Factors impacting on tacit knowledge transfer within Scrum software development teams

A Masters Dissertation Presented to the Department of Information Systems

University of Cape Town

in partial fulfilment of the requirements for the INF5005W course

BY:

Deon Takpuie (TKPDEO001)

NRF FUNDED RESEARCH

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the APA convention for citation and referencing. Each contribution to, and quotation in, this literature review of *"Factors impacting on tacit knowledge transfer within Scrum software development teams"* from the work(s) of other people has been attributed, and has been cited and referenced.
3. This technical report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.
5. I acknowledge that copying someone else's assignment, essay or paper, or part of it, is wrong, and declare that this is my own work.

Deon Takpuie (TKPDEO001) Date: 25/11/2014

Acknowledgements

I would like to thank God, for giving me the strength and wisdom to complete this Masters research and for the wonderful people that surrounded me. I pass my heartfelt thanks to **Dr. Maureen Tanner**, my supervisor, for all the direction and guidance given, as well as the continual support throughout the thesis.

I would also like to express my sincere thanks to the Department of Information Systems at the University of Cape Town, Professor Michael Kyobe, our charismatic lecturer, the administrative staff, Commerce IT (Information Technology) staff, and other department staff.

I hereby pass my sincere thanks to my 2013 Information Systems Masters coursework classmates: this journey would not have been as much fun without you. I cannot even believe we have come this far.

I would also like to thank my mother and family and friends, for their support and interest in my thesis, as I knew that I was not alone in this. Especially my mom, without whose sacrifices and encouragement, I would not have made it to where I am today.

Abstract

Over time, there has been a high failure rate of information systems development (ISD) projects, although Agile development has brought recent improvements. Knowledge management is also known to be one of the critical factors to Agile and project success; however, there are some knowledge transfer studies in Agile development. Therefore, the purpose of this research is to present a theoretical model examining what makes individuals successful at transferring knowledge in teams using Scrum, Agile's most popular methodology.

Twelve semi-structured interviews were conducted at two Scrum companies in Cape Town. Participants interviewed ranged from project managers and Scrum masters to software developers, business analyst and testers. The interviews were all transcribed, then analysed using thematic analysis.

The findings produced new relationships between characteristics already well known to impact knowledge transfer as well as newly defined characteristics impacting knowledge transfer in Scrum teams: empathy and articulation skills of the source. The results have shown that the recipient should perceive the person wanting to transfer knowledge as having these characteristics to enable successful knowledge transfer: empathy, motivation, capability, credibility, articulate and ability to communicate enough. The contribution of this study to practice is a list of attributes for HR managers to help improve the knowledge transfer of Scrum team members. The contribution to Scrum research is a new theoretical model that suggests which characteristics a person needs to transfer knowledge successfully in Scrum teams, adapted from Joshi, Sarker and Sarker's (2007) knowledge transfer model. This model can also be extended in the future by looking more deeply into the new relationships between constructs, such as how motivation together with capability of the source affect knowledge transfer in Scrum teams.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Problem Statement

Waterfall software development, the first software engineering methodology (Kasser, 2002), is characterised by its sequential nature and extensive documentation produced throughout the software development process (Boström, Wäyrynen, Bodén, Beznosov, and Kruchten, 2006). However, nearly two-thirds of all IT projects that fail, use the Waterfall methodology, as reported in the 2006 Standish Group Report (Rubinstein, 2007). Moreover, approximately the cost of terminated and failed projects is $80-$145 billion per year (Haas, 2007).

The main reason for these failures was the fact that the Waterfall methodology is inflexible to changing customer requirements. This has led to the emergence of more iterative methodologies, of which Agile methodologies are the most prominent (Larman and Basili, 2003). The term *"Agile"* relates to a family of methodologies characterised by light documentation, iterative and rapid software delivery and quick response to change (Conboy and Fitzgerald, 2004; Larman and Basili, 2003). Although established in the 1990s, Agile methodologies are now becoming mainstream (West, Grant, Gerush, and D'Silva, 2010). The 2010 Standish Group Reports show that between 1996 and 2008 Agile software development projects were around 1.5 times more successful than Waterfall projects, (Group, 2010). One of the key differences between Agile methodologies and Waterfall is the knowledge management principles underpinning them (Chau, Maurer, and Melnik, 2003).

According to Lyntinen and Robey (1999), knowledge management is one of the main remedies to project failures in information systems development (ISD). Knowledge transfer in particular is essential to knowledge management. Knowledge transfer is the process of transmitting a message from a source or sender to a recipient in a specific context (Szulanski, 1995, p.438).

There is limited evidence of success factors to knowledge transfer in general (Levin, Lesser, Cross, and Abrams, 2005). Most published articles recommend successful knowledge management practices without empirical evaluations such as the use of questionnaires or interviews to test claims made (Davenport, David, and Beers, 1998). Knowledge transfer is often

looked at in terms of inter- and intra-organisational transfer, but there has been little research done within Agile and ISD teams on that topic (Zellmer-Bruhn, 2003). ISD teams build information systems to meet business requirements and usually comprise at least programmers, project manager and designers (Joshi, Sarker, and Sarker, 2007). A more in-depth understanding of knowledge transfer in software teams is important for ISD firms to adapt to ever complicated environments (Sense, 2008). Furthermore, knowledge transfer studies in teams using Agile methodologies have been said to lack theoretical underpinnings (Wang and Noe, 2010), so this research aims to address that.

Knowledge transfer is important, particularly in Agile software development as its core principles include the involvement of collocated teams to increase communication and more efficiently disseminate project information (Chau et al., 2003). Indeed, Agile projects emphasise tacit (i.e. difficult to articulate or write down) knowledge transfer through frequent verbal communication and team interaction (Cockburn and Highsmith, 2001) since most software development knowledge is tacit (Chau et al., 2003).

One example of an Agile method that incorporates knowledge transfer throughout the project is Scrum. Scrum is one of the most popular Agile methodology according, to the 7th State of Agile Development survey (VersionOne, 2013). In Scrum, regular Sprint meetings are used to leverage knowledge transfer between the team, Scrum master, product owner and sometimes client (Karlsen, Hagman, and Pedersen, 2011). Knowledge transfer also happens in Scrum through artefacts like the Scrum burndown chart which is always readily available and usable for team members to keep track of progress (Cho, 2008). However, South Africa, in particular, has low stakeholder satisfaction in Agile development projects and case study research in knowledge transfer in these companies has been recommended to mitigate this (Ferreira and Cohen, 2008).

1.2 Research Purpose

As mentioned in section 1.1, there is limited evidence of success factors to knowledge transfer within software teams (Levin, Lesser, Cross, and Abrams, 2005), yet knowledge transfer is a core component of Agile methodologies (Chau et al., 2003) and consequently, project success. Scrum development, which is the most popular Agile methodology (VersionOne, 2013) integrates knowledge transfer throughout a project (Schwaber, 1995). Therefore, the purpose of

this study is to investigate the characteristics of Scrum team members which promote successful tacit knowledge transfer. Indeed, Agile projects emphasize tacit (or verbal) knowledge over explicit (written down) knowledge, as verbal communication has been said to be more effective (Cockburn and Highsmith, 2001). So investigating tacit knowledge might improve the team dynamics of future projects and it will enable human resource personnel to train individuals with the right knowledge transfer skills (Joshi, et al, 2007). The aim is to do this using Joshi, Sarker and Sarker's 2007 model of knowledge transfer within software development teams. The model states that the extent to which a team member is perceived to be capable, credible and has communicated regularly with others, will determine the knowledge they are able to transfer to them. Furthermore, this current research extends the model by investigating the role of motivation in helping a person transfer knowledge. Motivation is said to be essential to knowledge transfer, as a team member might not transfer knowledge if they do not feel motivated enough (Szulanski, 1996). Subsequently, this study has the following main and sub-objectives:

1.3 Research Objectives

Main Objective:

- Develop a theoretical model depicting the perceived characteristics that a Scrum team member should have to successfully transfer tacit knowledge throughout the project.

Sub-objectives:

- Determine how the perceived motivation of a team member affects the perceived extent of tacit knowledge transferred by them.
- Determine how the perceived capability of a team member affects the perceived extent of tacit knowledge transferred by them.
- Determine how perceived credibility of a team member affects the perceived extent of tacit knowledge transferred by them.

- Determine how the perceived communication extent of a team member affects the perceived extent of tacit knowledge transferred by them.
- Determine what other characteristics are associated with a team member that can successfully transfer tacit knowledge in Scrum projects.

These perceived characteristics will be looked at from the recipient's perspective, as tacit knowledge transfer is difficult to measure so it is usually determined by the knowledge that the recipient has perceived to have gained from the transfer (Karlsen et al., 2011).

1.4 Importance of the Research

The contribution to practice of this research is, firstly, for Agile managers/Scrum masters to have a better insight into how to enhance knowledge transfer in Scrum teams (Ko, Kirsch, and King, 2005). It will enable them to identify team members with the right characteristics to initiate knowledge transfer and with the help of human resource personnel to train others with these skills.

The contribution to research of this study is a theoretical model on source attributes affecting knowledge transfer within Scrum teams adapted from Joshi et al. (2007)'s model (Joshi, et al., 2007). Knowledge transfer is one of the key components to Scrum's success, so understanding how it works in Scrum is crucial (Karlsen et al., 2011).

Even though Joshi et al.'s (2007) knowledge transfer model was unique because it was actually specific to software development, it was not specific to Scrum. Therefore, this current research goes further to extend the model to a Scrum-specific setting, which does not seem to have been done before. Since the present research interpretatively generates a new theoretical model of knowledge transfer in Scrum teams, it would still need to be validated quantitatively in the future, maybe through a questionnaire (Levy, 2006). Subsequently, this model can be extended through looking more closely into relationships between constructs, such as motivation and knowledge transfer in Scrum development.

1.5 Research Context

The study has been conducted in two Scrum software development companies in Cape Town, South Africa. Cape Town is one of the three largest metropolitan cities in South Africa, a hub of technological creativity and also known for its diverse cultures (Booyens, 2012).The sample of participants interviewed was also diverse, consisting of twelve Scrum team members with roles varying from software developers, business analysts to Scrum masters and project managers. The team members' level of experience varied from a few years to decades of experience in industries such as finance, banking and manufacturing. Also, the Scrum setting was a co-located environment as Agile promotes team members working together in the same office, as this improves tacit knowledge transfer through communication (Boehm, 2002).

1.6 Research Method

This research was interpretative and qualitative, as the extent of tacit knowledge transferred, which is dominant in Scrum, is difficult to measure quantitatively (Karlsen et al., 2011). The data collection and analysis were guided by an adapted model of knowledge transfer in software development teams by Joshi et al. (2007). The research strategy used semi-structured interviews of Scrum team members in South Africa. Data analysis was done with thematic analysis using ATLAS.ti qualitative coding software.

1.7 Definition of Terms

The various terms that are used repeatedly throughout this paper will be described in Table 1:

Table 1: Definition of terms

Term	Definition
Agile software Development	Is an iterative software development methodology characterized by constant deliverables and high team interaction (Fowler and Highsmith, 2001).
Scrum Software Development	The most popular Agile methodology (VersionOne, 2013) during which software development is organised in Sprints

	which are 1 to 4-week time box periods to deliver software artefacts (Schwaber, 1995).
Term	**Definition**
Scrum team	Is a cross-functional team of programmers, testers and documenters consisting of at least five members (Schwaber, 1995).
Scrum master	The Scrum master, is there to make sure that Scrum is properly implemented, and that its values and rules are enforced within the team (Cho, 2008).
Product owner	The product owner is mainly responsible for conveying customer requirements to the development team and also getting funding for the project (Cho, 2008). For this research, the role of a project owner can be likened to that of a project manager (Sutherland, Schoonheim, Rustenburg, and Rijk, 2008).
Tacit knowledge	Knowledge that is difficult to articulate or write down and is the dominant knowledge type in Scrum development (Droege and Hoobler, 2003).
Explicit knowledge	Knowledge that can be written down and easily transferred (Droege and Hoobler, 2003)

1.8 Outline

This dissertation consists of six chapters. In Chapter 2, previous studies on knowledge transfer are explored, from which a theoretical model is derived, as well as research questions and objectives. In Chapter 3, the research methodology is described, paying attention to important

issues such as bias, reliability and validity. Chapter 4 presents the empirical findings of the study in light of the research questions. Chapter 5 discusses the findings in the context of the research questions, literature and their subsequent implications to theory and practice. Lastly, Chapter 6 will conclude the thesis by restating the aim and what was achieved and then pointing to future research.

2 Literature Review

To begin the research, a literature review was undertaken in order to review theoretical work and identify gaps in the literature to be addressed in this study (Hart, 2001). The literature review will first start in the broader context of software engineering and then explain the various software development methodologies, finally ending at Scrum development. Thereafter, fundamental concepts of knowledge, knowledge management and knowledge transfer will be covered. This is to be followed by Agile and Scrum development studies on knowledge transfer, a review of relevant theoretical work and a presentation of a theoretical model to address the gaps in the literature and theory. Then, the research questions and objectives that guided the research are generated from this model. The structure of the literature review was largely based on a similar study by Karlsen et al. (2011) on knowledge transfer in information systems development teams. This structure is shown in Figure 2-1 below.

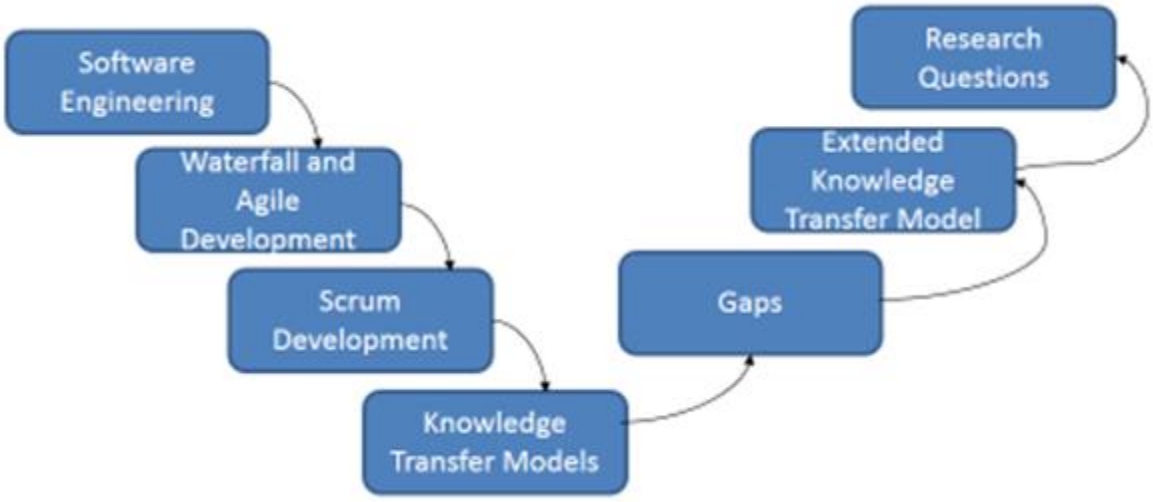


Figure 2-1: Literature review structure, adapted from Karlsen et al. (2011).

2.1 Software Engineering

Software engineering has been defined as the *"disciplined application of engineering, scientific and mathematical principles and methods to the economic production of quality software"* (Humphrey, 1988, p.82). This comprises a set of tasks that transfer customer specifications into working software, known as the software engineering process (Humphrey, 1988; Pressman and Ince, 1992). Some authors feel sound software engineering principles should lead to project

success (Cunningham, 1999; Miller and Mingins, 1998). However, following software engineering principles does not always lead to project success and this can be seen from the shortcomings of Waterfall development methodology which led to the formation of the Agile methodologies (Larman and Basili, 2003). These two software engineering methodologies will be discussed in turn.

2.1.1 Waterfall Methodology

The Waterfall methodology is the earliest methodology to document the software engineering/development process (Kasser, 2002). It first emerged in the literature in the 1970s, as a formal and sequential description of how to develop large software systems (Royce, 1970). Indeed, it is renowned for its sequential nature and extensive documentation throughout the software engineering process (Boström et al., 2006). Figure 2-2 below illustrates the linear nature of the Waterfall methodology, starting from planning through to maintenance without any iteration:

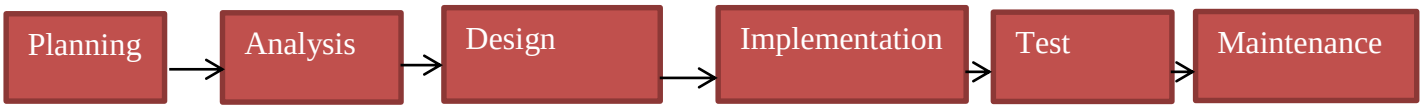


Figure 2-2: The Waterfall Model *(Royce, 1970)*

The Waterfall model is useful when customer requirements are well understood (Kasser, 2002), but this is not often the case (Burns and Klashner, 2005; Schwaber, 1995). Indeed, this lack of response to change and agility is one of the main criticisms of the model (Nerur, Mahapatra, and Mangalaraj, 2005; Schwaber, 1995). Approximately two-thirds of all IT projects which use the Waterfall methodology either fail or experience serious challenges, as reported by the Standish Group, 2006 (Rubinstein, 2007). Approximately $80-$145 billion per year is the cost of these terminated and failed projects (Haas, 2007). The next methodology to be described aims to address these project failures due to a lack of agility and responsiveness to change (Awad, 2005).

2.1.2 Agile Methodologies

Agility has been defined as the ability to detect and react quickly (Conboy and Fitzgerald, 2004) to technical changes and business opportunities (Lyytinen and Rose, 2006). This ability to adapt to business opportunities is shown in the iterative nature of agility where

software is re-developed and tested until the customer is satisfied (Boehm and Turner, 2005). This is in contrast with the linear flow of Waterfall and this is shown visually in Figure 2-3, below:

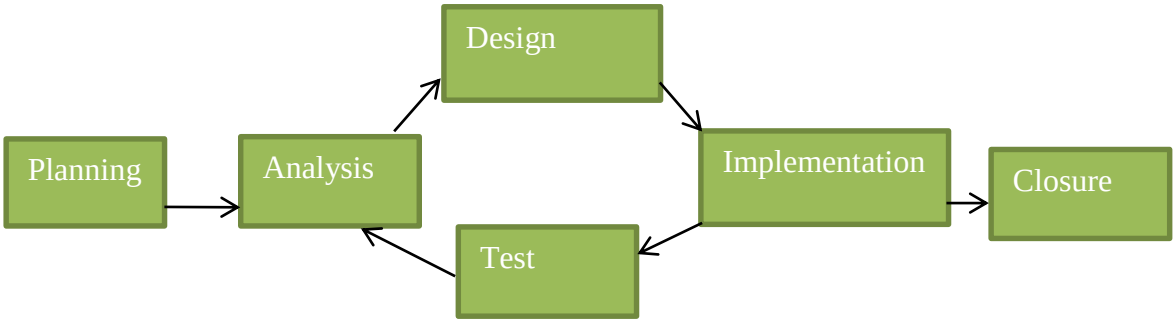


Figure 2-3: Typical Agile development flow (Huo, Verner, Zhu, and Babar, 2004; Schwaber, 1995)

Furthermore, agility about being light (especially in software documentation) but barely sufficient (Cockburn, 2007). These core characteristics are the basis for Agile methodologies, such as Scrum (Schwaber, 1995), XP (eXtreme Programming) (Beck, 1999) and Crystal(Cockburn, 2004). These methodologies arouse in the 1990s (Larman and Basili, 2003) but became formally described by a group of software developers only in 2001, in the Agile Manifesto (Fowler and Highsmith, 2001). The main principles of the Agile Manifesto are software agility (rapid delivery whilst responding to rapid changes in customer requirements), team autonomy (self-organized teams that make their own decisions) and team diversity (broad experience and roles within teams) (Lee and Xia, 2010). Pair programming, is a principle, from XP but used in many other Agile methodologies such as Scrum (Sutherland, Viktorov, Blount, and Puntikov, 2007) as it entails two software developers or team mates working side by side,

solving the same programming problem and sharing knowledge in the process (Chau et al., 2003).

Some of the main benefits of Agile methodologies reported in the literature and industry are high responsiveness to change and rapid delivery of software, but many times these claims are not empirically validated (Dybå and Dingsøyr, 2008; Lee and Xia, 2010). Some of the main challenges in Agile methodologies from case studies, are in aspects of little documentation, communication barriers due to physical distance and working environment not facilitating Agile practices (Cho, 2008; Kajko-Mattsson, 2008). However, the 2010 Standish Group Reports show that agile software development projects have been around 1.5 times more (see Figure 2-4) successful than Waterfall projects, from their 1994 to 2008 database (Group, 2010).

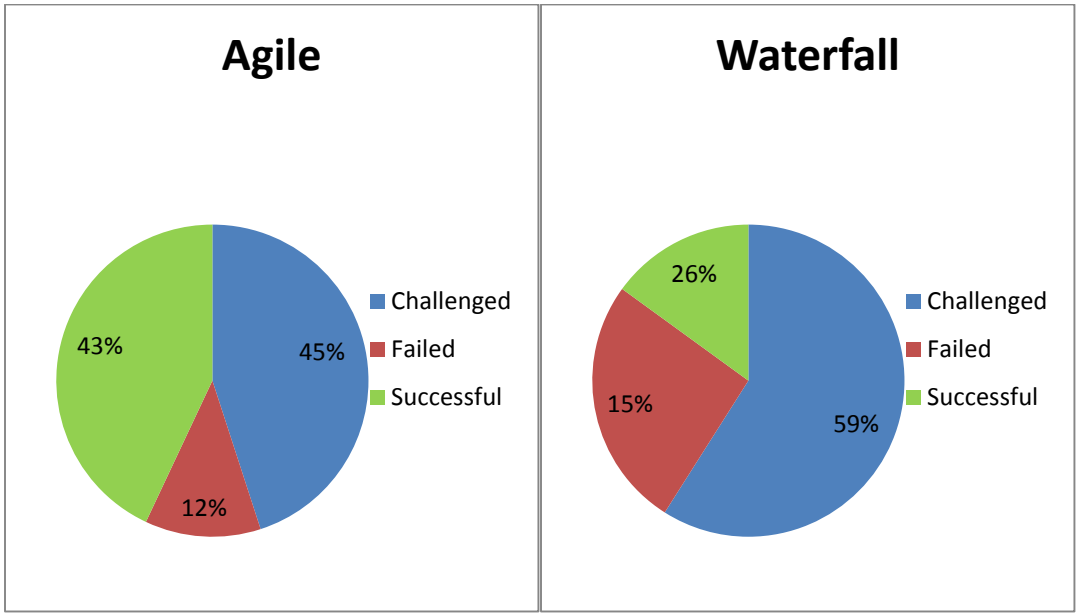


Figure 2-4: Agile vs Waterfall Projects from Chaos Project Databases from 1994 to 2008 *(Group, 2010)*

Indeed these reports have been criticised by some. These reports are said to only consider the three main project success criteria: whether the project was on time, on budget and met user requirements but excludes other project success criteria like customer satisfaction and risk. However, it is still the most prolific industry report about software project success (Dominguez, 2009).

To follow is a summary of Waterfall and Agile, in Table 2-1 and then a further explanation of one of Agile's most popular methodologies, Scrum development due to its suitability for industry (Cardozo, Neto, Barza, França, and Da Silva, 2010).

Table 2-1: Waterfall versus Agile software development summary

Waterfall	**Agile**
Sequential	Iterative
Plan orientated and not responsive to change	High customer collaboration and responsiveness to change
High Documentation	Low Documentation

Adapted from (Boehm, 2002; Awad, 2005)

2.1.3 Scrum Methodology

Takeuchi and Nonaka (1986) were the first to observe the Scrum approach used in highly productive small development teams such as Xerox, Canon and Honda (Takeuchi and Nonaka, 1986). They argued that the Scrum process be done by a cross-functional team spanning many overlapping stages, where the team *"tries to go the distance as a unit, passing the ball back and forth"* (Takeuchi and Nonaka, 1986, p.137).

Therefore, the name, 'Scrum', coined by Ken Schwaber (1995), is derived from the game of rugby, as rugby originated as a breakaway from soccer by adapting to the environment. So too, Scrum is meant to adapt to external environments a development project progresses (Schwaber, 1995).

Scrum is an empirical Agile project management methodology that reacts to constant change in an environment based on industrial process control theory (Paetsch, Eberlein, and Maurer, 2003; Schwaber, 1995). It is empirical in the sense that it treats the analysis, design and development phases as unpredictable and therefore needs a control mechanism. Each cycle of this set of phases is controlled using risk management techniques, unlike the Waterfall methodology which ignores the unpredictability of these phases (Awad, 2005; Schwaber, 1995).

The major roles in Scrum are product owner, Scrum master and development team. The product owner is mainly responsible for conveying customer requirements to the development team and also getting funding for the project. The Scrum master is important for enforcing the Scrum rules, values and principles. Furthermore, the Scrum master also tries to deal with obstacles faced by the development team (Cho, 2008). The development team (of usually 3 to 6 members) typically consisting of programmers, quality control testers and documenters, is responsible for implementing these customer requirements within the given time constraints and quality demands (Cho, 2008; Schwaber, 1995). This cross-functional/inter-disciplinary team is also meant to be self-organising and self-managing. This means the teams are autonomous in that they do their own task planning, decision making and assigning tasks to each other (Lee and Xia, 2010; Takeuchi and Nonaka, 1986).

The following Figure 2-5 summarises the Scrum process described here. An immediate difference between this and the Waterfall model is the iterative Agile nature.

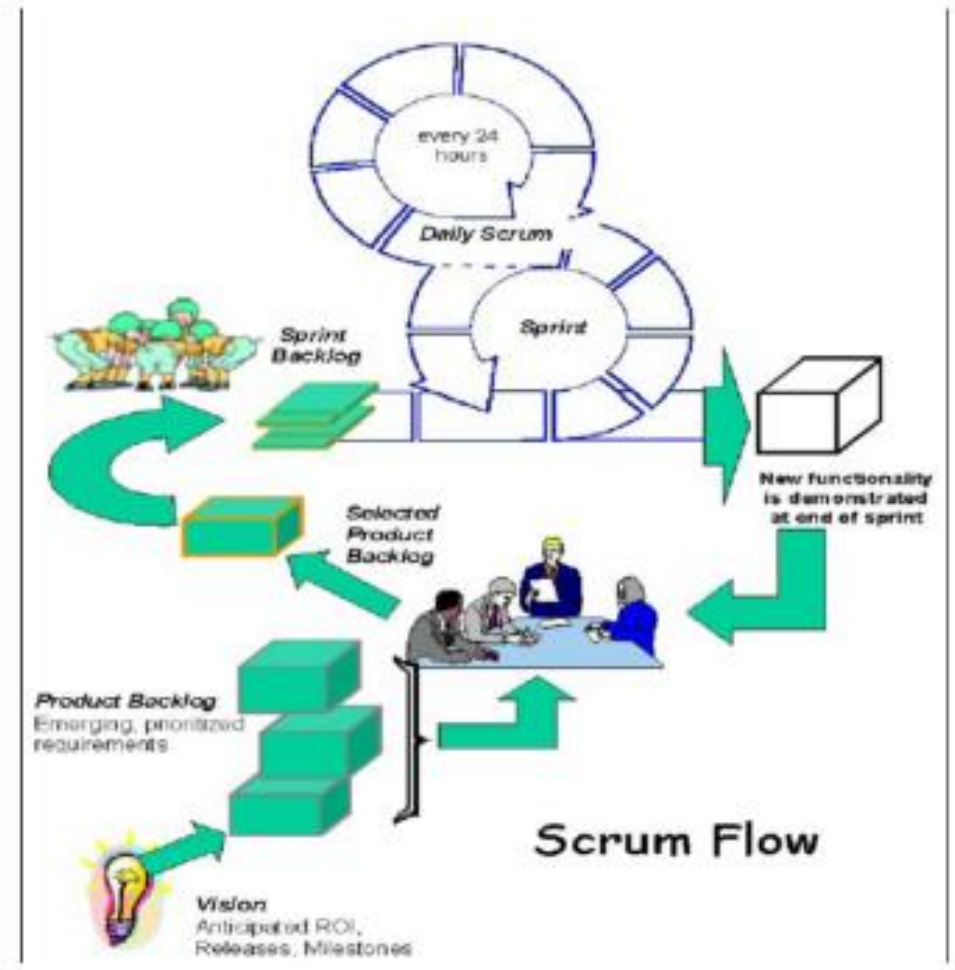


Figure 2-5: The Scrum process flow (Cho, 2008; Schwaber, 1995)

The Scrum process starts, a general project vision is set by the stakeholders/client, then this is streamlined into a prioritised list by the product owner, called the product backlog, of all the necessary product requirements such as features, functions and bugs to fix (Cho, 2008; Paetsch et al., 2003). Thereafter, a set of the most highly prioritised tasks are selected from the product backlog and put into the sprint backlog. These tasks, also known as user stories (Cohn, 2004; Mahnič and Hovelja, 2012) must be accomplished within the Sprint cycle which will further divide tasks to be done each Daily Scrum. At the Daily Scrum meeting, the development team report what they have done, any issues and planned work for next meeting, with the Scrum master (Cho, 2008). Thereafter, the accomplished tasks are demonstrated as a software prototype

during the Sprint Review Meeting, to the product owner and the product backlog is then updated and the cycle continues until the client is satisfied with the product (Cho, 2008; Paetsch et al., 2003) or the client cancels the project (Racheva, Daneva, and Buglione, 2008).

Furthermore, the Scrum development process has been empirically shown in case studies to improve communication in teams (Petersen and Wohlin, 2009) and also to increase customer satisfaction (Mann and Maurer, 2005). However, many case studies also report drawbacks to the Scrum process such as a lack of documentation to transfer knowledge from old to new software developers (Cho, 2008; Kajko-Mattsson, 2008), and the difficulty to actually get regular client feedback due to their busy schedules (Karlsen et al., 2011). Most of these types of issues are knowledge-related issues, and this concept will be discussed next.

2.2 Knowledge Management

This section will cover the fundamental aspects of the study of knowledge in the context of knowledge management. Knowledge transfer, falls under the field of knowledge management, making knowledge management an essential foundation to this study (Davenport and Pruzak, 2000). Therefore, the field of knowledge management will also be introduced along with its various schools of study and types of knowledge.

2.2.1 Introduction

In the literature many authors distinguish between data, information and knowledge (Karlsen et al., 2011). Data seems to be clearly defined in most fields and is usually known as a set of facts, observations or measurements which have no value on their own. An example of data could be a list of all the customer records in a database (Davenport et al., 1998).

