. Like, I trusted their testing and did not bring in external testing as well, which is probably the other major learning from that particular project." Product owner 3*

Trust was important in Agile software development teams as these teams were self-managing teams. However, it was reported that when the product owner trusted the team without any observations the results were not as expected. The team were expected to show some results to earn trust from the product owner. This could happens because the team may have not shown their confidence in the task or may have misplaced confidence during other projects. For example, the product owner trusted the senior member of the team as they would have gained confidence on the projects. This may impact on the motivation of the junior team members. Two respondents for this study were involved in observing and recognizing the activities and performance of team members (Alchian & Demsetz, 1972). Some of the respondents for this study reported that the product owner was involved in observing the team's performance.

> *"I am attending, I try to attend every stand-up and so if they are discussing about having done something I sometimes go to the individual person afterwards and ask them to show me and if its good I tell them right away or sometimes even during the sprint they have completed the milestone that they talk about in the stand-up and you know, then we just say that's well done, that kind of stuff." Product owner 6*

Agile software projects were developed around motivated and trusted individuals (McHugh et al., 2012). However, the product owner respondents for this study reported that too much confidence in the team without some observations of team performance was found to have an impact on the project's outcome. As well as saying that, the same respondent also said that development team should not be dominated. This meant that the development team should not be trusted completely and also should not be dominated during the project development. This information also suggested that the product owner may need to balance between the trust and dominance on the team.

"Where I trusted the team too much I didn't implement enough quality control and enough testing. Like, I trusted in their testing and did not bring in external testing as well, which is probably the other major learning from that particular project." Product owner 3.

Of the seven respondents, two of the respondents reported that they did not have confidence in the teams. This was because the product owner had observed that the team did not make correct estimations. Challenges were reported by McHugh et al. (2012) due to a lack of trust in the team. These challenges were that the team may feel pressure on themselves to deliver the project. Tension may also develop with the product owner and the development team during project development (McHugh et al., 2012).

In Agile software development, the teams are self-managed. Responses from the product owner respondents suggested that the product owner did have some control of the team. This could mean that the product owner was making sure that the project was successful and all the work was undertaken appropriately. It was stated by Moe et al. (2012) that, in a self-managing team, the team was responsible for observing and managing their performance. They also mentioned that observation was important for the Agile teams, based on the project their study was based on, as there were cases where the observer found that team members were working on other things; in some cases, they even found members sleeping. However, such activities were not reported in this current study. Two products owners in this study indicated that they were involved observing the development teams during a project to make be aware of the progress made and that the work was done in the correct way. This could mean that the product owner did not have confidence in the team or product owner wanted to be aware of the project being developed.

5.5.7 Organizational culture
Organizational culture in Agile software development refers to a self-managing team with a culture of shared responsibility (Dybå & Dingsøyr, 2008). This enabled development team members, including the product owner, to organize their work in a way that gave the development team common ownership of the project and control over how it was achieved. Respondents reported that the team have freedom to explore on the product.

One respondent said:

"So I sit with the apps team because, when they've got questions about functionality features and how they are doing stuff, they can just come and talk to me. They go, I am going to do it this way. Five minutes of looking and then I go, 'Yep, I can totally understand why you are doing that, that makes a lot of sense, thank you.' And they know they're on the right track and that's what it took." Product owner 1

And another said:

"Each team can choose how they work so you know the scrum processes from team to team can be quite different. As long as it works for that team, as long as they can deliver what we need to deliver, then they have total freedom." Product owner 7

This information suggested that the teams were completely self-managing and have control on the projects. Cockburn & Highsmith (2001) reported that when different members with different skills worked together, the people, environment and organizational culture all influenced one another. In the projects reported by two respondents, the organizational culture was found to be supportive of the team members. This could improve the coordination and motivation among the team members

Shared responsibility was important in Agile software development projects as the software was developed by self-managing teams. Respondents for this study reported that the product owner was there to assist the development team to perform the necessary processes for achieving a quality product. McHugh et al (2012) reported that shared responsibilities within development teams was important. As the team members worked as a group, everyone may need to take responsibility for the project developed. They also said that the people who take part in the project's development should be able to talk about what they did in the project with other members, and others can give feedback on the work. However, not all team members want to have that responsibility as it could be that they did not have confidence on the work they had done. The product owner expected the team to have the ownership of the work that the team had done.

One of the respondent said:

"The responsibility, ownership, because I think the question currently I meet is more related to the people. So I will like to have a better team because so previous[ly], I based [it] on my previous experience. I worked with a very good team and they ask a

question but they will also do things with a high quality. So this is very, good, like balanced. They will give you a challenge but they will give you the things they have, the responsibility of the ownership of all of those things." Product Owner 2

A cross-functional team was an important feature of Agile software development, as cross-functional teams have all the skills necessary to create product increments. However, despite its importance, it was not always possible to have cross-functional teams. This could mean that sometimes team members have to take multiple roles. One respondent mentioned that he tried to combine the skill from two teams in order to achieve the required skills. However, he said that he faced some difficulties because of the working cultures of the two separate teams had impacted the sprint timeline. As these people had different ways of managing the work, such as using the sprint board, it required some work to understand how the team wanted to work together. There needed to be some demonstration of the work to help the team members to follow similar approaches to complete the task. All team members, including the product owner, were involved in the open retrospective to improve the way of doing the tasks in each sprint.

"The main challenge I faced was that I have got two teams and they all specialize in specific components of the product and in order to resolve this user story or this epic I need skill from both teams. So we put in place a working group composed of people from both teams in order to work together, but they don't have exactly the same culture, they are not managing the sprint board the same way, they don't have same definition of ready the same definition of done, which had an impact on sprint timeline. " Product Owner 4

Another respondent reported that management sometimes tried to use their position to influence the team but were not prepared to talk with the product owner. This could create conflicts in the team during project development. The team, however, followed the Agile process and referred management to the product owner.

"What tends to happen is people will try to come from the site to the team and get them to work on what they want them to work on instead of going through what's on the board. But the team is really good at saying, 'That's not on the board.' Most of

the team is pretty good, some of the people can't do this, oh someone next to me oh look I'll do what you say, um, but overall it's pretty good." Product Owner 4

Organizational culture was one of the important factors in Agile practices since Agile practices emphasize individual and interactions over the processes and tools and put the emphasis on the collaboration. The organizational culture should be supportive of working in a collaborative environment on software development teams. The product owner and other people involved in the project needed to take an ownership of the project. Human activities and software development were influenced by the culture within the organization (Misra et al., 2009). This meant that the organisational culture impacted on communication, interpersonal relationships, coordination and feedback.

There were few cases reported where the team was composed of different people who had different working cultures. There were some instances reported in this study where management tried to influence the team to do their work beside/alongside? or as well as? the product owner. Such activities will impact on the software development process. This could lead to conflict between the management, the product owner and the development team as the Agile process encouraged changes on the project to come through the product owner. This finding goes some way to suggesting that organizational culture can have an impact on different factors, such as communication, team support, back-up and team motivation that, together, contribute to effective teamwork.

5.6   Summary of interview findings

The respondents who took part in this study reported about projects they were involved as product owner. These projects were initiated either by a customer, senior management, or the development teams. From the beginning, the product owners were involved in different activities, such as prioritization and determining the timeframe of the project. The majority of the respondents reported that these product owner were mainly involved in Agile process activities, such as prioritization, stand-ups, retrospectives, and giving direction to the development team. This suggested that the product owner was involved in the different practices used in Agile software development. This involvement of the product owner could impact on the relationship with the software development team. The product owner in some projects had more than one role, and this was reported to be one of the challenges. These product owners worked together with the team to achieve the project goals. Having

multiple roles could impact the relationship with the development team as the product owner needed to be able to fit between these two roles.

During product development, four of the product owners had confidence in their teams. There were two product owners who mentioned they did not have confidence in the team to complete the task. This was because the development team was not able to complete a correct estimation of the task. This suggested that the not all product owners have confidence in their teams. When the product owner did not have confidence in the team, then the team may need some monitoring from the product owner.

During project development, there were also exchanges of information between the product owner and the team members. Communication, in general, was found to be good in most of the projects. There were some cases where there were miscommunications and difficulties in communication because of language and time zone differences. Team members collaborated to perform the tasks. And team members, as well as customers, were involved in providing feedback about the product to the product owner. However, there was a case where the product owner was not able to get direct feedback from the end users of the product. This meant that communication was important for improving coordination and team support, which then led to effective teamwork.

Most of the organizations were following the Agile culture. However, some of the organizations did have some issues with their senior management approaching the team without going through the product owner. This meant the senior management either did not follow the proper Agile process or they still used the concept of traditional organizational culture where senior manager can approach the team at any time and request a change. Three of the product owners mentioned that they were quite new to the Agile environment and, because of this, they were still in the learning phase.

As part of this study was to understand the relationship between the Agile teams and the product owner, two separate studies were undertaken, the product owner and the software development teams. In the following chapter the findings from both the software development team study and the product owner study are compared and discussed, along with the teamwork model proposed by Dickinson and McIntyre (1997).

Chapter 6:   Combined Discussion Chapter

6.1   Introduction

In the previous chapters the findings from both the software development team study and the product owner study were presented. In this chapter, the research questions will be discussed together with the findings outlined in Chapter 2: .

The research questions are:

RQ1: The literature suggests that there are challenges between the software development team and the customer representative in Agile software development projects. To what extent do these challenges exist?

RQ2: What types of challenges are present in Agile software development projects?

RQ3: What is the impact of challenges on the relationship between the customer and the Agile software development team?

RQ4: Does the nature of the relationship between the customer and the development team have an impact on the project's outcome?

RQ5: Can the Teamwork Model adequately explain the nature of the relationship between the customer and the development team in an Agile development context?

Research Questions 1-4 are addressed using quantitative and qualitative data collected from both software development team study and the product owner study. The Teamwork model in Agile software development is discussed, along with the findings from this study, to answer Research Question 5. The findings from both studies go some way towards answering the first research question.

RQ1: The literature suggests that there are challenges between the software development team and the customer representative in Agile software development projects. To what extent do these challenges exist?

The data presented section 4.9 and 5.3.2 identified challenges in communication, project priorities, timeframe of projects and project requirements, as well as interpersonal challenges between team members. This meant that there were challenges in Agile software development projects. Such challenges were reported by the majority (73%) of

respondents from the software development team study and sixty-seven per cent of the product owners. Both the software development team study respondents and the product owner respondents believed that these could be common challenges in Agile software development.

Information obtained from both the software development study and the product owner study suggested that 53% of respondents faced at least one to three challenges about project priorities, project requirements, timeframe of the project, interpersonal challenges or communication. These findings suggested that the challenges existing in Agile software development were observable and could impact on the project's outcome. The implications of these observable challenges were that some of these challenges had an impact on project outcomes. This was confirmed by calculating the correlation between challenges, such as communication, interpersonal challenges, project priorities, sharing of ideas and project completion time, as presented in detail in section 4.3.

RQ2: What types of challenges are present in Agile software development projects?

Data collected from the online questionnaire suggested that challenges existed within the software development teams as well as between the product owner and the software development team. The implications of such challenges could impact the outcome of the project. Such challenges were in-line with those identified by Chow & Cao (2008) and Misra et al. (2009). As part of this study, in addition to the online questionnaire, data were collected using semi-structured interviews. The purpose of this data collection was to explore the information in greater detail than allowed in a questionnaire format. The analysis of interview data revealed challenges not previously identified in previous studies on Agile software development. The additional challenges included organizational culture, something a team cannot control. Challenges identified from this study are:

- Communication
- Availability of the product owner
- Team motivation
- Organisational culture
- Interpersonal challenges
- Disagreement with customers about project priorities and requirements

The challenges in communication found by Hoda et al (2011) were reportedly due to a lack of availability of the product owner during project development. The implications of such challenges were problems in gathering and clarifying requirements because the product owner was not there when the team required greater clarification about the requirements. This resulted in delays in making decisions. In this current study, the respondents reported that the availability of the product owner was a problem and caused delays in communication. This finding was discussed in depth in 6.2.3. Chow & Cao (2008) and Neur et al. (2005) reported defining the scope of a project as one of the main challenges in Agile software. This was because Agile software development methods relied on planning with an understanding that everything was against the rapid the development of the projects. In this study, respondents were asked how many times they faced a particular challenge in communication, project priorities, project requirements and interpersonal issues during a typical sprint. This information helped in indicating the depth of common challenges in Agile software development. The information obtained from the respondents on the number of challenges faced in each sprint are presented in section 4.3

Team motivation, in the context of this study, referred to the development team members' desire to complete tasks assigned to them. Similarly, Chow & Cao (2008) investigated the literature relating to Agile development and identified team motivation as one of the potential success factors for Agile software development projects. After the identification of these factors from the literature, they conducted a web-based survey, gathering feedback from 109 Agile projects from 25 different countries. The findings from this survey helped them further refine their list of critical success factors. The final list did not contain team motivation as one of the success factors in Agile software development. Their results were based on a questionnaire in which a seven point Likert scale was used to reflect the level of perception of the respondents about success factors. Their study could not confirm team motivation as a success factor in Agile software development. This could be because the respondents may have thought the Agile methods themselves resulted in team motivation. Ceschi et al., (2005) included the questionnaire the question, "What is the developer's most important quality in Agile software development?" Thirty per cent of the respondents in their study mentioned team motivation as an important quality in Agile software development. However, in this current study respondents were asked why they were

motivated when working in a Agile software development approach. This question was included to find out why people were interested in working on Agile software development projects. The finding from the software development team from this current study suggested that, like the literature review first undertaken by Chow & Cao (2008), team motivation was one of the success factors necessary in Agile software development.

Challenges with external factors and organizational culture were found to be new challenges identified from this study that had not been reported in the previous study (Chow & Cao, 2008; Misra et al., 2009). The external factors identified in this study included the customer not having experience of an Agile process or reliance on a number of outside stakeholders and contractors for integral parts of the project. Such challenges could have an impact on completing the task on time.

Challenges within organizational culture included a product owner having multiple roles. In this study two respondents reported that there were some issues with the multiple roles given to the product owner in some of the organizations. Sometimes these challenge were from the project owner's point of view rather than from the team's point of view. The project owner was conflicted by multiple roles and not being able to adequately prioritize these. This was reported by only two product owner respondents and this may or may not be common for all the projects. The Agile approach supported team work; however, there were some instances reported in this study where one member had to work alone because of the skill sets this person possessed. This meant that only one person had the skill set required for that project in the organization. This suggested that, in this particular case, they were not following Agile principles as Agile suggested having multi-functional teams during project development. There were three organizations where there was only one individual in a team. In each of these three cases the respondents stated that they followed an Agile approach. This suggested that some of the organizations may believe they have implemented an Agile process; however, there may have been difficulties in following this approach with only one team member. This suggested that organizations may think they were Agile but they were only using one or two Agile practices in parts of the process they want, or can (due to staff numbers), follow. It was difficult, with results such as these, to make generalizations about the Agile process, when it was not being followed correctly.

RQ3: What is the impact of challenges on the relationship between the customer and Agile software development teams?

Software development team respondents said that they had the "freedom to explore" solutions in the requirement gathering stage of a project. This may mean that they did not have to follow Agile approach to complete the task as it should be done. When the product owner assigned the list of priorities for the projects, the development team can have a discussion among themselves and then come up with the solution. This could also mean that the product owner had confidence in the team completing the given task efficiently. In the product owner study, some projects, where multiple teams worked together, had teams using different work approaches. The impact of this was on the distribution of the work. For example, one team may talk about the distribution of the work and the other may not. This was found to be a challenge because the team members had different ways of dealing with the problem and completing the task. Dahlin et al. (2005) reported that team members in a different team may have different abilities to identify solutions and make use of relevant information. When the members from different teams were brought together this increased the diversity and they then needed time to build a common understanding (Dahlin, Weingart, & Hinds, 2005). Such difficulties may cause delays in completing tasks as both teams needed time to understand the way each team worked to complete a task. The understanding between team members in the later stages may help in the coordination between the team members. Respondents also reported that the lack of availability of the product owner resulted in difficulties in communication and, because of this, there was a delay in decision-making about the project. Pikkarainen et al. (2008) found, during a case study on Agile software development teams, that having an open space in an office, daily meetings and iteration planning would improve communication. They also suggested that effective communication in a team required having a common understanding of the projects (Pikkarainen et al., 2008).

Communication in the team also depended on the interpersonal skills of team members. These skills included helpful suggestions, active listening and recognizing the interests and achievement of other members in a team. One respondent reflected on a project where management were involved in influencing the team. In this case, management would occasionally directly approach the team to ask for some changes to the project rather than

approaching the product owner. However, in this case, the development team did not agree with the management and referred them to the product owner. Such activities were not in line with Agile processes and this kind of culture in an organization may create more conflicts between the team and the product owner, and also with management. Such conflicts may impact on the relationship between the product owner and the Agile team. In a team, relationships were formed through interactions between team members (Chatman & Flynn, 2001).

Team relationships included the sharing of information, feedback, and recognizing the needs and feelings of team members. In some projects there were difficulties in obtaining feedback, team backup and interaction between the team members as a result. So, because of this, there were problems in the relationships in the team. This suggested that when there was problem in the relationship between the product owner and the software development team and this may have an impact on the project.

RQ4: Does the nature of the relationship between the customer and the development team impact on the project outcomes?

To understand the impact of challenges on project outcomes, linear dependencies between pairs of study variables were calculated using Pearson's Product Moment Correlation Coefficient. The results presented in section 4.4 suggested that that there was a positive linear relationship between the project completion time and disagreements with the customer about project priorities (r=0.329 n= 73 p = 0.004). This suggested, if there were an increase in disagreements with the product owner about the priorities then this was likely to increase the project completion time. The linear dependencies between the project completion time and interpersonal challenges between the team member(s) and the customer were positive (r=0.394 n =73 p =0.001). This suggested that when the interpersonal challenges between the software development team members increased the project completion time may also increase. In order to mitigate the implication of these relationships it may be necessary for organizations to create an environment that will help minimize interpersonal challenges.

A correlation coefficient was also calculated between the product owner's availability and disagreements between the development team members and the customer about the timeframe of the project. The data suggested a negative linear relationship between

product owner availability and disagreements with the customer about the timeframe of the project. (r=-0.348 n =73 p =0.001). This suggested that if there were an increase in the availability of the project owner there tended to be a decrease in disagreements between the development team members and the customer about the timeframe of the project. This finding was in line with others; for example, Turk et al. (2004) found that the availability of the customer was important in Agile software development in helping to maintain good communication with the team as it helped in tracking the project's status, inviting ideas and critiques and identifying the issues in the projects (Ramesh et al., 2006). The implications of this relationship in this study suggested that the availability of the product owner during project development was important for reducing disagreements between the product owner and the development teams.

A correlation was calculated with other factors, such as the availability of the product owner and timeframe of the project (r=-0.099). This showed that there was no significant relationship between availability of the product owner and the team members, even though there was an indication that these two factors could have a negative relationship. A correlation coefficient was also calculated between difficulties in sharing ideas within the team and project time and between Interpersonal challenges within the team, project time, and project budget. The data suggested that there were no significant relationship between these variables. Interpersonal challenges were expected to impact on the timeline and budget of the project. However, the data suggested that there were no significant relationships between these variables. Table 4-12 with correlation coefficient results is presented in Chapter 4: .

These findings go some way in addressing Research Question 4 for this study by showing that the nature of the relationship between the Agile software development team and customer had an impact on the project's outcome. The impact of the factors identified from this study of software development projects are summarized in Table 6-1

Table 6-1 Impact of factors on the project

Factors	Impact
Disagreement with the customer of project priorities	Project completion time
Product owner availability	Disagreement of the customer with the project's priorities
Interpersonal challenges between the team member(s) and customer	Project completion time
Disagreement between the team member(s) and customer about the project's requirements	Project completion time

6.2 Teamwork Model in Agile software development projects

This section presents a discussion using the Teamwork model to answer Research Question 5**:**

"Can the Teamwork Model adequately explain the nature of the relationship between the customer and the development team in an Agile development context?"

After analysis of the interview data from both software development team and the product owner study seven factors were identified from the product owner study and eight factors were identified from the software development team study. These factors were identified as challenges which may affect the success of Agile software development projects. Table 6-2 presents the factors from both the software development team and the product owner study. Table 6-3 presents the description of each factors identified in this current study.

Table 6-2 Themes identified from this study

Factors identified from the product owner study	Factors identified from the software development team study
Communication	Communication
Team support	Team support
Feedback	Feedback
Product owner engagement	Product owner engagement
Team engagement	Team engagement
Confidence	Confidence
Organizational culture	Organizational culture
	Team motivation

Table 6-3 Description of factors identified from this current study

Factors	Definition
Communication	Communication is the process of transferring information between individuals who are involved in a project.
Feedback	Feedback refers to providing information about the performance and acceptance of the suggestions given on the project's work.
Team support	Team support involves being available and willing to assist other members in the team. Development team members, including the product owner, work together to solve problems within the scope of the projects.
Organizational culture	A self-managing team with a culture of shared responsibility. Successful Agile development teams require product ownership, both from within the team and within the organization, that understands and supports the iterative nature of the Agile process.
Team engagement	Team member's activities during the projects. The activities of each of the team members are aligned in such a way that the targets associated with a project goal are achieved.
Product owner engagement	The product owner's role on the different activities of the project from starting of the project to deployment.
Confidence	Confidence, in the context of this study, refers to the belief on the competence of the team member in completing the task
Team motivation	Motivation, in the context of this study, refers to the development team members desire to complete the task

Dickinson and McIntyre (1997) constructed their Teamwork model with seven components: communication, team orientation, team leadership, monitoring, feedback, backup and coordination. They proposed that team leadership and team orientation impacted on monitoring, where this referred to the awareness of activities and the performance of team members. Monitoring was helpful in ensuring that team members were capable of their individual tasks and had a substantive understanding of the tasks of other members.

Dickinson and McIntyre (1997) suggested that monitoring a team's performance provided both feedback and timely backup behaviour. When the team members monitored each other they can provide feedback on each other's work. The team members will also be able to help other members in the team if they faced any difficulties as each member was aware of the other members' work through monitoring. In this study, team members were found not to be involved in monitoring each other's work. However, three product owner respondents reported that they were involved in monitoring the team's performance. This suggested that monitoring was not a common approach used observe the work of team members and provide the feedback on their work. Table 6-4 presents the factors along with the description from the Dickinson and McIntyre Teamwork Model (1997).

A mapping exercise was undertaken to identify common factors from the Teamwork model, the software development team study and the product owners' study. Figure 6-1 presents the mapping of the Teamwork model with this current study. The description of each of these factors in relation to the Teamwork model is described in the following section

Table 6-4 Components of the Teamwork model

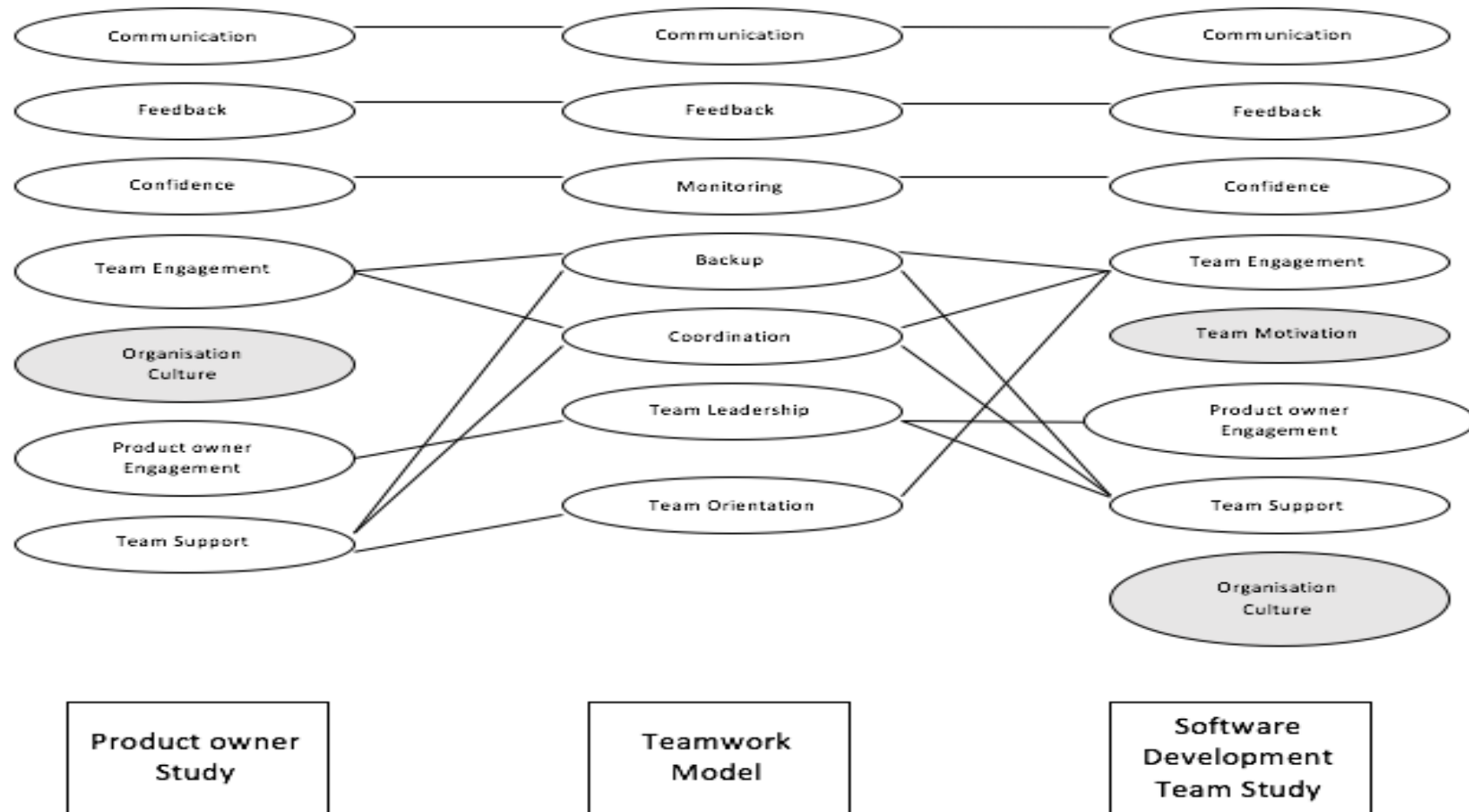|	
Communication	Communication involves the active exchange of information between the members of the team. Communication creates the link between the other components of the teamwork model.
Team orientation	This component includes team members' self-awareness and attitude towards the leadership.
Team leadership	This component of a teamwork model consists of direction by the team members and implies the planning and organizing of activities to enable team members to respond.
Monitoring	This component of the Teamwork model refers to the awareness of activities and the performance of the team members. Monitoring is helpful in ensuring that team members are capable of their individual tasks and to have a substantive understanding of the task of other members.
Feedback:	This component refers to learning from the team member's performance, which requires receiving, giving and seeking feedback from other members working in a team.
Back up	This component refers to a team member helping other team members to complete a task.
Coordination	This component refers to the activities that will help the team member to complete the functions (Dickinson & McIntyre, 1997).

Figure 6-1 Mapping the Teamwork model with results from the software development team and product owner studies

6.2.1   Team support

Team support was one of the important factors in both software development team and product owner studies. The team support theme from this study related to the willingness of team members to help each other to complete a task. Both the product owner and the software development team respondents reported that team support was demonstrated through shared leadership, collaboration, and having a helping attitude towards the other team members. However, team members may not necessarily be observing the work of other team member for the support they can provide to the other team members. This meant the team members may take a long time to understand what other members were doing and before supporting them. Another impact of this was that there was little evidence of backup (Team support) between the team and the product owner as the team members were not aware of the other members' work. For example, team members were, in fact, willing to provide support to other team members when needed. There were some projects where team members were not willing to get help from other members of the team as these members liked to work on their own. This suggested that team support overlapped with the team orientation, backup and coordination constructs of Dickinson and McIntyre's (1997) Teamwork Model. Team support was important for effective teamwork. Sheng et al. (2010) mentioned that team support was found to have significantly influenced teamwork behaviour and trust in the team. It was also found to have influenced the attitudes and behaviour of the team members within the team, including their commitment (Sheng, Tian, & Chen, 2010). This suggested that when an individual cannot accomplish the assigned tasks individually then they can request support from their colleagues to complete the task.

6.2.2   Feedback

Thirteen respondents from the software development team study and three respondents from the product owner study reported that team members were involved in giving and receiving information about the project. The Teamwork model suggested that feedback within a team can provide input about coordination in the team. This suggested the importance of feedback for effective teamwork. When the team members felt that their work and they, themselves, were important and valued for the project, this introduced positive emotions that encouraged the team to complete the given tasks (Amabile & Kramer, 2007). Four team members reported that the product owner encouraged them to provide feedback about the projects by listening to the feedback a team member gave.

Katzenbach & Smith (2005) found that feedback from team members worked because it promoted individual, as well as team performance. Respondents for this study stated that acceptance of the feedback was dependent on the experience of the team members providing the feedback. They also reported that there were some instances where the product owner accepted the feedback but changed that decision later. This suggested that product owner was not confident about some of the requirements for the projects. One product owner respondent said that they never received feedback from the end users of their product because they had no contact with them. This may mean that feedback from the user was not received, the developers were not provided with exactly what was required or the product owner was not sure what exactly was required. This lack of feedback may hinder improvements in the product if it was required. In saying this, though, it was difficult to know exactly how this feedback could be received. This was not the case in the other projects in this study where the product owner had access to the end users. This suggested that the feedback process on the project was dependent on factors, such as access to the end user and acceptance of feedback by the product owner. This also suggested that when team members have access to end users they will be able to receive feedback for the project as it was being developed.

Due to the fact that product owner did not monitor the team in most of the projects there was a little feedback from the product owner about the team members' performance. This suggested that that when the product owner was aware of the team members' work then feedback can be provided when needed. In such organizations where there was little or no feedback, team members were working on their own work and were not fully engaged in teamwork. Some teams did provide feedback to the product owner but the acceptance of the feedback by the product owner depended on the experience of the team member who provided the feedback. This suggested that the product owner was willing to listen to the experienced members rather than the inexperienced members. Ramesh et al. (2006) found that senior members, who had gained the trust of the product owner from earlier working relationships, were listened to by the product owner. This also suggested that there were some organizations where the level of product owner engagement needed to be improved and the product owner needed to believe in their team to complete the task. Belief in the team may increase with good performance and by completing the tasks on time. When the

team were aware of each other's work through monitoring then they can understand each other's tasks, what they are working on and what were they trying to solve. Information from the twenty-two respondents from both the software development and the product owner's study suggested that team members were not involved in observing other team members' work. This may mean that the team members may have focused on their own work and worked as individuals. Langfred (2000) stated that there can be negative effects on team performance when teams were trying to function as self-managing teams while were working as individuals (Langfred, 2000). When individual autonomy was low there was an increase in confidence and this resulted in better performance in the team (Langfred, 2004). Marks et al. (2001) found that that team members needed to be aware of each other's work in order to provide assistance to the team. This suggested that a team, including the product owner, needed to work closely with each other so they were aware of each other's work and can provide support and feedback on their work and performance. In this study, this was found not to be a common practice in all projects

In the Dickinson and McIntyre Teamwork Model (1997) the constructs of feedback and team backup were noted as being important for the coordination of team members. This was important because when team members knew about the other team members' work and can provide feedback and backup support there was more likely to be a good connection between the team members that will facilitate good coordination. Respondents for this study reported that the Agile software development team's involvement in a project differed between organizations. In sixty-four per cent of the projects, software development team members were involved in breaking down the user stories to detail a work plan. Thirty-six per cent of the software development study participants were in teams that were assigned tasks rather than being involved in the planning. The division of work was based on the skills team members had, which provided team members to do the work in an effective way. In four projects described by respondents in this study the team members were directed to work as individuals. This individual way of working, even though they belonged to the team, created difficulties in the teams concerned with respect to coordination and following the Agile process. The problems observed were that the team members were not aware of the other team members when they were working on their own. In these situations, the team members may not have been aware of the progress being made with a

particular task or of any difficulties occurring during the process of completing that task. The implication of this for successful outcomes is where team members can give the feedback and work as a group. Three respondents from the product owner study reported that the team did not have any say on the priorities of the projects. This may mean that the product owner was not used to working with Agile processes. The implications of this on the outcome of a project may be that the product owner needed to be following the correct Agile approach and coordinating with the team.