Information has been defined as data that has been processed with the knowledge of people to be given relevance and purpose (Drucker, 1988). An example of information, following from the data example, could be a list of all the customers that have made a purchase in the last 30 days (Rygielski, Wang, and Yen, 2002). However, knowledge itself is not simple to explain as it has different meanings in different context (Dingsøyr, 2002), but this study will use the research of

knowledge management authority (Dyba and Dingsøyr, 2009), Torgeir Dingsøyr , to define what knowledge is, using the Oxford Dictionary (Dingsøyr, 2002).

The online Oxford Dictionary (2013) defines knowledge as *"facts, information, and skills acquired through experience or education; the theoretical or practical understanding of a subject)" or "awareness or familiarity gained by experience of a fact or situation"* (Oxford Dictionary, 2013). This is in line, with many knowledge management scholars, such as Davenport et al. (1998), who define knowledge as information combined with experience, interpretation and reflection that enables the person to benefit from new information and experiences. Nonaka and Konno (1998) said that knowledge consists of two major components: tacit and explicit.

2.2.2 Knowledge Management

Knowledge management was first introduced as a term in a European Management conference in 1986 (Wiig, 1997). Knowledge management has been defined by Davenport et al. (1998) essentially as a technique to easily create, capture, share/transfer, and utilise knowledge in a company so as to better competitive advantages (Davenport & Pruzak, 2000). Indeed, a study of 70 international companies found knowledge management to be one of the key factors to project success (Cooke-Davies, 2002). However, since knowledge management is an emerging discipline some feel that there is no consensus in terms of classifying models (McAdam and McCreedy, 1999; Timbrell, Delaney, Chan, Yue, and Gable, 2005). For example, some categorise knowledge management into knowledge category models, intellectual capital models and social models (McAdam and McCreedy, 1999).

2.2.3 Schools of Knowledge Management

Yet, the two types of knowledge management seem to be more typically defined as technocratic schools and behavioural schools, in a systematic literature review of knowledge management in software engineering (Bjørnson and Dingsøyr, 2008).

The technocratic school sees knowledge as an object that can be created by information systems and retrieved by people. The behavioural school sees the human and social aspects key to

creating and managing knowledge although information systems can also play a part (Bjørnson and Dingsøyr, 2008; Karlsen et al., 2011).

The behavioural or people-centred track of knowledge management is particularly relevant in the case of Agile development where most of the knowledge transfer is verbal and tacit knowledge (Cockburn and Highsmith, 2001), requiring people skills. Indeed, knowledge transfer is a particularly important aspect of knowledge management and well-studied in the management and psychology literature (Argote, Ingram, Levine, and Moreland, 2000). The fundamental types of knowledge used in knowledge management (and therefore also in knowledge transfer) will now be discussed.

2.2.4 Types of Knowledge in Knowledge Management

Tacit knowledge is knowledge that is naturally difficult to articulate/write down as it is gained by an individual from experience (Nonaka and Konno, 1998; Venzin, M., von Krogh, G., and Roos, 2000) and this complexity makes it the focus of most knowledge management studies (Kyobe, 2010). Tacit knowledge is the dominant type of knowledge in Agile (including Scrum) methodologies (Karlsen et al., 2011). This is because these methodologies focus more on face-to-face communications to transfer information (Hoda, Noble, and Marshall, 2012), which many fields such as human computer interaction and psychology see as a very effective form of communication (Melnik and Maurer, 2004). An empirical psychology study of dialogues between 64 undergraduate students found face-to-face communication to be more effective than video-facilitated communication (Doherty-Sneddon et al., 1997). Also, since tacit knowledge is not written down, it makes it more difficult to imitate by competitors, but this also makes it more difficult to transfer than explicit knowledge (Droege and Hoobler, 2003).

Explicit knowledge is knowledge that can easily be written down and documented in such mediums like manual, reports, process guidelines and databases (Goh, 2002). Though, the codification/written-down nature of this knowledge also makes it easy for competitors to copy, if it is not protected (Droege and Hoobler, 2003). Waterfall software development focuses heavily on the transfer of explicit knowledge in the form of extensive process documentation (Bahli and

Zeid, 2005). However, explicit knowledge also needs to be managed in an effective way in order to be useful to a company (Zack, 1999).

Managing knowledge (both tacit and explicit) in organisations is the focus of the knowledge management which has been recommended by information systems (IS) authorities Lyytinen and Robey (1999) as a key remedy to IS project failures (Lyytinen and Robey, 1999).

2.3 Knowledge Transfer

This section will be covering knowledge transfer fundamentals, the various epistemological stances and its current use in Agile development studies. This will lead on to the discussion of new applications of theoretical work to Agile (in particular Scrum) development studies, in the next section.

2.3.1 Introduction

The study of knowledge transfer originated in the 1960s when product knowledge of products in the United States of America (USA) was documented on card books and sent to resellers in Japan. Thereafter, knowledge transfer has been studied at the individual, group and organisational level.

Indeed, one definition of knowledge transfer is: *"Knowledge transfer in organisations is the process through which one unit (e.g. individual, group or organisation) is affected by the experience of another"*(Argote and Ingram, 2000, p.151). The term 'affected' here, is fairly broad, but a similar and more explicit definition is: *"the knowledge transfer between two or more actors (individuals or organisations) can be defined as the process by which the knowledge of one actor is acquired by another"* (Albino, Garavelli, and Schiuma, 1998, p.54).

However, a more suitable definition of knowledge transfer for this research is *"communication of knowledge from a source to a recipient so that it is learned and applied by the recipient"* (Ko et al., 2005, p.62). Indeed, this is similar to the previous ones, but it also emphasizes how the knowledge has been re-applied, which is vital to knowledge transfer success (Szulanski, 1996).

Furthermore, this is also the chosen definition because Agile methodologies emphasizes knowledge transfer through communication (Cockburn and Highsmith, 2001), so this is a suitable definition for this research. On a more conceptual level, knowledge transfer is studied through epistemological and ontological models (Gebert, Geib, Kolbe, and Brenner, 2003).

Epistemological models focus on the nature knowledge transfer in isolation of environmental factors (Kyobe, 2010). Epistemology is concerned with understanding the true nature of knowledge/something and believes that it is definable (Cunningham and Fitzgerald, 1996). It recognises that knowledge transfer is naturally difficult and breaks it down into two categories (Kyobe, 2010). Knowledge that is easy to document and transfer is known as explicit knowledge(Goh, 2002) whilst knowledge that is difficult to transfer is known as tacit knowledge (Nonaka and Konno, 1998).

On the other hand, ontological models examine knowledge with respect to its environment, irrespective of its nature (Abdel-Moteleb and Woodman, 2007). Ontology in philosophy is the study of things that exists therefore it takes a more holistic approach than epistemology (Chandrasekaran, Josephson, and Benjamins, 1999). In terms of examples, an epistemological model such as Nonaka and Takeuchi (1995) looks at the nature of knowledge creation through the interaction of tacit and explicit knowledge, but omits the ontological dimension which would examine this knowledge creation at various environmental levels (such as individual, group and organizational levels) (García Muiña, De Castro, and López Sáez, 2002).

Some say it can be favourable to examine both the epistemological and ontological perspectives for a more holistic study (Onions and Orange, 2002). However, this research will examine knowledge transfer from a epistemological perspective as other researchers have also argued the importance of viewing knowledge management from this perspective (Christensen and Bang, 2003) and in the context of information systems research (Joshi et al., 2007). Therefore the various epistemological perspectives of knowledge transfer will now be discussed.

2.3.2 Epistemological Perspectives

Knowledge transfer issues are looked at from the following epistemological perspectives: connectionist, cognitivist, and autopoietic (Joshi, Sarker, and Sarker, 2007).

The connectionist perspective believes knowledge exists in the connections of experts/people and views knowledge as being contextual and non-universal therefore naturally difficult to convey between source and recipient (Kogut and Zander, 1992). Whereas the cognitivist perspective believes knowledge is a fixed and representable entity called data thus views knowledge as being universal usually stored in databases, manuals and computers thus easily transferable from source to recipient (Venzin, von Krogh, and Roos, 2000). However, the autopoietic perspective believes knowledge exists in the mind, body and social system (Venzin, von Krogh, and Roos, 2000) and differs from both connectionist and cognitivist perspectives in that it does not believe that knowledge can ever be shared, but rather it is only created (Maturana and Varela, 1980) . This is because knowledge is seen as subjective and can mean different things to different people, a 'justified true belief' (Nonaka and Takeuchi, 1995, p.21). Since autopoietic perspective deals with knowledge creation and not knowledge transfer, so its theories such as the very popular Nonaka's theory of knowledge creation are not applicable to this study. Indeed, none of the views are more superior than the other however since information systems (IS) is a social discipline the connectionist perspective to knowledge transfer is typically used as it recognises human connections (Hirschheim and Klein, 1989), but cognitive theories have also been used to explain knowledge transfer in information systems (Jacobson and Spiro, 1995). To follow is a summary of these epistemological perspectives of knowledge transfer, in Table 2-2.

Table 2-2: Summary of Knowledge Transfer Epistemological perspectives

Epistemologies	Connectionist	Cognitivist	Autopoietic
Belief of knowledge	Knowledge is contextual and non-universal therefore difficult to convey (Kogut and Zander, 1992) .	Knowledge is universal data usually stored in databases and computers hence easily transferable (Venzin, von Krogh, and Roos, 2000).	Knowledge exists in the mind and therefore can never be shared but rather created (Ikujir Nonaka and Takeuchi, 1995).

2.3.3 Knowledge Transfer in Agile teams

There is a lack of studies of knowledge transfer in information systems development (ISD) teams and more so, Agile development (Karlsen et al., 2011). However, there are many studies on knowledge transfers in distributed Agile teams (Da Silva, Costa, França, and Prikladinicki, 2010; Holz and Maurer, 2003; Hossain, Babar, and Paik, 2009; Jalali and Wohlin, 2010). This is because distributed Agile teams are seen to have more difficulties in transferring tacit knowledge as they are not co-located (Kontio, Hoglund, Ryden, and Abrahamsson, 2004). A systematic literature review of distributed software development found effective communication, cultural differences and co-ordination to be the three main knowledge transfer barriers (Da Silva, Costa, França, and Prikladinicki, 2010). Yet another global software development literature review reported knowledge transfer barriers which are applicable to both global and local teams to be: relationship between sender and receiver, lack of recognition/rewards and time constraints of employees (Wendling, Oliveira, and Maçada, 2013). However, no systematic literature reviews could be found on Agile/Scrum development in co-located/local teams.

Although, there was rare study of knowledge transfer in co-located Scrum teams found (Karlsen et al., 2011). This case study examined an information technology services company in Norway with over 100 employees working in system development and integration as well as infrastructural projects. The research methods used included in-depth interviews (analysed with content analysis) of various experienced Scrum team members such as project manager, Scrum

master and development team members. This was supplemented by observations and questionnaires. The main findings were that Scrum teams did in fact use minimal but quality documentation to transfer knowledge and the rotation of team roles (self-organisation) made it easier for team members to acquire different types of project knowledge (Karlsen et al., 2011). However, a limitation to this is that it did not use a theory or theoretical framework to guide the case study, but just made reference to the Scrum literature, in general. Furthermore, it did not come up with a framework of the findings. In addition, only 20% of knowledge management studies are said to have theoretical underpinnings (Wang and Noe, 2010). Therefore, the following section will cover theoretical work that can be applied to knowledge transfer within Scrum teams.

2.4 Communication and Knowledge Transfer

Most of the literature does say the explicit knowledge (like technical/software knowledge) is transferred through written communication, but tacit knowledge (like managerial knowledge such as leadership and relationship skills) is transferred through face-to-face communication (Chau et al., 2003; Joshi et al., 2007).

Likewise, in Agile methodologies, face-to-face communication is emphasised as this is said to be the best way to achieve knowledge transfer in teams, using Media Richness theory. Indeed, it has been found in a renowned enterprise resource planning (ERP) knowledge transfer study that the greater the communication skills of an individual, the greater the knowledge that will be transferred by that individual. These communication skills, are both the ability to communicate clearly (encoding competence) and the ability to listen and re-act appropriately (decoding competence).

Indeed, as mentioned above, this study views knowledge transfer as the *"communication of knowledge from a source to recipient so that it is learned and applied by the recipient"*(Ko et al., 2005, p.62). So, this study will look at communication attributes that enable successful knowledge transfer.

2.5 Motivation and Knowledge Transfer

2.5.1 Introduction

Motivation is what moves a person to do something. Self-determination theory, an influential theory in motivation, distinguishes between autonomous and controlled motivation (Ryan and Deci, 2000). A person can either be motivated to do a task, by the task itself (intrinsically/autonomously) or by factors other than the task such as money and peer recognition (extrinsically/controlled) (Ryan and Deci, 2000). Likewise, in knowledge transfer, people are either motivated to transfer knowledge because of interest in the task itself or because they have been incentivized by monetary or social rewards (Ko et al., 2005). Indeed, motivation is a key aspect to knowledge transfer in Agile development teams (Chau et al., 2003), so this will also be looked at in this section.

2.5.2 Intrinsic Motivation and Knowledge Transfer

Intrinsic motivation has been said to be important to tacit knowledge transfer, which is the focus of Agile team's (Karlsen et al., 2011), in organizations, as managers need to rely on employee's natural desire and intrinsic motivation to transfer unseen/tacit knowledge between each other (Osterloh and Frey, 2000). As a result, intrinsic motivation has also been found to increase knowledge transfer in enterprise resource planning (ERP) projects which has predominantly tacit and difficult to articulate knowledge such as project management and systems integration knowledge (Ko et al., 2005).

Also, intrinsic motivation has also been to influence knowledge transfer more strongly than extrinsic motivation, such as the motivational crowding out concept, which says that if someone is incentivized to do something, they will have less interest in it and potentially work less hard at it (Frey and Jegen, 2001). For example, in the previously mentioned study of knowledge transfer between ERP consultants and clients, intrinsic motivation was said to have an influence on knowledge transfer, but extrinsic motivation had none (Ko et al., 2005). Nonetheless, extrinsic motivation such as social recognition have been said to increase knowledge transfer, in some instances (Bock, Zmud, Kim, and Lee, 2005; He and Wei, 2009; Hung, Durcikova, Lai, and Lin, 2011) and will be discussed next.

2.5.3 Extrinsic Motivation and Knowledge Transfer

People are often also motivated to transfer knowledge based on external rewards, especially such as money and social/peer recognition (Bock, Zmud, Kim, and Lee, 2005; Hung, Durcikova, Lai, and Lin, 2011). Financial rewards have often been said to successfully to motivate employees to transfer knowledge, however many recent studies have found this not to be the case (Bock, Zmud, Kim, and Lee, 2005; He and Wei, 2009; Hung, Durcikova, Lai, and Lin, 2011). Furthermore, financial rewards have also been said to be controlling of people and remove the purpose of the activity in the first place.

Therefore, social rewards such as badges or the possibility of being recognized as an expert in a field, have also been seen as good motivators for knowledge transfer (Hung et al., 2011; Schenk and Lungu, 2013). For, example, in a study of 140 senior students at a Taiwanese university, it was the feedback and reputation benefits that people got from transferring knowledge that increased knowledge transfer, not financial rewards (Hung et al., 2011). In yet another example, a question and answer forum called StackOverflow, which generates over 100 000 answers a year, rewards users with reputation feedback mainly through virtual badges for answers given (Schenk and Lungu, 2013).

2.5.4 Motivation and Knowledge Transfer in Teams

Motivation is not just important for individual knowledge transfer, but also knowledge transfer within teams, as a whole (McHugh, Conboy, and Lang, 2011). Teams are said to be a social unit in which members of the team transfer/share knowledge and learn from each other (Bereby-Meyer, Moran, and Unger-Aviram, 2004). Therefore, knowledge transfer is key to teams functioning well, and motivation has already been shown to be key to knowledge transfer (Ko et al., 2005).

Knowledge transfer has often been a problem in teams due to lack of motivation, therefore incentives/extrinsic motivation have often been used to promote knowledge transfer in teams (Bock et al., 2005; Davenport et al., 1998; Hung et al., 2011). As mentioned, they sometimes are successful and other times not.

Most of the knowledge in software development teams, is tacit and therefore difficult to articulate/write-down (Ko et al., 2005; Ryan and O'Connor, 2013). So intrinsic motivation is important, for team members to transfer this unseen knowledge to each other, as it may be difficult for mangers to reward this unseen knowledge transfer (Osterloh and Frey, 2000).

Motivational needs amongst team members also differs slightly between co-located and global software development teams (Boland and Fitzgerald, 2004). Due to the geo-graphical distance in global teams, extrinsic motivation measures are often necessary to get the team to transfer knowledge. For example, team-based rewards (like free movie tickets for the entire team) as opposed to individual rewards have been used to reward exceptional team performance, to foster greater team-spirit and trust, which usually leads to greater knowledge transfer (Gupta, 2008; Hertel, Konradt, and Orlikowski, 2004).

2.5.5 Motivation and Knowledge Transfer in Agile and Scrum Teams

The motivation of an Agile team has always been said to be key to their knowledge transfer (McHugh et al., 2011; Nikitina, Kajko-Mattsson, and Strale, 2012), as motivation is one of the core principles in the Agile manifesto (Fowler and Highsmith, 2001). Both intrinsic and extrinsic motivation appear to influence knowledge transfer in Agile teams (McHugh et al., 2011).

Intrinsic motivation to transfer knowledge, for instance in Agile teams, through making the process for knowledge transfer more enjoyable, for instance, by letting people program together with Extreme Programming (XP) processes like pair programming (McDowell, Werner, Bullock, and Fernald, 2003; Salleh, Mendes, and Grundy, 2011). Indeed, these do work, as Agile teams have been said to experience better knowledge transfer than Tayloristic (including Waterfall) teams (Chau et al., 2003; Melnik and Maurer, 2004). Also, the cross-functionality of Agile teams, such as the variety of roles in which team members have to operate in, have also been said to improve knowledge transfer in Agile teams (Karlsen et al., 2011). This means that team members from different roles such as business analyst can also get involved in software development (Chau and Maurer, 2004) and software developers, can also gather requirements from the customer (Cho, 2008). This spreads the project knowledge throughout the team and reducing risk if a team member happens to leave the organization (Benedicenti and Paranjape, 2001).

Extrinsic motivation to transfer knowledge within Agile teams, such as peoples' need for social recognition for contributions made in meetings, have also been said to improve knowledge transfer in these settings (McHugh et al., 2011). Also, Agile teams increase the sense of autonomy for team members, as they are allowed to direct the projects themselves, and this sense of autonomy, gives them a sense of satisfaction allowing them to perform better on tasks such as knowledge transfer and collaboration (Alavi, Kayworth, and Leidner, 2006; Barney, Moe, Dybå, Aurum, and Winata, 2009).

Within Scrum teams, practices such as the daily Scrum meeting, as well as sprint planning and review meetings have been shown to improve individual's intrinsic motivations to transfer knowledge within Scrum teams as they seem to find them enjoyable (Karlsen et al., 2011; Melnik and Maurer, 2004).

The opportunity to provide feedback in daily Scrum meeting in order to be recognized as an expert and to help the team collectively meet their work targets, are extrinsic motivators that also enable knowledge transfer within Scrum teams (McHugh et al., 2011).

2.6 Relevant Theoretical Work

This section will discuss the various theoretical works that can be used to find factors that affect knowledge transfer in Scrum teams. Firstly, connectionist theories will be discussed as they are more relevant to information systems studies followed by cognitive theories which also have some potential. This section will also show the gaps in the current theoretical work which will be addressed later in the literature review.

2.6.1 Connectionist Perspective

The connectionist perspective as mentioned, sees knowledge as contextual and naturally difficult to convey (Venzin, von Krogh, and Roos, 2000). Only models based on communication theory within the connectionist perspective will be discussed as these theories are relevant to information systems studies (Joshi et al., 2007).

Communication theory originates from many theories, namely: mathematical theory of communication (Shannon, 1948), theory of mass communication (Schramm, 1971), and a model

of communication (Berlo, 1960). Communication theory in knowledge transfer is *"transfer of knowledge is likened to the transmission of a message from a source to a recipient in a given context"* (Szulanski, 1995, p.438). This is based on the work of Shannon (1948) in which the author was explaining machine-to-machine communication and assumed there were no barriers in communication. Therefore, this would initially be considered a cognitivist theory (Joshi et al., 2007) however later works on communication theory considered human-to-human communication (Schramm, 1971) and found communication to have natural difficulties/barriers, making it fit a connectionist perspective (Joshi et al., 2007).

The two models to be introduced – Szulanski (1996) knowledge impediment model and Joshi et al. (2007) knowledge transfer within information systems teams – are underpinned by communication theory.

2.6.1.1 Model of Knowledge Impediments within Firms

Szulanski (1996) model looks at knowledge transfer impediments within the firms/companies and is one of the most influential in knowledge management literature (G. Timbrell et al., 2005). This model is mainly based on communication theory and says a *"transfer of knowledge is likened to the transmission of a message from a source to a recipient in a given context. Characteristics of the message or the situation that limit the amount of knowledge that can be transferred render the transfer stickier"* (Szulanski, 1995, p.438). Szulanski's model provides the different types of knowledge stickiness that occurs in a practice and their effects on the different knowledge transfer stages (Elwyn, Taubert, Kowalczuk, et al., 2007). The four different stages of knowledge transfer are: initiation, implementation, ramp-up and integration (Szulanski, 1996).

The initiation stage consists of all the activities that precede the knowledge transfer and the stickiness here relates to difficulties in finding opportunities to transfer knowledge. The implementation phase is when the decision to transfer the knowledge is made and stickiness here is the technical and communication differences between the sender and receiver. The ramp-up phase starts when the recipient begins using the transferred knowledge and stickiness that occurs in this stage is related to casual ambiguity, which is the difficulty in discerning how to successfully use the knowledge. Then, the integration stage occurs when satisfactory performance is achieved as a result. Stickiness that occurs in this stage is the difficulties or adjustments made to the company/team in using the knowledge and can result in a reversion to

previous knowledge practices (Szulanski, 1996). These stages are also illustrated in Figure 2-6 below:

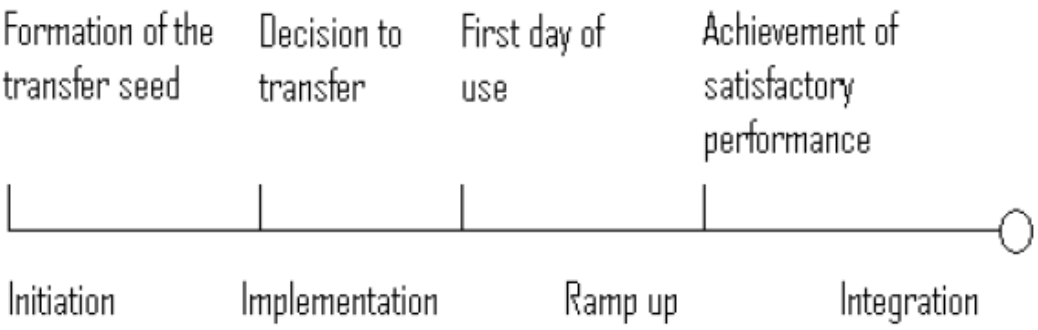


Figure 2-6: Stages of knowledge transfer *(Szulanski, 1996)*

There also four types of knowledge stickiness in the model, namely: knowledge, source, recipient and context (Szulanski, 1996). Here, the source or recipient can be a Scrum development team member, or the Scrum master or product owner (Karlsen et al., 2011). However, a field study of 17 large information systems development (ISD) projects found the most important characteristics for ISD organisations are source and (relational) context (Curtis, Krasner, and Iscoe, 1988) and therefore some authors just focus on these (Joshi et al., 2007).

The source characteristics of knowledge transfer in Szulanski's (1996) model are the: source motivation and credibility/perceived reliability of source by recipient. The lack of motivation of a source may impede him/her from sharing knowledge for reasons such as losing control over the knowledge and potentially making their jobs easier to be replaced. Also, the lack of rewards or incentives is a common deterrent for a person to not transfer knowledge (Cress and Martin, 2006; Siemsen, Balasubramanian, and Roth, 2007). However, even if the source engages in knowledge transfer but is seen as unreliable/not credible by the recipient, the information may be rejected or challenged (Szulanski, 1996).

The relational context characteristics in Szulanski's (1996) model, is the arduousness of the relationship between sender and receiver. This has to do with the ease and frequency of communication between the sender and receiver which was shown to have a strong influence on

knowledge transfer success (Szulanski, 1996) as supported in other studies (G. W. Bock and Kim, 2002; Joshi et al., 2007).

The entire model of Szulanski's (1996) knowledge transfer impediments can be shown in the Table 2-3 to follow:

Table 2-3: Szulanski's (1996) stickiness predictors at the different stages of knowledge transfer

Communication Elements	**Predictors of stickiness**
Knowledge	Causal ambiguity
	Unproven knowledge
Source	Source lacks motivation
	Source not perceived as reliable(or credible)
Recipient	Recipient lacks motivation
	Recipient lacks absorptive capacity
	Recipient lacks retentive capacity
Context	Barren organizational context
	Arduous relationship between source and recipient

The model is relevant to knowledge transfer within firms or companies therefore it is also relevant for knowledge transfer within Agile/Scrum companies. However, the model is not made specific to knowledge transfer within information systems teams (Curtis et al., 1988) which means many of its constructs are not relevant for this study (Joshi, Sarker, and Sarker, 2004). Furthermore, when the model was empirically validated with a questionnaire on 271 best practices transfers, of eight companies, only three (out of nine) constructs significantly affected knowledge transfer. These were casual ambiguity, recipient's lack of absorptive capacity and arduous relationship between source and recipients. Surprisingly motivation of the source was

not a major barrier to the knowledge transfer, as many prior knowledge management scholars thought (Szulanski, 1996).

However, though motivation was not shown to be statistically significant in Szulanski's (1996) study, other authors in the future did find it supports knowledge transfer when considering both intrinsic and extrinsic motivation (Osterloh and Frey, 2000).

In addition, the author mentions that the correlational design of the study limits the results from having casual relationships. Even though, Szulanski (1996) mentions that the study is generalisable to knowledge transfer within companies due to the theoretical sample selection, the actual theory was not named in the study. Nonetheless, this is still a very influential model in the literature of knowledge transfer (G. T. Timbrell, Delaney, Chan, Yue, and Gable, 2005).

2.6.1.2 Model of Knowledge Transfer in Information Systems Development (ISD) Teams

A study examined the role of source attributes in successful knowledge transfer within information systems development (ISD) teams (Joshi et al., 2007). In the context of Scrum, this team includes the development team (such as programmers, technical writers and quality control testers) (Schwaber, 1995), the Scrum master and product owner (Cho, 2008; Karlsen et al., 2011). The source and recipient can be anyone of these roles (Karlsen et al., 2011). This study chose to examine knowledge transfer in teams as most knowledge transfer studies were organisational (Joshi et al., 2007). Therefore, it is more relevant to this study than Szulanski's (1996) model as it focusses on specifically on teams and not intra-organisational transfer as a whole.

Just like the Szulanski (1996) model above, the resulting model by Joshi et al. (2007) is also based on communication theory(Joshi et al., 2007). The researchers identified three key attributes (although they said more could be applicable) of a source/sender which influence knowledge transfer from the communication theory literature and these are: capability, credibility (source characteristics) and communication extent (relational characteristics) (Joshi et al., 2007). These characteristics are appropriate knowledge transfer characteristics for information systems teams, as they cover both source and relational characteristics (Curtis et al., 1988) and shall be discussed in turn. However, the actual type of knowledge transferred is quite specific in this study, and must be explained first.

Characteristics of the Knowledge Transfer

There are various knowledge types in information systems development (ISD), such as technical, managerial, organisational and behavioural (Lindvall and Rus, 2002). However, only technical and managerial knowledge are studied in Joshi et al. (2007)'s model. Technical knowledge refers to programming, testing, designing, database and requirements gathering knowledge (Lindvall and Rus, 2002) and is explicit as it is easily transferable (Goh, 2002). Managerial knowledge refers to planning, deadline handling, dealing with staff and directing a project (Lindvall and Rus, 2002). This knowledge is more tacit as it gained with experience like how to handle staff and is usually transferred through observations (Bassellier, Reich, and Benbasat, 2003).

The transfer of technical knowledge in Agile (including Scrum) teams is encouraged as teams are cross-functional in order for developer/programming knowledge not to remain exclusive to some members (Tesch, Sobol, Klein, and Jiang, 2009).

The transfer of managerial knowledge amongst the team members is important in Scrum teams since these teams must be self-managing. This relates to the core characteristic of Agile (including Scrum) teams in that they need to be self-organizing and self-managing by delegating various aspects of work between all members such as planning and scheduling (Moe, Dingsoyr, and Dyba, 2008).

Characteristics of the Source

Credibility

The credibility of a source, as perceived by a recipient, has been shown empirically to be critical to knowledge transfer (Ko et al., 2005). Credibility consists of trust/trustworthiness and reputation, as proposed in Szulanski (1996) model of intra-firm knowledge impediments. When the source is not perceived as trustworthy, which is someone who can be trusted (Tsai and Ghoshal, 1998) or credible by the recipient, their knowledge transfer may be questioned or even rejected (Szulanski, 1996). Indeed, in Agile (so too Scrum) methodologies, trustworthiness of a source has been found to be key for the transfer of tacit (or difficult to explain) knowledge which is the dominant knowledge type in Agile methodologies (Karlsen et al., 2011). The perceived reputation of a source (by the recipient) is also key to any knowledge transfer, including

information systems development (ISD) and therefore in Agile information systems development too (Joshi et al., 2007).

Capability

The amount of knowledge transferred by a source has been empirically shown to be positively related to the source's knowledge (Joshi et al., 2007). Szulanski (1996) has proposed and studies have empirically shown that the experience and knowledge gained over time improves a person's ability to transfer knowledge (Faraj and Sproull, 2000). Furthermore, in Szulanski's (1996) popular model of knowledge impediment, casual ambiguity, which is when a source does not have the capabilities to reproduce knowledge, was empirically found in that study to be of the top three knowledge transfer impediments within companies.

On the other hand, the knowledge engineering paradox also mentions that experts sometimes find it difficult to communicate their knowledge as they are so immersed in it (Liebowitz and Chen, 2003). However, Joshi et al. (2007) still felt this was a rare occurrence in the literature. Also, once again, in the context of information systems development (including Agile development), these capabilities are technical knowledge (such as programming and database knowledge) as well as project management knowledge (Joshi et al., 2007).

Characteristics of the Relational Context between Source and Recipient

Communication Extent

The frequency and extent of communication between a source and recipient has been empirically shown to greatly influence the amount of knowledge transferred between them (Szulanski, 1996). Indeed, the literature finds team communication to be one of the core factors to knowledge transfer success (Chau et al., 2003; Cockburn, 2001). For example, communication frequency has been empirically found to be critical in knowledge transfer of product development teams (Leenders, Van Engelen, and Kratzer, 2003). Even in Szulanski's (1996) study of knowledge impediments, it was empirically found that a lack frequency and ease of communication between the source and recipient was one of the top three impediments to knowledge transfer within firms. In the case of Scrum, the Daily Scrum meetings have been said to foster frequent communication but when teams are far apart in the company, this does not

usually happen (Cho, 2008). However, it has also been shown that excessive face-to-face communication between team members can also have negative impacts like reduced team productivity (Joshi et al., 2007).

The above-mentioned constructs and hypothesises can be seen in the Figure 2-7 below. These hypothesises are that the source's capability/credibility/communication extent with the recipient will be positively related to the extent of knowledge (technical and managerial) transferred.

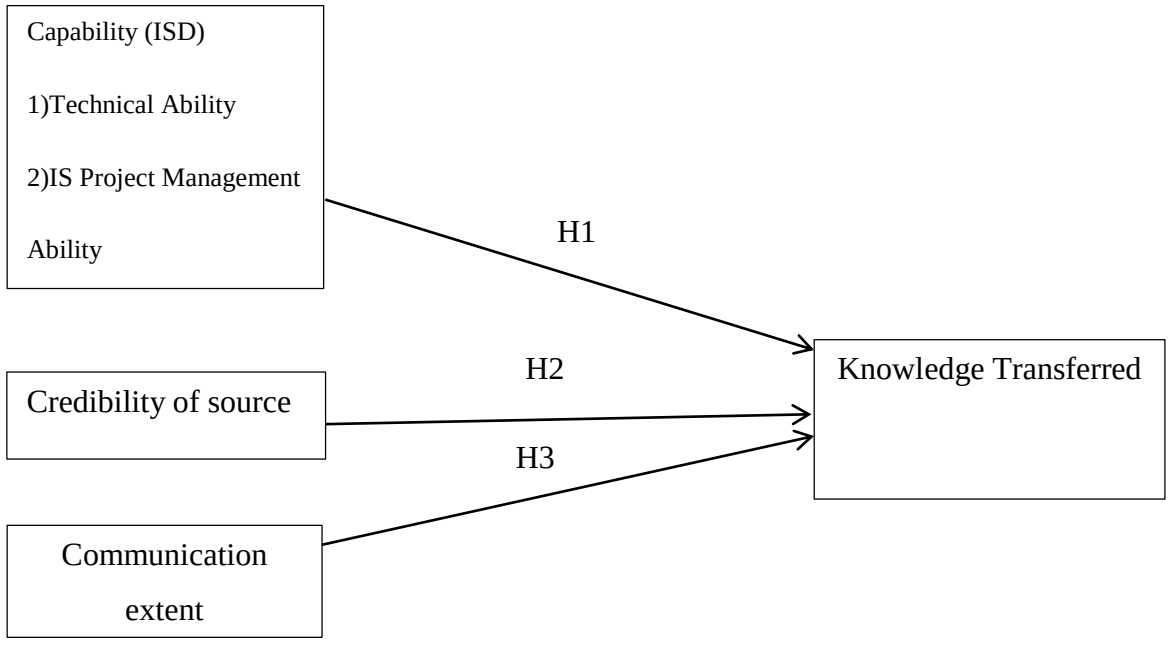


Figure 2-7: Source attributes affecting knowledge transfer in information systems development (ISD) teams*(Joshi et al., 2007)*

The main limitations to the study, as reported by the authors, Joshi et al. (2007), is the use of a student sample which many scientist feel limit external generalizability of results. An empirical psychology study of managerial decision making, with over 50 subjects, showed that there is no substantial difference between using a student sample instead of a professional sample (Remus, 1996). However, even Joshi et al. (2007) do call for future research with a professional sample to limit this threat of external generalizability, to many scientists. Also, Joshi et al. (2007) mention that the model can be extended to examine more factors that contribute to knowledge transfer such as motivation.

2.6.2 Cognitive Perspective

The cognitivist perspective views knowledge as being universal data thus easily transferable (in contrast to the connectionist perspective) from source to recipient (Venzin, von Krogh, and Roos, 2000).

2.6.2.1 Theory of Planned Behaviour

The theory of planned behaviour (TPB) is one such theory and is one of the most popular psychological models that explains and predicts human behaviour (Ajzen, 1991). TPB is an extension of a previous theory by the same author, theory of reasoned action (TRA) (Ajzen and Fishbein, 1980). This came as a result of a criticism to TRA that it did not consider the case of when a person has little control over their behaviour (Chennamaneni, 2006). Therefore, TPB consists of the following constructs: perceived behavioural control (new construct), attitude and subjective norm (Ajzen, 1991).

Perceived behavioural control is an individual's belief of the control they have over their behaviour. The attitude towards a behaviour (such as transferring knowledge) is based on the individual's belief of that behaviour and consequences of transferring knowledge (Ajzen, 1991). For instance, in the case of Szulanski (1996) model of knowledge impediments, an individual would not transfer knowledge for fear of loss of power. The subject norm is the perceived social pressure the individual has from the organisation to transfer knowledge. These will all drive an individual's intention to transfer knowledge and whether they actually transfer the knowledge (Ajzen, 1991) as shown in Figure 2-8 below :

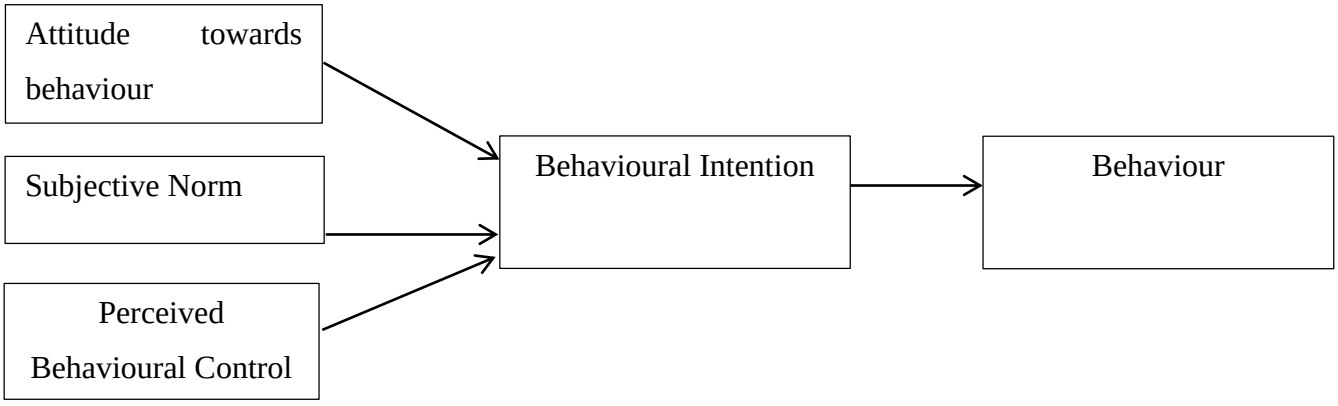


Figure 2-8: Theory of Planned Behaviour (Taylor and Todd, 1995)**.**

However, a problem with cognitive models such as these is that they do not consider the complexities of the actual knowledge transferred (Kogut and Zander, 1992). Therefore, they do not consider the capabilities of the knowledge sender as important, yet some empirical studies found them critical (Faraj and Sproull, 2000).

2.7 Identified Gaps in the Literature

This section presents the gaps identified from the literature review, which will be addressed in the next section.

2.7.1 Gaps Identified

Knowledge transfer studies in information systems development (ISD) teams is low, especially in Agile teams (Karlsen et al., 2011). Many studies have looked at knowledge transfer in global/distributed agile teams (Da Silva et al., 2010; Holz and Maurer, 2003; Hossain et al., 2009; Jalali and Wohlin, 2010). This is because knowledge transfer barriers are more prominent due to team distance and even culture differences (Da Silva, Costa, França, and Prikladinicki, 2010). Indeed, there have been studies found that have examined knowledge transfer in co-located teams but they too have their limitations.

There was a case study of intra-team knowledge transfer in Scrum development teams by Karlsen et al. (2011) but it had not stated its theoretical underpinnings. However, there was another model, by Joshi et al. (2007) found which was based on communication theory, investigated knowledge transfer in information systems development teams but it was only tested on a student sample. Furthermore, this model is not specific to Agile/Scrum development and it can still be extended to include more constructs, such as motivation of the source, according to Joshi et al. (2007).

Therefore, from the literature review conducted, it is clear there is knowledge transfer studies of information systems teams (ISD) teams are scarce especially in Agile development, as also found by Karlsen et al. (2011) and that they lack theoretical underpinnings (Wang and Noe, 2010).

2.8 Theoretical Model

In this section, the theoretical model of the study will be discussed as emergent from the literature. The main parts of this section will be the reasons for choosing and extending the model, a presentation of the model which is to be followed by associated research questions and objectives in the next section.

2.8.1 Introduction

The theoretical model chosen for this study is Joshi et al. (2007) model of knowledge transfer within information systems development teams. This is shown again, in Figure 2-9:

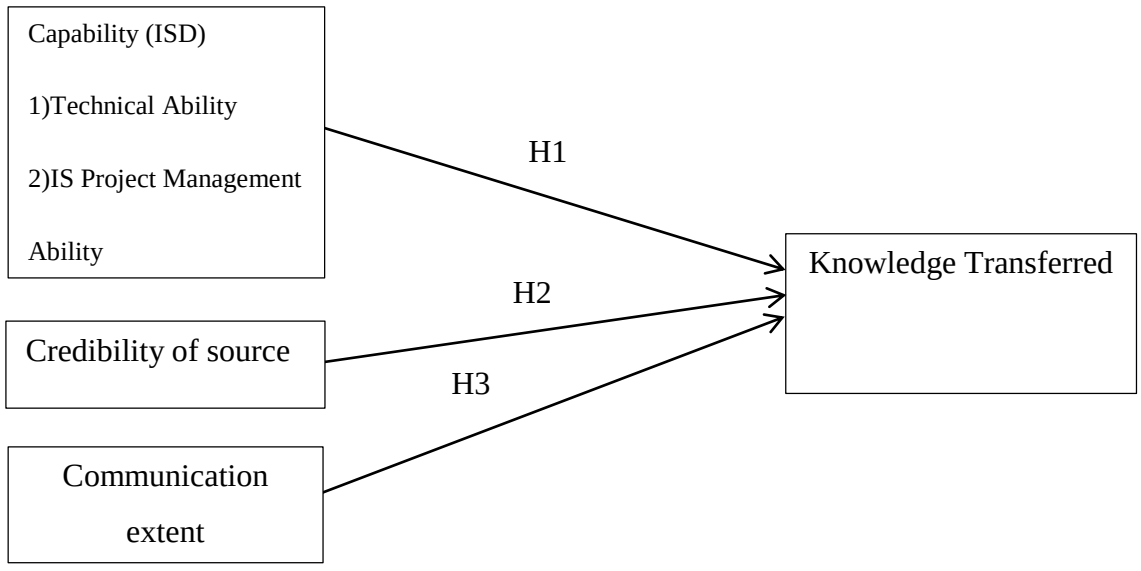


Figure 2-9: Source attributes affecting knowledge transfer in information systems development (ISD) teams*(Joshi et al., 2007)*

The reasons for choosing this model and how it is to be adapted for this research are to follow.

Firstly, Joshi et al. (2007) model was chosen as it is the only model reviewed that specifically investigates knowledge transfer within software development teams. Though, this model is not as comprehensive as Szulanski's (1996) knowledge management model, it covers the relevant attributes of knowledge transfer for information systems development (ISD) teams. Furthermore, the simplicity of this model makes it is seemingly easy to apply and adapt, as Joshi et al. (2007) did intend for future researchers to extend it. Indeed, it will be extended and adapted in the following ways.

2.8.2 Model Extension

The model will be extended theoretically in that the motivation factor of the source/sender will be added, even as the authors, Joshi et al. (2007) suggested.

2.8.2.1 Adding Motivation

Motivation is what moves a person to do something (Ryan and Deci, 2000). Motivation is important to knowledge transfer, as people have been said to be motivated to transfer knowledge because of interest in the task itself or sometimes they have been incentivised by monetary or social rewards (Ryan and Deci, 2000). Therefore, there have been many studies in software development that have either tried to increase participation in knowledge transfer by making the activity itself more enjoyable by maybe adding game-like features (Pandey and Dutta, 2013) or they have incentivized, through either social or economic rewards for the best contributors (Hung et al., 2011). Furthermore, motivation is important to Agile teams, as motivation is one of the core principles in the Agile Manifesto (Fowler and Highsmith, 2001), and if teams are not motivated, they will not be able to take advantage of the numerous opportunities to transfer knowledge in Scrum teams, like the daily Scrum meeting, sprint planning and review meetings, etc. (McHugh et al., 2011; Moe, Dingsøyr, and Dybå, 2010).

2.8.2.2 Only measuring the recipient's perspective

Since it is difficult to measure knowledge transfer, the recipient's perception of the knowledge they have gained will be measured (Karlsen et al., 2011). However, where this research model again differs from Joshi et al. (2007)'s, is that the source will not be interviewed but just the recipient, as the source might have left the company/business, whereas they were interviewing students. Indeed, it is acceptable to just look at the recipient's perspective in a knowledge

transfer study (Szulanski, Cappetta, and Jensen, 2004). In Joshi et al. (2007) knowledge transfer model of information systems development (ISD) teams, credibility and communication extent and the total knowledge transferred is already evaluated from the recipient's perspective, therefore only motivation and capability construct needs to be further modified.

The construct of reciprocity from Social Exchange Theory mentions how people/agents reward others for kind actions and punish them for unkind ones (Stanca, Bruni, and Corazzini, 2009; Trauth, 2010). Likewise, it has been found that people will reward those they perceive are kind based on their genuine/intrinsically motivation, therefore they will accept and transfer knowledge to them. But people will not want to go to gain knowledge from those who are perceived to be unkind to them and show only external motivation (Stanca et al., 2009).

In terms of perceived capability, it has been shown that if people perceive a source to be capable, they will be like to learn from it and re-using its knowledge in the future (Kankanhalli, Lee, and Lim, 2011), which is even a step better than initial knowledge transfer (Szulanski, 1996).

2.8.2.3 Contextualizing

The model will also be contextualised to co-located Scrum development teams in companies. Also, research questions will be posed instead of hypotheses as the model will be tested using semi-structured interviews in a Scrum organization in South Africa, to allow simpler sampling than a survey (Gable, 1994). South Africa, in particular, has low stakeholder satisfaction in Agile development projects and more research in companies has been recommended (Ferreira and Cohen, 2008). Furthermore, the dependent variable of knowledge transferred will now be more explicitly articulated as the 'perceived knowledge transferred' as perceived by the recipient, based on the questionnaire that validated the model (Joshi et al., 2007). Also, the term 'Scrum team' consists of a programmer, tester and documenter as well as the Product Owner and Scrum Master (Schwaber, 1995).

2.8.3 Theoretical Model for study

The contextualisation of these constructs to Scrum projects will be presented in Table 2-4 next:

Table 2-4: Theoretical Model applied to Scrum projects

Theoretical Model Concepts	**Application**
The team	Usually software developers, documenters and testers (Schwaber, 1995).
Perceived tacit knowledge transferred	The tacit (or difficult to articulate) knowledge that one Scrum team member perceives to have acquired from another team member (Chau et al., 2003; Joshi et al., 2007). Agile projects often emphasis tacit knowledge transfer through frequent verbal communication and team interaction (Cockburn and Highsmith, 2001).
Motivation of source (as perceived by recipient).	Motivation is what moves a person to do something. Motivation can be either intrinsic or extrinsic (Ryan and Deci, 2000).
Perceived Intrinsic motivation	The natural motivation that a Scrum team member has to transfer knowledge, because they find it interesting or enjoyable (Ryan and Deci, 2000).
Perceived Extrinsic motivation	Factors other than the act of transferring knowledge itself, that motivates a Scrum team member to transfer knowledge (Ryan and Deci, 2000).
Capability of source (as perceived by recipient)	This is the information systems development (ISD) knowledge of a team member and consists of technical and managerial capability (Joshi et al., 2007).

Theoretical Model Concepts	**Application**
Perceived Technical capability	This refers to programming, database and general computer knowledge of a Scrum team member (Lindvall and Rus, 2002).
Perceived Managerial capability	This refers to the leadership, planning and relationship management abilities (inter-personal skills) of the Scrum team member (Lindvall and Rus, 2002).
Credibility of source (as perceived by recipient)	This refers to the extent to which a Scrum team member is perceived by other team members to be trustworthy and reputable (Joshi et al., 2007; Szulanski, 1996).
Trustworthiness	This is the extent to which one Scrum team member is trusted by another Scrum team member (Szulanski, 1996).
Reputation	This is the extent to which one Scrum team member finds another team member reputable (Szulanski, 1996).
Communication extent of source (as perceived by recipient)	The extent to which a Scrum team member has communicated, as perceived by other team members (Leenders et al., 2003; Szulanski, 1996).

This leads to the following theoretical model, in Figure 2-10 which is adapted from Joshi et al. (2007) by adding the motivation construct.

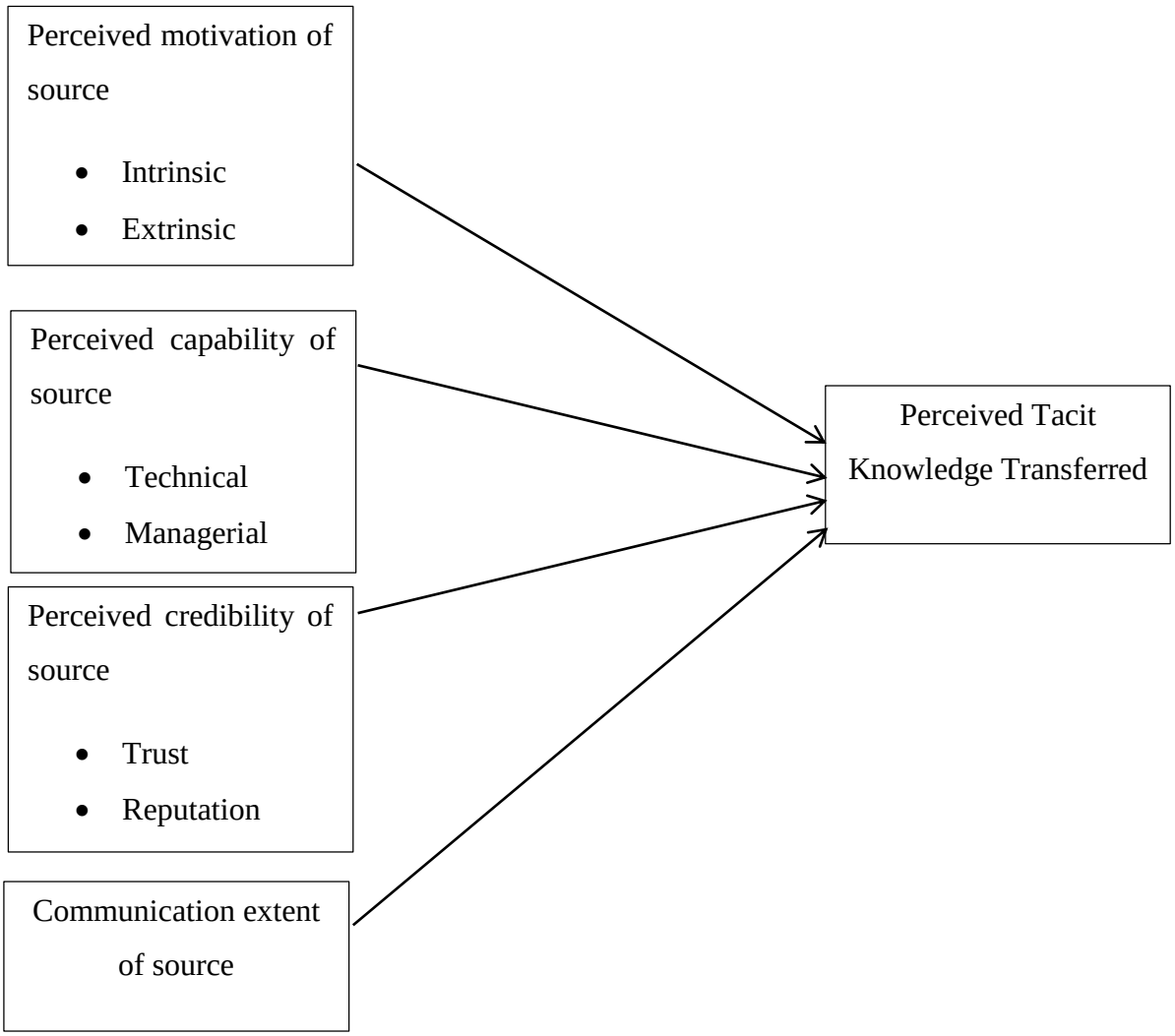


Figure 2-10: Theoretical Model for Research

2.9 Research Questions and Objectives

Based on the literature review, the following research questions and objectives were obtained.

Main Research Question:

- What are the perceived characteristics that a Scrum team member should have to successfully transfer tacit knowledge throughout the project?

Sub-Research Questions:

- How does the perceived motivation of a team member affect the perceived extent of tacit knowledge transferred by them?
- How does the perceived capability of a team member affect the perceived extent of tacit knowledge transferred by them?
- How does perceived credibility of a team member affect the perceived extent of tacit knowledge transferred by them?
- How does the perceived communication extent of a team member affect the perceived extent of tacit knowledge transferred by them?
- What other characteristics are associated with team members that can successfully transfer tacit knowledge in Scrum projects?

Main Objective:

- Develop a theoretical model depicting the perceived characteristics that a Scrum team member should have to successfully transfer tacit knowledge throughout the project.

Sub-objectives:

- Determine how the perceived motivation of a team member affects the perceived extent of tacit knowledge transferred by them.
- Determine how the perceived capability of a team member affects the perceived extent of tacit knowledge transferred by them.
- Determine how perceived credibility of a team member affects the perceived extent of tacit knowledge transferred by them.
- Determine how the perceived communication extent of a team member affects the perceived extent of tacit knowledge transferred by them.
- Determine what other characteristics are associated with a team member that can successfully transfer tacit knowledge in Scrum projects.

3 Research Methodology

Based on the research questions derived from the literature review, this section describes the research methodology followed to answer these questions. Firstly, the underlining research paradigm and approach are described, followed by a description of the semi-structured interview research strategy employed. Thereafter, the data collection prerequisites (such as target population and sample), data collection and analysis are described. Then, the issues of reliability and validity, and ethical issues are discussed. Finally, the section will conclude with a table summarising the research methodology.

3.1 Underlying Philosophy and Approach

Information system (IS) is an inter-disciplinary field which results in a number of philosophical approaches. Therefore, IS researchers must state the ontological and epistemological views underpinning their research methodology (Kanellis and Papadopoulos, 2008). The main epistemologies in IS research are positivist, interpretivist and critical realism (Myers, 1997). Positivism believes in a single objective reality with single answers (its ontology) and aims to produce measurable results (in numbers/quantitatively) which predict behaviour and are generalisable to a population.

Interpretivism believes in a subjective reality with multiple answers (its ontology) and aims to socially construct/interpret people's subjective/personal view of reality (in words/qualitatively). Critical realism believes that an objective reality can only be known once the subjective/personal biases are removed (its ontology) and aims to remove these biases in order to emancipate the subjects/people from their situation (Myers, 1997).

This study used an interpretivist epistemology as it is well suited to complex human issues such as knowledge transfer. Interpretivism can construct/interpret how the team members perceive this knowledge transfer (Rodriguez-Ulloa and Paucar-Caceres, 2005). In contrast, a positivistic paradigm might not be as suitable as it is difficult to measure knowledge transfer quantitatively (Karlsen, Hagman, and Pedersen, 2011).

The study was interpretive but also employed a preliminary framework to guide the research. Walsham (1995) acknowledges that, even though interpretivist research typically builds theory

from data, a preliminary framework might be used to guide interpretive studies. Hence, an a priori theoretical model was formulated from the literature. Indeed, there have been other interpretative studies which were driven by a priori theoretical models (Fereday and Muir-Cochrane, 2008; Prifling, Gregory, and Beck, 2008; Walsham and Sahay, 1999). The aim of this research was not to test the framework but to use it as a basis for data collection and analysis. In addition, the researcher sought to inductively identify more constructs to add richness and expand that framework during data analysis.

The research approach was mainly descriptive in nature. This is because this research aimed to describe the characteristics that a team member should have to successfully transfer knowledge to other Scrum team members. This allowed the first 5 sub-research questions, which are descriptive in nature, to be answered.

The research was also exploratory in that it enhanced the theoretical framework based on findings that emerged (Baxter and Jack, 2008). This allowed the final research question which was exploratory in nature, to be answered.

Only qualitative data was used as this corresponded with the interpretative epistemology, which supports getting rich and in-depth qualitative (or wordy) information on a small sample (Voss, Tsikriktsis, and Frohlich, 2002).

3.2 Research Strategy

A clear definition of a research strategy is necessary for any good empirical research (Amaratunga, Baldry, Sarshar, and Newton, 2002). This study used semi-structured interviews as research strategy since knowledge transfer can be measured only by what people perceive to have gained (Karlsen et al., 2011). Semi-structured interviews have been used by other knowledge transfer studies for this purpose (Karlsen et al., 2011; Moe, Dingsoyr, and Dyba, 2008). These interviews were first piloted with three participants. The results of the pilot led to questions being refined and new questions added so richer information could be gathered (Barriball and While, 1994). Members of Scrum teams from two Scrum software development companies were then interviewed. Each interview was then transcribed (written into text) and analysed using thematic analysis to produce a final report. Given the hermeneutic nature of

interpretive studies (Klein and Myers, 1999), the analysis process was iterative and was revisited even during the report writing. This process is illustrated in Figure 3-1:

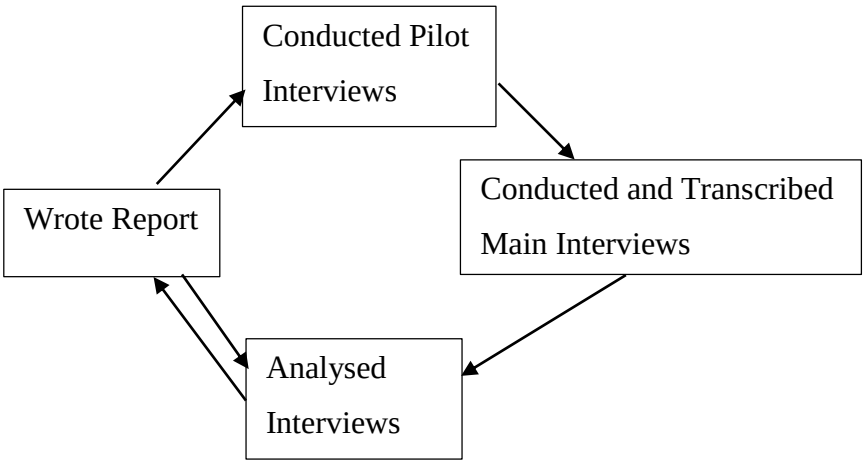


Figure 3-1: Research Process

3.3 Target Population

The target population is the group or individuals to whom the research results apply. The target population was all the co-located Scrum teams in the world. However, since this research was qualitative, the intention was not to generalise to this population statistically but rather to theory, known as analytical generalisability (Yin, 1994). In this research, the results of the findings should be used to analyse other related contexts (Walsham, 1995) but a full theory was not developed, due to time constraints (Bonoma, 1985).

3.4 Sample

As mentioned, in qualitative research the aim is to build theory, and information rich findings are needed as a building block for these. Thus, the careful selection of a sample is still imperative in a qualitative study, in order to assure the best transferability of results to other similar contexts,

which is important for theory creation (Flyvbjerg, 2006; Marshall, 1996). Therefore, this section is divided into two parts, the sample choice and the sample profile.

3.4.1 Sample Choice

Purposeful sampling, as opposed to random sampling, is used in qualitative research to obtain candidates/participants that are strategically related to the purposes of the study and is necessary for the transferability of results to other contexts (Flyvbjerg, 2006; Patton, 1990). Purposeful sampling does introduce a sampling bias in that the sample is not representative of the entire population (Thompson, 1999), but that is not a concern for qualitative research, which is about analytical/theoretical generalisation as opposed to statistical generalisation (Flyvbjerg, 2006) .

In particular, maximum variation sampling was the type of purposeful sampling that was used to select the twelve participants who were interviewed in this study. These participants were split equally between two companies. Maximum variation sampling is a technique that selects a broad range of subjects, in order to get a holistic understanding of the phenomena (Patton, 1990). Therefore, in this study, team members from all Scrum roles were interviewed in each company, as selected by their head project managers. This included software developers, business analysts, quality assurance tester (though only in one company), project managers and Scrum masters (Schwaber, 1995). The group of participants also consisted of both male and female, as well as African, Asian and Caucasian ethnicities.

Twelve participants were chosen as studies have shown that usually not much new information emerges after conducting twelve interviews (Guest, Bunce, and Johnson, 2006), as was also the case with this research. The literature often samples different companies when looking at knowledge transfer in software development teams as different companies may have different project types (like business intelligence and cloud computing) which may influence team context (Bjørnson and Dingsøyr, 2008). To this end, the participants for this study were selected from two Scrum companies in Cape Town, each focusing on different project types.

Cape Town was a suitable city being one of the three largest metropolitan cities in South Africa, diverse and technologically advanced (Booyens, 2012). Organisational permission was obtained from them in an email letter inviting them into the research (see Appendix B). For

confidentiality, they are called Company A and B (Walsham, 2006). Company A focuses on providing business consulting and software solutions to corporate organisations. Company B focuses on software solutions, business intelligence as well as usability and design services to mainly insurance and financial companies.

Using two companies, allowed for space triangulation, which is using multiple locations to collect the same data (Hussein, 2009) and helped confirm results found in one company by testing at the other (Baxter and Jack, 2008; Hussein, 2009).

One team of 6, was interviewed in each company. These teams were purposefully chosen by the project managers/software managers in each company, based on their diversity in experience and roles.

3.4.2 Sample Profile

This section describes the profile of the sample and a narration of their company context, to give this study greater context as a whole (Braun and Clarke, 2006). The team from Company A said they were focussed on making custom software solutions for a bank client whereas the team from Company B said they were focussed on business intelligence for a financial services client. Both teams said they were following the Scrum process entirely, mentioning the relevant roles, ceremonies and artefacts (Cho, 2008) however since the team from company B, had a certified Scrum master, that team might have been more systematic in their approach. Here is the sample description, provided in Table 3-1 below:

Table 3-1: Sample Profile

Alias	Company	Job Position	Project Team Role	Experience
P1	A	Junior Software Developer	Software Developer	<1 year
P2	A	Test Manager	Test Manager and Quality Assurance Tester	8 years
P3	A	Head Project Manager	Project Manager and was a former Software Developer	6 years
P4	A	Project Manager and	Project Manager	4 years

		Business Analyst		
Alias	**Company**	**Job Position**	**Project Team Role**	**Experience**
P5	A	Business Analyst	Business Analyst and Scrum Master	2 years
P6	A	Project Manager	Project Manager	20 years
P7	B	Scrum Master	Scrum Master	4 years
P8	B	Software Developer	Software Developer	3 years
P9	B	Lead Software Developer	Software Developer	3 years
P10	B	Software Architect	Software Architect and Software Developer	8 years
P11	B	Intermediate Software Developer	Software Developer	3 years
P12	B	Head Project Manager	Project Manager	20 years

Each Participants answered questions on their experiences working within these companies; however, where necessary they also drew on past experiences of work in companies that were mostly from the finance, banking, and manufacturing industries. This was because most of them have not been working at their current companies for a long time.