6.2.3   Product owner engagement

Product owner engagement was reported to be another factor in Agile software from both the software development and the product owner study. Engagement by a product owner here referred to the participation of the product owner during the different activities of a project, from the beginning to the release stage. From this study, 19 of 29 (66%) of respondents reported that the product owner was engaged with the team in refining the requirements, backlog planning and clarifying aspects not understood by a team member. There were three organizations where product owners were not taking part in such activities. It could be that they were not following the Agile process or their role was to look over all the projects as well as the development team, and then approach the team if they were issues. The engagement of the product owner in the project was crucial in Agile software development (Beck et al., 2001a). For example, Moe et al. (2008) found that the involvement of the team members, including the product owner in a project improved the emotional attachment to the organization, the motivation to perform better and a desire to take responsibility. This could mean that when the product owner was engaged in the project the team members felt better in completing the tasks as they can received feedback and direction quickly.

Not all product owners involved in this study had full authority to make decisions about the projects they were assigned to; senior management in some organizations were, at times, involved in making such decisions. In these cases, the lack of authority may create confusion within the teams and increase the chance of missing deadlines. Moe et al. (2012) found that the authority given to a product owner will help in making decisions, such as which backlog items should be developed in the following sprint, how the problem was to be solved and where to allocate the resources. When the product owner did not have full authority on a

project it took longer for decision-making during the project's development. In some projects in this current study, the availability of the product owner was an issue. When the product owner was unavailable there were delays in the completion of the project's tasks. This was because team had to wait to hear back from the product owner and this could increase the time to complete the task. At times, this could create frustration in the teams.

Three respondents from the product owners reported that the product owner was given multiple roles on several occasions and this was reported as one of their challenges. For example, in one case, a respondent was fulfilling their normal role of software developer in the team while they were also acting as product owner. This meant that the product owner, in this particular case, was involved in a project with multiple roles. When the product owner has multiple roles in a project they can face difficulties while looking at the project from both the team members' and product owners' perspectives. This may cause conflict within their roles. (Mahnic, 2012) found that it was advisable to separate the multiple roles of the product owner, so the person can focus on the role outside his usual commitments. The findings from both software development and product owner studies suggested that the product owner engagement had some overlap with the coordination, team orientation from the Dickinson and McIntyre Teamwork Model (1997) as the product owner was involved in activities that will help team members complete their functions. The product owners were also involved in giving direction to team members for planning and organizing activities to enable them to complete their tasks.

The implications on the level of engagement of the product owner can have two impacts, one positive, one negative. The positive effect on the team was that there may be better communication as the product owner can be available when the team needed them. If the team has any concerns, such as the timeline of the given sprint or the complexity of the work, then they can approach the product owner. On the negative side the effect from the product owner being less involved with the team could delay the communication process with the team. Also, the product owner may not be aware of what was going on with the project in detail.

6.2.4 Team engagement

It was interesting to note that the Dickinson and McIntyre Teamwork Model (1997) did not contain team engagement as a separate component. A team was often found to be engaged in activities such as stand-ups and retrospectives in Agile software development (Drury et al., 2012). These activities required a high level of engagement The level of involvement of the team members may have had an impact on communication and collaboration and the feedback component of the teamwork model. This suggested an overlap between the team engagement theme from this study and the team orientation, backup and coordination constructs from the Dickinson and McIntyre Teamwork Model (1997). As team members were involved in activities that will help the team to complete their functions there must be a level of coordination and engagement. Team members were also involved in planning and organizing activities to complete the tasks. It was interesting that in this study the team engagement theme was strong and the people involved in the software development spoke about its importance in completing the project as a team. The positive impact from team engagement on different Agile practices during the software development projects helped the team become aware of the progress being made in the projects. This also provided opportunities for the team members to talk about the projects in detail if they had any issues. If the team members were not fully engaged in the projects then there may be difficulties in communication and coordination in the team.

6.2.5 Confidence

In this section, two terms used were confidence and trust. Team confidence, in the context of this study, referred to the belief that the product owner has in the team member in completing the task. Trust referred to the ability of the team members to complete the tasks in the correct way. Trust also referred to the dependence on the character, strength and ability of the team member in completing the task (Dingsøyr et al., 2012b; Gefen & Straub, 2004). Seven software development team members involved in this study reported that the product owners had trust in the team because the product owner listened to their concerns In saying that, there were two respondents from the software development team study who reported that confidence in the team was dependent on the team members' experience. In saying that, the perception within the team was that more junior team members were not trusted as much by the product owner as the more senior team members. This may mean that the product owner had more confidence in team members

who have more experience working in a software development project and as they have demonstrated working relationships in previous projects. Whereas, they did not know as much about the more junior members and their abilities. There were projects where the product owner closely observed the teamwork because they were not confident in the team's ability to complete their tasks. It could be that the team members have not yet demonstrated their work or are new to the working relationship with the product owner. The implication of this level of team monitoring could presumably have two effects, one negative, one positive. The negative effect being that the teams grew to resent the perceived level of micro managing they were receiving and became less happy about working with that particular product owner. The reverse side of that was a positive effect with the product owners getting to know their teams at a much deeper level, being able to observe what they were capable of and, therefore, giving them more trust in the future.

Five product owner respondents reported that they trusted their team to complete their given tasks. However, there were two product owner respondents who reported that they did not trust their team as they thought the team had not estimated the tasks correctly. This could be because the product owner may have seen the impact from their estimations of the work in other projects, which may not have meet the expectations of the product owner. A lack of confidence in the team may increase the timeline of the project as there may be more time required to observe the work of an individual or the team. To achieve trust, there needed to be shared leadership, feedback and communication, with team members being allowed to demonstrate what they can do and also given the opportunity to extend their skills (Salas, Sims, & Burke, 2005). They also mentioned that trust can be achieved when the team members were willing to admit mistakes, acknowledge the message received and clarify the messages received. Previous research by Langfred (2004) found that trust in the team was associated with better team performance. However, they also mentioned that having too much trust in a self-managing team may be harmful as the team's individual autonomy will be high. This suggested that there needed to be some monitoring of the team to maintain individual autonomy. It also suggested that trust was important for effective teamwork and the development of successful projects. Confidence was found to have some overlap with the monitoring construct from the Dickinson and McIntyre Teamwork Model (Dickinson & McIntyre, 1997). This overlap was found in

situations where a product owner reported an awareness of the activities and performance of the team members along with having confidence in the team to complete a task.

According to the Dickinson and McIntyre Teamwork Model (Dickinson & McIntyre, 1997), monitoring was helpful in ensuring that team members were capable of their individual tasks and had a substantive understanding of the tasks of the other members. Monitoring may provide an opportunity to a product owner to see the progress of the team member. When the product owner was aware of the team members' work and progress that may help the team members gain the confidence of the product owner. Only one respondent from the product owner study mentioned that the product owner was involved in monitoring the team members' performance because the team product owner did not have confidence in the team. This suggested that monitoring was not a common practice used in Agile software development projects. The positive aspect of this could be that the product owners had confidence in the teams. When the product owner had confidence in the team the team became motivated. However, the negative impact of not monitoring the team may result in that the team not meeting the expectations of the product owner and there may be some rework required on the project that could delay project completion.

6.2.6 Communication

Communication was the most (64%) reported challenges faced both by the software development and product owners in the projects. Challenges, such as disagreement about the project requirements and availability of the product owner, were found to have an impact on the projects' timelines. Both studies reported that communications were achieved using different media, such as emails, using some application or face-to face. Both studies also reported that challenges can occur during communication. Such challenges were miscommunication because of differences in language and culture. A study undertaken by (Matveev & Nelson, 2004) with 124 upper and middle managers found that there was a relationship between cross cultural communication and team performance. During project development, the development team and the product owner may have had some informal discussions about the projects. This discussion could be on the features of the projects, resources allocation or changes to the projects. Some of these informal discussions can be missed, as they were not documented. This suggested that the information shared during informal communication was forgotten or it could be that the

important discussion was not made at the correct place, such as, during stand-ups. Dickinson and McIntyre (1997) said that communication was involved in teamwork during monitoring, feedback, back-up, team leadership, team orientation and team coordination. They also mentioned that it was important to have effective communication during the process for better teamwork. In the Dickinson and McIntyre Teamwork Model (1997), communication was the main component that linked the teamwork processes together. It had been reported in this study that the daily stand-ups were the most commonly used mechanism for communication in Agile projects (Paasivaara, Durasiewicz, & Lassenius, 2008). This was because during the stand-ups all team members came together and shared information. This included the progress of the project, difficulties in the project and other information about the project. During software development, team members may be working from different places. There may be challenges in communication with working between different time zones, in different languages and the lack of availability of the product owner. Problems with time zone differences could be minimized by finding a common time when all the people were available. If this problem persisted then there was a problem in communication. Problems in communication could result difficulties in coordination, feedback and backup behaviour, according to the Teamwork model. This suggested that communication was important in bringing all the components of the Teamwork model together to achieve effective teamwork in Agile software teams. In the above section, different factors, identified from both the studies, were discussed in the context of Dickinson and McIntyre's model (1997). There was some overlap with the themes identified from this study, such as team confidence, product owner and team engagement. However, the Teamwork model did not model organizational culture or team motivation. These two additional factors identified from this study were not identified in a previous study by Moe et al (2010). In their study, they suggested that an extension to the Teamwork model would help in allowing a mature Agile software development team achieve a better understanding of the teamwork. Each of the factors identified in this study, but not present in the Teamwork model, will be discussed now in detail.

6.2.7   Team motivation

Eleven respondents reported that team members were motivated by the success of the team and the project's success. Team members were also motivated because they felt they were being sufficiently rewarded in their salary for their involvement in projects. Another

important factor that motivated people was the organization's social culture. In organizations where there were regular staff events, motivational levels were higher than in organizations that did not have a social culture. Two of the twenty-two participants reported that the some of the management's activities, such as not listening to team concerns, were impacting on the team members' motivation. For example, one respondent said that their product owner did not listen to the feedback and, because of this, they were not motivated to engage in further communication with the product owner about any concerns that arose during project development. A study carried out by Moe et al. (2012) mentioned that increased motivation in the team members' resulted in employees having more care about their work and it encouraged greater creativity, helping behaviour and higher productivity. Another study undertaken by Ceschi et al.(2005) mentioned that continuous training and regular communication improved teamwork ability and motivation (Ceschi et al., 2005). This suggested that team motivation in Agile software development could be an important component of the Teamwork model for ensuring effective teamwork. Team motivation also helped to achieve the goals with good quality results (Turk et al., 2014a).

6.2.8   Organizational culture

Another additional construct in the Teamwork model identified from this study was organizational culture. In Agile software development the organizational culture should be encouraging self-managing teams and shared responsibility (Dybå & Dingsøyr, 2008). Good organizational culture enabled people to organize their work in a way that gave the development team common ownership and a shared responsibility for the project (Dingsøyr et al., 2012a). Good organizational culture also provided an environment for effective teamwork in Agile teams. Goodman, Zammuto & Gifford (2001) found that the culture of an organization can influence the quality of work life as it had an influence on individual attitudes, commitment, motivation and moral satisfaction, as these human factors impact on teamwork. A study by Leonard et al.(2004) on a team mentioned that an organizational culture where individuals were motivated to complete tasks, and where they can speak up and express concerns in the team, made a positive impact on the teamwork (Leonard, Graham, & Bonacum, 2004). This suggested that organizational culture could have an impact on the relationship between the Agile software development teams and the product owner.

When the teamwork model was mapped with the findings from this study two factors, communication and feedback, mapped strongly to those found in this study. The mapping of common factors from software development team study, the product owner study and the Dickinson and McIntyre Teamwork Model (1997) is shown in Figure 25

Factors such as team confidence, team involvement, product owner engagement and team support have similar descriptions with monitoring, back-up, coordination, team leadership and team orientation. Two factors, team motivation and organizational culture, did not have sufficient similarities to map with factors from the Teamwork model. Hence, these two factors were proposed to be additional constructs in a revised team work model.

6.3   Summary

In this chapter, the findings from both the software development team study and the product owner study were brought together and discussed. Each of the research questions on which this study was based were answered using the findings from this study. The (Dickinson & McIntyre, 1997) Teamwork Model was used to explain the relationships in Agile software development teams, and between the teams and the product owner. Mapping was carried out with the Teamwork model and this study. Two additional constructs of the Teamwork model were identified from this study. The following chapter presents a summary of the research undertaken, research questions addressed, findings and contribution.

Chapter 7:   Conclusions and Future Work

7.1   Introduction

Many software projects are undertaken using an Agile approach. Agile is an increasingly common approach to software development reported to have a better success rate than a plan driven approach. Agile software development includes a much stronger relationship with the customer, often in the role of product owner, than the plan driven approach.

When reviewing the literature, success factors, failure factors, teamwork, relationships in agile software development, were explored. It became clear that the impact of the relationship between the product owner and software development team on the outcome of software development projects had not been previously studied. This may be because the impact of the relationship between the product owner and the software development team on the project's outcome was often overlooked, or neglected, when software development projects were planned. Factors such as communication and coordination played important roles in terms of establishing a good working relationship between software development teams and the product owner. This thesis explores this idea, while focusing on the Agile software development approach. The goal of this research was to Identify the challenges in Agile software development and their impacts on the relationships between the product owner and the software development team. This research also aims to understand the impacts of relationships on a project's outcome, and to identify a mean to address the nature of the relationships to help improve the project's outcome. The Teamwork model was used to explain the relationship between the product owner and the software development team. The research questions for this research, as presented in section 2.9.1, are:

RQ1: The literature suggests that there are challenges between the software development team and the customer representative in Agile software development projects. To what extent do these challenges exist?

RQ2: What types of challenges are present in Agile software development projects?

RQ3: What is the impact of challenges on the relationship between the customer and the Agile software development team?

RQ4: Does the nature of the relationship between the customer and the development team impact on the project outcome?

RQ5: Can the Teamwork model adequately explain the nature of the relationship between the customer and the development team in an Agile development context?

A two study mixed methods research design was adopted. One study collected the experiences and perceptions of software development team members through a questionnaire and with a sub-sample of the members interviewed. A further study collected the experiences and perceptions of product owners through a short questionnaire and in-depth interviews.

The data were analysed using descriptive and inferential statistical methods, and thematic analysis methods, as reported in section 3.2.

7.2   Research findings

RQ1: The literature suggests that there are challenges between the software development team and the customer representative in Agile software development projects. To what extent do these challenges exist?

Both the software development team and the product owners' study confirmed that challenges in communication, interpersonal challenges, project priorities, the sharing of ideas and project completion time, were able to be observed in Agile software development projects. The extent of such challenges are presented in **Table 4-9**.

RQ2: What types of challenges are present in Agile software development projects?

During Agile software development there was involvement between team members and the product owner. This study confirmed there were challenges during the software development process. The challenges identified from this study are:

- Communication
- Availability of the product owner
- Team motivation
- Organizational culture
- Interpersonal challenges
- Disagreements with the customer about project priorities and requirements
- Team motivation

- Team support

RQ3: What is the impact of challenges on the relationship between the customer and the Agile software development team?

The findings from this current study suggested that challenges in communication, team engagement, product owner engagement, confidence and team motivation, have an impact on the relationships between the software development team and the product owner. This research has found in some projects there were difficulties getting feedback, team backups and interactions between the team members, so there was a problem in the relationships in the team. This is explained in detail in section 6.1.