3.5 Unit of Analysis

The unit of analysis which was the main entity that was analysed was of great importance to any research. Essentially, it is the *who* or *what* that is examined (Yin, 1994). The unit of analysis was each individual team member, which consisted of the programmer/software developer, tester, documenter/business analyst, product owner, project manager and Scrum master (Schwaber, 1995). There would be no inclusion or exclusion criteria (such as age) as this was not found relevant in other Scrum studies (Karlsen et al., 2011; Moe, Dingsøyr, and Dybå, 2010).

3.6 Data Collection

The following will be addressed in the data collection section: the time-frame, the pilot study strategy and the data collection techniques employed. These are important elements needed to conduct a comprehensive data collection (Rowlands, 2005).

3.6.1 Cross-sectional Research

A cross-sectional study is a snapshot at a particular moment in time, it allows the researcher to find associations between variables but not causality as the time period is short. In contrast, longitudinal studies are better for identifying casual relationships as they are done over a extended period of time. Indeed, interpretative studies can be cross-sectional as they do not usually seek to establish causal relationships (Kakkuri-Knuuttila, Lukka, and Kuorikoski, 2008).

This study was cross-sectional as the data was gathered in three months, from November 2013 to February 2014. This was aligned to the descriptive nature of the study which aimed to describe the successful knowledge transfer factors and not explain why they occur. Furthermore, a cross-sectional study was less costly in terms of time and money (Levin, 2006), which was essential given the financial limitations of the researcher. Similar knowledge transfer studies in information systems development (ISD) teams have also been cross-sectional (Joshi, Sarker, and Sarker, 2007; Karlsen et al., 2011).

3.6.2 Pilot Study Strategy

Pilot studies can be used as a means to formulate interview questions and test research questions. Therefore, after interview questions are formulating on the basis of the literature review and research questions, they were further developed in a pilot study (Noor, 2008).

The pilot study was conducted in a Scrum development company in Cape Town, the city where the researcher works. A pilot study of only three interviews was done as pilots are exploratory in nature (Gable, 1994) and their purpose is mainly to find ambiguities in questions and wording as well as to establish additions or deletions to questions (Noor, 2008).

After this pilot, some of the questions were refined to make them more understandable, and new ones were added, in order to capture richer information more relevant to the research question and objectives (Barriball and While, 1994). For example, one of the participants felt that it was too vague to ask if they trusted someone, so suggested that an example of trust should also be supplied. Likewise, the original interview questions did not have any context-specific questions, such as about role and years of experience, yet it was found this would be crucial in reporting the study (Braun and Clarke, 2006).

3.6.3 Data-collection Techniques

As mentioned before, semi-structured interviews were the only data-collection technique used as it allows knowledge to emerge from individuals (making it interpretative), yet still addresses the research questions. This limitation was also overcome using other triangulations, as twelve different participants were interviewed in two different companies (Hussein, 2009). Its closed questions allowed the research question to be addressed, and its allowance for unplanned questions allowed for new knowledge to emerge from individuals (Louise Barriball and While, 1994).

Before the interviews occurred, the participants were given an interview consent form to sign (see Appendix C) which detailed the research objectives and their rights to confidentiality. This also mentioned that they were free to stop the interview at whichever point they wanted. Once the participants accepted these conditions, the interviews took place. When the interview began, the researcher asked the respondent whether the interview could be recorded (with a tape recorder) so that it can be analysed completely later on.

In terms of interview length, interviews in software development studies usually fall between 15 and 60 minutes (Fritz and Murphy, 2010), but some research has found that at least 30 minutes makes for an effective interview (Basri and O'Connor, 2010; Svensson, Parker, and Regnell, 2011). Therefore, each interview lasted at least 30 minutes in the study and a total of 12 interviews were conducted. As mentioned, this was because after twelve interviews, usually not much new information will emerge (Guest et al., 2006) and this was also the case for this research.

At the end of each interview, the researcher transcribed it into a Microsoft Word document. The transcription process is said to be the start of the interpretative process so the researcher also jotted down some notes during this stage (Bailey, 2008).

Indeed, interviews do have their strengths as well as weaknesses and this is shown in Table 3-2 below.

Table 3-2: Strengths and weaknesses of using interviews as a data collection technique (Tellis, 1997)

Source of Evidence	Strengths	Weaknesses
Interviews	• Targeted – Focuses on study topic • Insightful – provides perceived casual inferences.	• Response bias (interviewees telling the interviewer what they want to hear) • Bias due to poor interview questions • Incomplete recollection of participants

Adapted from (Tellis, 1997).

To reduce the response bias, the subjects were assured at the beginning that their responses were anonymous, so they had liberty to say what they wanted (Walsham, 2006). Furthermore, the interviewer (researcher) made the interview as smooth as possible by avoiding over-direction or excess passivity. The interviewer directed the interview with a set of questions but still allowed participants to express their views outside of these. The interviewer also avoided passivity, by providing thoughts on issues to thus showing interest to the interviewee/participant (Walsham, 1995). Also, interview questions were not be poor as they were pilot-tested to ensure their reliability (Louise Barriball and While, 1994). However, as with any data collection sources, there were weaknesses that could not be removed, such as the incomplete recollection of participants.

3.7 Research Instrument

A copy of the interview protocol is attached in Appendix A. The interview questions were based mainly on a questionnaire from Joshi et al. (2007) who conducted a similar study. They were based on a theoretical framework which guided the formulation of the research questions. However, the questions were made more open-ended due to the interpretative nature of this study. Furthermore, questions for the new construct, motivation, were also derived from the literature review (Ko, Kirsch, and King, 2005).

Accordingly, the interview questions were divided into four sections which correspond to the constructs in the theoretical model. The aim was to have a balance between exploration of the participant's ideas and standardisation/focus of questions, to make them easily comparable with other researchers. The questions were also pilot-tested in order to detect ambiguities in them, and add or remove questions were necessary (Louise Barriball and While, 1994).

3.8 Data Analysis

Thematic analysis was used to analyse each interview. Thematic analysis is a systematic process for categorising text and forming relationships around categories (Berg, 1995). Codes are the most basic elements that label the data (Boyatzis, 1998), in order to organise or categorise the data so that themes can emerge (Braun and Clarke, 2006).

Thematic analysis involves identifying research themes or patterns that relate to one or many research questions. Themes are important abstract ideas found from significant patterns in the research data (Braun and Clarke, 2006). Thematic analysis can either be deductive/theory-led or inductive. Deductive thematic analysis (mostly used in this research) analyses data in the context of pre-defined themes derived from a theoretical/conceptual model (Boyatzis, 1998), yet inductive analysis allows themes (constantly occurring concept) to emerge from data itself (Braun and Clarke, 2006). Indeed, deductive thematic analysis may seem non-interpretative as it is theory-driven; however, at the level of analysis it is interpretative as the researchers interpret the findings and it is therefore still subjective (Braun and Clarke, 2006; Walsham, 1995).

Thus deductive thematic analysis was used in addressing the first five descriptive research questions but inductive thematic analysis was used to generate new constructs to address the exploratory research question (Braun and Clarke, 2006). The unit of analysis was each individual that was interviewed as they would provide the insight into what makes team members successful at knowledge transfer (Yin, 1994). The process of thematic analysis undertaken will be described, but first an overview of the phases will be shown in Table 3-3:

Table 3-3: Phases of Thematic Analysis (Adapted from Braun and Clarke, 2006)

Phase	**Description of Phase**
1. Familiarising oneself with one's data	Transcribe and read over data.
2. Generating initial codes	Generate codebook after re-reading data.
3. Searching for themes	Combine codes into potential themes.
4. Reviewing themes	Compare themes against each other (level 1) and the entire data set (level 2).
5. Defining and naming themes	Name themes so as to create a story effect.
6. Producing the report	Produce a succinct and interest report of results, which are related back to research questions and literature.

Therefore, the deductive and inductive process will follow the same steps, with the differences being in phases 2–5, where in deductive thematic analysis the codes and themes are generated in the context of theoretical model (Boyatzis, 1998) but in inductive thematic analysis they are generated from the data (Braun and Clarke, 2006). It is also worth noting that thematic analysis, although shown sequentially here, is iterative in that the researcher may repeat the process as necessary (Braun and Clarke, 2006).

3.8.1 Familiarising oneself with one's data

In this stage, the researcher is meant to be immersed in the data in order to get a feel for it (Braun and Clarke, 2006). For the researcher, this involved transcribing each interview verbatim into

Microsoft word after recording it on a smartphone. The transcription process is said to be the start of the interpretative process and indeed, meanings of the data began to emerge in the researcher's head and notes were taken for future coding (Bailey, 2008).

However, an issue that emerged here was that not all the audio was clear, but good inferences could be made due to the researcher's knowledge of the context area, Agile software development. By the end of the transcription process, the researcher felt well accustomed to the data and could transition into the coding process almost seamlessly.

3.8.2 Generating initial codes

As mentioned above, codes, usually just a few words long, are the most basic elements that label the data (Boyatzis, 1998), in order to arrange the data so that potential themes can emerge (Braun and Clarke, 2006). Atlas TI (version 7) was the qualitative data analysis tool used above and chosen for this research as it is among the best qualitative data analysis tools around, has a good user interface, and is also rich in features. Although it just helps manage the data rather than do the actual analysis, it does not code the data but tracks each code to each interview and more (Lewis, 2004). An example of a coded extract is shown in Figure 3-2:

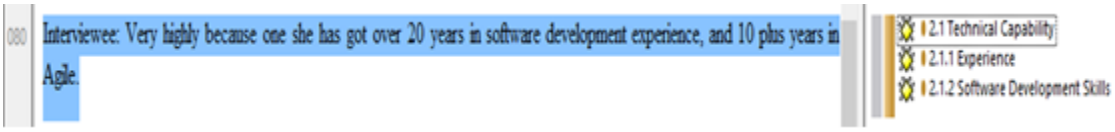


Figure 3-2: Assigning codes to a sentence

From above it can be seen in the right-hand column, that multiple codes (such as Technical Capability and Experience and Software Development Skills) were assigned to the same segment, so that as many potential themes can be generated in the future (Braun and Clarke, 2006).

Also, the context of the quote can also be retrieved (Braun and Clarke, 2006) as the software tracks the paragraph of each code (080, in this case on the left of the screenshot) and its corresponding interview number.

This coding process was also largely by whether the themes were data- or theory-driven (Boyatzis, 1998; Braun and Clarke, 2006). The coding process was done in two different ways,

inductively generated from the data or assigned from the theoretical-framework (theory-driven), depending on the research question (Boyatzis, 1998; Braun and Clarke, 2006). For the last research question, which was inductive, where codes were data-driven, the researcher had liberty to assign codes to a data, without any framework constraint. For example, codes generated in this way included empathy, willingness to listen and patience, all of which were not linked to the framework of Joshi et al. (2007).

To answer the four deductive research questions, theory-driven codes had to be generated in the context of the constructs in the theoretical framework. For example, codes found in this way were peer recognition and financial motivation, a sub-set of extrinsic motivation in the framework of Joshi et al. (2007).

In terms of the coding process, the researcher read through the related part on the interview to obtain a code list for each particular research question/construct. Thereafter, for each section of the interviews relating to a research question, the interviews were analysed line by line and codes were applied where necessary. Then the interviews were read over again and some new codes emerged relevant to a research question, but found in a different part of the interview. Some of the codes that were generated can be seen from Figure 3-3 that follows. Appending tags like, "#A1" to certain codes was a way for the researcher to label these codes as potentially important for later in the research (Braun and Clarke, 2006).

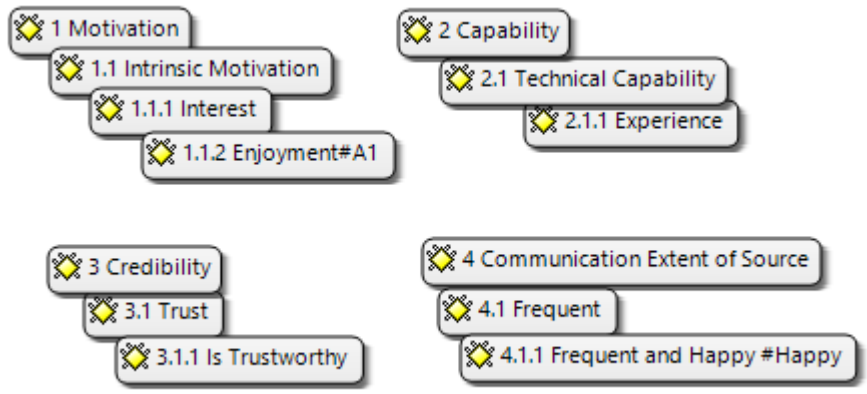


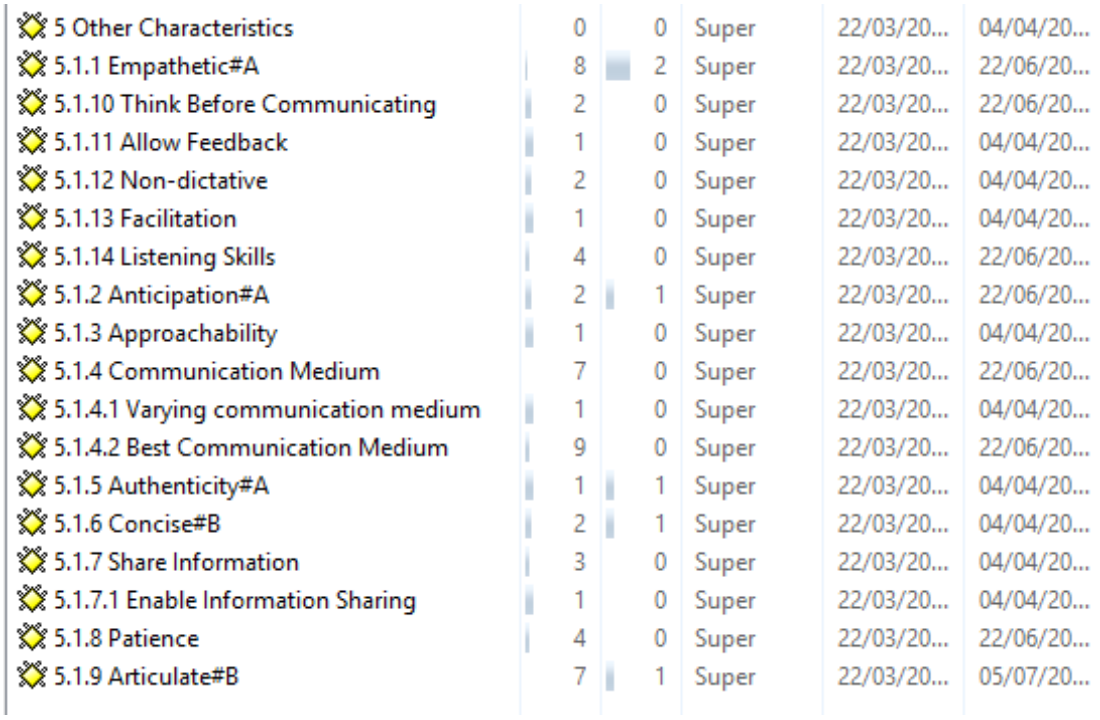
Figure 3-3: Sample of codes initially generated

A comprehensive list of all the codes used in the analysis can be found in the codebook, in Appendix D.

3.8.3 Searching for themes

Themes are significant patterns in the data that are related to a research question (Braun and Clarke, 2006). In this section codes were being combined to find potential themes and sub-themes (or smaller themes). This was done based on the researcher's judgement and similarity between codes (Braun and Clarke, 2006). Furthermore, codes were combined differently for the deductive and inductive research questions. In the inductive case, the themes did not have to relate to constructs in a pre-existing theoretical framework, whereas in the deductive/theory-driven case, they had to (Braun and Clarke, 2006).

In thematic analysis, a theme can be persistent across the dataset or just say something important about the data (Braun and Clarke, 2006). Similar codes were grouped together to form themes based on the researcher's interpretation (Braun and Clarke, 2006; Walsham, 2006). An example which best illustrates this is how the inductive themes were found. Figure 3-4 lists the codes that emerged from recipients' suggestions of what makes a team member good at knowledge transfer (the inductive research question).



5 Other Characteristics	0	0	Super	22/03/20...	04/04/20...
5.1.1 Empathetic#A	8	2	Super	22/03/20...	22/06/20...
5.1.10 Think Before Communicating	2	0	Super	22/03/20...	22/06/20...
5.1.11 Allow Feedback	1	0	Super	22/03/20...	04/04/20...
5.1.12 Non-dictative	2	0	Super	22/03/20...	04/04/20...
5.1.13 Facilitation	1	0	Super	22/03/20...	04/04/20...
5.1.14 Listening Skills	4	0	Super	22/03/20...	22/06/20...
5.1.2 Anticipation#A	2	1	Super	22/03/20...	22/06/20...
5.1.3 Approachability	1	0	Super	22/03/20...	04/04/20...
5.1.4 Communication Medium	7	0	Super	22/03/20...	22/06/20...
5.1.4.1 Varying communication medium	1	0	Super	22/03/20...	04/04/20...
5.1.4.2 Best Communication Medium	9	0	Super	22/03/20...	22/06/20...
5.1.5 Authenticity#A	1	1	Super	22/03/20...	04/04/20...
5.1.6 Concise#B	2	1	Super	22/03/20...	04/04/20...
5.1.7 Share Information	3	0	Super	22/03/20...	04/04/20...
5.1.7.1 Enable Information Sharing	1	0	Super	22/03/20...	04/04/20...
5.1.8 Patience	4	0	Super	22/03/20...	22/06/20...
5.1.9 Articulate#B	7	1	Super	22/03/20...	05/07/20...

Figure 3-4: Codes for suggested characteristics of good knowledge transfer

In the diagram above, it is clear that the best communication medium, empathy and articulation, are the most persistent codes across the data and so looked like good candidates for potential themes (Braun and Clarke, 2006). However, the communication medium was excluded, as tacit knowledge in Agile methodologies focuses on face-to-face communication only (Cockburn and Highsmith, 2001). This is followed by, listening skills, patience, and willingness to share information which were also considered as potential themes as they seemed important. The hashtags like those appended to the 'articulate' and 'concise' codes, were to represent that they could be part of the same family. After this initial analysis, it would later emerge from the literature that patience and willingness to listen were sub-themes of empathy.

In the deductive thematic analysis for the deductive research question, the same process was followed; however, each construct was a theme in itself, such as intrinsic and extrinsic motivation, so the search was more to find sub-themes within these. For example, Figure 3-5 below shows some of the codes for motivation and themes that were starting to emerge from them.

Name	Gro...	De...	Author	Created	Modified	Families
1 Motivation	0	0	Super	27/03/20...	27/03/20...	
1.1 Intrinsic Motivation	0	0	Super	27/03/20...	27/03/20...	
1.1.1 Interest	0	0	Super	01/04/20...	01/04/20...	
1.1.2 Enjoyment#A1	26	1	Super	01/04/20...	03/04/20...	
1.1.2.1 Enjoy Job in General	5	0	Super	02/04/20...	03/04/20...	
1.1.2.2 Enjoy Working in Scrum	15	0	Super	02/04/20...	03/04/20...	
1.1.2.3 Enjoy Learning Scrum	6	0	Super	02/04/20...	03/04/20...	
1.1.2.4 Enjoy Learning	4	0	Super	02/04/20...	03/04/20...	
1.1.3 Passion#A2	4	1	Super	01/04/20...	03/04/20...	
1.1.4 Motivation to Teach	0	0	Super	02/04/20...	02/04/20...	
1.1.5 Embraces Challenges#B	4	0	Super	02/04/20...	03/04/20...	
1.1.6 Motivation Validity#C	7	0	Super	02/04/20...	09/04/20...	
1.2 Extrinsic Motivation	0	0	Super	27/03/20...	27/03/20...	
1.2.1 Financially Motivated	11	0	Super	01/04/20...	03/04/20...	
1.2.2 Goal Driven	0	0	Super	01/04/20...	01/04/20...	
1.2.3 Peer Recognition	2	0	Super	01/04/20...	03/04/20...	

Figure 3-5: Codes for perceived motivation

The above codes in Figure 3-5 were combined into themes and sub-themes, as follows in Table 3-4:

Table 3-4: Generating themes of motivation

Characteristic	**Themes derived**
Intrinsic motivation	Enjoyment and its related-themes (such enjoying working in Scrum) are the main themes under intrinsic motivation, based on their popularity with participants as well as how they relate to intrinsic motivation.
Extrinsic motivation	Financial motivation and peer recognition, emerged as the main themes for extrinsic motivation, based on what participants said as well as their popularity in the literature.

The following Figure 3-6 will now show the initial list of motivation themes found. This list is refined in the next section by merging similar themes, such as merging enjoyment of working in job and enjoyment of working in a Scrum role into one theme. The original list of themes will now be shown as an initial thematic map for perceived motivation.

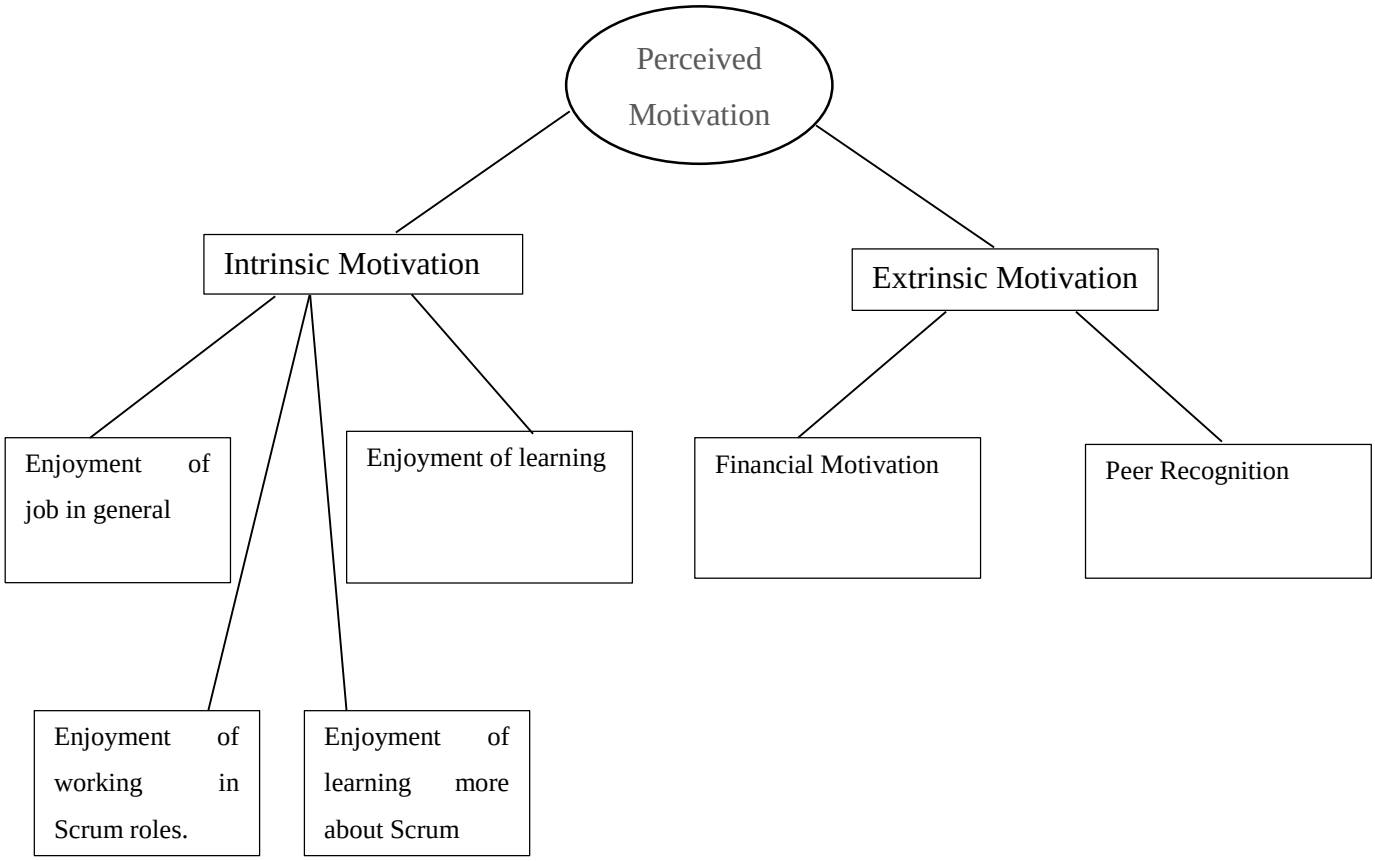


Figure 3-6: Initial thematic map for Perceived Motivation

3.8.4 Reviewing themes

In this section themes would be refined by being checked against the coded extracts, each other (level 1) and in relation to the entire data set (level 2), in order to generate a thematic map of the analysis. The key here is that the themes found are meant to be distinct/different from each other, coherent/fitting with the data and research questions (Braun and Clarke, 2006). This was done for both themes found inductively and deductively (same process).

With the themes found for the inductive research question, the coded extracts were first checked against each theme, to see if they were coherent with each other (level 1). Then the themes were checked against each other to see if any could be combined into one (level 1) (Braun and Clarke, 2006). The researcher had a feeling that willingness to listen and patience could be combined under empathy. A brief literature search confirmed this (Nonaka and Takeuchi, 2011; Zhou, 2014), and these themes were regrouped as sub-themes of empathy.

Thereafter, the themes were also checked in relation to the entire data set (level 2). The data were re-read to see if the themes found, were consistent across the set, and most were, particularly empathy. Likewise, similar terms were also combined in themes found deductively, such as a refinement to the intrinsic motivation themes, combing four small themes into two bigger themes, to produce a refined list of themes for perceived motivation as illustrated in a refined thematic map from Figure 3-7 below:

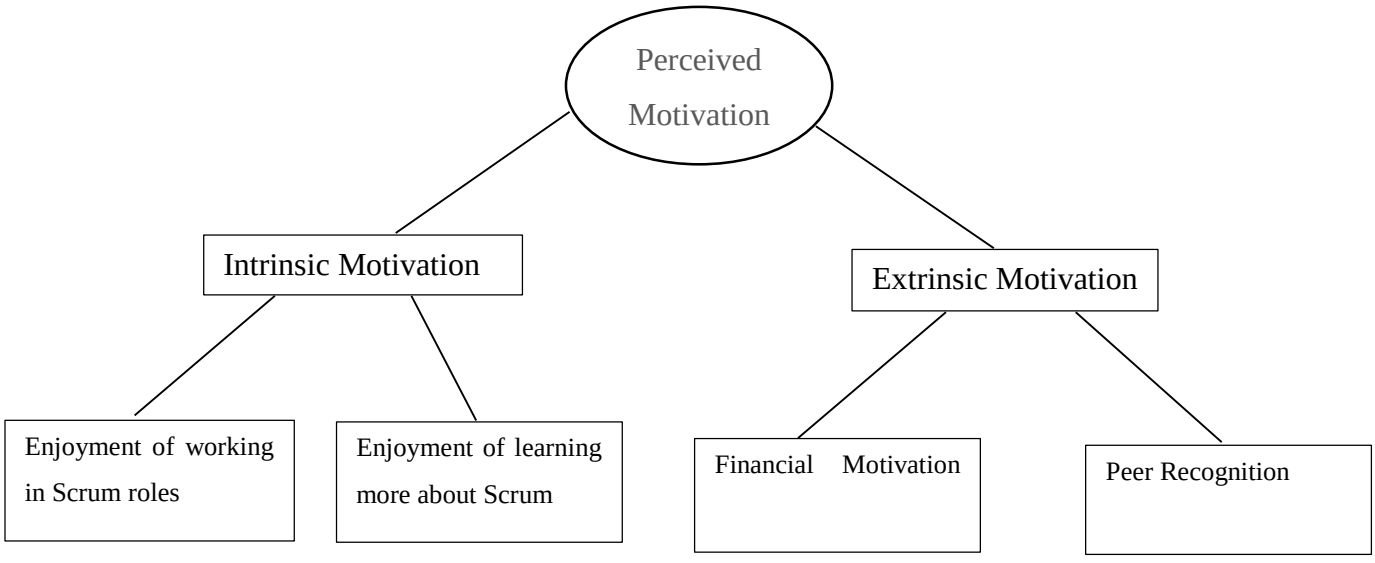


Figure 3-7: Refined thematic map for Perceived Motivation

Also, when all the themes were finalised, the researcher had to check whether these themes spoke for the data (level 2) (Braun and Clarke, 2006) which appeared to be the case as they were the most occurring patterns.

3.8.5 Defining and naming themes

In this stage, each theme was refined in order to collectively form a story. Consequently, what was important here was to make sure that themes flowed with each other, did not claim too much or too little and were related to the research questions (Braun and Clarke, 2006). An example of this follows below.

When thinking about presenting the themes in a story form, it was found that the theme of trust had no sub-themes. However, trust was too complex for a single theme, so had to be broken down into sub-themes (Braun and Clarke, 2006). Therefore, when looking at the codes generated for trust, it appeared that trust could be broken down into personal and professional trust, so it was. After even further evaluation, this theme of trust to speak more of the sources ability to be trusted, so renamed it to the original term, 'trustworthiness' as referred to in Szulanski's (1996, p.31) model of knowledge transfer, from which Joshi et al (2007, p.326) derived the term, 'trust' from.

3.8.6 Producing the report

The reporting of the data was done in a succinct way so as to illustrate the main points, interesting findings relevant to the research questions at hand and later situated in the context of the literature in the discussion section.

Also, in writing quotes in the report, the context was always explained and the participant number and role in Scrum team would almost always be listed. Furthermore, the report was reviewed by another person (the researcher's supervisor) and refined accordingly, as per recommended practice (Braun and Clarke, 2006).

3.9 Validity and Reliability

The researcher made every effort to ensure the validity and reliability of this study, as these are crucial criterion to any qualitative research (Riege, 2003). Validity is broken down into three components, in qualitative research: construct, internal validity and external validity (Anfara, Brown, and Mangione, 2002; Voss et al., 2002). Measures taken in these criterion will be discussed, to be followed by the reliability measures taken.

Construct validity

Construct validity is the degree to which the right measures/procedures are used to identify a concept (Voss et al., 2002). Construct validity was ensured by using data and investigator triangulation.