RQ4: Does the nature of the relationship between the customer and the development team impact on the project outcome?

The results presented in section 4.4 show the impact of the relationship between the product owner and the software development teams. These findings go some way in addressing Research Question 4 in this study by showing that the nature of the relationships between the Agile software development team and the customer have an impact on the project's outcome. For example, the results from this study suggested that there was a positive, linear relationship between the project's completion time and disagreements with the customer about project priorities ($r=0.329$ $n= 73$ $p = 0.004$). This suggested that if there were increased disagreements with the product owner about priorities then this was likely to increase the project completion time.

RQ5: Can the Teamwork model adequately explain the nature of the relationship between the customer and the development team in an Agile development context?

The relationships in the Agile software development team were explored using the Dickinson and McIntyre Teamwork Model (1997). This model was chosen because the constructs of this model were found to be relevant to the Agile software development team. In doing so, this research uncovered factors not previously thought to be relevant in this type of study and tested the robustness of the Dickinson and McIntyre Teamwork Model (1997)

When mapping the findings from this study and the Teamwork model, there was some overlap with the factors identified from this study, such as team confidence, and product

owner and team engagement. However, the Teamwork model did not model any organizational culture or team motivation. Results from this study suggested that the Teamwork model can explain the nature of the relationships. However, for effective teamwork additional constructs, as identified from this study, were proposed to be included in the current Teamwork model. This is explained in section 6.2.

7.3 Contribution of this research

This study found an impact on project outcome based on the relationship between the product owner and the software development team. This had not previously been reported, and, specifically, answered Research Question 4 (RQ4).

This research used the Dickinson and McIntyre Teamwork Model (1997), which has been used to study Agile software development teams in previous studies. This research identified two additional constructs for the Teamwork model: team motivation and organizational culture. These additional constructs were found to be important for effective teamwork in Agile software development teams. Team motivation impacted on the coordination, communication and team support in the project. Team motivation from working as a team resulted in employees taking more care of their work and increased greater interaction within the team to help behaviour and produce higher productivity. Team performance not only depended on the competence of the team alone, but also on organizational culture and the context. Good organizational culture enabled people to organize their work in ways that gave the development team common ownership and a shared responsibility of the project. Hence, organizational culture had an overall impact on the relationship between the software development team and the customer in achieving effective teamwork. The proposed teamwork model from this study can viewed in Figure 7-1.

A product owner can be from the client's organization or internal to the software development organization (in this case they are called proxy product owners) (Kettunen, 2009). In this study, most of the respondents from the software development team (65; 89%) mentioned that they had proxy product owners in their projects. This study confirmed that the proxy product owner had a stake in guiding the end deliverables of the projects either by providing input, coordinating with the development teams, or reviewing and

approving the product. This study has also confirmed that there were relationship challenges in Agile software development. These challenges included challenges in communication, project priorities, project requirements and interpersonal challenges. In a previous study on Agile software development, interpersonal challenges were found to have had an impact on a project's outcome (Liu et al., 2011). However, this study could not confirm a significant relationship between interpersonal challenges and the project's outcome.
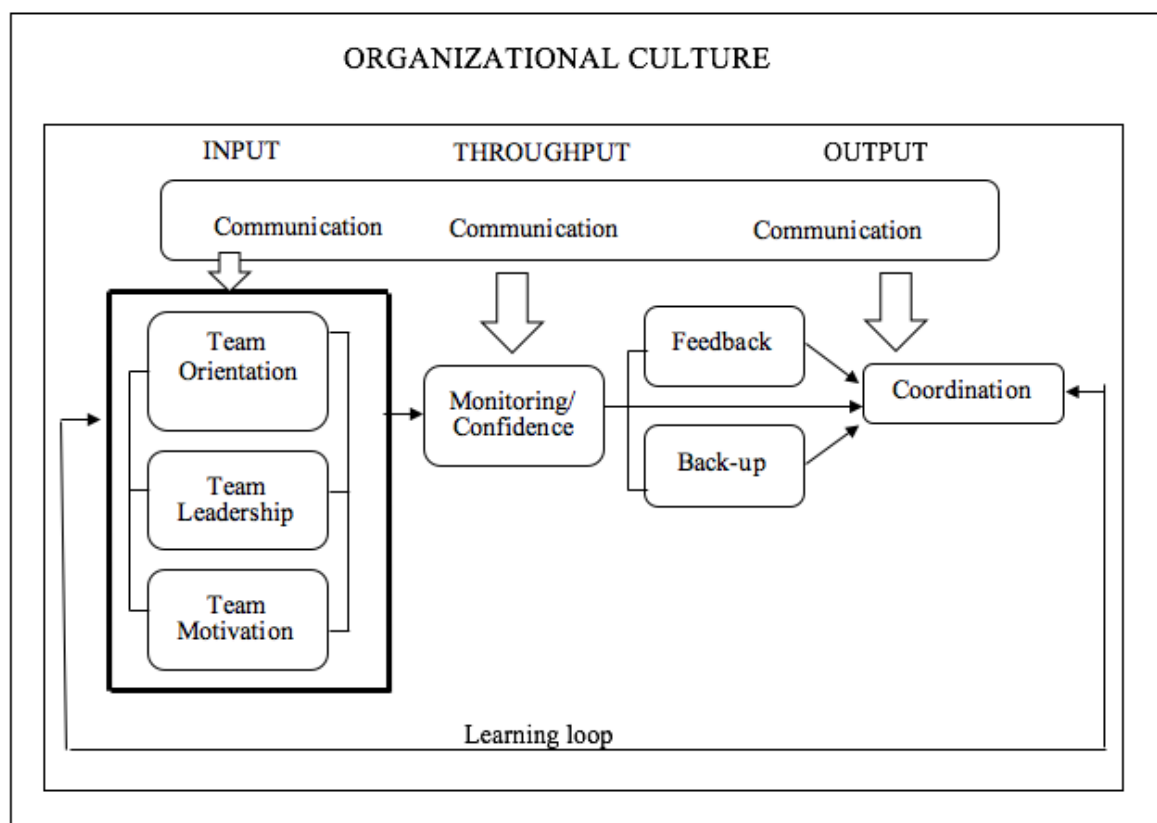
Figure 7-1 Proposed Teamwork Model

7.4   Limitations of the study

The limitations of this study included those common to quantitative and qualitative research. First, the widespread location of professional people in software development made it difficult to establish the location of companies using an Agile software development approach. Second, there were difficulties in locating the actual customers who were involved during the Agile software development projects. Proxy product owners were customer representatives in many Agile software developments so this may mean that the

data collected from the proxy product owners may not reflect the information of the client organization itself.

Results from this study may not be representative of all the environments, we considered Agile software approach used by a software development team who are collocated. Hence the results could be different with an open source software and projects with virtual or remote software development teams.

7.5 Future Work

The results of this study presented a number of directions for future work. First, this study highlighted several challenges that existed in Agile software development teams. Accordingly, future work should focus on identifying challenges when there was an actual customer taking the role of product owner.

Secondly, future work would be useful to confirm the effectiveness of the Teamwork model proposed in this thesis. Future work can also be carried out to confirm the additional constructs identified from this study.

7.6 Summary

The literature has recognized the importance of team relationships in Agile software development. Using an Agile approach for software development was seen as a way to improve the likelihood of successful outcomes in software development projects. A number of factors were found to impact on the relationship between the product owner and the software development team and these were found have an impact on the outcome of Agile projects.

This study provided a framework to assist practitioners and researchers to understand and manage effective teamwork in Agile software development projects. This research investigation has summarized the positive and negative impacts of different factors, such as communication, feedback, team support, confidence in implementing an agile software development approach, and proposed a theoretical framework for effective teamwork. This research also helped the software development organization to understand the importance of team motivation and organizational culture in Agile software development teams.

References:

Alchian, A. A., & Demsetz, H. (1972). Production, information costs, and economic organization. *The American Economic Review, 62*(5), 777-795.

Alliance, A. (2001). Agile alliance. *Internet URL.'' www. agilealliance. com*.

Amabile, T. M., & Kramer, S. J. (2007). Inner work life. *Harvard Business Review, 85*(5), 72-83.

Anderson, D. J. (2003). *Agile management for software engineering: Applying the theory of constraints for business results*: Prentice Hall Professional.

Arksey, H., & Knight, P. T. (1999). *Interviewing for social scientists: An introductory resource with examples*: Sage.

Atkinson, R., & Flint, J. (2001). Accessing hidden and hard-to-reach populations: Snowball research strategies. *Social research update, 33*(1), 1-4.

Aubert, B. A., Patry, M., & Rivard, S. (2005). A framework for information technology outsourcing risk management. *ACM SIGMIS Database, 36*(4), 9-28.

Awad, M. (2005). A comparison between agile and traditional software development methodologies. *University of Western Australia*.

Axelsson, K., Melin, U., & Lindgren, I. (2013). Public e-services for agency efficiency and citizen benefit — Findings from a stakeholder centered analysis. *Government Information Quarterly, 30*(1), 10-22. doi:10.1016/j.giq.2012.08.002

Bahli, B., & Rivard, S. (2003). The information technology outsourcing risk: a transaction cost and agency theory-based perspective. *Journal of Information Technology, 18*(3), 211-221.

Balijepally, V., Mahapatra, R., & Nerur, S. P. (2006). Assessing personality profiles of software developers in agile development teams. *Communications of the Association for Information Systems, 18*(1), 4.

Bass, J. M. (2015). How product owner teams scale agile methods to large distributed enterprises. *Empirical Software Engineering, 20*(6), 1525-1557.

Batra, D., Xia, W., VanderMeer, D., & Dutta, K. (2010). Balancing agile and structured development approaches to successfully manage large distributed software projects: A case study from the cruise line industry. *Communications of the Association for Information Systems, 27*(1), 21.

Beck, K. (2000). *Extreme programming explained: embrace change*: Addison-Wesley Professional.

Beck, K., & Andres, C. (2004). *Extreme programming explained: embrace change*: Addison-Wesley Professional.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Jeffries, R. (2001a). The agile manifesto.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Jeffries, R. (2001b). Manifesto for agile software development.

Begel, A., & Nagappan, N. (2007a). Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study. 255-264. doi:10.1109/esem.2007.12

Begel, A., & Nagappan, N. (2007b). *Usage and perceptions of agile software development in an industrial context: An exploratory study.* Paper presented at the Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on.

Bell, D. E., Raiffa, H., & Tversky, A. (1988). *Decision making: Descriptive, normative, and prescriptive interactions*: Cambridge University Press.

Bengtsson, M., & Kock, S. (2014). Coopetition—Quo vadis? Past accomplishments and future challenges. *Industrial Marketing Management, 43*(2), 180-188.

Bernard, H. R. (2011). *Research methods in anthropology: Qualitative and quantitative approaches*: Rowman Altamira.

Berscheid, E. (1999). The greening of relationship science. *American Psychologist, 54*(4), 260.

Biehl, J. T., Czerwinski, M., Smith, G., & Robertson, G. G. (2007). *FASTDash: a visual dashboard for fostering awareness in software teams.* Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems.

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

Boehm, B., & Turner, R. (2003a). *Balancing agility and discipline: A guide for the perplexed*: Addison-Wesley Professional.

Boehm, B., & Turner, R. (2003b). People factors in software management: lessons from comparing agile and plan-driven methods. *Crosstalk-The Journal of Defense Software Engineering,(Dec 2003)*.

Boehm, B., & Turner, R. (2003c). Using risk to balance agile and plan-driven methods. *Computer, 36*(6), 57-66.

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *Software, IEEE, 22*(5), 30-39.

Booch, G. (1986). Object-oriented development. *Software Engineering, IEEE Transactions on*(2), 211-221.

Boudreau, M.-C., Gefen, D., & Straub, D. W. (2001). Validation in information systems research: a state-of-the-art assessment. *MIS Quarterly*, 1-16.

Brannick, M. T., Salas, E., & Prince, C. W. (1997). *Team performance assessment and measurement: Theory, methods, and applications*: Psychology Press.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77-101.

Brown, A. D., & Starkey, K. (1994). The effect of organizational culture on communication and information. *Journal of Management studies, 31*(6), 807-828.

Bryman, A. (2012). *Social research methods*: Oxford university press.

Bustard, D., Wilkie, G., & Greer, D. (2013). *The maturation of agile software development principles and practice: Observations on successive industrial studies in 2010 and 2012.* Paper presented at the Engineering of Computer Based Systems (ECBS), 2013 20th IEEE International Conference and Workshops on the.

Cao, L., & Ramesh, B. (2008). Agile requirements engineering practices: An empirical study. *Software, IEEE, 25*(1), 60-67.

Ceschi, M., Sillitti, A., Succi, G., & De Panfilis, S. (2005). Project management in plan-based and agile companies. *Software, IEEE, 22*(3), 21-27.

Chatman, J. A., & Flynn, F. J. (2001). The influence of demographic heterogeneity on the emergence and consequences of cooperative norms in work teams. *Academy of management journal, 44*(5), 956-974.

Chen, I. J., & Popovich, K. (2003). Understanding customer relationship management (CRM) People, process and technology. *Business process management journal, 9*(5), 672-688.

Chow, T., & Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software, 81*(6), 961-971. doi:10.1016/j.jss.2007.08.020

Cochran, W. G. (2007). *Sampling techniques*: John Wiley & Sons.

Cockburn, A. (2006). *Agile software development: the cooperative game*: Pearson Education.

Cockburn, A., & Highsmith, J. (2001a). Agile software development, the people factor. *Computer, 34*(11), 131-133.

Cockburn, A., & Highsmith, J. (2001b). Agile software development: The people factor. *Computer*(11), 131-133.

Cohen, D., & Crabtree, B. (2006). Qualitative research guidelines project.

Cohen, S. G., & Bailey, D. E. (1997). What makes teams work: Group effectiveness research from the shop floor to the executive suite. *Journal of management, 23*(3), 239-290.

Cohn, M. (2004). *User stories applied: For agile software development*: Addison-Wesley Professional.

Conboy, K., Coyle, S., Wang, X., & Pikkarainen, M. (2010). People over process: key people challenges in agile development. *IEEE software, 99*(1), 47-57.

Conboy, K., & Fitzgerald, B. (2010). Method and developer characteristics for effective agile method tailoring: A study of XP expert opinion. *ACM Transactions on Software Engineering and Methodology (TOSEM), 20*(1), 2.

Coram, M., & Bohner, S. (2005). *The impact of agile methods on software project management.* Paper presented at the Engineering of Computer-Based Systems, 2005. ECBS'05. 12th IEEE International Conference and Workshops on the.

Cramton, C. D., & Webber, S. S. (2005). Relationships among geographic dispersion, team processes, and effectiveness in software development work teams. *Journal of Business Research, 58*(6), 758-765.

Creswell, J. (2011). Controversies in mixed methods research. *The Sage handbook of qualitative research, 4*, 269-284.

Creswell, J. W. (2013). *Research design: Qualitative, quantitative, and mixed methods approaches*: Sage publications.

Cuevas-Rodríguez, G., Gomez-Mejia, L. R., & Wiseman, R. M. (2012). Has Agency Theory Run its Course?: Making the Theory more Flexible to Inform the Management of Reward Systems. *Corporate Governance: An International Review, 20*(6), 526-546. doi:10.1111/corg.12004

Daft, R. L., Lengel, R. H., & Trevino, L. K. (1987). Message equivocality, media selection, and manager performance: Implications for information systems. *MIS Quarterly*, 355-366.

Dahlin, K. B., Weingart, L. R., & Hinds, P. J. (2005). Team diversity and information use. *Academy of management journal, 48*(6), 1107-1123.

Davies, D., & Dodd, J. (2002). Qualitative research and the question of rigor. *Qualitative health research, 12*(2), 279-289.

Davis, J. H., Schoorman, F. D., & Donaldson, L. (1997). Toward a stewardship theory of management. *Academy of management review, 22*(1), 20-47.