Data triangulation is the use of sources to gather the same data. This was done by person and by space (Hussein, 2009). By person, data were collected on the same information from multiple sources. In particular, each source ranging from software developers through to project managers, gave their input on the same/similar interview question (though they were some impromptu questions in each interview). By space, multiple locations (two companies) were used to conduct the interviews.

Investigator triangulation is the use of multiple researchers at some point during a study. In the analysis of this research, the researcher's supervisor reviewed the analysis to add to its consistency (Hussein, 2009) .

Internal Validity

Internal validity in qualitative research deals with the credibility of findings, how similar the findings are to reality. Internal validity is usually only demonstrated at the analysis stage. Thematic analysis was used to ensure internal validity, making sure that the concepts that were reported on, occurred across multiple interviews (Shenton, 2004). A description of how the analysis process was conducted to achieve internal validity was provided in section 3.8.

External Validity

External validity in qualitative research refers to transferability of findings, which is the extent to which they can be applied to other settings (Shenton, 2004). To ensure this, purposive sampling was used (Anfara et al., 2002) and a rich description of the results was done with many direct quotations included (Myers and Newman, 2007).

Reliability

Reliability in qualitative research refers to dependability of the research, which is the extent to which the study can be repeated by other researchers (Shenton, 2004). Reliability will be assured in the following ways. An audit trail was used (Anfara et al., 2002) by the researcher through documenting all activities, data collection chronology and data analysis procedures comprehensively (Creswell and Miller, 2000).

The use of triangulation as mentioned in the construct validity section also enhances the reliability of the research (Anfara et al., 2002). Also, to help other researchers to depend on the findings, the effectiveness of the study is demonstrated in the discussion and limitations sections (Shenton, 2004).

Objectivity

Objectivity in qualitative research refers to the confirmability of the research, which entails ensuring as far as possible that the findings emerge from participants' experiences and ideas, instead of the researcher's preferences (Shenton, 2004). This was achieved mainly through multiple information sources being asked the same questions, like using software developers through to project managers (data triangulation) and constant practice reflexivity (Anfara et al., 2002). To be reflexive is to be self-aware. So the researcher was always conscious of influencing the research. For instance, when coding interviews, it was not just based on the researcher's thoughts but also on the literature (Finlay, 2002). Likewise, even the audit trail/clear description of the research process mentioned in reliability section adds to the confirmability of the research as others can check exactly how data were gathered. Furthermore, the researcher has stated underlying philosophical beliefs and assumptions which underpin the research, giving further transparency. Finally, confirmability is also shown in the admission of the research's limitations at the end of this paper (Shenton, 2004).

3.10 Ethical Issues

It is important for a researcher to state his ethical considerations (Roode, 2008) and this is also of great importance to the Department of Information Systems at the University of Cape Town (UCT).

Particularly in interpretative research, ethics and values play an integral part, as no group's values are assumed to be more important than another (Fossey, Harvey, McDermott, and Davidson, 2002). Therefore, this research made sure that people's values were considered at all times.

The research was designed not to cause any harm or pain to any participants (Johnston, 2011) and this was confirmed by ethics approval. This was obtained from the Commerce Faculty Ethics in Research Committee at UCT before data collection could begin. This required the Research Design as well as the set of interview questions (see Appendix A) which were to be asked.

The researcher had to request written permission from the organisation/s by email to conduct the interviews (see Appendix B). This email requested permission to at least conduct interviews with members of the organisation, whilst assuring anonymity. Participant and organisation names (or any identifying elements) remained anonymous in this study (Ellis, 2007). Furthermore, the organisation was promised a copy of the final publication of the study, to encourage their participation.

The participants were treated with respect at all times. Also, an interview consent form (see Appendix C) detailing the purpose of the research (such as its non-commercial nature) and ethical precautions taken, was emailed to each participant before the interview, and they had to sign it to be interviewed. Participants were also allowed to leave/cancel the interview at any point, if they so wished. Furthermore, they were not incentivised financially to participate in the research, further increasing the voluntary nature of their participation (Dresser, 2001).

In terms of access precautions, the data collected from participants was also kept confidential by not exposing them to external parties and storing them securely (in password-locked files) on the researcher's computer.

3.11 Research Methodology Summary

The research methodology can now be summarised in Table 3-5, as follows:

Table 3-5: Research Methodology Summary

Criteria	**Choice**
Purpose	Descriptive with exploratory
Ontology	Subjective
Epistemology	Interpretative
Research approach	Qualitative
Research strategy	Semi-structured Interviews
Data collection	Semi-structured Interviews
Sampling	Maximum Variation Sampling
Data analysis	Deductive thematic analysis with some inductive analysis.
Timeframe	Cross-Sectional

4 Findings

This section presents the findings of the study identified from the thematic analysis of the data, in the context of the research question and sub-research questions. To recap, the main research question of the study was:

- What are the perceived characteristics that a Scrum team member should have to successfully transfer tacit knowledge throughout the project?

The main research question was broken down into four sub-research questions. The first three were addressed using deductive thematic analysis since they were questions derived from the theoretical model. The last question was addressed using inductive thematic analysis from which new constructs emerged, suggested by the participants. This section specifically describes the empirical findings and will not discuss the results in the context of literature. This will instead be addressed in the Discussion chapter, to follow.

4.1. Perceived Motivation

The first sub-research question was formulated as follows: *"How does the perceived motivation of a team member affect the perceived extent of tacit knowledge transferred by them?"*. As specified in the literature review, there are two types of motivation: intrinsic and extrinsic motivation. The themes identified relative to this construct and their corresponding sample evidence are displayed in Table 4-1. This table further stipulates the source alias and the number of times the themes appeared in interview transcripts as incidents:

Table 4-1: Perceived Motivation

Sub-category	Themes	Incidents	Sample evidence	Source
Intrinsic Motivation	Enjoyment of working in Scrum roles.	15	*"With __he loves it, he lives it, he breathes it, he enjoys Scrum"*	P1
	Enjoyment of learning more about Scrum	6	*"Yeah, I think he enjoys it, he is always on his phone and all that kind of stuff, he is always monitoring his phone with the news on Scrum"*	P2
Extrinsic Motivation	Financial	11	*"Financial rewards to a degree, there is obviously a trade-off between enjoying your job and how much are you earning"*	P11
	Peer recognition	2	*"Yeah, I think peer recognition motivates anyone, so definitely"*	P2

4.1.1 Intrinsic Motivation

As mentioned in the literature review , intrinsic motivation in doing a task relates to motivation that comes from the task itself (Ryan and Deci, 2000). It relates to the natural motivation that a Scrum team member has to transfer knowledge, because they find it interesting or enjoyable (Ryan and Deci, 2000). The two themes related to the enjoyment of working in Scrum roles and also learning about Scrum, were found to enable individuals to transfer more knowledge.

4.1.1.1 Enjoyment of working in Scrum roles

Many participants felt that their team mates were motivated by the very nature of their role within the Scrum team and also by Scrum itself. They perceived that these particular motivated individuals were able to successfully transfer both technical and managerial knowledge to them, given their passion about Scrum. These intrinsically motivated individuals wanted others to be as passionate about Scrum.

For example, Participant 11, a software developer, learned many Agile development-related skills like automated testing and test-driven development, from the lead developer, whom he said simply loved programming and was always motivated to transfer such types of knowledge. In particular, whenever Participant 11 would ask the lead developer for help, the latter would gladly

explain the concept whilst referring to an exact page in the relevant textbook. In doing so, the lead developer went the extra mile, to ensure what Participant 11, his junior, would be on the right track and understood the correct fundamentals. Participant 11 also mentioned that he learned the principle of separating dependences between components when writing unit tests through the same approach, from the lead developer.

Participants also felt individuals who were in Scrum roles could not help but be motivated and wanted to transfer managerial/methodological knowledge about Scrum (Participant 2, Test Manager). Particularly, people learned about the *"philosophy and principles behind Scrum"* and why it works, from their Scrum masters. They found such knowledge helpful in even motivating themselves to be passionate about Scrum as they actually understood the principles behind it, and not just doing it for the sake of doing it. Participant 4, a project manager and former business analyst put it, as follows:

"*You know, it went from just a couple of people's understanding of things; just a couple of sticky notes on the board too; this is why we actually do that and I understand it, so I am going to put more effort into doing it correctly.*"

Sticky notes are one of the concepts in Scrum that are used to remind the team of project tasks to be completed. When they understood its purpose from their Scrum master, they would see greater value in using them.

Then, there was a case of people who were just passionate about Scrum and wanted others to feel likewise. For example, participants mentioned someone who was always online, reading about Scrum and would even organise house chores using a Scrum board. As Participant 2 put it, *"he is literally living Scrum and Agile*". Participant 2 also learned managerial skills from this person pertaining to how to perform cost estimations for Agile/Scrum projects, a task which many people thought could not be done in Scrum.

4.1.1.2 Enjoyment of learning more about Scrum

Participants also felt that people who enjoyed learning about Scrum where able to transfer more knowledge about it. As Participant 11, a software developer put it, *"if someone is motivated to learn, you can ask him a question, and even if they do not know the answer, they will still look it*

up and give you an answer". Therefore, when someone enjoys learning about Scrum, participants feel that they will always be able to acquire new knowledge about it and be able to transfer it.

Likewise, Participant 2, a software test manager, felt the same of his project manager of whom he thought very highly. Participant 2 described the project manager as someone who was able to deploy an entire project on his own, given his competencies in programming, testing and management. Participant 2 also said that the Project Manager "lived and breathed" Scrum, as he was always on his phone monitoring Scrum news. This enabled that participant to learn many skills from his project manager, including how to accurately estimate tasks in Scrum projects, which he said was an uncommon skill to learn.

Similarly, research and development seem to feature frequently in the participants' companies, a process where the team must research into how Scrum can be used better from a technical perspective. Participant 9, a software developer, mentioned how he and his team mates enjoyed researching a different Scrum backlog management framework and presenting them to each other. Even though only one framework was ultimately chosen, the more passionately someone engaged in their research, the more knowledge they would have to share with their peers (Participant 11, software developer).

4.1.2 Extrinsic Motivation

Extrinsic motivation is any motivation for doing a task, other than the task itself (Ryan and Deci, 2000). The two types of extrinsic motivation participant identified from the study are financial motivation and peer-recognition.

4.1.2.1 Financial Motivation

Most participants perceived that financial motivation could have an impact on the extent to which knowledge was successfully transferred by a team member, given their degree of intrinsic motivation (see Figure 4-1). As Participant 8, a software developer, put it, *"there is a trade-off between enjoying your job and how much you are earning".* As reported in the previous section, people who enjoyed their jobs and Scrum would be likely to successfully transfer knowledge about Scrum. However, if they were not being paid well enough, participants felt that money

might become a worry, and a person would feel less inclined to transfer knowledge in a team (Participant 3, project manager and former software developer).

Participant 9 put it quite honestly, in saying that, *"it is difficult to determine someone else's financial motivation but a person can enjoy a bit of both"* (inherent or financial motivation). Indeed, participants find it easier to recognise when a person is intrinsically/naturally motivated to transfer knowledge, but they also realise that there could be some financial motives for people to transfer knowledge in teams. Therefore, financial motivation appears to be an intervening factor in the successful transfer of knowledge given a team member intrinsic motivation level.

4.1.2.2 Peer recognition

Peer recognition was another type of extrinsic motivating factor that appeared to impact knowledge transfer, as Participant 2 stated: *"Yeah, I think peer recognition motivates anyone, so definitely".* Therefore, Scrum team members were aware that peer recognition motivated their peers but could not say whether this affected the knowledge they could transfer to them. However, there was one person, Participant 1, a software developer, who felt that if someone was truly motivated by a task, then peer recognition would not play a part. In his words:

"No, this is an old-fashioned interest ... So the qualities he has in that, is that he is passionate, so he gets excited about that, and I want to know what is he excited."

He felt such people would transfer knowledge simply because they were excited about Scrum, making others want to be as knowledgeable and excited about the methodology.

Therefore, if a Scrum team member was perceived to be motivated by peer recognition it did not clearly seem to affect the perceived extent of knowledge they could transfer. If anything, it seemed that Scrum team members who find others to be externally motivated, might find them less passionate/attractive to learn from, as they do not look genuinely motivated.

4.1.3 Summary

In general, the perceived motivation, particularly the perceived intrinsic motivation of an individual, seemed to increase the perceived extent of knowledge transferred. Such is the case as participants believed that a motivated person is more likely to want to transfer knowledge as

compared to an unmotivated one. In addition, people seemed more likely to want to gain knowledge from them.

It appears that recipients could easily identify when team members were intrinsically/naturally motivated, and they were attracted to gain knowledge from such individuals. However, they did admit that these team members could very well be financially and/or socially motivated to transfer knowledge (extrinsic motivation). Therefore, the financial motivation of team mates seemed to play a moderating role in their intrinsic/natural motivation to transfer knowledge. Therefore, though participants recognize that their peers could be motivated financial, they attribute true motivation to intrinsic motivation, so financial/extrinsic motivation would only aid/moderate this motivation. Furthermore, it appears that perceived intrinsic motivation had a stronger impact on the perceived success of knowledge transfer because the participants reported that they learned a lot from such types of motivated people. Therefore there is an arrow drawn from extrinsic motivation to intrinsic motivation, as it does not seem to directly affect knowledge transfer but rather seems to affect intrinsic motivation which in turn affects knowledge transfer. These results are as illustrated in Figure 4-1 below:

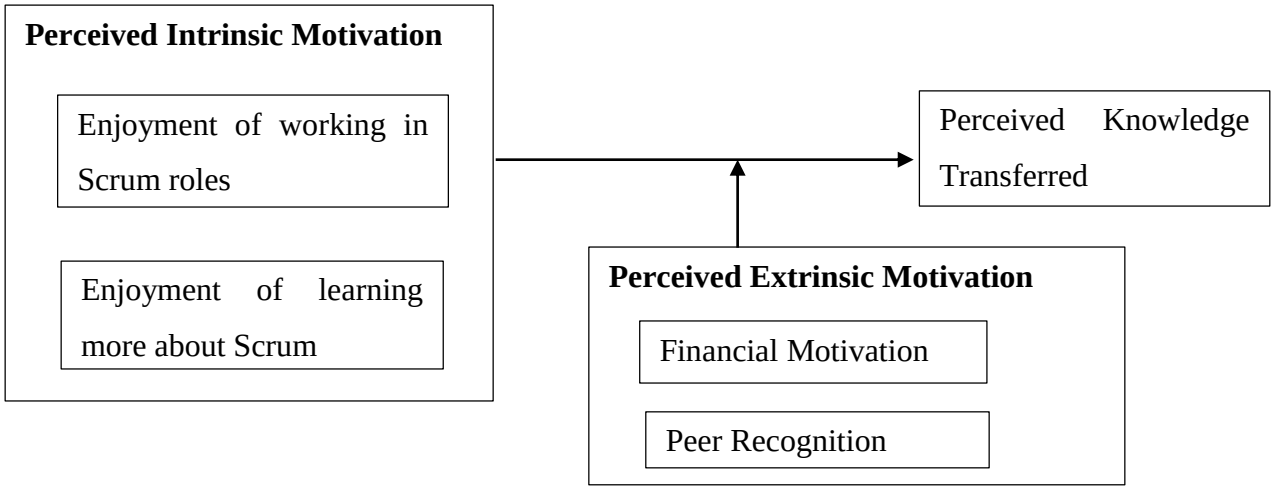


Figure 4-1: Impact of Perceived Motivation on the Perceived Knowledge Transferred

4.2 Perceived Capability

The second sub-research question was, *"How does the perceived capability of a team member affect the perceived extent of tacit knowledge transferred by them?"*. As discussed in the literature review, perceived capability was split up into technical and managerial capability. Themes, supporting evidence, and number of incidents for each sub-category can be found in Table 4-2:

Table 4-2: Perceived Capability

Sub-category	Themes	Incidents	Sample evidence	Source
Technical Capability	Software development expertise.	14	*"Yeah, some of the things I learned or we spoke about, was test driven development, and automated testing…"*	P5
Managerial Capability	Agile Project Management	14	*"I learned how the Scrum infrastructure in Agile worked, that was a big learning curve"*	P10
	Agile Leadership	11	*"Yes, definitely, part of being on the team, we also got to see what his leadership style is, we also took part in the actual process of Scrum, so we also took turns to run the planning sessions"*	P3

4.2.1 Technical Capability

Technical capabilities are any capabilities relating to software development, database and other computer knowledge of a Scrum team member (Lindvall and Rus, 2002). However, the participants associated technical capabilities almost entirely with software development expertise. The "software development expertise" theme is further described below.

4.2.1.1 Software Development Experience

Team members with high technical knowledge and experience, who were also motivated to share their knowledge, were said to be able to transfer a high amount of technical knowledge to others in the team. Several participants judged whether the people had technical expertise by the experience they had. If someone had a lot of experience in a particular software or technology,

people knew they could go to them for help. This is as illustrated by Participant 1, a junior software developer, talking about the lead developer:

"… was a really nice guy, with lots of experience because this J2EE type architecture based on Linux CentOS, I had to learn a lot of that type of stuff and that is where his experience fits in."

Participant 4, a project manager, thought highly of a Scrum master's technical expertise based on her technical experience:

"Very highly because one she has got over 20 years in software development experience, and 10 plus years in Agile."

Participants also reported on the case of a software developer who also learned technical skills specific to Agile software development (for example automated testing, and test-driven development) from a lead developer who had vast Agile development experience (Participant 11, software developer). Test-driven development is an iterative fast-paced software practice that involves solving a solution by incrementally writing and implementing smaller unit test cases (Beck, 2002).

The Agile working environment/setup also appeared to play a role in technical knowledge transfer of software development knowledge. For instance, it was mentioned that this transfer of knowledge was enabled through the pair programming environment, where a senior and more experienced team member would always be paired with a more junior team member (Participant 11, software developer). In that work set-up, Participant 3, a project manager, learned about modular design skills when he was still a software developer from his chief technical officer along the lines of test driven-development. He learned that Agile and Scrum methodologies promote a modular way of design in order to easily adapt to changing requirements.

4.2.2 Managerial Capabilities

Managerial capabilities according to the literature can essentially be split into project management and project leadership(Lindvall and Rus, 2002) . All the sub-categories are described below, given the empirical findings.

4.2.2.1 Agile Project Management Capabilities

It was also found that the Agile project management ability/knowledge of team members increased the perceived extent of knowledge transferred by them. Participants mostly felt that individuals possessed management-related knowledge on the basis of their role and experience.

For example, many participants mentioned that they had learned managerial skills from their Scrum masters relative to the underlying principles of Scrum, how it works and how to quote projects (Participant 2, test manager). Learning how to quote/assign development costs in an Agile project was said to be very difficult due to the iterative nature of projects, but Participant 2 said he could still learn this from his highly experienced Scrum master. Generally, if someone was a Scrum master, participants believed that they could gain managerial knowledge from them, as the role requires them to know a lot about Agile and Scrum management. Participant 2 said as much:

"_____ worked really hard to become an expert in Agile and in Scrum in particular and he, on this project, he was a Scrum master."

Also, this participant felt he had learnt from the Scrum master:

"Go and learn technical skills from developers and things like that but from ____ perspective his role was a Scrum master and our coach and that's really where the learning is from."

In addition to this, participants also felt that Scrum masters with more experience, could transfer greater managerial knowledge. This can be seen by the statement from Participant 4, a project manager, who rated his former Scrum master as highly capable:

"Facilitator: Yes, the coach?

Interview (Participant 4): Very highly because one she has got over 20 years in software development experience, and 10 plus years in Agile".

Also, team members did learn managerial knowledge such as user story/task estimations from other team mates during Scrum planning and review sessions. Not only did the participants get to see other team mates, estimating tasks for each Sprint cycle, but they would also take turns in running Sprint planning session and estimating tasks themselves with the team's (development team's) support (Participant 3, project manager).

4.2.2.2 Leadership Capabilities

Participants also said they were able to easily gain Scrum leadership knowledge from their Scrum masters who possessed such knowledge, enabling them to successfully transfer it to others. According to participants, Scrum leadership is about effective communication, facilitation, and making it easy for team members to express themselves as best as they can.

It appears that Scrum masters were usually very good at communicating with clients and people in general. Many of the participants' Scrum masters had a business analyst background, where they had to communicate with clients and development teams. This was echoed by Participant 9, a software developer, who mentioned that he learned the following from his Scrum master in terms of communication with clients:

"*You can't be mean to them, even though I have good intentions, or just trying to correct them, you must approach … I guess business is business, you have to have certain tact and what no t… he is really good at that."*

This was in relation to a case where the software developer had a tense relationship with a client. That developer could not help writing aggressive emails to an indecisive client, at which point he asked the Scrum master to screen and tone down each email he would later send to that client.

Another example, of Scrum master's leadership abilities and the potential for others to learn from it, was shown by Participant 12, a Scrum master and project manager. He spoke of a time when an unhappy team member was shouting at him in front of the whole team. The Scrum master took the person outside and asked them what was wrong. The level of composure of the Scrum master, would have spoken volumes to the rest of the team.

4.2.3 Relationship between Experience, Perceived Capability and Perceived Knowledge Transferred

It is evident from the above results that participants perceive a person with experience to be highly capable and able to transfer knowledge to them. This is illustrated by Participant 1's comment, who was a junior developer that learned a lot from his lead developer:

"… was a really nice guy, with lots of experience because this J2EE type architecture based on Linux CentOS, I had to learn a lot of that type of stuff and that is where his experience fits in."

So here it can be seen that the lead developer's experience made him capable of transferring technical knowledge to the junior developer.

Furthermore, as mentioned before, Scrum masters with more experience were also highly regarded, and therefore were perceived as possibly being able to transfer a great amount of managerial knowledge to team mates. This can be seen from the quote below, where a Scrum coach (and master) is spoken of by her team mate:

"Facilitator: Yes, the coach?

Interview (Participant 4): Very highly because one she has got over 20 years in software development experience, and 10 plus years in Agile."

In addition to this, Scrum team members also felt that they gained managerial knowledge on Agile methods like self-organisation of teams, based on the experience they had interacting with Scrum masters, as can be seen from the quote below:

"… he had a good understanding of the process, he had communicated it to the team and essentially the team self-organises to deliver what needs to happen, so from the limited experience I had it was clearly good."

This relationship can be illustrated in Figure 4-2:

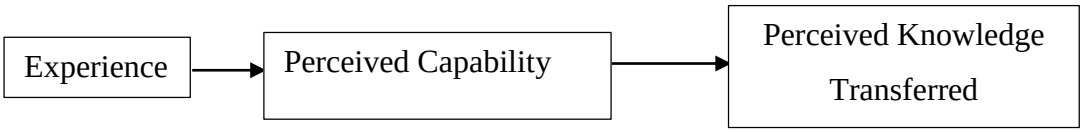


Figure 4-2: Relationship between Experience, Perceived Capability and Perceived Knowledge transferred

4.2.4 Relationship between Qualification, Perceived Capability and Perceived Knowledge Transferred

Also visible in this section is the relationship between qualification and perceived capability. Participant 7, a project manager, mentioned that as a new graduate, being one of the few software developers with a degree at his work, he already knew more than most of his seniors and struggled to gain knowledge from them. As he put it:

"So my first actual job as a developer as a graduate, I arrived in a small little team where I knew unfortunately a bit more than most people who were working there for years and you do not actually learn, you need somebody that is stronger than you to learn from them."

He thus left that company and joined the current one so that he could learn from more qualified and experienced developers.

Another example of this relationship was given by Participant 11, a software developer, who mentioned that at his company it was compulsory for a business analyst to be Prince II project management-certified. This is because that company realised the credibility and perceived capability that having such a qualification would bring. As he put it:

"I refused to take the Prince II examination at Company B but business analyst are required to take that and some developers do take Prince II, I said no I do not want to take it, it is probably because when I was doing my Masters before I started working… It was a critique from some

guys in the UK, they were criticising the current and dominant models in project management, things like your Prince II."

He refused to take the Prince II certification as his Masters research led him to be against such accreditations. The company probably made such an examination compulsory as they realised the perceived managerial capability of having such a qualification would show to other Scrum team members (McHugh and Hogan, 2011). This relationship will now be shown in Figure 4-3 below:

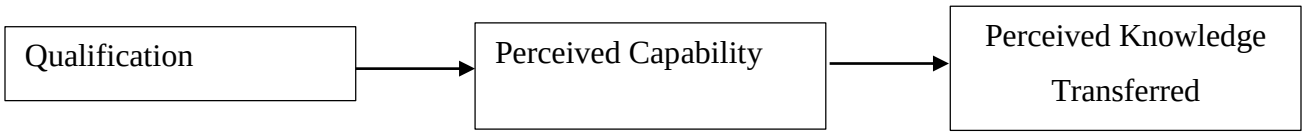


Figure 4-3: Relationship between qualification, perceived capability and perceived knowledge transferred

4.2.5 Relationship between Perceived Motivation and Perceived Capability

As previously mentioned, perceived technical capability increases the perceived extent of tacit knowledge transferred. Participants perceived that when they approached a person with high technical capabilities, they will always find an answer (Participant 11, software developer). Participants usually determined this by looking at the experience that the team member had.

The findings revealed that a team member can also be highly technically capable and experienced (or in general) but might not be motivated (intrinsically) to transfer knowledge maybe due to power issues. In particular, they may feel exclusive knowledge ensures job security. As Participant 6, a project manager says:

"Technically competent people believe they must protect their IP, but they do not realize in the fact that they are sharing the knowledge there in lies the power, their asset is worth nothing if it only sits with you, it is only when you start sharing it out, do people start referring to you, they will pay more money to get to speak to you."

Indeed, team members may see it logical to withhold knowledge to ensure job security but then again, they will not be considered to be experts in their areas.

Furthermore, another software developer, Participant 11, also picked up on this lack of willingness/intrinsic motivation of capable team members to share information. He mentions how when they gained new knowledge on a subject through research for the company, they are unwilling to transfer it to others:

"Interviewee: They do not believe in knowledge sharing, you did the research on your own, spent your own time so why should you?

Facilitator: [laughs]

Interviewee: Teach other people who are lazy or whatever."

Therefore, highly capable and experienced Scrum team members must also be intrinsically/genuinely motivated to share knowledge in order for knowledge transfer to happen, as the above quotes indicate.

This relationship is shown in Figure 4-4 to follow. Once again, arrows pointing to other arrows, show that the characteristic (such as intrinsic motivation) has been shown in the findings to influence another (such as capability).

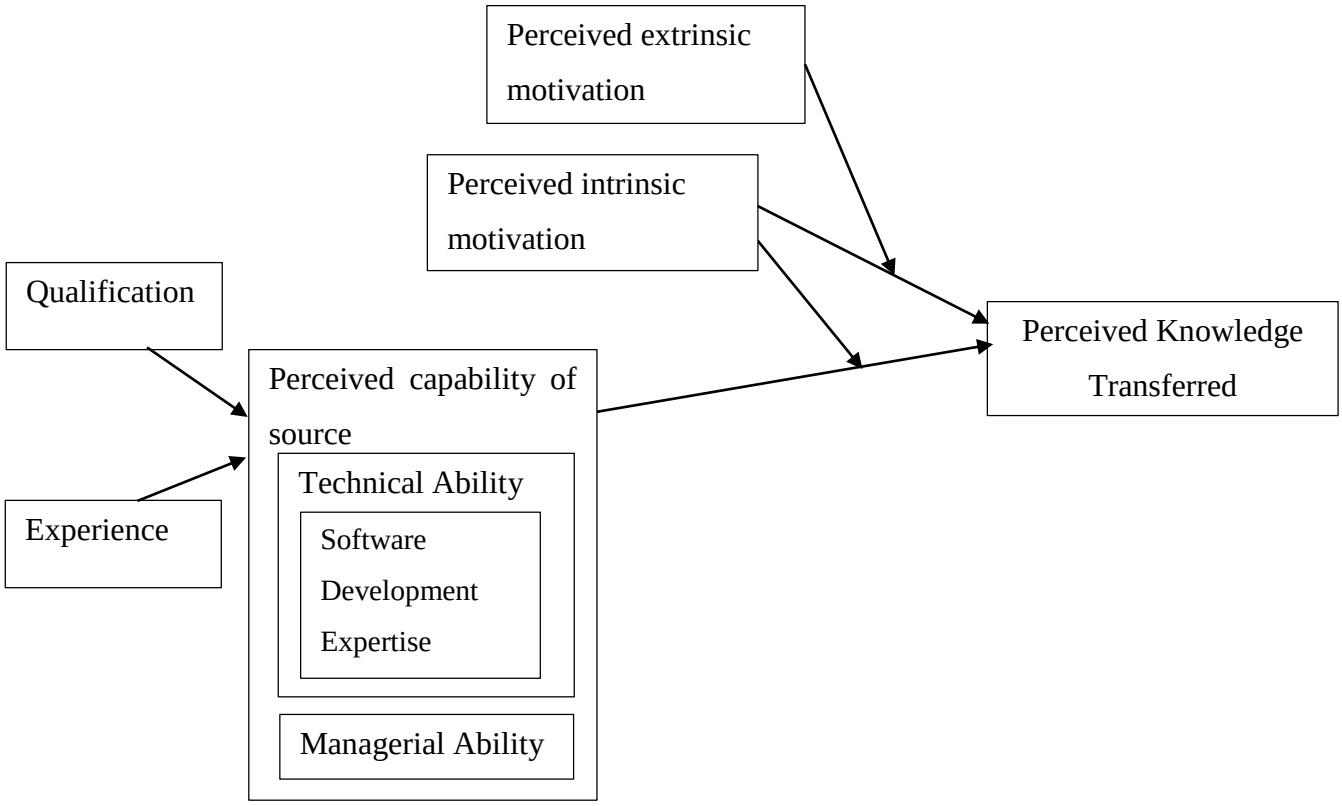


Figure 4-4: Relationship between Perceived Motivation and Perceived Capability

4.2.6 Summary

The findings reveal that the perceived capability of a team member does improve the perceived extent of knowledge transfer. Participants perceived team mates to be technically/managerially capable if they were experienced and qualified in that area. Experience was often what participants would mention about what made their colleagues good at what they do.

Furthermore, participants also frequently mentioned how they gained managerial and leadership knowledge from Scrum masters and sometimes their own peers. Scrum masters are typically well versed with Agile and Scrum methodology and were thus able to give team mates a genuine appreciation of Scrum and its principles. Furthermore, their leadership abilities, such as how to deal with conflict and clients, was something that many participants gained from them. Indeed, participants did also learn from other team mates on how to estimates user stories/tasks, as the development team is responsible for task estimations in Scrum.

It was also evident that a team member must not only be perceived as capable but also be motivated to transfer knowledge. This is because participants were sometimes said to be unwilling and unmotivated to transfer knowledge, as they felt that they could lose power. Furthermore, participants said they found positive and motivated (or rather intrinsically/ genuinely motivated) people attractive, so such a person would be unattractive to them.

The results are summarised in Figure 4-5 below:

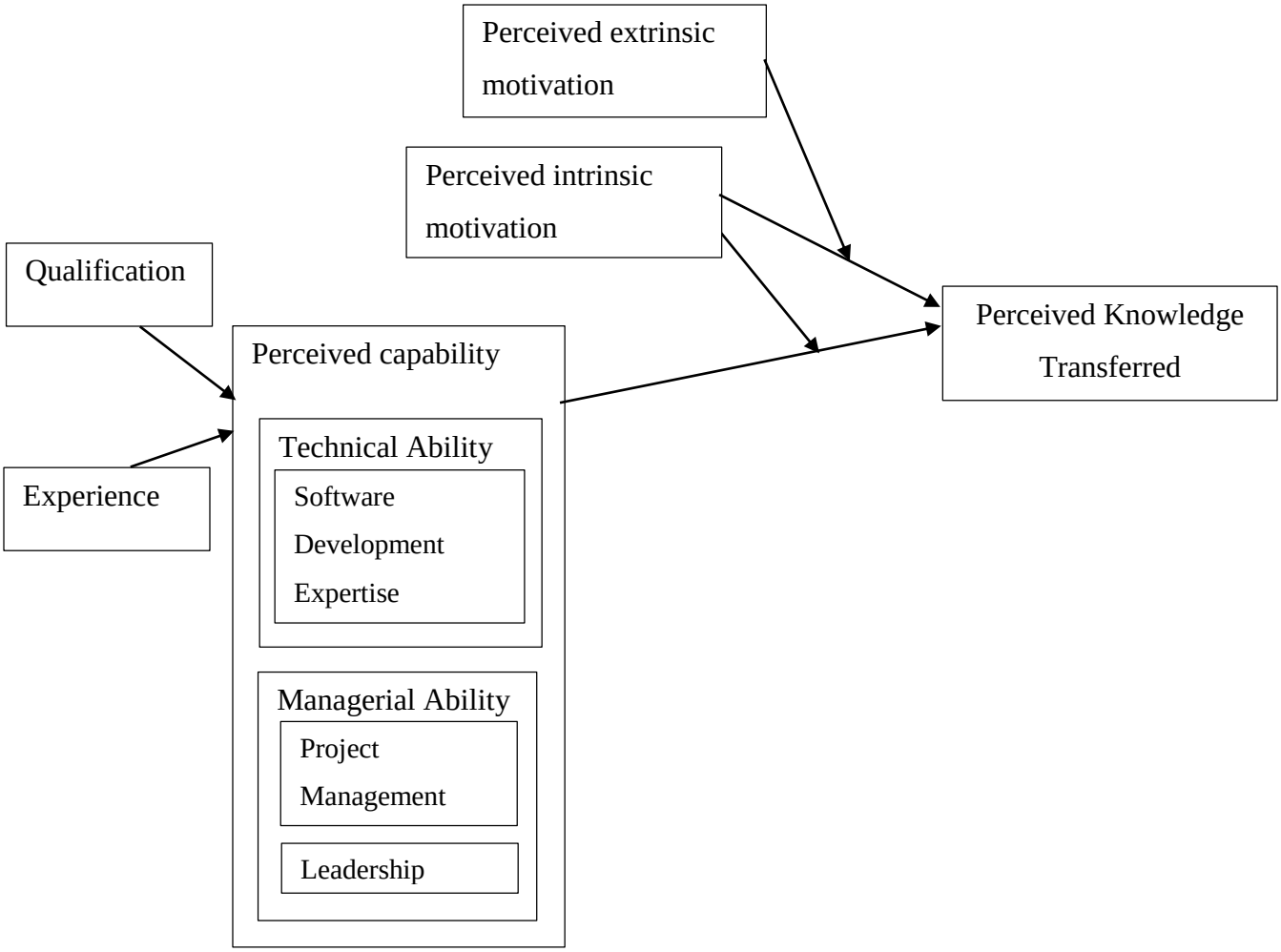


Figure 4-5: Perceived Capability's influence on the Perceived Knowledge transferred

4.3 Perceived Credibility

The second sub-research question was, *"How does perceived credibility of a team member affect the perceived extent of tacit knowledge transferred by them?"*. In most cases team members found each other trustworthy and reputable. Table 4-3 illustrates the themes found in this section and their supporting evidence:

Table 4-3: Perceived Capability

Sub-category	Themes	Incidents	Sample evidence	Source
Trustworthiness	Professional trustworthiness	17	*"Trust again is another pillar of Agile, another one of the 12 principles is to build teams around individuals, motivated individuals who can be trusted"*	P12
	Personal trustworthiness	2	*"We always saw him as an incredible leader, today he is one of my best friends, he was groomsman, at my wedding."*	P3
Reputation	Reputation	5	*"Very highly because one she has got over 20 years in software development experience, and 10 plus years in Agile."*	P4

4.3.1 Trustworthiness

From the interviews, two types of trust seemed to emerge, professional and personal trustworthiness. These are further elaborated below.

4.3.1.1 Professional Trustworthiness

Trust was described as one of the necessary pillars behind Agile Software Development (Participant 3, project manager) and was perceived as essential for knowledge transfer by recipients (Participant 4, project manager). Therefore, the trustworthiness of the source does increase the perceived extent of knowledge transferred to other Scrum team members. The lower it is, the more a recipient will doubt the knowledge received. All participants said that they did trust their team mates professionally, in the work they did. Most felt that trust should be earned

but some reported that they just gave their trust until proven otherwise (Participant 9, software developer).

Participants trusted their team members based on the fact that they demonstrated some degree of integrity, did what they said they would do, proved that they were competent (Participant 12, project manager), and on how well they handled issues throughout the projects (Participant 9, software developer). Furthermore, participants also trusted senior team members or managers on their level of transparency, i.e. the extent to which they did not hide anything senior management said from the team. Here is such an illustration by Participant 8, a software developer, talking about how the team had been granted access to the scheduled backlog/sprint items by their Scrum master:

"He never hid the figures from us, so there was always visibility, there was no reason to doubt what he was saying, I could always open the backlog myself and see how big it is and see the deadline, so if I ever had any doubt, which I did not, all the data was always available to me, that I could verify it."

Therefore, since the participant had no reason to doubt the Scrum Master, he was more prone to accept the relevance of good agile project management skills and subsequently acquire those skills from the Scrum Master. For example, he learned how to track progress in a Scrum project, as can be seen from the quote below:

"So I mean you take it delicately, he was very good with explaining the facts and the numbers, ah …so I mean he took a very factual approach, how many story points have we completed in whatever period, how much we need to complete the project, discuss the impact it will have if we are late, ummm, and then try to make decisions now on the way ahead."

Furthermore, Participant 9, a software developer, gained customer relationship knowledge (managerial knowledge) from his Scrum master, after he trusted him enough to allow him to help in dealing with a difficult client by censoring the emails he sent. As he put it:

"I ended up screening all my mails through him, to just take the bite out of it, because you know in client communication, you can't, ultimately the client pays for whatever… you can't be [rude]".

Indeed, one could also say that trust worked both ways here, as the Scrum master also trusted the participant's decision to have his emails censored. Through this experience, he (Participant 9, the software developer) also recognised how to carefully and delicately communicate with clients as they are key to a company's business.

4.3.1.2 Personal Trustworthiness

It also appeared that some team members trusted each other at a personal level as they even became friends, would go out for drinks together and could share personal information with each other. As Participant 9, a software developer had to say about his Scrum master:

"I guess it is just the social, outside of work he got relationships with people and it is never, I have been out with drinks for a few times with him and we have been on projects … I know he is a good guy. I know I can trust him with work related stuff as well."

So from this above quote, it can be seen that as a result of the personal and social nature of the Scrum master, Participant 9, felt that he could trust him with work-related matters as well. Indeed, this participant even trusted the Scrum master with helping deal with a difficult project client, and as a result, he learned how to better deal with difficult clients from his Scrum master.

Furthermore, Participant 10, a software architect, had this to say about his Scrum master:

"We always saw him as an incredible leader, today he is one of my best friends, he was groomsman at my wedding."

This is an example of a trust that not only enabled knowledge transfer but also enabled the software architect to allow the Scrum master to become his friend. Though they were friends, he also learned how also to manage and motivate teams from his Scrum master, as he put it:

"So from his perspective, I would say that I learned the most. Just sort of how do you manage teams, how do you motivate a team, how do you motivate people in a Scrum environment if not obviously controlled properly can easily get out of control."

4.3.2 Reputation

The reputation of a source was shown to be important to successful knowledge transfer. Many participants reported felt their team mates were highly capable because the latter's reputation of having a lot of experience and having seen them produce good work before. Reputation was also based on the qualification of an individual. Formal qualifications made participants view their team mates as more reputable sources of knowledge. These findings will be elaborated on in the sections to come.

4.3.2.1 Relationship between perceived reputation and experience

The reputation of team members seems to be strongly influenced by how experienced people perceive them to be and this also affects knowledge transfer. For example, participants seem to be willing to learn from team members who have a high level of experience and subsequently gain knowledge from them; such is also the case for Participant 4, a project manager, highly regard his former Scrum master's capability because of her extensive experience:

"Very highly because one she has got over 20 years in software development experience, and 10 plus years in Agile."

This participant later said that he had learned a lot from this Scrum master, as can be seen from the following quote:

"Facilitator: …did you learn anything in that regards?

Interviewee: Absolutely, you get Agile for project managers which is about how to build a backlog of requirements, how to groom a backlog, how to split the release plan or the backlog into sprints, how to estimate, how to prioritise, how to manage scope."

Furthermore, based on people's experience working with a team member, they perceive them to be credible and therefore reputable as well (Joshi et al., 2007). Participant 12, a project manager, gave this statement about why he found a Scrum master/coach he worked with to be reputable:

"Yes, I had already worked with this person in a previous organisation, they had credibility for me, but so it was quite easy for me to trust."

Furthermore, because the Scrum master/coach was found to be credible and reputable by the project manager, he was able to learn how to become a Scrum master himself from the coach:

"So the best way to learn thinking like a Scrum master or Scrum coach is the way you learn generally, is they observe you in a situation and then they give you advice on how best to tackle that situation."

The amount of experience that a team member has or their experience with others, will determine how reputable and credible a source they will appear to others. Furthermore, as in the above quote, where someone learned how to be a Scrum master, it is imperative that the person that they are learning from is found to be credible, so their knowledge can be trusted and transferred. These relationships can now be shown in Figure 4-6 below:

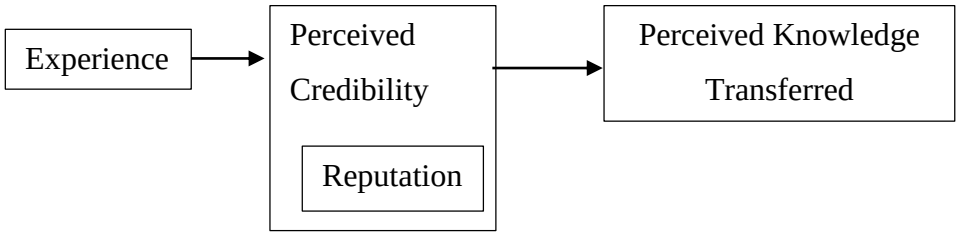


Figure 4-6: Relationship between perceived capability and credibility

4.3.2.2 Relationship between reputation and qualification

The reputation of a source also seemed to be based on their qualification. As mentioned before, Participant 7, a project manager, also brought up the issue of qualification and a person's ability to transfer knowledge. He mentioned that as new graduate, being one of the few software developers with a degree at his work, he already (or felt he did) know more than most of his seniors and struggled to gain knowledge from them as he viewed them as less reputable and therefore would find their knowledge less credible/trustworthy.

Likewise, Participant 12, a software developer, also took note of the qualifications that his fellow Scrum team mates had, as can be seen from the quote below:

"People somehow tend to form relationships with different people ... we pair in a Scrum as a very cross-functional team … that has people from a different backgrounds, some did computer

science, some did information systems, others do not have a university qualification at all, you find that our passion for the future are different."

Therefore, the qualification of his fellow team members also influenced what he could expect of them and even the relationship he had with them, which could have influenced the knowledge he could gain from them. So overall, as can be seen, particularly from the first case above (by Participant 7, the only graduate software developer in his old company) that qualification of a Scrum team member does indeed influence their reputation and the potential amount of knowledge they can transfer to others.

This is illustrated in Figure 4-7 as follows:

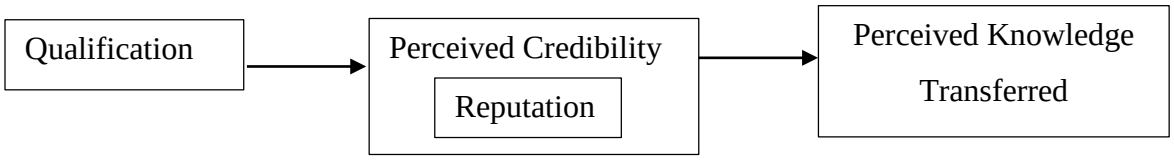


Figure 4-7: Relationship between Qualification and Perceived Credibility

4.3.2.3 Relationship between reputation and trustworthiness

From the findings, it was found that that there is a relationship between reputation and trustworthiness. When people found their fellow Scrum team members to be reputable (through either their experience or qualification) they seemed to trust their knowledge. Using a previous example, when Participant 12, a project manager trusted his current Scrum team mate as they had worked together at a previous organisation and hence found him to be reputable:

"Yes, I had already worked with this person in a previous organisation, they had credibility for me but so it was quite easy for me to trust."

Also, in another example, Participant 7, a project manager, spoke about how he found his Scrum master's software development knowledge as trustworthy, since he knew that he had over 10 years of experience in that area:

"He also has a developer background, he has 10 years total more experienced than me…he understands how software works."

This is summarised in Figure 4-8 to follow:

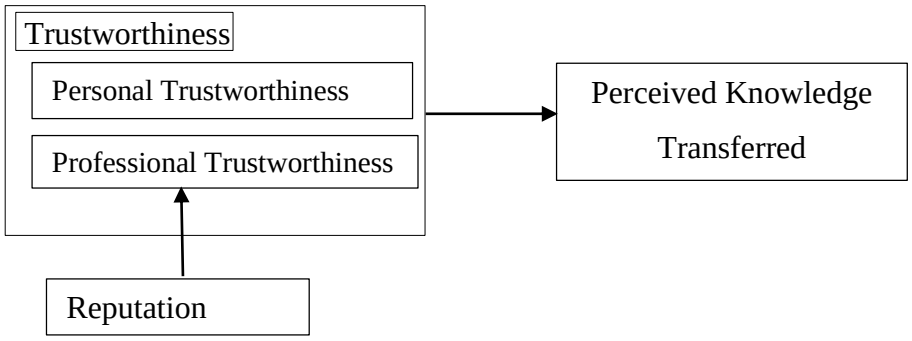


Figure 4-8: Relationship between Reputation and Trustworthiness

4.3.3 Summary

From these results, it is clear that the more credible a source is to the recipient, the greater the knowledge transferred. The source was perceived as credible the more the recipient both trusted the message, and found the source to be reputable. Hence, from the results, credibility was found to comprise elements of reputation and trustworthiness. The reputation of a team member was influenced by how qualified and experienced people viewed them. Essentially, if they viewed the team member as qualified and experienced, then they would be seen as reputable and their knowledge as trustworthy. It was also found that knowledge transfer cannot happen without trust, as the recipient will doubt the credibility of the knowledge.

There were also two types of trustworthiness reported by participants: professional and personal trustworthiness. Professional trustworthiness could be seen as the trust that participants had in

their team mates' professional abilities. Personal trustworthiness appeared to be the trust that friends had for each other, as some participants would relax with colleagues after work. The results of this section is depicted in Figure 4-9 below:

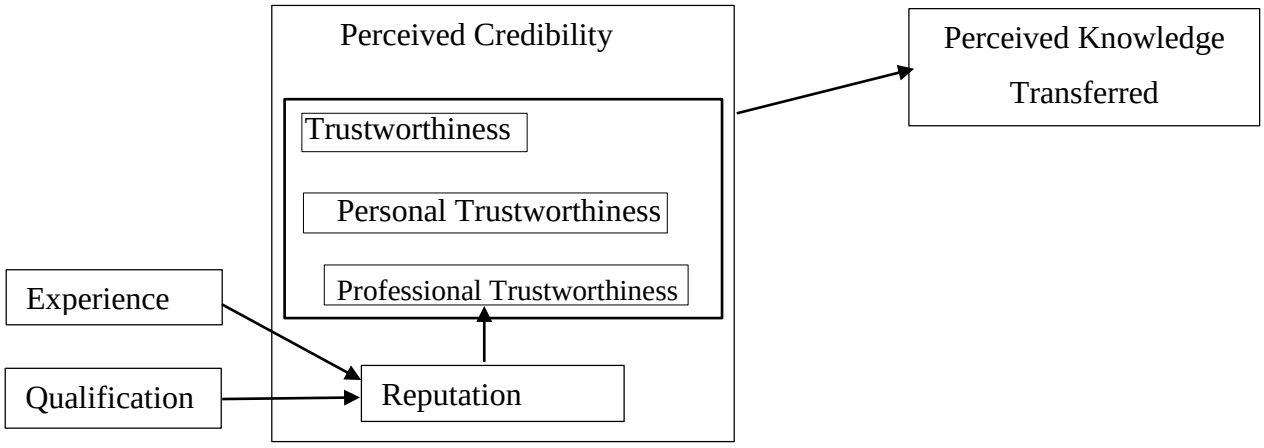


Figure 4-9: Credibility's influence on the Perceived Knowledge Transferred

4.4 Perceived Communication Extent

The fourth sub-research question was, *"How does the perceived communication extent of a team member affect the perceived extent of tacit knowledge transferred by them? "*. The following Table 4-4 is a thematic map of this section:

Table 4-4: Perceived Communication Extent

Sub-category	Themes	Incidents	Sample evidence	Source
Perceived Communication Extent	Frequency of Communication	11	*"Yeah, definitely, co-location is necessary in Scrum, if you are in a room with someone, sitting next to someone, you get to know how they think, you get to know their skillsets and they get to know yours"*	P3
	Communication Balance	11	*"Yes and no, with communication you need to have a balance… ummm… too little and you will not understand what you are doing and you will be lost and too much you will get to a point where you will just end up getting frustrated."*	P2

4.4.1 Frequency of Communication

A team member's ability to communicate frequently was said to generally increase the perceived extent of knowledge transferred. Most participants reported frequent communication between team members and this was typically due to the nature of Scrum. Knowledge transfer and communication in Scrum they say, occurs at many points including: the daily Sprint meeting, Sprint planning session, demonstrations, reviews and retrospectives (Participant 5, business analyst and occasional Scrum master). During the daily Sprint planning meeting, for example, the Scrum master encouraged everyone to contribute and share knowledge. This can be seen from Participant 12, a project manager, who had this to say:

"...when you have a planning session…you can encourage or force, well force is the wrong word, but help that person to actually contribute, so even though they were not comfortable, yes, for you to draw them out of their shell."

So, in this case, the Scrum master was intent on getting all members to communicate in the planning session, whether shy or not. The ability of a team member to communicate during these sessions determined the extent of knowledge that they would transfer. Also, Scrum masters

would also always communicate with their team in all these Scrum ceremonies, so participants generally learned from them. However, in one Agile team, the communication between the team and Scrum master was not always as frequent as it should have been until the Scrum master realised the team was always exceeding time on deliverables. Therefore, the communication between the Scrum master and the team increased, and it allowed knowledge related to problem solving to be transferred to the team members. This was the case since they could discuss how problems were to be solved early on (Participant 5, business analyst).

Another reason for a team member to communicate regularly in Scrum to transfer knowledge, is that Scrum prescribes this through co-located teams. Participant 11, a software developer, mentioned that in their company, the walls between offices had just been removed, enabling easy communication between team members. This meant that a developer could easily communicate with a tester or business analyst in the same building and not even have to leave their seat. In this way the developer could easily gain requirements knowledge from the analyst and the analyst could easily gain knowledge about technical difficulties in the project from the developer.

Pair-programming was another practice from Agile development's XP methodology (often applied during Scrum) that also showed the need for frequent communication to transfer knowledge. Pair-programming allowed developers to easily bounce ideas off each other (verbally) as they sat next to each other. This allowed for developers to solve problems/bugs together and easily educate/transfer technical knowledge to each other. Furthermore, since software developers pair-programmed with business analyst, it also allowed them show analyst their code and easily transfer technical knowledge such as SQL scriptwriting to them (Participant 5, business analyst). However, not all participants enjoyed having frequent communication, as will be discussed in the next section.

4.4.2 Balanced Communication

Most participants said frequent communication is good for learning and knowledge transfer, but there should be a balance. Therefore, they perceived that anyone seeking to transfer knowledge should not overdo communication. Many felt that without communication, someone could be lost and since they did not communicate their problems, they could never learn and gain some

knowledge. Yet, on the other hand, someone might want to ask others questions all the time and ultimately end up irritating them. The following quote sums it up quite nicely:

"Too little and you will not understand what you are doing and you will be lost and too much you will get to a point where you will just frustrate me" (Participant 2, test manager).

Indeed, people said too much communication could have the effect of disrupting someone's progress and also irritate that person (Participant 5, business analyst). Therefore this could harden the relationship between the sender and recipient, reducing the knowledge transferred as the recipient would reject/ignore the knowledge being transferred to them.

4.4.3 Summary

To summarise, a team member should communicate regularly to improve knowledge transfer, but too much communication may harden the relationship between them and the recipient, then it would decrease the amount of knowledge transferred. Finding a balance may be common sense, like not sending someone 50 emails a day, and it could also be determined over time, based on the recipient's feedback. Therefore, communication extent does affect knowledge transfer in Scrum teams, but it should be balanced as illustrated in Figure 4-10 below.

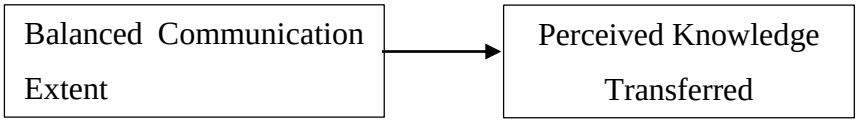


Figure 4-10: The influence of Communication Extent on the Perceived Knowledge Transferred

4.5 Other suggested characteristics needed to transfer knowledge

The final sub-research question was, *'What other characteristics are associated with team members that can successfully transfer tacit knowledge in Scrum projects?'* This section describes the participants' perception of what makes someone good at transferring knowledge through communication.

Literature states that the greater the communication skills of an individual, the greater the knowledge that will be transferred by that individual (Ko et al., 2005). Therefore, participants were asked what characteristics make people good at communicating. The most popular characteristics suggested were: empathy, articulation and some characteristics that empathy produces (Nonaka and Takeuchi, 2011; Zhou, 2014) – willingness to listen and patience. Sample evidence of these findings is shown in Table 4-5 to follow. Table 4-5

Table 4-5: Other suggested characteristics needed to transfer knowledge

Sub-category	Themes	Incidents	Sample evidence	Source
Suggested characteristics of a person who can transfer knowledge well	Empathy	8	*"He is clear and he is communicating on my level, so he is empathetic"*	P1
	• Willingness to listen	4	*"Some people say that a good communicator is a good listener, so, somebody who is able to listen to you when you talk is mostly likely going to be able to answer and give you what you are looking for."*	P11
	• Patience	4	*"Must have the patience , to think about stuff that he communicates"*	P6
	Ability to be articulate	*7*	*"So the basics, is that you must be articulate, there is no point in communicating if you are not going to*	*P7*

			make sense when you open your mouth."	

4.5.1 Empathy

Participant 3, a project manager, defined empathy as the *'ability to relate with the people you work with'* and this was found to be essential for knowledge transfer to succeed. Others mentioned how they benefited from empathetic team members who communicated without jargon as they understood the recipient's perspective (Participants 1 and 11). For example, Participant 1, a software developer, mentioned about a senior developer:

"He is clear and he is communicating on my level, so he is empathetic, he is not going to talk over my head, just because he knows more, he comes down to my level of understanding, he has understood me well."

Yet on the other hand, some developers complained about technically brilliant colleagues who assume that you know a lot already, and speak in ways that are hard to follow (Participant 11, software developer). To be empathetic, one would actually take time to ask the recipient whether they have understood, what has been communicated to them and have them even explain it back (Participant 6, project manager).

Indeed, some point out that empathetic team mates are also usually willing to transfer knowledge, as Participant 1, said about a senior developer he called empathetic in the previous quote above:

"*Yes, very much so, very keen to explain things , patient, look I think if he tells me something and I do not remember it, that might stretch his patience but that is understandable.* "

The following points are consequences of empathy: willingness to listen, patience. These points will now be discussed in turn.

4.5.1.1 Listening Skills

One of the most popular characteristics of a good communicator (and ultimately someone who is able to transfer knowledge) mentioned in all the interviews was listening skills. Participant 11, a

software developer, explained that the benefit of listening is that you can give someone the answers they are looking for:

"You know some people say that a good communicator is a good listener, so somebody who is able to listen to you when you talk is mostly likely going to be able to answer and give you what you are looking for."

Also, a few participants also pointed out that listening also helps someone to adapt what they need to say based on the audience/recipient's response. Participant 12, a program manager and former developer with 18 years' experience, mentions:

"Ah, I think a good communicator should be, well the most important thing is that they need to have good listening skills, so that they can tailor what they want to say to speak to the audience."

From the above quotes, it can thus be posited that listening is important in transferring knowledge as one becomes more aware of what the recipient needs and can provide them with the knowledge and information they require. Otherwise, the wrong information will be transferred to the recipient and they might become irritated, and not gain any knowledge.

4.5.1.2 Patience

Patience, in general, was also a concept that participants found necessary in order to transfer knowledge. Participant 6, a project manager, points out that one needs to demonstrate some degree of patience while interacting with people and transferring knowledge:

"Umm, often times we assume people know stuff, we have to take time to get people to articulate back to us what we have told them so that we can make sure they understand clearly what we intended them to understand. But I think probably the biggest crime follows, we do not have the patience for that."

Furthermore, another interesting point was also mentioned about patience, in that a person: *"must have the patience, to think about what he communicates."* (Participant 6, project manager). Indeed, as the saying goes, "you should think before you speak". This is probably so as to not hurt or offend the recipient (Participant 9, software developer), to help ensure a good

relationship between the sender and recipient, which would result in better knowledge transfer (Szulanski, 1996).

4.5.2 Ability to be Articulate

Many participants, 7 out of 12, mentioned the need for a person to be able to clearly articulate/express what they need to say in order to transfer knowledge. They said that it was also important for a person to consider the audience, in order to express their message in a way that the recipients can understand, as described by Participant 12:

"They need to be able to clearly articulate their mission or goals, in a way that people, the person they talk to understands it, so that might differ depending on the person you are talking to, that makes a good communicator".

For that matter, participants mentioned the need to be brief and concise in order to effectively transfer knowledge (Participant 5). However, a participant also added that a person's language background or accent could be a potential barrier to a person being able to articulate themselves clearly:

"Okay, my challenges were more personal than anything, I think I have always had issues I think with my English, I have always been like maybe this guy does not understand what I am saying with my accent" (Participant 11, software developer).

4.6 Summary

This section discussed the characteristics needed of a team member needed for knowledge transfer, as suggested by participants: empathy, articulation, and some characteristics that empathy produces, such as willingness to listen, and patience. Of these, it appears that willingness to listen, and patience, should all be brought about by empathy. However, in the following chapter, the discussion chapter, the findings will all be justified from the literature.

5 Discussion

This chapter discusses the findings of the research, in the light of the research questions and related literature.

The literature uses tacit knowledge transfer and knowledge transfer interchangeably, as knowledge transfer, in general, usually incorporates this difficult-to-articulate (tacit) knowledge (Bassellier et al., 2003; Karlsen et al., 2011) that is dominant in Agile development teams and usually transferred through face-to-face interactions and verbal communication (Hoda et al., 2012; Melnik and Maurer, 2004).

One of the main contributions of this research is the introduction of moderating variables between already well-known characteristics that influence knowledge transfer. In addition, a new characteristic, empathy, has been identified as impacting knowledge transfer. In essence, the results found the following perceived attributes of the source key to influence knowledge transfer in Scrum teams: motivation, capability (qualification and experience), credibility, communication extent and articulation of the source and their empathy.

Empathy with the source is a new characteristic found to be important in tacit knowledge transfer in Scrum teams and should be integrated in future models. Empathy was shown to make Scrum team members more motivated and willing to transfer knowledge. Indeed, Van Lange's study (2008) found that empathy improves a person's intrinsic motivation to help others. Furthermore, empathy was also found in the present research to improve knowledge transfer as a person who is empathetic will want to express themselves in a way that the recipient can best understand. It has also been found in another study that empathy in software development teams, leads to effective communication, collaboration and faster software development (Akgün, Keskin, Byrne, and Gunsel, 2011).

Perceived motivation, together with the perceived capability of the source, improves the perceived extent of tacit knowledge transferred, which was predicted by Joshi et al. (2007). A Scrum team member with good qualification and experience was perceived to be reputable and capable by others, hence seen as having valuable knowledge to transfer. This is in line with Szulanski (1996), who stated that if someone is not viewed as an expert, their knowledge would not be seen as reliable and might be questioned (Szulanski, 1996). Furthermore, in the present

study, if a Scrum team member was seen as reputable, their professional knowledge (technical and managerial) was found to be more trustworthy (Joshi et al., 2007).