De Dreu, C. K., & Weingart, L. R. (2003). Task versus relationship conflict, team performance, and team member satisfaction: a meta-analysis. *Journal of applied Psychology, 88*(4), 741.

DeChurch, L. A., & Mesmer-Magnus, J. R. (2010). The cognitive underpinnings of effective teamwork: a meta-analysis: American Psychological Association.

Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2010). The scrum primer. *Scrum Primer is an in-depth introduction to the theory and practice of Scrum, albeit primarily from a software development perspective, available at: http://assets. scrumtraininginstitute. com/downloads/1/scrumprimer121. pdf, 1285931497*, 15.

DiCicco-Bloom, B., & Crabtree, B. F. (2006). The qualitative research interview. *Medical education, 40*(4), 314-321.

Dickinson, T. L., & McIntyre, R. M. (1997). A conceptual framework for teamwork measurement. *Team performance assessment and measurement*, 19-43.

Dillon, S. M. (1998). Descriptive decision making: Comparing theory with practice. *Unpublished manuscript, Department of Management Systems, University of Waikato, New Zealand.*

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012a). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software, 85*(6), 1213-1221.

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012b). A decade of agile methodologies: Towards explaining agile software development: Elsevier.

Donaldson, L., & Davis, J. H. (1991). Stewardship theory or agency theory: CEO governance and shareholder returns. *Australian Journal of management, 16*(1), 49-64.

Drury, M., Conboy, K., & Power, K. (2012). Obstacles to decision making in Agile software development teams. *Journal of Systems and Software, 85*(6), 1239-1254.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology, 50*(9-10), 833-859. doi:10.1016/j.infsof.2008.01.006

Dybå, T., Dingsøyr, T., & Moe, N. B. (2014). Agile project management *Software project management in a changing world* (pp. 277-300): Springer.

Eby, L. T., Rhodes, J. E., & Allen, T. D. (2007). Definition and evolution of mentoring. *The Blackwell handbook of mentoring: A multiple perspectives approach*, 7-20.

Eckfeldt, B., Madden, R., & Horowitz, J. (2005). *Selling agile: target-cost contracts.* Paper presented at the Agile Conference, 2005. Proceedings.

Eisenhardt, K. M. (1989). Agency theory: An assessment and review. *Academy of management review, 14*(1), 57-74.

Faraj, S., & Sproull, L. (2000). Coordinating expertise in software development teams. *Management Science, 46*(12), 1554-1568.

Fenton, N., & Bieman, J. (2014). *Software metrics: a rigorous and practical approach*: CRC Press.

Ferran, C., & Watts, S. (2008). Videoconferencing in the field: A heuristic processing model. *Management Science, 54*(9), 1565-1578.

Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems, 15*(2), 200-213.

Fitzgerald, B., Russo, N. L., & O'Kane, T. (2003). Software development method tailoring at Motorola. *Communications of the ACM, 46*(4), 64-70.

Fornell, C. (1992). A national customer satisfaction barometer: the Swedish experience. *the Journal of Marketing*, 6-21.

Fowler, M. (2001). The new methodology. *Wuhan University Journal of Natural Sciences, 6*(1-2), 12-24.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development, 9*(8), 28-35.

Franklin, T. (2008). *Adventures in agile contracting: Evolving from time and materials to fixed price, fixed scope contracts.* Paper presented at the Agile, 2008. AGILE'08. Conference.

Fuggetta, A., & Di Nitto, E. (2014). *Software process.* Paper presented at the Proceedings of the on Future of Software Engineering.

Gabarro, J. J. (1990). The development of working relationships. *Intellectual teamwork: Social and technological foundations of cooperative work*, 79-110.

Garbarino, E., & Johnson, M. S. (1999). The different roles of satisfaction, trust, and commitment in customer relationships. *the Journal of Marketing*, 70-87.

Gefen, D., & Straub, D. W. (2004). Consumer trust in B2C e-Commerce and the importance of social presence: experiments in e-Products and e-Services. *Omega, 32*(6), 407-424.

Ghazawneh, A., & Henfridsson, O. (2013). Balancing platform control and external contribution in third-party development: the boundary resources model. *Information systems journal, 23*(2), 173-192.

Ghimire, D., Gibbs, S., & Charters, S. Software Development Team Views of Success Factors in Agile projects.

Ghobadi, S., & Mathiassen, L. (2016). Perceived barriers to effective knowledge sharing in agile software teams. *Information systems journal, 26*(2), 95-125.

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

Gollenia, L. A. (2016). *Business Transformation Management Methodology*: Routledge.

Gopal, A., & Sivaramakrishnan, K. (2008). Research Note-On Vendor Preferences for Contract Types in Offshore Software Projects: The Case of Fixed Price vs. Time and Materials Contracts. *Information Systems Research, 19*(2), 202-220.

Gordon, J., & Hartman, R. L. (2009). Affinity-seeking strategies and open communication in peer workplace relationships. *Atlantic Journal of Communication, 17*(3), 115-125.

Greenhaus, J. H., Collins, K. M., & Shaw, J. D. (2003). The relation between work–family balance and quality of life. *Journal of vocational behavior, 63*(3), 510-531.

Greer, D., & Hamon, Y. (2011). Agile Software Development. *Software: Practice and Experience, 41*(9), 943-944.

Grisham, P. S., & Perry, D. E. (2005). *Customer relationships and extreme programming.* Paper presented at the ACM SIGSOFT Software Engineering Notes.

Groves, R. M., Fowler Jr, F. J., Couper, M. P., Lepkowski, J. M., Singer, E., & Tourangeau, R. (2011). *Survey methodology* (Vol. 561): John Wiley & Sons.

Guba, E. G., & Lincoln, Y. S. (1994). Competing paradigms in qualitative research. *Handbook of qualitative research, 2*(163-194), 105.

Guzzo, R. A., & Dickson, M. W. (1996). Teams in organizations: Recent research on performance and effectiveness. *Annual review of psychology, 47*(1), 307-338.

Guzzo, R. A., & Shea, G. P. (1992). Group performance and intergroup relations in organizations. *Handbook of industrial and organizational psychology, 3*, 269-313.

Hackman, J. (1987). The design of work teams. Inj. w. lorsch (ed.), Handbook of organizational behavior (pp. 315-342): englewood cliffs, nj: prentice hall.

Hannafey, F. T., & Vitulano, L. A. (2013). Ethics and Executive Coaching: An Agency Theory Approach. *Journal of business ethics, 115*(3), 599-603.

Hannay, J. E., & Benestad, H. C. (2010). *Perceived productivity threats in large agile development projects.* Paper presented at the Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement.

Harrison, M., & Norman, C. (2001). What trust means in e-commerce customer relationships: an interdisciplinary conceptual typology. *International journal of electronic commerce, 6*(2), 35-59.

Harter, J. K., Schmidt, F. L., & Hayes, T. L. (2002). Business-unit-level relationship between employee satisfaction, employee engagement, and business outcomes: a meta-analysis. *Journal of applied Psychology, 87*(2), 268.

Haynes, S. N., Richard, D., & Kubany, E. S. (1995). Content validity in psychological assessment: A functional approach to concepts and methods. *Psychological assessment, 7*(3), 238.

Henttonen, K., & Blomqvist, K. (2005). Managing distance in a global virtual team: the evolution of trust through technology-mediated relational communication. *Strategic Change, 14*(2), 107-119.

Herrnstein, R. J., & Murray, C. (2010). *Bell curve: Intelligence and class structure in American life*: Simon and Schuster.

Highsmith, J. (2002). *Agile software development ecosystems* (Vol. 13): Addison-Wesley Professional.

Highsmith, J. (2009). *Agile project management: creating innovative products*: Pearson Education.

Highsmith, J. (2013). *Adaptive software development: a collaborative approach to managing complex systems*: Addison-Wesley.

Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer, 34*(9), 120-127.

Hill, R. (1998). What sample size is "enough" in internet survey research. *Interpersonal Computing and Technology: An electronic journal for the 21st century, 6*(3-4), 1-12.

Hirsch, P., Michaels, S., & Friedman, R. (1987). Dirty hands" versus "clean models. *Theory and society, 16*(3), 317-336.

Hochmüller, E. (2011). *The requirements engineer as a liaison officer in agile software development*. Paper presented at the Proceedings of the 1st workshop on agile requirements engineering.

Hoda, R., Noble, J., & Marshall, S. (2010). *Organizing self-organizing teams*. Paper presented at the Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1.

Hoda, R., Noble, J., & Marshall, S. (2011). The impact of inadequate customer collaboration on self-organizing Agile teams. *Information and Software Technology, 53*(5), 521-534. doi:http://dx.doi.org/10.1016/j.infsof.2010.10.009

Hoda, R., Noble, J., & Marshall, S. (2013). Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering, 39*(3), 422-444.

Hoegl, M., & Gemuenden, H. G. (2001). Teamwork quality and the success of innovative projects: A theoretical concept and empirical evidence. *Organization Science, 12*(4), 435-449.

Hoegl, M., & Parboteeah, P. (2006). Autonomy and teamwork in innovative projects. *Human Resource Management, 45*(1), 67-79.

Hole, S., & Moe, N. B. (2008). *A case study of coordination in distributed agile software development*. Paper presented at the European Conference on Software process improvement.

Holmstrom, H., Conchúir, E. Ó., Agerfalk, J., & Fitzgerald, B. (2006). *Global software development challenges: A case study on temporal, geographical and socio-cultural distance*. Paper presented at the Global Software Engineering, 2006. ICGSE'06. International Conference on.

Hossain, E., Babar, M. A., & Paik, H.-y. (2009). *Using scrum in global software development: a systematic literature review*. Paper presented at the Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on.

Hossain, E., Babar, M. A., Paik, H.-y., & Verner, J. (2009). *Risk identification and mitigation processes for using scrum in global software development: A conceptual framework*. Paper presented at the Software Engineering Conference, 2009. APSEC'09. Asia-Pacific.

Huda, M. N. (2011). *Comparison between adaptive software development and feature driven development*. Paper presented at the Computer Science and Network Technology (ICCSNT), 2011 International Conference on.

Hummel, M., & Epp, A. (2015). *Success factors of agile information systems development: A qualitative study*. Paper presented at the System Sciences (HICSS), 2015 48th Hawaii International Conference on.

Hummel, M., Rosenkranz, C., & Holten, R. (2013). The role of communication in agile systems development. *Business & Information Systems Engineering, 5*(5), 343-355.

Huo, M., Verner, J., Zhu, L., & Babar, M. A. (2004). *Software quality and agile methods.* Paper presented at the Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International.

Jacobson, I., Booch, G., Rumbaugh, J., Rumbaugh, J., & Booch, G. (1999). *The unified software development process* (Vol. 1): Addison-wesley Reading.

Jakobsen, C. R., & Johnson, K. A. (2008). *Mature Agile with a Twist of CMMI.* Paper presented at the Agile, 2008. AGILE'08. Conference.

Janzen, D. S., & Saiedian, H. (2005). Test-driven development: Concepts, taxonomy, and future direction. *Computer, 38*(9), 43-50.

Jeong, H., Mason, S. P., Barabási, A.-L., & Oltvai, Z. N. (2001). Lethality and centrality in protein networks. *Nature, 411*(6833), 41-42.

Johnson, B., & Onwuegbuzie, A. (2004a). Mixed Methods Research: A Research Paradigm Whose Time Has Come. *EDUCATIONAL RESEARCHER*, 33: 14. doi:10.3102/0013189X033007014

Johnson, R. B., & Onwuegbuzie, A. J. (2004b). Mixed methods research: A research paradigm whose time has come. *EDUCATIONAL RESEARCHER, 33*(7), 14-26.

Kahkonen, T. (2004). *Agile methods for large organizations-building communities of practice.* Paper presented at the Agile Development Conference, 2004.

Katzenbach, J. R., & Smith, D. K. (2005). The discipline of teams. *Harvard Business Review, 83*(7), 162.

Kaur, R., & Sengupta, J. (2013). Software process models and analysis on failure of software development projects. *arXiv preprint arXiv:1306.1068.*

Keegan, A., & Turner, J. R. (2001). Quantity versus quality in project-based learning practices. *Management learning, 32*(1), 77-98.

Kettunen, P. (2009). Adopting key lessons from agile manufacturing to agile software product development—A comparative study. *Technovation, 29*(6), 408-422.

Khalid, A., Zahra, S., & Khan, M. F. (2014). Suitability and Contribution of Agile Methods in Mobile Software Development. *International Journal of Modern Education and Computer Science, 6*(2), 56.

Kitto, S. C., Chesters, J., & Grbich, C. (2008). Quality in qualitative research. *Medical Journal of Australia, 188*(4), 243.

Kniberg, H. (2010). *Kanban and Scrum-making the most of both*: Lulu. com.

Koh, C., Ang, S., & Straub, D. W. (2004). IT outsourcing success: A psychological contract perspective. *Information Systems Research, 15*(4), 356-373.

Korkala, M., Abrahamsson, P., & Kyllönen, P. (2006). *A case study on the impact of customer communication on defects in agile software development.* Paper presented at the Agile Conference, 2006.

Kothari, C. R. (2004). *Research methodology: Methods and techniques*: New Age International.

Lalsing, V., Kishnah, S., & Pudaruth, S. (2012). People factors in agile software development and project management. *International Journal of Software Engineering & Applications, 3*(1), 117.

Langfred, C. W. (2000). The paradox of self-management: Individual and group autonomy in work groups. *Journal of Organizational Behavior*, 563-585.

Langfred, C. W. (2004). Too much of a good thing? Negative effects of high trust and individual autonomy in self-managing teams. *Academy of management journal, 47*(3), 385-399.

Larman, C., & Basili, V. R. (2003). Iterative and incremental developments. a brief history. *Computer, 36*(6), 47-56.

Lawshe, C. H. (1975). A quantitative approach to content validity1. *Personnel psychology, 28*(4), 563-575.

Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012a). *Software Development Life Cycle AGILE vs Traditional Approaches.* Paper presented at the 2012 International Conference on Information and Network Technology (ICINT 2012) IPCSIT.

Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012b). *Software development life cycle AGILE vs traditional approaches.* Paper presented at the International Conference on Information and Network Technology.

Lee, A. S. (1991). Integrating positivist and interpretive approaches to organizational research. *Organization Science, 2*(4), 342-365.

Lee, G., & Xia, W. (2010). Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly, 34*(1), 87-114.

Lee Rodgers, J., & Nicewander, W. A. (1988). Thirteen ways to look at the correlation coefficient. *The American Statistician, 42*(1), 59-66.

Leffingwell, D. (2007). Scaling software agility; best practices for large enterprises. . *31*(2, p. 0).

Leonard, M., Graham, S., & Bonacum, D. (2004). The human factor: the critical importance of effective teamwork and communication in providing safe care. *Quality and Safety in Health Care, 13*(suppl 1), i85-i90.

Lin, J. (2015). Human Factors in Agile Software Development. *arXiv preprint arXiv:1502.04170.*

Lindsjørn, Y., Sjøberg, D. I., Dingsøyr, T., Bergersen, G. R., & Dybå, T. (2016). Teamwork quality and project success in software development: A survey of agile development teams. *Journal of Systems and Software, 122*, 274-286.

Lindstrom, L., & Jeffries, R. (2004). Extreme programming and agile software development methodologies. *Information systems management, 21*(3), 41-52.

Liu, J. Y.-C., Chen, H.-G., Chen, C. C., & Sheu, T. S. (2011). Relationships among interpersonal conflict, requirements uncertainty, and software project performance. *International Journal of Project Management, 29*(5), 547-556. doi:10.1016/j.ijproman.2010.04.007

Lovelace, K., Shapiro, D. L., & Weingart, L. R. (2001). Maximizing cross-functional new product teams' innovativeness and constraint adherence: A conflict communications perspective. *Academy of management journal, 44*(4), 779-793.

Lu, Y., Xiang, C., Wang, B., & Wang, X. (2011). What affects information systems development team performance? An exploratory study from the perspective of combined socio-technical theory and coordination theory. *Computers in Human Behavior, 27*(2), 811-822.