It was also found that in Scrum teams, it is not just the communication extent of the source, but also verbal communication skills/articulation that improves tacit knowledge transfer, a finding that Joshi et al. (2007) also predicted as possible. All of these findings can be summarised in Figure 5-1 below, with new findings to the model of Joshi et al. (2007) and the literature highlighted in orange:

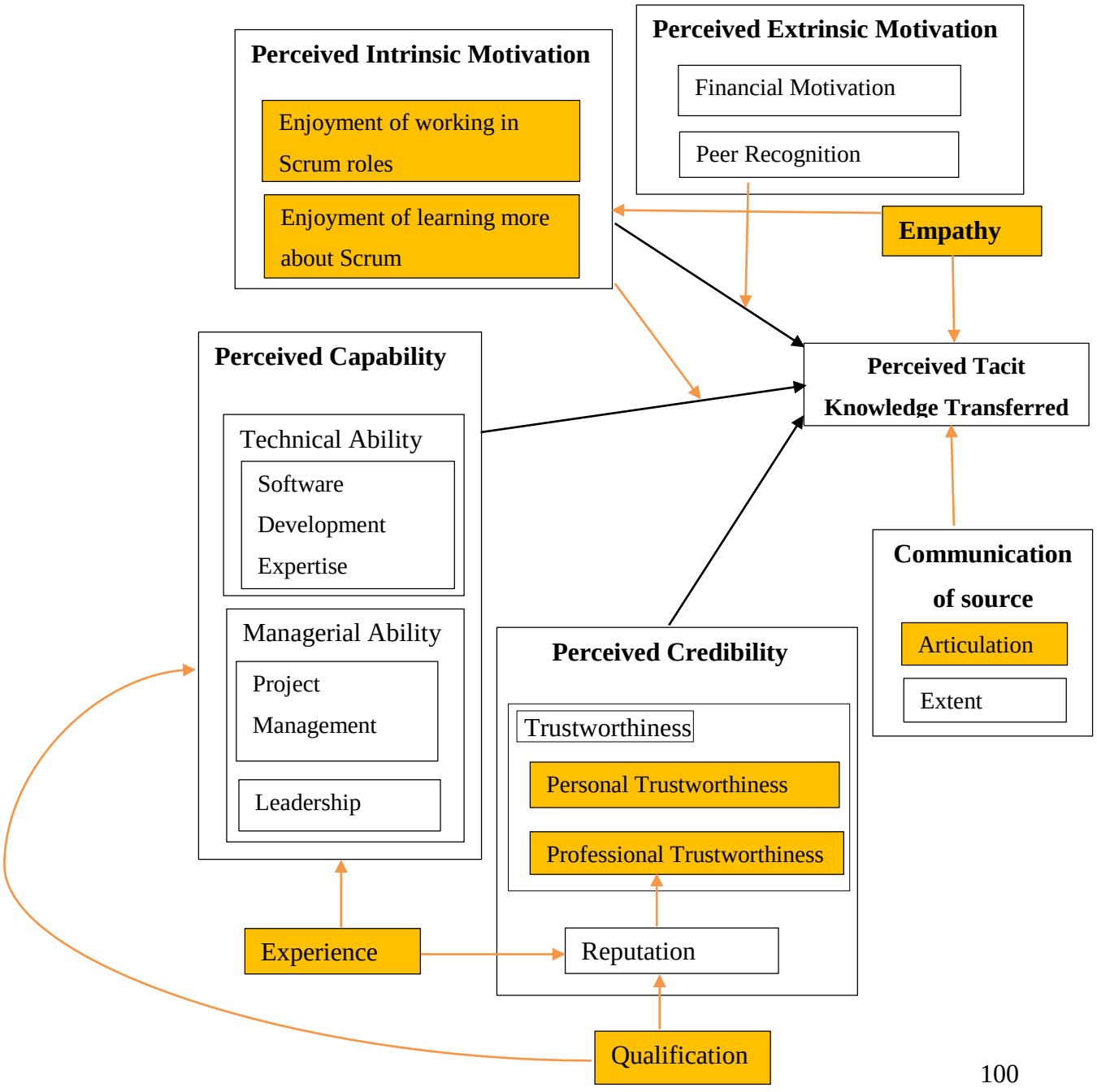


Figure 5-1: Final Theoretical Model for Research – suggested characteristics that a Scrum team member needs to transfer knowledge successfully within the team

These findings will now be compared and contrasted with existing literature. The relevant implications to industry and practice of this model are then discussed in the last section of this chapter.

5.1 Perceived Motivation and Knowledge Transfer

In this research, it was found that a Scrum team member who is perceived to be motivated will be able to transfer tacit knowledge to people in the team, as people find motivation attractive and sought to gain knowledge from such motivated individuals (Stanca et al., 2009; Trauth, 2010).

Team members were inclined to acquire knowledge from their peers due to their natural/intrinsic motivation to use Scrum and their passion for learning about the methodology. Indeed, the literature defines signs of intrinsic motivation more broadly such as enjoyment of job and enjoyment of learning (Nikitina, Kajko-Mattsson, and Strale, 2012), but does not seem to have related intrinsic motivation to Scrum teams before.

In terms of the importance of whether someone appears to be extrinsically motivated (by rewards), Scrum team members also felt it could play a small/moderating part in how internally/naturally motivated they appear to be. Indeed, the literature does say that a person can be both intrinsically and extrinsically motivated at the same time (Frey, 1997).

Knowledge transfer depends a lot on how the recipient perceived the source, and theories that illustrate this well, are theories of reciprocity. Intention-based reciprocity theories, say that people punish others for unkind/insincere actions and reward them for kind/sincere actions (Stanca et al., 2009). It can be inferred, using a theory of intention-based reciprocity, that someone who is perceived to be more intrinsically/genuinely motivated will transfer more knowledge than a person perceived to be extrinsically motivated (Stanca et al., 2009). This is

because the recipient perceives such a person's intentions as more genuine/sincere and will reward them by being willing to listen to and accept their knowledge (Stanca et al., 2009; Trauth, 2010).

They seemed to be of the opinion that team members who appear to be intrinsically motivated are more passionate and would put more effort into transferring knowledge and assisting others, than someone who appears to be extrinsically motivated by something such as money or regulation (Frey and Jegen, 2001). Frey and Jegen (2001) empirically verified this effect (through a survey) which is called 'motivational crowding out'. The results of the survey found that extrinsic motivation such as money would make someone less intrinsically motivated or interested in the task itself, hence would work less at it (Burroughs, Dahl, Moreau, Chattopadhyay, and Gorn, 2011). This difference in passion therefore becomes obvious to recipients and they would be less inclined to learn/gain knowledge (Argote, 1999) from someone who is less motivated and less inspiring to learn from (Barnes, 2005).

In Agile methodologies (including Scrum), it has been said that projects should evolve around motivated teams as they should be trusted enough to complete their work unmanaged (Fowler and Highsmith, 2001; Lee and Xia, 2010). Motivated (especially intrinsically motivated) individuals will be driven to work hard and not mind transferring knowledge to others (Deci and Ryan, 2008). However, even though Scrum tries to improve the intrinsic motivation of its individuals through team cohesion practices like the daily Scrum meeting, where team members can feel like they are making a contribution to the entire team (McHugh, Conboy, and Lang, 2011), team members do not always end up motivated (Cottmeyer, 2008; Karlsen et al., 2011). This is because of issues such as team tension and frustration adjusting to the Scrum methodology (Cohn and Ford, 2003; Cottmeyer, 2008). Therefore, it is imperative for the Scrum master to deal with these issues, so that everyone will be motivated to transfer knowledge to each other and be inspired to gain knowledge from each other (Moe et al., 2010).

Also, according to literature, extrinsic motivation such as introducing rewards to encourage knowledge transfer could be used, as they have been known to be effective (Bock and Kim, 2001). Social rewards that improves team member's recognition amongst peers like badges have been known to be good for knowledge transfer (Schenk and Lungu, 2013), but financial incentives have been warned against due to their controlling nature (Cabrera and Cabrera, 2005).

Furthermore, recent studies have shown that financial incentives have little impact on knowledge transfer (Bock, Zmud, Kim, and Lee, 2005; He and Wei, 2009; Hung, Durcikova, Lai, and Lin, 2011). Indeed, the present study has found that Scrum team members also could not say whether the financial motivation of their peers would affect the knowledge they could transfer to them. However, Scrum team members found that extrinsic motivation had a moderating effect on the extent to which intrinsic motivation impacted the perceived extent of knowledge transfer. This is in line with Frey and Jegen's (2001) motivational crowding-out effect, which showed that extrinsic motivation, such as rewards, is given to a person, usually reduces the intrinsic motivation such as the enjoyment of a task for what it is.

Furthermore, empathy was found to be a characteristic that enabled knowledge transfer in Scrum teams. Empathy can be defined as seeing the world from someone else's perspective and sharing in their emotions (Duan and Hill, 1996). It was also found in the present research that senior Scrum team members who did not use jargon due to their empathy (Puck, Rygl, and Kittler, 2007) and instead used language that their peers could understand, were able to transfer more tacit knowledge (Ko et al., 2005). This is in line with Ko et al.'s (2005) ERP study which found that encoding/clarity of communication did increase knowledge transferred between clients and consultants.

Also, when Scrum team members felt another member was empathetic, they also seemed to find them to be intrinsically motivated to transfer knowledge as they always mentioned how willing they were to help them gain understanding and knowledge. Indeed, this finding is in line with a Van Lange (2008) study which found that an empathetic person would also be altruistically/intrinsically motivated to help others. Altruistic motivation (or altruism) is a form of intrinsic motivation and it depicts the need to want to help others without expecting anything in return (Hars and Ou, 2001).

In addition, empathy is also relevant to Scrum teams because there have been studies that say those who show empathy are also likely to be trusted (Ford, 2004). Since trust between team members is key for knowledge transfer in Agile teams (Karlsen et al., 2011), an empathetic individual is also likely to transfer more knowledge as they would appear more sincere to others, so what they have to say would actually be considered (Stanca et al., 2009).

In line with the findings of this study, an experiment with 84 participants in various hypothetical situations, empirically validated that empathy improves a person's intrinsic motivation to help others (Van Lange, 2008). If a team member appears to be empathetic they will also appear to be motivated, as shown in the motivational theory of charismatic leadership (Lin, 2007). This is beneficial in that Scrum team members are attracted to those who are genuinely motivated and would be willing to accept their knowledge (McHugh et al., 2011; Stanca et al., 2009). Indeed, motivated Scrum teams have been said to enable extensive knowledge transfer, as they are able to transfer knowledge in numerous circumstances such as the planning, review and daily Scrum meetings (Karlsen et al., 2011).

5.2 Perceived Capability and Knowledge Transfer

The perceived capability of a Scrum team member improves their perceived ability to transfer knowledge. If the recipient perceives the team member to be capable, they will go to them to gain knowledge, as was mentioned by Joshi et al. (2007).

In particular, this research found that the experience and formal qualification of a Scrum team member influence how capable people find them to be. Likewise, Ko et al. (2005) found that well-trained consultants are good at knowledge transfer. Indeed, it is recommended that Agile teams have some experienced personnel since Agile principles are quite imprecise (Boehm, 2002; Lindvall et al., 2002). These experienced personnel could transfer knowledge to junior team members through Agile Extreme Programming practices such as pair programming on a daily basis (Chau et al., 2003), which is often used in Scrum (Sutherland et al., 2007).

Knowledge transfer is also said to be high in Agile and Scrum teams of similar experience, since people assume they can trust others who are of similar experience and expertise (Cross, Laseter, Parker, and Velasquez, 2006; Levin and Cross, 2004). In contrast, in the present study people were of varied experience, with the least experienced team members often looking up to those of higher experience and hence viewing their knowledge as trustworthy (Szulanski, 1996). Furthermore, the present research found that people with experience in a particular area of software development like Agile test-driven development or experience in a particular managerial area, like Scrum project management, are believed to able to transfer knowledge in that area. Team members believed they could gain managerial knowledge from someone if

he/she was a Scrum master who had led by example. In addition, Scrum masters had the ability to influence others because of their extensive experience in Scrum, their ability to show how it works properly and their ability to make their team see the importance of Scrum and trust it more. Likewise, Szulanski (1996) posited that experts in an area are likely to be trusted and this was found to be the case for software development teams as well (Joshi et al., 2007).

The perceived qualification of a Scrum team member is also essential for the knowledge transferred to be seen as credible, as qualification usually means the person is well-trained and their knowledge would be viewed as reliable. This has also been demonstrated by Ko et al., (2005) and Szulanski (1996). Software development qualifications, like Java-certified developers, show that a person is well trained in Java, and that their knowledge is credible. Likewise, a Scrum master certification ensures that a person is indeed knowledgeable in the managerial aspects of Scrum and others would feel they can gain from that (Levin and Cross, 2012; Moe et al., 2010; Ortiz, 2003). This is in line with knowledge-based theory that says that qualification of a source influences knowledge transfer (Sveiby, 2001).

Similarly, a team member who is perceived to be capable and who is actually capable (technically or managerially) will transfer more knowledge as there is a greater depth to their knowledge (Joshi et al., 2007). However, similar to the knowledge engineering paradox which states that highly capable people sometimes struggle to transfer knowledge (Liebowitz and Chen, 2003), in this research it was reported that such people often withhold knowledge for fear of losing power/exclusivity. This was also reported by Szulanski (1996). Indeed, in Szulanski's (1996) model of impediments to knowledge transfer this was described as a lack of motivation of a source due to lack of incentives. This would also make these people look less motivated to their Scrum team, hence a less attractive source to gain knowledge from (Stanca et al., 2009).

Therefore, even though Agile teams are supposed to be highly capable (Boehm, 2002), incentives such as rewarding the best individual at transferring knowledge (as voted by peers) with a badge and a prize, should be used to help motivate them to transfer knowledge to each other (Antin and Churchill, 2011).

A Scrum team member who is not just capable but also motivated will transfer even more knowledge as they will be willing to transfer knowledge they otherwise would have wanted to

withhold for fear of losing power (Szulanski, 1996). Therefore, team members will also be attracted to their willingness (or perceived intrinsic motivation) to transfer knowledge and at the same time will be able to trust their knowledge if they find them highly capable (Joshi et al., 2007; Stanca et al., 2009). Indeed, Joshi et al (2007) did predict that there are probably moderating factors that make a person with good capabilities, better at transferring knowledge and this research has found motivation to be one of them.

Likewise, if a Scrum team member is perceived to be just motivated but not capable, their knowledge may be doubted by the recipient (Joshi et al., 2007). For example, people will more likely trust the knowledge of someone with over 10 years' experience than someone with less than one year's experience (Gupta and Govindarajan, 2000). Also, if a Scrum team member was Java-certified or Scrum Master-certified, this could also make people trust their knowledge more (Moe et al., 2010; Ortiz, 2003; Sveiby, 2001).

5.3 Perceived Credibility and Knowledge Transfer

Credibility is determined by both how reputable the recipient finds a team member, and how trustworthy they find their knowledge. The perceived credibility of a team members, did affect the knowledge they could transfer to others, which was also found in Joshi et al.'s (2007) study of knowledge transfer in information systems development teams (ISD).

The experience and qualifications of a team member were shown in this research to influence their perceived reputation to the eyes of others who would view their knowledge as credible. Indeed, in Szulanski's (1996) theoretical model of knowledge transfer barriers, he mentioned that if someone is seen as experienced and well-trained, recipients would not doubt the knowledge that they transfer. Experienced people would also be part of a Scrum team, in any case, as it would ensure rapid software delivery due to members past experience (Sutherland, Schoonheim, Rustenburg, and Rijk, 2008).

It has been said that trust is the 'magic' ingredient for knowledge transfer (Conboy and Fitzgerald, 2004). Indeed, trust was found in the present research to be one of the pillars for Agile teams working well together, in agreement with the literature (Fowler and Highsmith, 2001). Likewise, the literature says that Agile methods such as pair programming, which is often

used in Scrum (Sutherland et al., 2007), promotes mutual trust between team members and enables knowledge transfer between them (Chau et al., 2003). Indeed, Scrum team members in the present research did seem naturally to trust each other until proven otherwise, and if analysed using social theory, this was maybe because of the professional setting in which they worked in. Hence, they did not have to doubt each other's credibility (Meyer, Ward, Coveney, and Rogers, 2008).

Also, from the present study, it appears that some Scrum team members trusted themselves at a personal level as well, since they even relaxed with each other after work. Indeed, Gillespsie (2003) also found that there were two different types of trustworthiness, based on personal and professional trust. Personal trust is a trust that allows people to disclose sensitive issues to each other (Gillespie, 2003), whilst professional trust is the usual trust in other people's skills, knowledge and professional judgement (Gillespie, 2003; Zand, 1972).

Trust, according to social exchange theory (Cropanzano and Mitchell, 2005), is an important component for strong relationships to occur (Emerson, 1976). In particular, personal trust allows for easy communication and knowledge transfer. Indeed, the Scrum methodology acknowledges the fact that knowledge transfer occurs through strong personal ties and frequent interaction, and so facilitates this by allowing regular opportunities for tacit knowledge transfer like the daily Scrum meetings (Karlsen et al., 2011; Melnik and Maurer, 2004). Also, both personal and professional trust have been shown to improve knowledge transfer in a previous survey study of 135 knowledge recipients at a workplace, in agreement with findings from the present study (Alexopoulos and Buckley, 2013).

The perceived reputation of a Scrum team member also seemed to influence trust, whereby people who were reputable were regarded as knowledgeable experts. Indeed, if team members find someone's knowledge to be reputable they will trust their knowledge (Joshi et al., 2007). This is in line with a study of 46 small and medium enterprises (SMEs) using Agile development, which found that the credibility of a team member strongly influences the tacit knowledge they transferred to others (Ryan and O'Connor, 2013). Ultimately, what can be deduced from the findings is that credibility is key to knowledge transfer in Scrum teams. Without credibility, knowledge transfer cannot happen as the recipient will doubt/not trust the knowledge being transferred (Szulanski, 1996).

5.4 Communication and Knowledge Transfer

In Joshi et al. (2007) related model of knowledge transfer in information systems development (ISD) teams, it was found that the communication extent of the source improves knowledge transfer. However, the findings of the present research reveal that it is not only the communication extent that improves knowledge transfer but the quality of communication/articulation is also important.

The need to be articulate was said to be essential by most of the Scrum team members in this research. Indeed, a big complaint reported in this research from developers about senior developers, was that they would use jargon terms, assuming others already knew them (Cohn, 2004). In the literature, this is known as the communication encoding competence, which is one's ability to express themselves clearly and be understood (Monge, Bachman, Dillard, and Eisenberg, 1982). A knowledge transfer study between ERP clients and consultants also found support for the fact that good communication skills increases the knowledge transferred (Ko et al., 2005). Likewise, Agile software development teams in the past that tried to improve their clarity of communication by speaking more slowly and intentionally, seemed to experience improved knowledge transfer (Dorairaj, Noble, and Malik, 2011). Indeed, Scrum also has its own jargon like user stories and sprints (Fernández-Vara and Tan, 2008; Mikael Lindvall et al., 2004), so any team member should make sure that the recipient is also aware of these terms in their communication with them (Cohn, 2004).

Nonetheless, regular face-to-face communication by a Scrum team member was still found to be essential to knowledge transfer, as regular communication opens up opportunities for tacit knowledge transfer and resolving issues quickly (Karlsen et al., 2011; Ko et al., 2005). Regular communication is key to Agile software development, as one of the fundamental principles in the Agile manifesto is "individuals and interactions over processes and tools" (Fowler and Highsmith, 2001, p.29). This is why Agile development also emphasises having co-located teams as this promotes regular face-to-face communication and tacit knowledge transfer (Chau et al., 2003; Eccles, Smith, Tanner, Van Belle, and Van der Watt, 2010). Therefore, in Scrum software development, team members have the opportunity to transfer knowledge regularly through daily

Scrum meetings (Karlsen et al., 2011) and in pair programming, which is an Agile Extreme Programming practice often used in Scrum (Sutherland et al., 2007).

However, this present research also found that too frequent communication in Scrum teams also irritated the recipient (Moe and Šmite, 2007), therefore Scrum team members should communicate sufficiently and not excessively in order to transfer knowledge (Svensson and Höst, 2005). Indeed, this can be linked back to Szulanski's (1996) arduous/poor relationship concept, in that it is important that team members maintain a good relationship and do not irritate each other. For example, in Scrum teams, it is very easy for product owner/anyone else to speak to the software developers due to the co-located environment, so they need to limit their communication in order to not disturb them from their highly focussd work and end up irritating them (Cho, 2008).

Overall, it can be seen that, it is not just extent of communication that improves knowledge transfer within Scrum teams but also how articulate the source is. Indeed, Szulanski's (1996) model in knowledge management, ease of communication was spoken of, in an arduous relationship, and could be measured by more constructs in communication, such as how regular the communication is and how skilled/articulate the source at communicating is and whether the simplest communication medium was used – which in Agile should be face to face, as it is tacit knowledge that is shared mostly (Chau et al., 2003).

5.5 Summary

This section will now summarise the results and discussion in

Table 5-1, which compares and contrasts the findings of this study with the relevant literature. Overall, these findings agreed with the literature in general, but there was some literature which did not occur in a Scrum or Agile context, such as the impact of empathy on knowledge transfer. To follow, is the table comparing the findings with the literature.

Table 5-1: Comparison of findings with literature

Study Findings	**Literature Findings**
Motivation of the source	
Scrum team members are inclined to gain knowledge from those they find intrinsically motivated.	Slightly similar findings in a study using intention-based theory of reciprocity, verified that people will reward actions they perceive to be kind/sincere but punish unkind/insincere actions (Stanca et al., 2009)..
Scrum team members perceive those who seem to have weaker/non-genuine motivation to transfer knowledge, as extrinsically motivated	Slightly similar findings by Frey and Jegen (2001) who verified the motivational crowding out effect which says that an extrinsic reward such as money makes a person less intrinsically/naturally motivated in the task itself.
Scrum team members seemed to find intrinsic motivation of others to have more of an impact than extrinsic motivation.	Some past studies also identified that extrinsic motivation does not have much of an impact on knowledge transfer (Bock, Zmud, Kim, and Lee, 2005; He and Wei, 2009; Hung, Durcikova, Lai, and Lin, 2011)
Signs of intrinsic motivation in Scrum teams include enjoyment of working in Scrum roles and enjoyment of learning about Scrum	Very similar but general signs of intrinsic motivation, such as enjoyment of job and enjoyment of learning were used to determine intrinsic motivation in a survey on Scrum teams (Nikitina, Kajko-Mattsson, and Strale, 2012). However, they did not relate intrinsic motivation in Scrum teams to knowledge transfer within these teams.

Study Findings	**Literature Findings**
Signs of extrinsic motivation in Scrum team members include financial motivation and peer recognition.	Financial motivation and peer recognition are typical characteristics known to impact on knowledge transfer in teams (Ko et al., 2005), however the impact of financial incentives in Scrum teams has rarely been investigated.
Empathy	
Empathetic Scrum team members were said to be able to transfer knowledge well, as they were concerned with the recipient understanding their message.	Somewhat similar findings by Van Lange's (2008) study found that an empathetic person will be naturally/intrinsically motivated to help but the literature surveyed did not speak about empathy's impact on knowledge transfer in teams.
Empathetic Scrum team members were said to be more willing to transfer knowledge as well.	Empathy of the source has been shown to beneficial to knowledge transfer (Zarraga and Bonache, 2005) but from the literature surveyed, not within Scrum teams.
Perceived Capability of the source	
Scrum team members considered those with high experience to be capable and therefore able to transfer knowledge	Experience, has been shown to influence knowledge transfer in Agile and Scrum teams (Cross, Laseter, Parker, and Velasquez, 2006; Levin and Cross, 2004).
Scrum team members considered those who are highly qualified (like Scrum masters) to be capable and therefore able to transfer knowledge.	Qualifications of team members has been known to influence knowledge transfer, in general and in Scrum teams (Levin and Cross, 2012; Moe et al., 2010; Ortiz, 2003).
Scrum team members who are perceived to be capable but not motivated, struggle to transfer knowledge to others	Joshi et al. (2007) predicted that there are probably moderating factors to a source's capability in knowledge transfer in cross-functional information systems development

	teams, such as Scrum teams.
Study Findings	**Literature Findings**
Scrum team members consider technical capabilities to be solely software development expertise.	Slightly similar though technical knowledge is usually a lot broader in information systems development teams to include knowledge about applications or tools, in general (Joshi et al., 2007).
Scrum team members consider managerial capabilities to be Scrum project management and Scrum leadership abilities.	Managerial capabilities usually do relate to management and leadership/relationship capabilities (Joshi et al., 2007).
Perceived Credibility of the source	
Scrum team members who were perceived to be capable, were also found to be trustworthy by others, and could transfer knowledge within the team.	Perceived capability and expertise has been shown to be important in knowledge transfer in Agile and Scrum teams (Cross, Laseter, Parker, and Velasquez, 2006; Levin and Cross, 2004).
Scrum team members felt that it is not the usual professional trust in other's skills but also personal trust (like that of a friend) that influences knowledge transfer in Scrum teams.	Agile and Scrum teams are said to work and collaborate very well when they have strong personal ties (Karlsen et al., 2011; Melnik and Maurer, 2004).
Scrum team members whom others found to be reputable (maybe by being highly experienced), were also seen as trustworthy.	Indeed, Agile and Scrum team members of similar expertise, also trust each other as they perceive each other as reputable (Cross, Laseter, Parker, and Velasquez, 2006; Levin and Cross, 2004).
Scrum team members also regarded those who were highly experienced as credible and would hence their knowledge as trustworthy.	Indeed, it has been recommended that Agile teams have some highly experienced team members, so they can transfer knowledge to those of lower experience (Chau et al., 2003),

Study Findings	**Literature Findings**
Scrum team members also regarded those who were highly qualified as reputable and hence would trust their knowledge	Qualifications of team members have been known to influence knowledge transfer, in general and in Scrum teams (Levin and Cross, 2012; Moe et al., 2010; Ortiz, 2003).
Perceived Communication of the source	
It is not just communication extent but also communication quality/articulation of the source that improves knowledge transfer in Scrum teams.	There is much literature focusing on communication extent in Scrum and Agile teams (Chau et al., 2003; Melnik and Maurer, 2004), but from the literature surveyed, there is little on the effect of quality communication/articulation of the source on knowledge transfer in Scrum teams.
Scrum team members who were perceived to be articulate as they spoke clearly and without jargon, could transfer a great deal of knowledge.	Indeed, communication encoding competence (articulation of the source) has been shown to be important to knowledge transfer in teams (Ko et al., 2005).
Scrum team members felt excessive communication within teams, sometimes irritates the recipient. Therefore, they felt that a person/the source should communicate just enough to transfer knowledge.	Slightly different to popular Scrum and Agile literature which promotes regular and frequent communication to increase knowledge transfer (Chau et al., 2003; Melnik and Maurer, 2004). However, there have also been some findings that excessive communication leads to team member distraction and lower productivity (Leenders et al., 2003; Oertig and Buergi, 2006).

6 Conclusion

This chapter concludes the research, by re-stating the research questions and what was achieved in each case, as well stating the research contributions of these findings. The limitations and relevant areas for future research are then stated.

6.1 Research Questions

To re-cap, the research questions as stated after the literature review were:

Main Research Question:

- What are the perceived characteristics that a Scrum team member should have to successfully transfer tacit knowledge throughout the project?

Sub-Research Questions:

- How does the perceived motivation of a team member affect the perceived extent of tacit knowledge transferred by them?
- How does the perceived capability of a team member affect the perceived extent of tacit knowledge transferred by them?
- How does perceived credibility of a team member affect the perceived extent of tacit knowledge transferred by them?
- How does the perceived communication extent of a team member affect the perceived extent of tacit knowledge transferred by them?
- What other characteristics are associated with team members that can successfully transfer tacit knowledge in Scrum projects?

The research did not only address how each of the four perceived characteristics of a team member influenced knowledge transfer in Scrum teams, but also identified new characteristics and relationships between characteristics from the data collection.

6.2 Summary of Findings

The study has found that the perceived motivation of a team member has an impact on the perceived extent of knowledge that the latter can transfer in a Scrum team. Scrum team members felt that they were naturally attracted to intrinsically motivated people and wanted to gain knowledge from them. This was because they knew that someone who was intrinsically motivated was more likely to want to transfer knowledge to them, than someone who appeared to be extrinsically motivated. Indeed, this is why Agile recommends having not just motivated but also self-driven individuals, as such individuals, who will go the extra mile in whatever they do, hence even in terms of knowledge transfer.

Empathy, which is seeing the world from another person's perspective, was also a new characteristic seen by participants which makes team members concerned about transferring knowledge in the best way, such as avoiding use of jargon. Also, empathetic Scrum team members were perceived to be willing/motivated to transfer and therefore others would be attracted to gain knowledge from them.

The perceived capability of a team member improves knowledge transfer, but usually if the team member is perceived to be motivated as well. Indeed, participants acknowledged that if someone looks like they are highly capable, because of a high qualification, experience or past evidence, they would like to learn from them. For example, if a Scrum team member is a certified Java software developer or a certified Scrum master, others will believe that they will be able to gain technical or managerial knowledge from them. However, if the person looks like they are unmotivated or has been unwilling to transfer knowledge, they would not be interested in going back to gain knowledge from them.

The perceived credibility, which is based on the trustworthiness and reputation people give to a person, is essential to knowledge transfer. The reputation of an individual, determined by their perceived experience and reputation, added to their credibility and perceived extent of knowledge that they could transfer. Team members found the knowledge of those who had a lot of experience, maybe over 10 years in software development or working in Agile methodologies, highly valuable. Indeed, this is why Agile recommends having teams of high expertise, so that team mates will not doubt the knowledge that is being transferred. Also, Agile teams who share

similar expertise and experience, have also been known to thrive, since team mates feel they can trust the knowledge of others with a similar amount of experience.

Trust was described by some as a pillar for Agile teams working together successfully. Scrum team members seemed to trust each other at a professional and personal level, and both of these seemed to impact the perceived extent of knowledge transferred. They felt that knowledge transfer could not work without trust. Agile promotes mutual trust through methods such as pair programming in order to enable natural knowledge transfer. The participants seemed to trust each other naturally as they often had no reason to doubt each other, maybe due to the professional setting. Therefore, credibility is key to knowledge transfer, and the team member should not give others a reason to doubt.

In terms of communication, it was found that a person's ability to communicate extensively improves knowledge transfer: a Scrum team member should communicate just enough so as not to irritate the recipient. Furthermore, it is not just the communication extent but also the quality of communication (or rather articulation of the source) that improves knowledge transfer in Scrum software development teams. Indeed, team members expressed the need for communication to be clear and concise, in order to easily understand what is being said. Likewise, Agile software development teams that tried to improve the clarity by speaking slowly, for example, seemed to initiate better knowledge transfer. Since Agile methodologies believe face to face communication to be the most effective form of communication, it must also be done effectively.

Fortunately, Scrum itself makes knowledge transfer easy to occur, according to participants and literature, due to the regular Scrum daily meetings and various Scrum ceremonies like the Sprint Planning and Sprint Review meetings. A lot of managerial knowledge, such as how to estimate user stories, would be shared during these ceremonies, as each team member was given a chance to run some of these meetings, such as the Sprint planning meeting. Also, the pair programming environment in both of these companies made it easy for developers to communicate with each other and the business analyst, all the time, making knowledge transfer much easier.

6.3 Research Contribution

This study contributes to Scrum and Agile research by producing a theoretical model extended from Joshi et al. (2007), describing the characteristics of the source needed for successful knowledge transfer in Scrum. Therefore, it also contributes a theoretical model to knowledge management in general, which is a field said to be lacking in theoretical underpinnings (Wang and Noe, 2010). There has also been said to be a lack of empirically based knowledge transfer studies in Agile teams (Karlsen et al., 2011), so this study has made a methodological contribution to Agile research in this regard as it used semi-structured interviews with a systematic qualitative analysis: the thematic analysis technique.

Mostly, this research contributes to the field of knowledge management as it brought up new moderating variables between characteristics already well known to influence knowledge transfer and new characteristics within Scrum teams, such as intrinsic and extrinsic motivation as well as empathy.