Lunn, D. J., Thomas, A., Best, N., & Spiegelhalter, D. (2000). WinBUGS-a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and computing, 10*(4), 325-337.

Lynn, M. R. (1986). Determination and quantification of content validity. *Nursing research, 35*(6), 382-386.

Mackenzie, N., & Knipe, S. (2006). Research dilemmas: Paradigms, methods and methodology. *Issues in educational research, 16*(2), 193-205.

Mahaney, R. C., & Lederer, A. L. (2003). Information systems project management: an agency theory interpretation. *Journal of Systems and Software, 68*(1), 1-9.

Mahnic, V. (2012). A capstone course on agile software development using Scrum. *IEEE Transactions on Education, 55*(1), 99-106.

Malone, T. W., & Crowston, K. (1990). *What is coordination theory and how can it help design cooperative work systems?* Paper presented at the Proceedings of the 1990 ACM conference on Computer-supported cooperative work.

Malone, T. W., & Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys (CSUR), 26*(1), 87-119.

Marks, M. A., Mathieu, J. E., & Zaccaro, S. J. (2001). A temporally based framework and taxonomy of team processes. *Academy of management review, 26*(3), 356-376.

Martin, F. (2005). The New Methodology.

Martin, R. (2003). *Agile software development: principles, patterns, and practices*: Prentice Hall PTR.

Maruping, L. M., Venkatesh, V., & Agarwal, R. (2009). A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research, 20*(3), 377-399.

Matveev, A. V., & Nelson, P. E. (2004). Cross cultural communication competence and multicultural team performance: Perceptions of American and Russian managers. *International Journal of Cross Cultural Management, 4*(2), 253-270.

McHugh, O., Conboy, K., & Lang, M. (2011). Using agile practices to influence motivation within it project teams.

McHugh, O., Conboy, K., & Lang, M. (2012). Agile practices: The impact on trust in software project teams. *IEEE software, 29*(3), 71-76.

Melnik, G., & Maurer, F. (2004). *Direct verbal communication as a catalyst of agile knowledge sharing.* Paper presented at the Agile Development Conference, 2004.

Melton, H. L., & Hartline, M. D. (2010). Customer and frontline employee influence on new service development performance. *Journal of Service Research, 13*(4), 411-425.

Meredith, J. R., & Mantel Jr, S. J. (2011). *Project management: a managerial approach*: John Wiley & Sons.

Miller, D., & Breton-Miller, L. (2006). Family governance and firm performance: Agency, stewardship, and capabilities. *Family Business Review, 19*(1), 73-87.

Mishra, D., Mishra, A., & Ostrovska, S. (2012). Impact of physical ambiance on communication, collaboration and coordination in agile software development: An empirical evaluation. *Information and Software Technology, 54*(10), 1067-1078.

Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software, 82*(11), 1869-1890.

Mittal, V., & Kamakura, W. A. (2001). Satisfaction, repurchase intent, and repurchase behavior: Investigating the moderating effect of customer characteristics. *Journal of marketing research, 38*(1), 131-142.

Moe, N. B., Dingsøyr, T., & Dybå, T. (2008). *Understanding self-organizing teams in agile software development.* Paper presented at the Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on.

Moe, N. B., Dingsøyr, T., & Dybå, T. (2009). Overcoming barriers to self-management in software teams. *IEEE software, 26*(6).

Moe, N. B., Dingsøyr, T., & Dybå, T. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology, 52*(5), 480-491.

Mohrman, S. A., Cohen, S. G., & Morhman Jr, A. M. (1995). *Designing team-based organizations: New forms for knowledge work*: Jossey-Bass.

Moniruzzaman, A., & Hossain, D. S. A. (2013). Comparative Study on Agile software development methodologies. *arXiv preprint arXiv:1307.3356.*

Morgan, G., & Ramirez, R. (1984). Action learning: A holographic metaphor for guiding social change. *Human relations, 37*(1), 1-27.

Munassar, N. M. A., & Govardhan, A. (2010). A comparison between five models of software engineering. *IJCSI, 5*, 95-101.

Muth, M. M., & Donaldson, L. (1998). Stewardship theory and board structure: A contingency approach. *Corporate governance, 6*(1), 5-28.

Natour, A., Kiridena, S., & Gibson, P. (2011). Supply chain integration and collaboration for performance improvement: an agency theory approach.

Nerur, S., & Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM, 50*(3), 79-83.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM, 48*(5), 72-78.

Neuman, W. L. (2005). *Social research methods: Quantitative and qualitative approaches* (Vol. 13): Allyn and Bacon Boston.

Noble, H., & Smith, J. (2015). Issues of validity and reliability in qualitative research. *Evidence Based Nursing, 18*(2), 34-35.

Noy, C. (2008). Sampling knowledge: The hermeneutics of snowball sampling in qualitative research. *International Journal of Social Research Methodology, 11*(4), 327-344.

Nuseibeh, B. (2001). Weaving together requirements and architectures. *Computer, 34*(3), 115-119.

Olsson, H. H., Alahyari, H., & 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.* Paper presented at the Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on.

Opelt, A., Gloger, B., Pfarl, W., & Mittermayr, R. (2013). *Agile Contracts: Creating and Managing Successful Projects with Scrum*: John Wiley & Sons.

Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2008). Using scrum in a globally distributed project: a case study. *Software Process: Improvement and Practice, 13*(6), 527-544.

Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2009). *Using scrum in distributed agile development: A multiple case study.* Paper presented at the Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on.

Paasivaara, M., Heikkilä, V. T., & Lassenius, C. (2012). *Experiences in scaling the product owner role in large-scale globally distributed scrum.* Paper presented at the Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on.

Paetsch, F., Eberlein, A., & Maurer, F. (2003a). *Requirements engineering and agile software development.* Paper presented at the Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on.

Paetsch, F., Eberlein, A., & Maurer, F. (2003b). *Requirements engineering and agile software development.* Paper presented at the null.

Paetsch, F., Eberlein, A., & Maurer, F. (2003c). *Requirements Engineering and Agile Software Development.* Paper presented at the WETICE.

Palmer, S. R., & Felsing, M. (2001). *A practical guide to feature-driven development*: Pearson Education.

Parnas, D. (2006). Agile methods and GSD: The wrong solution to an old but real problem. *Communications of the ACM, 49*(10), 29-29.

Patton, M. Q. (2005). *Qualitative research*: Wiley Online Library.

Perry, W. E. (2007). *Effective Methods for Software Testing: Includes Complete Guidelines, Checklists, and Templates*: John Wiley & Sons.

Petersen, K., & Wohlin, C. (2010). The effect of moving from a plan-driven to an incremental software development approach with agile practices. *Empirical Software Engineering, 15*(6), 654-693.

Phalnikar, R., Deshpande, V., & Joshi, S. (2009). *Applying agile principles for distributed software development.* Paper presented at the Advanced Computer Control, 2009. ICACC'09. International Conference on.

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering, 13*(3), 303-337.

Polit, D. F., Beck, C. T., & Owen, S. V. (2007). Is the CVI an acceptable indicator of content validity? Appraisal and recommendations. *Research in nursing & health, 30*(4), 459-467.

Poppendieck, M., & Poppendieck, T. (2003a). *Lean software development: An agile toolkit*: Addison-Wesley Professional.

Poppendieck, M., & Poppendieck, T. (2003b). *Lean Software Development: An Agile Toolkit: An Agile Toolkit*: Addison-Wesley.

Poppendieck, M., & Poppendieck, T. (2007). *Implementing lean software development: from concept to cash*: Pearson Education.

Power, K. (2010). *Stakeholder identification in agile software product development organizations: A model for understanding who and what really counts.* Paper presented at the AGILE Conference, 2010.

Pressman, R. S. (2005). *Software engineering: a practitioner's approach*: Palgrave Macmillan.

Pym, D., & Sadler, M. (2010). Information Stewardship in cloud computing. *International Journal of Service Science, Management, Engineering, and Technology (IJSSMET), 1*(1), 50-67.

Rai, A., Maruping, L. M., & Venkatesh, V. (2009). Offshore information systems project success: the role of social embeddedness and cultural characteristics. *MIS Quarterly*, 617-641.

Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: an empirical study. *Information systems journal, 20*(5), 449-480.

Ramesh, B., Cao, L., Mohan, K., & Xu, P. (2006). Can distributed software development be agile? *Communications of the ACM, 49*(10), 41-46.

Reh, F. J. (2005). Pareto's principle-The 80-20 rule. *BUSINESS CREDIT-NEW YORK THEN COLUMBIA MD-, 107*(7), 76.

Rising, L., & Janoff, N. S. (2000). The Scrum software development process for small teams. *IEEE software, 17*(4), 26.

Ross, S. A. (1973). AssociationThe Economic Theory of Agency: The Principal's Problem. *The American Economic Re, 63*(2), 134-139.

Royce, W. W. (1970). *Managing the development of large software systems.* Paper presented at the proceedings of IEEE WESCON.

Ryan, S., & O'Connor, R. V. (2013). Acquiring and sharing tacit knowledge in software development teams: An empirical study. *Information and Software Technology, 55*(9), 1614-1624.

Salas, E., Cooke, N. J., & Rosen, M. A. (2008). On teams, teamwork, and team performance: Discoveries and developments. *Human factors, 50*(3), 540-547.

Salas, E., Sims, D. E., & Burke, C. S. (2005). Is there a "Big Five" in teamwork? *Small group research, 36*(5), 555-599.

Santos, V., Goldman, A., & De Souza, C. R. (2015). Fostering effective inter-team knowledge sharing in agile software development. *Empirical Software Engineering, 20*(4), 1006-1051.

Sarantakos, S. (2012). *Social research*: Palgrave Macmillan.

Schatz, B., & Abdelshafi, I. (2005). Primavera gets agile: a successful transition to agile development. *IEEE software*(3), 36-42.

Schillemans, T. (2013). Moving Beyond The Clash of Interests: On stewardship theory and the relationships between central government departments and public agencies. *Public Management Review, 15*(4), 541-562.

Schwaber, K. (1997). Scrum development process *Business Object Design and Implementation* (pp. 117-134): Springer.

Schwaber, K. (2004). *Agile project management with Scrum*: Microsoft Press.

Schwaber, K., & Beedle, M. (2002). gilè Software Development with Scrum.

Schwaber, K., & Sutherland, J. (2013). The definitive guide to scrum: The rules of the game. *Improving the Profession of Software development*.

Seale, C. (1999). Quality in qualitative research. *Qualitative inquiry, 5*(4), 465-478.

Shank, G., & Brown, L. (2013). *Exploring educational research literacy*: Routledge.

Sheehan, K. B. (2001). E-mail survey response rates: A review. *Journal of Computer-Mediated Communication, 6*(2), 0-0.

Sheng, C.-W., Tian, Y.-F., & Chen, M.-C. (2010). Relationships among teamwork behavior, trust, perceived team support, and team commitment. *Social Behavior and Personality: an international journal, 38*(10), 1297-1305.

Shore, J. (2007). *The art of agile development*: " O'Reilly Media, Inc.".

Sias, P. M. (2005). Workplace relationship quality and employee information experiences. *Communication studies, 56*(4), 375-395.

Smith, H., & Fingar, P. (2003). *Business process management: the third wave* (Vol. 1): Meghan-Kiffer Press Tampa.

Sohaib, O., & Khan, K. (2010). *Integrating usability engineering and agile software development: A literature review.* Paper presented at the Computer design and applications (ICCDA), 2010 international conference on.

Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D.-B. (2013a). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *Journal of Systems and Software*.

Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D.-B. (2013b). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *Journal of Systems and Software, 86*(6), 1663-1678. doi:10.1016/j.jss.2013.02.027

Stapleton, J. (1997). *DSDM, dynamic systems development method: the method in practice*: Cambridge University Press.

Stenbacka, C. (2001). Qualitative research requires quality concepts of its own. *Management decision, 39*(7), 551-556.

Stoica, M., Mircea, M., & Ghilic-Micu, B. (2013). Software Development: Agile vs. Traditional. *Informatica Economica, 17*(4), 64.

Stol, K.-J., & Fitzgerald, B. (2014). *Two's company, three's a crowd: a case study of crowdsourcing software development.* Paper presented at the Proceedings of the 36th International Conference on Software Engineering.

Strauss, A., & Corbin, J. (1990). *Basics of qualitative research* (Vol. 15): Newbury Park, CA: Sage.

Sureshchandra, K., & Shrinivasavadhani, J. (2008). *Adopting agile in distributed development.* Paper presented at the Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on.

Sutherland, J., Jakobsen, C. R., & Johnson, K. (2008). *Scrum and CMMI level 5: The magic potion for code warriors.* Paper presented at the Hawaii International Conference on System Sciences, Proceedings of the 41st Annual.

Sutherland, J., Viktorov, A., Blount, J., & Puntikov, N. (2007). *Distributed scrum: Agile project management with outsourced development teams.* Paper presented at the System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on.

Svensson, H., & Höst, M. (2005). *Views from an organization on how agile development affects its collaboration with a software development team.* Paper presented at the International Conference on Product Focused Software Process Improvement.

Takeuchi, H., & Nonaka, I. (1986). The new new product development game. *Harvard Business Review, 64*(1), 137-146.

Tan, C.-W., Pan, S. L., & Lim, E. T. (2005). Managing stakeholder interests in e-government implementation: Lessons learned from a Singapore e-government project. *Journal of Global Information Management (JGIM), 13*(1), 31-53.

Tate, K. (2005). *Sustainable software development: an agile perspective*: Addison-Wesley Professional.

Taylor, R. (1990). Interpretation of the correlation coefficient: a basic review. *Journal of diagnostic medical sonography, 6*(1), 35-39.

Topp, L., Barker, B., & Degenhardt, L. (2004). The external validity of results derived from ecstasy users recruited using purposive sampling strategies. *Drug and alcohol dependence, 73*(1), 33-40.

Tuckman, B. W. (1965). Developmental sequence in small groups. *Psychological bulletin, 63*(6), 384.

Turk, D., France, R., & Rumpe, B. (2014a). Assumptions underlying agile software development processes. *arXiv preprint arXiv:1409.6610.*

Turk, D., France, R., & Rumpe, B. (2014b). Limitations of agile software processes. *arXiv preprint arXiv:1409.6600.*

Utting, M., & Legeard, B. (2010). *Practical model-based testing: a tools approach*: Morgan Kaufmann.

Van Lamsweerde, A. (2000). *Requirements engineering in the year 00: A research perspective.* Paper presented at the Proceedings of the 22nd international conference on Software engineering.

Van Selm, M., & Jankowski, N. W. (2006). Conducting online surveys. *Quality and Quantity, 40*(3), 435-456.

Van Teijlingen, E., & Hundley, V. (2002). The importance of pilot studies. *Nursing standard, 16*(40), 33-36.

Vidgen, R., & Wang, X. (2009). Coevolving systems and the organization of agile software development. *Information Systems Research, 20*(3), 355-376.

Vijayasarathy, L., & Turk, D. (2008). Agile software development: A survey of early adopters. *Journal of Information Technology Management, 19*(2), 1-8.

Voss, C., Tsikriktsis, N., & Frohlich, M. (2002). Case research in operations management. *International journal of operations & production management, 22*(2), 195-219.

West, D., Grant, T., Gerush, M., & D'silva, D. (2010). Agile development: Mainstream adoption has changed agility. *Forrester Research, 2*(1), 41.

Whitworth, E., & Biddle, R. (2007). *The social nature of agile teams.* Paper presented at the Agile conference (AGILE), 2007.

Wiegers, K., & Beatty, J. (2013). *Software requirements*: Pearson Education.

Wilding, R., Wagner, B., Fayezi, S., O'Loughlin, A., & Zutshi, A. (2012). Agency theory and supply chain management: a structured literature review. *Supply chain management: an international journal, 17*(5), 556-570.

Williams, L., & Cockburn, A. (2003). Guest Editors' Introduction: Agile Software Development: It's about Feedback and Change. *Computer, 36*(6), 39-43.

Witzel, A. (2000). *The problem-centered interview.* Paper presented at the Forum Qualitative Sozialforschung/Forum: Qualitative Social Research.

Wright, K. B. (2005). Researching Internet-based populations: Advantages and disadvantages of online survey research, online questionnaire authoring software packages, and web survey services. *Journal of Computer-Mediated Communication, 10*(3), 00-00.

Zelkowitz, M. V., & Wallace, D. R. (1998). Experimental models for validating technology. *Computer, 31*(5), 23-31.

Zhang, Y. (2000). Using the Internet for survey research: A case study. *Journal of the American society for information science, 51*(1), 57-68.

Zikmund, W., Babin, B., Carr, J., & Griffin, M. (2012). *Business research methods*: Cengage Learning.

Appendices

APPENDIX A: CONTENT VALIDATION TABLE

Item	Expert1	Expert 2	Expert3	Expert4	Expert 5	Total	CVI
Item 1.3	X	X	X	X	X	5	1
Item1.4	X	X	X	X	X	5	1
Item1.5	X	X	X		X	4	0.8
Item 1.6		X	X			2	0.4
Item 1.7	X	X	X	X	X	5	1
Item1.8	X	X	X	X	X	5	1
Item 1.9	X	X	X	X	X	5	1
Item 1.10	X	X	X	X	X	5	1
Item 1.11.	X	X	X	X	X	5	1
Item1.12	X	X	X	X	X	5	1
Item 1.13	X	X	X	X		4	0.8
Item 1.14	X	X	X	X		4	0.8
Item 1.15	X	X	X	X		4	0.8
Item 1.16	X	X	X	X		4	0.8
Item 1.17	X	X	X	X		4	0.8
Item 1.18	X	X	X	X	X	5	1
Item 1.19	X	X	X	X	X	5	1
Item 1.20	X	X	X		X	4	0.8
Item 1.21	X	X	X	X		4	0.8
Item 1.22	X		X	X		3	0.6
Item 1.23	X		X			2	0.4
Item 1.24	X		X		X	3	0.6
Item 1.25	X	X	X	X	X	5	1
Item 1.26		X	X	X	X	4	0.8
Item 1.27		X	X	X		4	0.8

					X		
Item 1.28	X	X	X	X	X	5	1
Item 1.29	X	X	X	X	X	5	1
Item 1.30	X	X	X	X	X	5	1
Item 1.31	X	X	X	X	X	5	1
Item 1.32	X	X	X	X	X	5	1
Item 1.33	X	X	X	X	X	5	1
Item 1.34	X	X	X	X	X	5	1
Item 1.35	X	X	X	X	X	5	1
Item 1.36	X	X	X		X	4	0.8
Item 1.37	X	X	X	X	X	5	1
Item 1.38	X		X		X	3	0.6
Item 1.39	X	X	X	X		4	0.8
Item 1.40	X	X	X	X	X	5	1
Item 2.1	X	X	X	X	X	4	0.8
Item 2.2	X	X	X	X	X	4	0.8
Item 2.3	X		X		X	3	0.6
Item 2.4	X	X	X	X	X	5	1
Item 2.5	X	X	X	X	X	5	1
Item 2.6	X		X	X	X	4	0.8
Item 2.7	X		X	X	X	4	0.8
Item 2.8	X		X	X	X	4	0.8
Item 2.9	X	X	X	X	X	5	1
Item 2.10	X	X	X	X	X	5	1
Item 4.1	X	X	X	X	X	5	1
Item 4.2	X		X		X	3	0.6
Item 4.3	X		X		X	3	0.6
Item 4.4	X	X	X	X	X	5	1

Item 4.5	X	X	X		X	4	0.8
Item 4.6	X	X	X		X	3	0.6
Item 4.7	X		X		X	3	0.6
Item 4.8	X		X		X	3	0.6
Item 4.9	X	X	X	X	X	5	1
Q4.10 Please list those challenges	X	X	X	X	X	5	1
Item 3.1	X	X	X	X	X	5	1
Proportion Relevant							

TOTAL 49.6

AVERAGE
CVI 0.8

APPENDIX B: PILOT STUDY QUESTIONNAIRE

Q1.1 We would like to invite you to participate in a pilot study to examine the relationship between customers and software development teams in Agile software development.
The research aims to understand the impact of the relationship on project outcomes. This pilot study aims to develop the research instruments to conduct the research. Which we anticipated will take you no longer than 20 minutes. We would appreciate your taking the time to complete this survey. The responses that you will provide for this survey will remain confidential. You can contact Dipendra Ghimire(Dipendra.Ghimire@lincolnuni.ac.nz) or my supervisors Stuart Charters (Stuart.Charters@lincoln.ac.nz) or Shirley Gibbs (Shirley.Gibbs@lincoln.ac.nz) if you have any questions about the research.

I have read and understood the description of this study. On this basis, I agree to participate and consent to the information that I provide being used to develop a research questionnaire. Please enter your email address. _____

Q1.2 In this section questions related to your organisation and your experiences on software development will be asked.

Q1.3 What is your current job title?

Q1.4 How long have you been employed in the IT sector?

Q1.5 How many people are employed in software development related roles in your organisation?
- Less than 5
- 5- 10
- 10 - 20
- 21-40
- 41 - 100
- 101 - 500
- 501 - 1000
- More than 1000

Q1.6 Are you currently involved in software development?
- Yes
- No

Q1.7 Have you undertaken any workplace training in software development?
- Yes
- No

Display This Question:
    If Have you undertaken any training in software development? Yes Is Selected
Q1.8 Please list the training you have done in software development

Q1.9 In this section you will be asked questions related to a recently completed project/product. The term project is used to indicate the complete software and product represents deliverable software. Please think about the recent completed software project/product and answer the following questions.

Q1.10 Which approaches were used on your project/ product?

- Scrum
- extreme programming (XP)
- Feature Driven Development (FDD)
- Dynamic Systems Development Method (DSDM)
- Kanban
- Adaptive Software Development (ASD)
- Lean software development
- Scrum-ban
- Crystal Methods
- Test Driven Development (TDD)
- Behaviour Driven Development (BDD)
- Waterfall development
- Spiral development
- Prototyping development
- Code and fix
- Other please write _____

Q1.11 Which of the following practices did you use?

- Stand-ups
- Continuous integration
- Backlog
- Pair Programming
- Burndown chart/ Burnup chart
- Definition Of Done
- Refactoring
- Scrum board
- Kanban board
- Retrospective
- Epic
- Sprint/Time box
- User Stories
- Planning Poker
- Personas
- Automated Test
- Online Tools
- Definition of Ready
- Unit test
- Continuous development
- None of the above
- Other please write below _____

Q1.12 Which best describes your role(s) in the team?

- Product owner
- Developer
- Tester
- Scrum master
- Coach
- Business Analyst
- Client representative
- Technical writer
- Agile mentor
- Team leader
- UI/UX
- Line manager
- Project manager
- Integrator
- Trainer
- Other please write below _____

Q1.13 From the list below which best describes your project

- In-house development
- Product development
- Contract bespoke development
- Other _____

Q1.14 How many people were in your team?

Q1.15 How was the team distributed geographically?

- Team all sit together
- Within the same building or location
- Within the same city
- Across multiple sites nationally
- Across multiple sites internationally

Q1.16 How long did the project/product take in months?

Q1.17 Roughly, what was the cost of project/product?
- Less than $50k
- $50k - $250k
- $251k - $1M
- $1M - $10M
- $11M - $50M
- $51M - $100M
- More than $100M
- Am not able to say
- I don't know

Q1.18 What type of contract model was used?
- Fixed price contract
- Time-and materials contract
- Target cost
- I don't know
- I am not able to say
- There was no contract
- Other please state _____

Q1.19 Which of the following did the project/product have?
- Product owner
- Project manager

Display This Question:
    If Which of the following did the project/product had? Product Owner Is Selected
Q1.20 Was the product owner from a client organisation?
Yes
No

Display This Question:
    If Which of the following did the project/product had? Product owner Is Selected
Q1.21 Was the product owner a representative from your organization?
Yes
No

Display This Question:
    If Which of the following did the project/product had? Product owner Is Selected
Q1.22 Was there more than one customer representative?
Yes
No

Q1.23 Did the product owner have sufficient delegated authority to make the decisions required for the project/product?

Yes

No

Q1.24 In your opinion was the customer/product owner available to answer any questions you had during the development of project in a timely manner?

Never

Rarely

Sometimes

Most of the Time

Always

Q1.25 Please rate how often the following occurred within the team

	Never	Rarely	Sometimes	Most of the Time	Always	N/A
Difficulties in communicating within the team						
Interpersonal challenges within the team						
Difficulties in sharing of ideas within the team						
Problems with distribution of the work within the team						

Q1.26 Please rate how often the following occurred between team and customer/product owner

	Never	Rarely	Sometimes	Most of the Time	Always	N/A
Disagreement with the customer regarding project priorities						
Disagreement with customer regarding project requirements						
Disagreement with the customer regarding the timeframe of the project						
Interpersonal challenges between the team member(s) and customer						
Difficulties in communicating with the customer						

Display This Question:

    If Which of the following did the project/product? Product Owner Is Selected

Q1.27 How do you communicate with the Product owner?
- Through email
- Through telephone
- Video conferencing
- In person
- Instant messaging
- Other please write _____

Q1.28 The customer in the project/product had a background in analysis or design
- Yes
- No
- Don't know

Q1.29 Do you think the customer/product owner needed more training in working with teams?
- Yes
- No
- Don't know

Q1.30 Was the project/product completed on time?
- Yes
- No
- Don't know

Q1.31 Was the project/product completed within the specified budget?
- Yes
- No
- Don't know

Q1.32 Was the project/project contract met?
- Yes
- No
- Don't know

Q1.33 In other projects/products you have worked on, have you encountered occurrences of what could be described as challenges?
- Yes
- No

Q1.34 Please list those challenges

Q1.35 Did you find the questions easy to answer?
- Yes
- No

Q1.36 Did you find these questions relevant to your software development role?

- Yes

- No

Q1.37 Would you like to give any feedback?

- Yes

- No

Q1.38 Please write the feedback

Pilot Study Interview Questions

- Can you please tell me about your current role?

   Prompt question

  - What do you do?

  - Which approach do you use for software development?

- Can you please describe how a new project starts and what process it goes through

   Prompt questions:

  - Please tell me how are the requirements gathered

  - Please tell me how are the requirements refined

  - Can you please tell me about the documentation done for the project

  - How much details does the documentation include

  - Can you please tell about the contract process (legal document)

  - What happens when there is change in scope after the contract?

3 Can you please tell me about the process of assigning roles on a project?

- Please tell me about how the work is divided

- What do you think is motivating the team member working in group

- Please tell me about the commitment of team members in achieving the project/product goal

4   Can you please tell me about customer involvement in your recently completed project/product?

     Prompt questions:

- Please tell me about the communication process with customer/Product owner

- Can you please tell me more about the product owner regarding their knowledge on working in Agile teams?

- Please tell me about some of the challenges you have faced with customer

- How often do you face those challenges (any numbering)

- In your view what are the major challenges with customer/ PO on software project/ product development

- How do you see the customer developer relationship impact on the project outcome

- Can you please tell me about your current project?

- Please tell me about some of the difficulties you have faced so far.

- What impact do u think those difficulties will make on project outcome?

6   "I'm interested in understanding why some projects go well and are successful and others struggle. Can you please describe what you see as the important differences in projects that you have worked on that have been a success vs. those that haven't been?"

APPENDIX C: ETHICS APPROVAL

Software Development Teams study

Application No: 2015-09   27 March 2015

Title: The impact of the relationship between a customer and a software development team in software development projects outcome.

Applicant: Dipendra Ghimire

The Lincoln University Human Ethics Committee has reviewed the above noted application.

Thank you for your responses to the questions which were forwarded to you on the Committee's behalf.

I am satisfied on the Committee's behalf that the issues of concern have been satisfactorily addressed.

I am pleased to give final approval to your project. Please note that this approval is valid for three years from today's date at which time you will need to reapply for renewal.

There are a couple of minor suggestions and corrections I suggest you make to the invitation letter and reminder email, but this is not a requirement. (Please see comments in the attachment.)

Once your field work has finished can you please advise the Human Ethics Secretary, Alison Hind, and confirm that you have complied with the terms of the ethical approval.

May I, on behalf of the Committee, wish you success in your research.

Yours sincerely

Caitriona Cameron
Acting Chair, Human Ethics Committee

PLEASE NOTE: The Human Ethics Committee has an audit process in place for applications. Please see 7.3 of the Human Ethics Committee Operating Procedures (ACHE) in the Lincoln University Policies and Procedures Manual for more information.

Product owner study:

Application No: 2015-43   27 October 2015

Title: The impact of the relationship between a customer and a software development team in software development projects outcome. This is a second of a pair of studies.

Applicant Dipendra Ghimire

The Lincoln University Human Ethics Committee has reviewed the above noted application.
Thank you for your response to the questions which were forwarded to you on the Committee's behalf.

I am satisfied on the Committee's behalf that the issues of concern have been satisfactorily addressed. I am pleased to give final approval to your project.

Please note that this approval is valid for three years from today's date at which time you will need to reapply for renewal.

Once your field work has finished can you please advise the Human Ethics Secretary, Alison Hind, and confirm that you have complied with the terms of the ethical approval.

May I, on behalf of the Committee, wish you success in your research.

Yours sincerely

Grant Tavinor
Chair, Human Ethics Committee

PLEASE NOTE: The Human Ethics Committee has an audit process in place for applications. Please see 7.3 of the Human Ethics Committee Operating Procedures (ACHE) in the Lincoln University Policies and Procedures Manual for more information.

Research Information sheet (Software development team study)

Lincoln University

Department of Informatics and Enabling Technologies, Faculty of Environment, Society and Design

Research Information Sheet

Name of project: The impact of the relationship between a customer and a software development team in determining the outcome of a software development project

You have been invited to participate in this research project because of your role in the software development sector. Your participation in this research, should you agree to participate, will involve the completion of an online questionnaire with the possibility of being invited for a short interview at a time and place to suit you.

The questionnaire covers a range of questions about software development. This process will take 20 min on average to complete. An email address is collected during the questionnaire. The email address collected will be used for two purposes. One is to invite participants for an interview and other is to locate participant records in case a participant wishes to withdraw their information from this study. The invitation for an interview will go to randomly selected participants

Participants for the interview section of this research will be randomly selected from participants who have completed the online questionnaire. The interview will take an estimated 30 to 40 minutes to complete. In order to participate in the interview a recording is required. This process can be done at a time, which best suits you. Participation in all stages of this this study are voluntary therefore you are welcome to decline a survey or interview invitation should you receive one.

None of the information you contribute to the study will lead to you being identified in any subsequent components of the study by the researcher. You may withdraw your responses

to this study until 31st December 2015. If you withdraw your response to this study all of your data will be destroyed. Withdrawal can be done by contacting the principal researcher or one his supervisors before the above date. The interview will be audio recorded. Interview transcriptions will be available on request.

The results of the research may be published, but you may be assured of your anonymity in this investigation. The identity of any participant will not be made public. To ensure anonymity and confidentiality no names or other identifying information will be collected on the questionnaire. Information will not be collected which will identify the workplace. All electronic copies of data and decoding system to link participants to the pseudonyms will be stored on the secure university network server and the principal researcher's personal computer at University with encoded passwords to login. Data files will also be encoded with a password.

The project is being carried out by: Dipendra Ghimire

Contact details: Dipendra.Ghimire@lincolnuni.ac.nz */ +64211407931*

He and his supervisors will be pleased to discuss any concerns you have about participation in the project.

Name of Supervisors:

Stuart Charters

Department of Informatics and Enabling Technologies

Phone: 03 34230415

Email: Stuart.Charters@lincoln.ac.nz

Shirley Gibbs

Department of Informatics and Enabling Technologies

Phone: 03 34230418

Email: Shirley.Gibbs@lincoln.ac.nz

The project has been reviewed and approved by the Lincoln University Human Ethics Committee.

Invitation letter for the participant whose email address are known already:

Dear,

My name is Dipendra Ghimire and I am a PhD student in the Department of Informatics and Enabling Technologies at Lincoln University. My project involves an examination of the relationship between customers and software development teams.

I have obtained your email address through your business card or through personal contact, or from web address. I am recruiting participants currently involved in software development in roles such as developer, tester, or business analyst to participate in this study. Participation will involve the completion of an online questionnaire that I anticipate will take no longer than 20 minutes to complete. Following this, I will interview a sample of participants to gather more in-depth data. Interview participants will be randomly selected from those who have completed the questionnaire. I anticipate that interviews, which will be conducted at a time and place to suit you and will take around 30 to 40 minutes. Participants who are invited to an interview are under no obligation to participate.

The responses that you provide will remain confidential to the researcher and research supervisors and anonymity will be preserved.

If, in your current role, you would not be able to participate but could pass on details of a colleague who could that would be appreciated. Please let me know if I need to seek permission from your organization. The employer will not be given any information about your participation unless you have given permission for this.

If you have already received this request please disregard this request.

For more information on the research please click here to access the Research Information Sheet

Click on the following link to start the online questionnaire

http://lincoln.az1.qualtrics.com/SE/?SID=SV_ea2QR4BOXzkKQdv

If you would like more information about this project please feel free to contact me or either of my supervisors:

Stuart Charters (stuart.charters@lincoln.ac.nz)

Shirley Gibbs (shirley.gibbs@lincoln.ac.nz).

Thank you

Dipendra Ghimire

Invitation letter for contacting the company for participants:

Dear Sir or Madam,

My name is Dipendra Ghimire and I am a PhD student in the Department of Informatics and Enabling Technologies at Lincoln University. My PhD project involves an examination of the relationships between customers and software development teams.

I have identified your organization as one where I may locate suitable participants for my project. Participants should be people who currently involved in software development in roles such as developer, tester, or business analyst. If you could forward the attachment email to colleagues who may be interested in helping that would be appreciated. If you have already received this request please disregard this message.

For more information about this study click here to access the research information sheet

Click on the following link to start the survey.

http://lincoln.az1.qualtrics.com/SE/?SID=SV_ea2QR4BOXzkKQdv

If you would like more information about this project please feel free to contact me or either of my supervisors:

Stuart Charters (stuart.charters@lincoln.ac.nz)

Shirley Gibbs (shirley.gibbs@lincoln.ac.nz).

Thank you

Dipendra

Reminder letter for contacting companies

Dear Sir or Madam,

My name is Dipendra Ghimire and I am a PhD student in the Department of Informatics and Enabling Technologies at Lincoln University. I contacted you a few weeks regarding my search for participants to take part in my research project.

In order complete this project in a timely manner the questionnaire will be closed on 31st of October 2015. If you would like to participate in the research, and have not yet completed the survey, it would be helpful if you could do so by 31st of October 2015

For more information on the research please click here to access the Research Information Sheet

Click on the following link to start the survey.

http://lincoln.az1.qualtrics.com/SE/?SID=SV_ea2QR4BOXzkKQdv

If you would like more information about this project please feel free to contact me or either of my supervisors:

Stuart Charters (stuart.charters@lincoln.ac.nz)

Shirley Gibbs (shirley.gibbs@lincoln.ac.nz).

Thank you

The impact of customer development team relationship on Agile software development project outcome

Q1.1 We would like to invite you to participate in a study to examine the relationship between customers and software development teams in Agile software development. This research aims to understand the impact of the relationship on project outcomes. We anticipate this will take you no longer than 20 minutes. We would appreciate you taking the time to complete this questionnaire. The responses that you will provide will remain confidential. You may also be invited for a follow up interview, which will take around 30 to 40 minutes but can decline this invitation if you so wish. You can contact Dipendra Ghimire (Dipendra.Ghimire@lincolnuni.ac.nz) or my supervisors Stuart Charters (Stuart.Charters@lincoln.ac.nz) and Shirley Gibbs (Shirley.Gibbs@lincoln.ac.nz) if you have any questions about the research.

- I have read and understood the description of the above-named project. On this basis, I agree to participate as a subject in the project, and I consent to publication of the results of the project with the understanding that anonymity will be preserved. I understand also that I may at any time withdraw from the project, including withdrawal of any information I have provided, up to 31st December 2015.

Please write your email address (The email address here will be used for two purposes. One is to invite the participant for interview and other is to locate participant records in case a participant wishes to withdraw their information from this study)

_____.

Q1.2 This section of question is related to your organization and your experiences in software development.

Q1.3 What is your current job title?

Q1.4 How long have you been employed in the IT sector?

Q1.5 How many people are employed in software development related roles in your organization?

5   Less than 5

6   5- 10

7   11 - 20

8   21-40

9   41 - 100

10 101 - 500

11 501 - 1000

12 More than 1000

Q1.6 Are you currently involved in software development?

13 Yes

14 No

Q1.7 Have you undertaken any workplace training in software development?

15 Yes

16 No

Answer If Have you undertaken any training in software development? Yes Is Selected

Q1.8 Please list the training you have done in software development

Q1.9 In this section you will be asked questions related to a recently completed project/product. The term project is used to indicate the complete software developed for specific objectives. A Product indicates a continuous development process for improvement

of the product through the delivery of new features. A **product** is designed for diverse market needs.

Q1.10 Which approaches were used on your project/ product? (tick more than one box, as applicable)

- Scrum
- eXtreme programming (XP)
- Feature Driven Development (FDD)
- Dynamic Systems Development Method (DSDM)
- Kanban
- Adaptive Software Development (ASD)
- Lean software development
- Scrum-ban
- Crystal Methods
- Test Driven Development (TDD)
- Behavior Driven Development (BDD)
- Waterfall development
- Spiral development
- Prototyping development
- Code and fix
- Other please write _____

Q1.11 which of the following practices did you use? (tick as many boxes as are applicable)

- Stand-ups
- Continuous integration
- Backlog
- Pair Programming
- Burndown chart/ Burnup chart
- Definition Of Done
- Refactoring
- Scrum board
- Kanban board
- Retrospective
- Epic
- Sprint/Time box
- User Stories
- Planning Poker
- Personas
- Automated Test
- Online Tools
- Definition of Ready
- Unit test
- Continuous development
- None of the above
- Other please write below _____

Q1.12 Which best describes your role(s) in the team? (tick as many boxes as are applicable)

- Product owner
- Developer
- Tester
- Scrum master
- Coach
- Business Analyst
- Client representative
- Technical writer
- Agile mentor
- Team leader
- UI/UX
- Line manager
- Project manager
- Integrator
- Trainer
- Other please write below _____

Q1.13 From the list below which best describes your project

17   In-house development

18   Product development

19   Contract bespoke development

20   Other _____

Q1.14 How many people were in your team?

Q1.15 How many Agile teams are in your company?

Q1.16 How was the team distributed geographically?

21   Team all sit together

22   Within the same building or location

23   Within the same city

24   Across multiple sites nationally

25   Across multiple sites internationally

Q1.17 How long did the project/product take in months?

Q1.18 Roughly, what was the cost of project/product?

26   Less than $50k

27   $50k - $250k

28   $251k - $1M

29   $1M - $10M

30   $11M - $50M

31   $51M - $100M

32   More than $100M

33   Am not able to say

34   I don't know

Q1.19 What type of contract model was used?

35   Fixed price contract

36   Time-and materials contract

37   Target cost

38   I don't know

39   I am not able to say

40   There was no contract

41   Other please state _____

Q1.20 Which of the following did the project/product have?

    a. Product owner

    b. Project manager

    c. Other Customer Representative please State _____

Q1.21 Was the product owner from a client organisation?

    42 Yes

    43 No

    44 I don't know

Q1.22 Was the product owner a representative from your organization?

    45 Yes

    46 No

    47 I don't know

Q1.23 Was there more than one customer representative?

    48 Yes

    49 No

    50 I don't know

Q1.24 Did the product owner have sufficient delegated authority to make the decisions required for the project/product?

    51 Yes

    52 No

    53 I don't know

Q1.25 In your opinion was the customer/product owner available to answer any questions you had during the development of project in a timely manner?

    54 Never

    55 Rarely

    56 Sometimes

    57 Most of the Time

    58 Always

    59 I don't know

Q1.26. Please rate how often the following occurred within the team during a typical sprint

	0	1- 3 times	4-6 times	7-9 times	More than 9 times	Don't know
Difficulties in communicating within the team						
Interpersonal challenges within the team						
Difficulties in sharing of ideas within the team						
Problems with distribution of the work within the team						

Q1.27. Please rate how often the following occurred during a typical sprint

	0	1-3 times	4-6 times	7-9 times	More than 9 times	Don't know
Disagreement with the customer regarding project priorities						
Disagreement with customer regarding project requirements						
Disagreement with the customer regarding the timeframe of the project						
Interpersonal challenges between the team member(s) and customer						
Difficulties in communicating with the customer						

Q1.28 How does product team communicate with the Product owner? (Tick as many boxes as are applicable.

- By email
- By telephone
- Video conferencing
- In person
- Instant messaging
- Other please write _____

Q1.29 The customer in the project/product had a background in analysis or design

   60 Yes

   61 No

   62 Don't know

Q1.30 Do you think the customer/product owner needed more training in working with teams?

   63 Yes

   64 No

   65 Don't know

Q1.31 Was the project/product completed on time?

   66 On time

   67 Before time

   68 Completed late

   69 Was cancelled

   70 Was not completed

   71 Don't know

Q1.32 Was the project/product completed within the specified budget?

    72 Less than budget

    73 Within a budget

    74 More than budget

    75 Don't know

Q1.33 Was the project/project original contract met?

    76 Yes

    77 No

    78 Don't know

Q1.34 "I'm interested in understanding why some projects go well and are successful while others struggle. Can you please describe what you see as the important differences in projects that you have worked on that have been a success versus. those that haven't been?"

Q1.35. In your opinion which of the following relationships work best in order for a project t to be successful? (Tick as many boxes as are applicable)

79 The team members in a software development collaborate with the customer/product owner to meet the project goal.

80 The motivation to succeed derives team member to complete software development projects.

81 The team members in a software development team are guided by the instruction given by the customer/ product owner with the focus on finishing the project on time.

82 The main goal of both the team and the customer is to finish the project within the agreed budget.

83 The main goal of both the team and the customer is to complete a project within the agreed timeframe,

84 Team members and the customer are guided by the project contract

85 The team members have trust in the decisions that the customer/product owner may make

86 The customer/product trusts the team to complete the project according to the agreed contract.

87 There is complete transparency in interactions between team members and the customer/ product owner during the project development process.

APPENDIX F: SOFTWARE DEVELOPMENT TEAM INTERVIEW QUESTIONS:

- Can you please tell me about your current role?

    Prompt question

    - What do you do?

    - Which approach do you use for software development?

- Can you please describe how a new project starts and what process it goes through

    Prompt questions:

    - Please tell me how are the requirements gathered

    - Please tell me how are the requirements refined

    - Can you please tell me about the documentation done for the project

    - How much detail does the documentation include about the project the project?

    - Can you please tell me about the contract process (legal document)

- What happens when there is change in scope after the contract?

3 Can you please tell me about the process of assigning roles on a project?

- Please tell me about how the work is divided

- What do you think is motivating the team member working in group

- Please tell me about the commitment of team members in achieving the project/product goal

4 Can you please tell me about customer involvement in your recently completed project/product?

Prompt questions:

- Please tell me about the communication process with customer/Product owner

- Can you please tell me more about the product owner regarding their knowledge on working in Agile teams?

- Please tell me about some of the challenges you have faced with customer

- How often do you face those challenges

- In your view what are the major challenges with customer/ PO on software project/ product development

- Can you please describe your relationship with customer/product owner

- How do you see the customer developer relationship impact on the project outcome

- Can you please tell me about your current project?

- Please tell me about some of the difficulties you have faced so far.

- What impact do u think those difficulties will make on project outcome?

APPENDIX G: PRODUCT OWNER STUDY QUESTIONNAIRE

The impact of customer development team relationship on Agile software development projects outcome

I would like to invite you to participate in a study to examine the relationship between customers/ Product owners and software development teams in Agile software

development projects. This research aims to understand the impact of the relationship on project outcomes.

This research has two components

- This questionnaire, which should take no longer than 10 minutes to complete
- A follow up interview that will take around 30 to 40 minutes

You can contact me, Dipendra Ghimire (Dipendra.Ghimire@lincolnuni.ac.nz), or my supervisors Stuart Charters (Stuart.Charters@lincoln.ac.nz) and Shirley Gibbs (Shirley.Gibbs@lincoln.ac.nz) if you have any questions about the research.

I have read and understood the description of the above-named project. On this basis I agree to participate as a subject in the project, and I consent to publication of the results of the project with the understanding that anonymity will be preserved. I understand also that I may at any time withdraw from the project, including withdrawal of any information I have provided, up to 15th March 2016.

Please write your email address (Your email address will be used for two purposes. To invite you for interview and to locate your records in case you wish to withdraw from this study)

_____.

In this section questions related to your organization and your experiences of Agile software development projects will be asked

1 What is your current job title?

2 How long have you been involved in IT projects?

3. Have you been a product owner or customer representative for a software development project?

    88 Yes

    89 No

Answer. If have you been a product owner or customer representative for a software development projects? Yes is selected then participant will continue the questionnaire Else will get out of the survey

4. Have you undertaken any product owner training?

    90 Yes

    91 No

Answer If Have you undertaken any training in software development? Yes Is Selected

5. Please list the product owner training you have done.

Please answer the questions below thinking about your most recent experience in a product owner / customer representative role

6. Please rate how often the following occurred with the development team during during a typical sprint

	Never	1-3 times	4- 6 times	7-9 times	More than 9 times	Don't know
Disagreement with the development team regarding project priorities						
Disagreement with development team regarding project requirements						
Disagreement with development team regarding the timeframe of the project						
Interpersonal challenges between you and development team members						
Difficulties in communicating with the development team						

7. Was the project/product completed on time?

Completed early

On time

Completed late

Was Cancelled

Was not completed

I Don't know

8. Was the project/product completed within the specified budget?

    92 Under budget

    93 Within budget

    94 Over budget

    95 Don't know

9. Was the project/project contract met?

    96 Yes

    97 No

    98 I Don't know

10. I am interested in understanding why some projects go well and are successful and others struggle. Can you please describe what you see as the important differences in projects that you have worked on that have been a success vs. those that haven't been?"

11. Which of the following roles related to you as product owner during the project development (Tick as many boxes as are applicable)

    Responsible for creating a product backlog and priorities requirements

    Responsible for what gets built and which activities will produce the most business value

    Communicating with stakeholders as well as the development team members

    Have delegated authority to make the decision regarding the scope of the project.

    The final say on when the software should be shipped

    Other

        Please describe the other roles you do as a product owner

APPENDIX H: PRODUCT OWNER STUDY INTERVIEW QUESTIONS:

1 Can you please tell me about your current role?

      Prompt question

- What do you do?

- Which approach do you use for software development?

2 Can you please describe how a new project starts and what process it goes through

   Prompt questions:

- Please tell me how are the requirements gathered

- Please tell me how are the requirements refined

- Can you please tell me about the documentation done for the project

- How much detail does the documentation include about the project the project?

- Can you please tell me about the contract process (legal document)

- What happens when there is change in scope after the contract?

3 Can you please tell me about the process of assigning roles on a project?

- Please tell me about how the work is divided

- What do you think is motivating the team member working in group

- Please tell me about the commitment of team members in achieving the project/product goal

4   Can you please tell me about customer involvement in your recently completed project/product?

   Prompt questions:

- Please tell me about the communication process with customer/Product owner

- Can you please tell me more about the product owner regarding their knowledge on working in Agile teams?

- Please tell me about some of the challenges you have faced with customer

- How often do you face those challenges

- In your view what are the major challenges with customer/ PO on software project/ product development

- Can you please describe your relationship with customer/product owner

- How do you see the customer developer relationship impact on the project outcome

- Can you please tell me about your current project?

- Please tell me about some of the difficulties you have faced so far.

- What impact do u think those difficulties will make on project outcome?