Indeed, this research extends the model of Joshi et al. (2007) by adding the construct, 'motivation', as suggested by the authors due to its importance to knowledge transfer. Furthermore, this study investigates both intrinsic and extrinsic motivation not just in Scrum teams (Nikitina, Kajko-Mattsson, and Strale, 2012) but in relation to knowledge transfer, which may be new to the literature. Indeed, its findings also add to the growing trend of results that show intrinsic motivation to influence knowledge transfer more than extrinsic motivation. Therefore, by looking at ways of improving intrinsic motivation of team members such as making the process of transferring knowledge more fun and enjoyable (Deterding, Dixon, Khaled, and Lennart, 2011), knowledge transfer could improve greatly.

Empathy has been found to be beneficial to knowledge transfer in the past studies (Zarraga and Bonache, 2005), but from the literature surveyed, not within Scrum teams. However, this research has shown that empathy of team members can improve knowledge transfer, so this could be looked into further.

Furthermore, both professional and personal trustworthiness of the source emerged from the data in this study as characteristics that impact on knowledge transfer. These two types of

trustworthiness have been shown to affect knowledge transfer in the past, but few theoretical models in knowledge management include these types of trustworthiness.

Also, the present research found that it is not just communication extent that improves knowledge transfer, which is why Agile and Scrum teams focus on this so much, but also the quality of communication/articulation of the source that improves knowledge transfer in Scrum teams. Therefore, this research has shown that Agile and Scrum studies should look into this more comprehensively.

The relationships between other constructs, such as the findings that the qualification and experience impact on perceived capability (and hence knowledge transfer) of a Scrum team member, can be investigated more deeply. For example, this research has shown there is now a need to find out what the best mixture of experience would be needed in a Scrum team to allow for optimal knowledge transfer.

This research also found motivation to be a moderating variable to the perceived capability of the source. As Joshi et al. (2007) found, the capability of a team member alone does not improve knowledge transfer, thus there must be a moderating variable. Indeed, if the source/person who is wanted to transfer knowledge is not motivated, others would not want to be around them and gain knowledge from them. Thus this relationship can also be investigated further.

6.4 Implication for Practice

This research produces a list of factors that Scrum managers can use to identify people who can initiate good knowledge transfer. For example, when it comes to hiring, the management/Scrum masters should ensure that candidates have a good reputation and qualifications, as this influences how credible the team members view them. It will also contribute to the human resources (HR) team, to help provide them with a list of factors that they can focus on when training people. This thesis will be emailed to the participating companies, so they can immediately adopt these factors that should allow for better knowledge transfer in their Scrum teams.

An understanding of knowledge transfer, is key to Scrum teams, since they follow an Agile methodology whose success is largely underpinned by regular face-to-face communication to

transfer tacit knowledge (Chau et al., 2003) . However, knowledge transfer is not always said to be successful in Scrum teams due to reasons such as team tension and resistance to Scrum adoption. Therefore, it is also the Scrum master's responsibility to resolve these issues (Moe et al., 2010), even if a team possesses good knowledge transfer attributes.

In one of the companies that were interviewed, it was reported that employees undergo a course in communication so that they can always be able to express themselves clearly. Companies could also go further and build a trusting culture within Scrum teams (Joshi et al., 2007), as trust has been shown to be crucial to knowledge transfer. This could be done through team-building exercises such as board-game evenings and sports days (Gordon and Esbjörn-Hargens, 2007).

Also, since motivation is clearly important for knowledge transfer, and not all members in Scrum teams are typically motivated, measures could be taken to incentivise knowledge transfer. A good example would be gamification, where game-elements are introduced into non-game activities (Deterding et al., 2011). This could be having a leaderboard showing those who were rated the best by their peers at transferring knowledge and reward the best person every week with a badge and a prize (Antin and Churchill, 2011).

Finally, since there was low stakeholder satisfaction in Scrum companies in South Africa (Ferreira and Cohen, 2008), which is the location of this study, this research is a step forward in providing evidence to improve knowledge transfer in Scrum teams which should therefore improve project success and stakeholder satisfaction in these companies (Lyytinen and Robey, 1999).

6.5 Limitations of study

The first limitation of this study would be that though it has looked at what characteristics make a person good at transferring knowledge in Scrum teams, future research should also look at characteristics needed by the recipient. The duration of the study could also be extended, to include interviewing recipients over time or even the researcher/s working with the participants to see if what is reported in interviews, happens at the workplace as well. The setting of the study is a limitation as it was only done in one country, and it only looked at two companies in-depth.

6.6 Future Work

Future research may extend this study, by looking in-depth into both types of knowledge transfer, explicit and tacit knowledge transfer within Scrum development teams. Knowledge transfer can also be looked at from the source's perspective. Future studies could also extend the sample size, number of interviews and/or study length, to obtain even richer data. Also, future research could look more deeply into the new findings, such as how empathy affects motivation of a source to transfer knowledge in software teams and also how motivation and capability affect knowledge transfer. Lastly, this model would also need to be empirically validated quantitatively so that it could be used in survey research in the future or so that relationships can be proved to be true.

7 References

Abdel-Moteleb, A.-B., & Woodman, M. (2007). Towards a Knowledge Management System Development Method: Critique of Some Relevant Theories and Methodologies. *Journal of Information & Knowledge Management*, *6*(01), 33–43.

Ajzen, I. (1991). The theory of planned behavior. *Organizational behavior and human decision processes*, *50*(2), 179–211.

Ajzen, I., & Fishbein, M. (1980). *Understanding Attitudes and Predicting Social Behavior*. Englewood Cliffs, NJ,: Prentice-Hall, Inc.

Akgün, A. E., Keskin, H., Byrne, J. C., & Gunsel, A. (2011). Antecedents and Results of Emotional Capability in Software Development Project Teams. *Journal of Product Innovation Management*, *28*(6), 957–973.

Alavi, M., Kayworth, T. R., & Leidner, D. E. (2006). An empirical examination of the influence of organizational culture on knowledge management practices. *Journal of management information systems*, *22*(3), 191–224.

Albino, V., Garavelli, A. C., & Schiuma, G. (1998). Knowledge transfer and inter-firm relationships in industrial districts: the role of the leader firm. *Technovation*, *19*(1), 53–63.

Alexopoulos, A. N., & Buckley, F. (2013). What Trust Matters When: The Temporal Value of Professional and Personal Trust for Effective Knowledge Transfer. *Group & Organization Management*, *38*(3), 1–31.

Amaratunga, D., Baldry, D., Sarshar, M., & Newton, R. (2002). Quantitative and qualitative research in the built environment: application of "mixed" research approach. *Work study*, *51*(1), 17–31.

Anfara, V. A., Brown, K. M., & Mangione, T. L. (2002). Qualitative analysis on stage: Making the research process more public. *Educational researcher*, *31*(7), 28–38.

Antin, J., & Churchill, E. (2011). Badges in Social Media : A Social Psychological Perspective. *Human Factors, Human Fact*, 1–4. Retrieved from http://research.yahoo.com/node/3469

Argote, L. (1999). *Organizational learning: Creating, retaining and transferring knowledge*. Kluwer Academic: Boston, MA.

Argote, L., & Ingram, P. (2000). Knowledge Transfer: A Basis for Competitive Advantage in Firms. *Organizational Behavior and Human Decision Processes*, *82*(1), 150–169. doi:http://dx.doi.org/10.1006/obhd.2000.2893

Argote, L., Ingram, P., Levine, J. M., & Moreland, R. L. (2000). Knowledge transfer in organizations: Learning from the experience of others. *Organizational Behavior and Human Decision Processes*, *82*(1), 1–8.

Awad, M. A. (2005). A comparison between agile and traditional software development methodologies. The University of Western Australia, Perth: School of Computer Science and Software Engineering.

Bahli, B., & Zeid, E. S. A. (2005). The role of knowledge creation in adopting extreme programming model: an empirical study. *Enabling Technologies for the New Knowledge Society: Iti 3rd International Conference on Information & Communications Technology* (pp. 75–87). Cairo, Egypt.

Bailey, J. (2008). First steps in qualitative data analysis: transcribing. *Family practice*, *25*(2), 127–131.

Barnes, A. (2005). A passion for languages: Motivation and preparation to teach modern foreign languages in eight cohorts of beginning teachers. *Research papers in education*, *20*(4), 349–369.

Barney, H. T., Moe, N. B., Dybå, T., Aurum, A., & Winata, M. (2009). Balancing individual and collaborative work in agile teams. *Agile Processes in Software Engineering and Extreme Programming* (pp. 53–62). Pula, Sardinia, Italy: Springer.

Barriball, K., & While, A. (1994). Collecting Data using a semi-structured interview: a discussion paper. *Journal of advanced nursing*, *19*(2), 328–335.

Basri, S., & O'Connor, R. V. (2010). Understanding the perception of very small software companies towards the adoption of process standards. *17th European Software Process Improvement (EuroSPI) Conference, 1-3 September*. Grenoble, France.

Bassellier, G., Reich, B. H., & Benbasat, I. (2003). Information technology competence of business managers: A definition and research model. *Journal of Management Information Systems*, *17*(4), 159–182.

Baxter, P., & Jack, S. (2008). Qualitative Case Study Methodology : Study Design and Implementation for Novice Researchers. *The Qualitative Report*, *13*(4), 544–559.

Beck, K. (1999). *Extreme programming explained: embrace change* (1st Ed.). Addison-Wesley Professional.

Beck, K. (2002). *Test Driven Development -- by Example*. Reading, MA.

Benedicenti, L., & Paranjape, R. (2001). Using Extreme Programming for Knowledge Transfer. *Proceedings of XP2001 Conference, May 23-30*. Cagliari, Villasimius, Sardinia.

Bereby-Meyer, Y., Moran, S., & Unger-Aviram, E. (2004). When performance goals deter performance: Transfer of skills in integrative negotiations. *Organizational Behavior and Human Decision Processes*, *93*(2), 142–154.

Berg, B. L. (1995). *Qualitative Research Methods for the Social Sciences.* Boston: Bacon & Allyn.

Berlo, D. K. (1960). The process of communication: An introduction to theory and practice. New York: Holt, Rinehart and Winston, 1960.

Bjørnson, F. O., & Dingsøyr, T. (2008a). Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Information and Software Technology*, *50*(11), 1055–1068.

Bjørnson, F. O., & Dingsøyr, T. (2008b). Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Information and Software Technology*, *50*(11), 1055–1068.

Bock, G. W., & Kim, Y.-G. (2002). Breaking the myths of rewards: an exploratory study of attitudes about knowledge sharing. *Information Resources Management Journal (IRMJ)*, *15*(2), 14–21.

Bock, G.-W., & Kim, Y.-G. (2001). Breaking the myths of rewards: An exploratory study of attitudes about knowledge sharing. *Pacis 2001 proceedings, June 20-22, 2001*. (p. 78). Seoul, Korea.

Bock, G.-W., Zmud, R. W., Kim, Y.-G., & Lee, J.-N. (2005). Behavioral intention formation in knowledge sharing: Examining the roles of extrinsic motivators, social-psychological forces, and organizational climate. *MIS quarterly*, *29*(1), 87–111.

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

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *Software, IEEE*, *22*(5), 30–39.

Boland, D., & Fitzgerald, B. (2004). Transitioning from a co-located to a globally-distributed software development team: A case study at Analog Devices Inc. *Proceedings of the International Conference on Software Engineering (ICSE 2004)*. Edinburgh, Scotland.

Bonoma, T. V. (1985). Case research in marketing: opportunities, problems, and a process. *Journal of marketing research*, *22*(2), 199–208.

Booyens, I. (2012). Creative industries, inequality and social development: developments, impacts and challenges in Cape Town. *Urban Forum* (Vol. 23, pp. 43–60).

Boström, G., Wäyrynen, J., Bodén, M., Beznosov, K., & Kruchten, P. (2006). Extending XP practices to support security requirements engineering. *Proceedings of the 2006 international workshop on Software engineering for secure systems, 20-28 May*. Shanghai, China.

Boyatzis, R. E. (1998). *Thematic Analysis and Code Development: Transforming Qualitative Information*. Thousand Oaks, CA: Sage Publications.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, *3*(2), 77–101.

Burns, T., & Klashner, R. (2005). A cross-collegiate analysis of software development course content. *Proceedings of the 6th conference on Information technology education* (pp. 333–337). Newark, NJ, USA.

Burroughs, J. E., Dahl, D. W., Moreau, C. P., Chattopadhyay, A., & Gorn, G. J. (2011). Facilitating and rewarding creativity during new product development. *Journal of Marketing*, *75*(4), 53–67.

Cabrera, E. F., & Cabrera, A. (2005). Fostering knowledge sharing through people management practices. *The International Journal of Human Resource Management*, *16*(5), 720–735.

Cardozo, E. S. F., Neto, J. B. F. A., Barza, A., França, A. C. C., & Da Silva, F. Q. B. (2010). SCRUM and productivity in software projects: a systematic literature review. *Proceedings of the 14th international conference on Evaluation and Assessment in Software Engineering* (pp. 131–134). Swinton, UK.

Chandrasekaran, B., Josephson, J. R., & Benjamins, V. R. (1999). What are ontologies, and why do we need them? *Intelligent Systems and Their Applications, IEEE*, *14*(1), 20–26.

Chau, T., & Maurer, F. (2004). Knowledge sharing in agile software teams. *Lecture Notes in Computer Science #3075: Logic Versus Approximation* (pp. 173–183). New York, NY, USA: Springer.

Chau, T., Maurer, F., & Melnik, G. (2003). Knowledge sharing: Agile methods vs. tayloristic methods. *Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises(WETICE 03), 9-11 June*. Linz, Austria.

Chennamaneni, A. (2006). *Determinants Of Knowledge Sharing Behaviors: Developing And Testing An Integrated Theoretical Model*. The University of Texas at Arlington.

Cho, J. (2008). Issues and Challenges of Agile Software Development with Scrum. *Issues in Information Systems*, *9*(2), 188–195.

Christensen, K. S., & Bang, H. K. (2003). Knowledge management in a project-oriented organization: three perspectives. *Journal of Knowledge Management*, *7*(3), 116–128.

Cockburn, A. (2001). *Agile Software Development*. Boston: Addison-Wesley.

Cockburn, A. (2004). *Crystal clear: a human-powered methodology for small teams*. Boston, MA: Addison-Wesley Professional.

Cockburn, A. (2007). *Agile Software Development: The Cooperative Game*. Boston: Addison-Wesley.

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, *34*(11), 131–133.

Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley Professional.

Cohn, M., & Ford, D. (2003). Introducing an agile process to an organization. *Computer*, *36*(6), 74–78.

Conboy, K., & Fitzgerald, B. (2004). Toward a Conceptual Framework of Agile Methods. *Proceedings of the 2004 ACM Workshop on Interdisciplinary Software Engineering Research, 5 November*. Newport Beach, CA.

Cooke-Davies, T. (2002). The "real" success factors on projects. *International journal of project management*, *20*(3), 185–190.

Cottmeyer, M. (2008). The good and bad of Agile offshore development. *Proc. of the AGILE 2008 Conference, 4-8 August* (pp. 362–367). Toronto,Canada.

Cress, U., & Martin, S. (2006). Knowledge sharing and rewards: a game-theoretical perspective. *Knowledge Management Research & Practice*, *4*(4), 283–292. Retrieved from http://www.palgrave-journals.com/doifinder/10.1057/palgrave.kmrp.8500112

Creswell, J. W., & Miller, D. L. (2000). Determining validity in qualitative inquiry. *Theory into practice*, *39*(3), 124–130.

Cropanzano, R., & Mitchell, M. S. (2005). Social exchange theory: An interdisciplinary review. *Journal of Management*, *31*(6), 874–900.

Cross, R., Laseter, T., Parker, A., & Velasquez, G. (2006). Using social network analysis to improve communities of practice. *California Management Review*, *49*(1), 32.

Cunningham, H. (1999). A definition and short history of Language Engineering. *Natural Language Engineering*, *5*(01), 1–16.

Cunningham, J. W., & Fitzgerald, J. (1996). Epistemology and reading. *Reading Research Quarterly*, *31*(1), 36–60.

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, *31*(11), 1268–1287.

Da Silva, F. Q. B., Costa, C., França, A. C. C., & Prikladinicki, R. (2010). Challenges and solutions in distributed software development project management: A systematic literature review. *Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on* (pp. 87–96). 23-26 August, Princeton, New Jersey.

Davenport, T. H., David, W., & Beers, M. C. (1998). Successful knowledge management projects. *Sloan management review*, *39*(2), 43–57.

Davenport, T. H., & Pruzak, L. (2000). *Working knowledge: How organizations manage what they know*. Harvard Business Press.

Deci, E. L., & Ryan, R. M. (2008). Self-determination theory: A macrotheory of human motivation, development, and health. *Canadian Psychology/Psychologie canadienne*, *49*(3), 182.

Deterding, S., Dixon, D., Khaled, R., & Lennart, N. (2011). From game design elements to gamefulness: defining gamification. *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments.* (pp. 9–15). Tampere, Finland.

Dingsøyr, T. (2002). *Knowledge Management in Medium- Sized Software Consulting Companies : An Investigation of Intranet-based Knowledge Management Tools for Knowledge Cartography and Knowledge Repositories for Learning Software Organisations*. Norwegian University of Science and Technology.

Doherty-Sneddon, G., Anderson, A., O'Malley, C., Langton, S., Garrod, S., & Bruce, V. (1997). Face-to-face and video-mediated communication: A comparison of dialogue structure and task performance. *Journal of Experimental Psychology: Applied*, *3*(2), 105.

Dominguez, J. (2009, July). The curious case of the CHAOS report, The Standish Group Report 2009.

Dorairaj, S., Noble, J., & Malik, P. (2011). Effective communication in distributed Agile software development teams. *Agile Processes in Software Engineering and Extreme Programming - 12th International Conference, XP, 10-13 May,2013*. Madrid, Spain: Springer.

Dresser, R. (2001). Payments to research participants: The importance of context. *American Journal of Bioethics*, *1*(2), 47.

Droege, S. B., & Hoobler, J. M. (2003). Employee turnover and tacit knowledge diffusion: a network perspective. *Journal of Managerial Issues*, *XV*(1), 50–64.

Drucker, P. (1988). The coming of the new organization. *Harvard Business Review*, 44–53.

Duan, C., & Hill, C. E. (1996). The current state of empathy research. *Journal of counseling psychology*, *43*(3), 261.

Dyba, T., & Dingsoyr, T. (2009). What do we know about agile software development? *Software, IEEE*, *26*(5), 6–9.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, *50*(9), 833–859.

Eccles, M., Smith, J., Tanner, M., Van Belle, J., & Van der Watt, S. (2010). The impact of collocation on the effectiveness of agile is development teams. *Communications of the IBIMA*, 1–11.

Ellis, C. (2007). Telling secrets, revealing lives relational ethics in research with intimate others. *Qualitative Inquiry*, *13*(1), 3–29.

Elwyn, G., Taubert, M., Kowalczuk, J., & others. (2007). Sticky knowledge: A possible model for investigating implementation in healthcare contexts. *Implementation Science*, *2*(1), 44.

Emerson, R. M. (1976). Social exchange theory. *Annual review of sociology*, 335–362.

Faraj, S., & Sproull, L. (2000). Coordinating expertise in software development teams. *Management science*, *46*(12), 1554–1568.

Fereday, J., & Muir-Cochrane, E. (2008). Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International journal of qualitative methods*, *5*(1), 80–92.

Fernández-Vara, C., & Tan, P. (2008). The game studies practicum: applying situated learning to teach professional practices. *Proceedings of the 2008 Conference on Future Play: Research, Play, Share . November 03 - 05* (pp. 25–32). Toronto,Canada.

Ferreira, C., & Cohen, J. (2008). Agile Systems Development and Stakeholder Satisfaction : A South African Empirical Study. *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology. 6-8 October* (pp. 48–55). Wilderness, South Africa.

Finlay, L. (2002). "Outing" the researcher: The provenance, process, and practice of reflexivity. *Qualitative Health Research*, *12*(4), 531–545.

Flyvbjerg, B. (2006). Five Misunderstandings About Case-Study Research. *Qualitative Inquiry*, *12*(2), 219–245.

Ford, D. P. (2004). Trust and knowledge management: the seeds of success. *Handbook on Knowledge Management 1* (pp. 553–575). Berlin, Germany: Springer.

Fossey, E., Harvey, C., McDermott, F., & Davidson, L. (2002). Understanding and evaluating qualitative research. *Australian and New Zealand journal of psychiatry*, *36*(6), 717–732.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development*, *9*(8), 28–35.

Frey, B. S., & Jegen, R. (2001). Motivation crowding theory. *Journal of economic surveys*, *15*(5), 589–611.

Frey, S. B. (1997). *Not Just for the Money: An Economic Theory of Personal Motivation*. Cheltenham, UK & Brookfield, US: Edward Elgar Publishing.

Fritz, T., & Murphy, G. C. (2010). Using information fragments to answer the questions developers ask. *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1, 1-8 May*. Cape Town, South Africa.

Gable, G. G. (1994). Integrating case study and survey research methods: an example in information systems. *European Journal of Information Systems*, *3*(2), 112–126.

García Muiña, F. E., De Castro, G., & López Sáez, P. (2002). The knowledge-creation process: a critical examination of the SECI model. *The Third European Conference on Organizational Knowledge, Learning and Capabilities (OKLC)*. 5-6 April, Athens.

Gebert, H., Geib, M., Kolbe, L., & Brenner, W. (2003). Knowledge-enabled customer relationship management: integrating customer relationship management and knowledge management concepts. *Journal of knowledge management*, *7*(5), 107–123.

Gillespie, N. (2003). Measuring trust in working relationships: The behavioral trust inventory. *Paper presented at the annual meeting of the Academy of Management Conference*. Seattle, WA.

Goh, S. C. (2002). Managing effective knowledge transfer: an integrative framework and some practice implications. *Journal of knowledge management*, *6*(1), 23–30.

Gordon, G., & Esbjörn-Hargens, S. (2007). Are we having fun yet? An exploration of the transformative power of play. *Journal of Humanistic Psychology*, *47*(2), 198–222.

Group, S. (2010). CHAOS Summary 2010. Retrieved July 7, 2014, from http://insyght.com.au/special/2010CHAOSSummary.pdf

Guest, G., Bunce, A., & Johnson, L. (2006). How Many Interviews Are Enough?: An Experiment with Data Saturation and Variability. *Field Methods*, *18*(1), 59–82.

Gupta, A. K., & Govindarajan, V. (2000). Knowledge flows within multinational corporations. *Strategic management journal*, *21*(4), 473–496.

Gupta, K. S. (2008). A comparative analysis of knowledge sharing climate. *Knowledge and Process Management*, *15*(3), 186–195.

Haas, K. P. (2007). The blending of traditional and agile project management. *PM World Today*, *9*(5), 1–8.

Hars, A., & Ou, S. (2001). Working for free? Motivations of participating in open source projects. *34th Annual Hawaii International Conference on System Sciences*. Hawaii, USA.

Hart, C. (2001). *Doing a literature search: a comprehensive guide for the social sciences*. London: Sage.

He, W., & Wei, K.-K. (2009). What drives continued knowledge sharing? An investigation of knowledge-contribution and-seeking beliefs. *Decision Support Systems*, *46*(4), 826–838.

Hertel, G., Konradt, U., & Orlikowski, B. (2004). Managing distance by interdependence: Goal setting, task interdependence, and team-based rewards in virtual teams. *European Journal of work and organizational psychology*, *13*(1), 1–28.

Hirschheim, R., & Klein, H. K. (1989). Four paradigms of information systems development. *Communications of the ACM*, *32*(10), 1199–1216.

Hoda, R., Noble, J., & Marshall, S. (2012). Developing a grounded theory to explain the practices of self-organizing Agile teams. *Empirical Software Engineering*, *17*(6), 609–639.

Holz, H., & Maurer, F. (2003). Knowledge management support for distributed agile software processes. In S. Henninger & F. Maurer (Eds.), *Advances in Learning Software Organizations* (pp. 60–80). Springer.

Hossain, E., Babar, M. A., & Paik, H. (2009). Using Scrum in global software development: A systematic literature review. *5th IEEE International Conference on Global Software Engineering (ICGSE), 13-16 July* (pp. 175–184). Limerick, Ireland.

Humphrey, W. S. (1988). The software engineering process: definition and scope. *Proceedings of the 4th international software process workshop on Representing and enacting the software process* (pp. 82–83). New York, NY, USA.

Hung, S.-Y., Durcikova, A., Lai, H.-M., & Lin, W.-M. (2011). The influence of intrinsic and extrinsic motivation on individuals' knowledge sharing behavior. *International Journal of Human-Computer Studies*, *69*(6), 415–427.

Huo, M., Verner, J., Zhu, L., & Babar, M. A. (2004). Software quality and agile methods. *Computer Software and Applications Conference, 2004. COMPSAC 2004.* (pp. 520–525). Hong Kong, China.

Hussein, A. (2009). The use of Triangulation in Social Sciences Research: Can qualitative and quantitative methods be combined. *Journal of Comparative Social Work*, *4*(1), 1–12.

Jacobson, M. J., & Spiro, R. J. (1995). Hypertext learning environments, cognitive flexibility, and the transfer of complex knowledge: An empirical investigation. *Journal of Educational Computing Research*, *12*(4), 301–333.

Jalali, S., & Wohlin, C. (2010). Agile practices in global software engineering-A systematic map. *5th IEEE International Conference on Global Software Engineering (ICGSE),13-16 July* (pp. 45–54). Limerick, Ireland.

Johnston, K. A. (2011). *An IS Perspective on Managing Change in a University*. University of Cape Town, Doctoral Thesis.

Joshi, K. D., Sarker, S., & Sarker, S. (2004). Knowledge transfer among face-to-face information systems development team members: Examining the role of knowledge, source, and relational context. *Proceedings of the 37th Hawaii International Conference on System Sciences*. Big Island, Hawaii.

Joshi, K. D., Sarker, S., & Sarker, S. (2007). Knowledge transfer within information systems development teams: Examining the role of knowledge source attributes. *Decision Support Systems*, *43*(2), 322–335.

Kajko-Mattsson, M. (2008). Problems in agile trenches. *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement* (pp. 111–119). New York, USA.

Kakkuri-Knuuttila, M.-L., Lukka, K., & Kuorikoski, J. (2008). Straddling between paradigms: a naturalistic philosophical case study on interpretive research in management accounting. *Accounting, Organizations and Society*, *33*(2), 267–291.

Kanellis, P., & Papadopoulos, T. (2008). Conducting research in information systems: an epistemological journey. *Information Systems Research: Public and Private Sector Applications, Idea Group Publishing*. Retrieved October 16, 2013, from http://www.igi-global.com/chapter/information-systems-research-methods-epistemology/23466

Kankanhalli, A., Lee, O.-K. D., & Lim, K. H. (2011). Knowledge reuse through electronic repositories: A study in the context of customer service support. *Information & Management*, *48*(2), 106–113.

Karlsen, J. T., Hagman, L., & Pedersen, T. (2011). Intra-project transfer of knowledge in information systems development firms. *Journal of Systems and Information Technology*, *13*(1), 66–80.

Kasser, J. E. (2002). The cataract methodology for systems and software acquisition. *Systems Engineering Test and Evaluation (SETE) Conference, 29-30 October*. Sydney, Australia.

Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS quarterly*, *23*(1), 67–93.

Ko, D.-G., Kirsch, L. J., & King, W. R. (2005). Antecedents of knowledge transfer from consultants to clients in enterprise system implementations. *MIS quarterly*, *29*(1), 59–85.

Kogut, B., & Zander, U. (1992). Knowledge of the firm, combinative capabilities, and the replication of technology. *Organization science*, *3*(3), 383–397.

Kontio, J., Hoglund, M., Ryden, J., & Abrahamsson, P. (2004). Managing commitments and risks: challenges in distributed agile development. *Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on* (pp. 732–733). May 23-28, Edinburgh, Scotland.

Kyobe, M. (2010). A knowledge management approach to resolving the crises in the information systems discipline. *Journal of Systems and Information Technology*, *12*(2), 161–173.

Larman, C., & Basili, V. R. (2003). Iterative and incremental developments. a brief history. *Computer*, *36*(6), 47–56.

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.

Leenders, R. T. A. J., Van Engelen, J. M. L., & Kratzer, J. (2003). Virtuality, communication, and new product team creativity: a social network perspective. *Journal of Engineering and Technology Management*, *20*(1), 69–92.

Levin, D., Lesser, E., Cross, R., & Abrams, L. (2005). Trust and knowledge sharing: a critical combination. In E. Lesser & L. Prusak (Eds.), *Creating Value with Knowledge* (pp. 36–41). Oxford: Oxford University Press.

Levin, D. Z., & Cross, R. (2004). The strength of weak ties you can trust: The mediating role of trust in effective knowledge transfer. *Management science*, *50*(11), 1477–1490.

Levin, D. Z., & Cross, R. (2012). Emergence of agile methods: perceptions from software practitioners in Malaysia. *AGILE India, 17 – 19 Feb* (pp. 30–39). Bengaluru, India.

Levin, K. A. (2006). Study design III: cross-sectional studies. *Evidence-based dentistry*, *7*(1), 24–25.

Levy, D. (2006). Qualitative methodology and grounded theory in property research. *Pacific Rim Property Research Journal*, *12*(4), 369–388.

Lewis, R. B. (2004). NVivo 2.0 and ATLAS.ti 5.0: A Comparative Review of Two Popular Qualitative Data-Analysis Programs. *Field Methods*, *16*(4), 439–464.

Liebowitz, J., & Chen, Y. (2003). Knowledge sharing proficiencies: the key to knowledge management. In C. W. Holsapple (Ed.), *Handbook of Knowledge Management: Knowledge Matters* (pp. 409–424). Berlin , Germany: Springer-Verlag.

Lin, H.-F. (2007). Effects of extrinsic and intrinsic motivation on employee knowledge sharing intentions. *Journal of information science*, *33*(2).

Lindvall, M., Basili, V. R., Boehm, B., Costa, P., Dangle, K., Shull, F., Tesoriero, R., et al. (2002). Empirical Findings in Agile Methods. *XP/Agile Universe, August 4–7*. Chicago, IL, USA.

Lindvall, Mikael, Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., May, J., et al. (2004). Agile software development in large organizations. *Computer*, *37*(12), 26–34.

Lindvall, Mikael, & Rus, I. (2002). Knowledge management in software engineering. *IEEE software*, *19*(3), 26–38.

Lyytinen, K., & Robey, D. (1999). Learning failure in information systems development. *Information Systems Journal*, *9*(2), 85–101.

Lyytinen, K., & Rose, G. M. (2006). Information system development agility as organizational learning. *European Journal of Information Systems*, *15*(2), 183–199.

Mahnič, V., & Hovelja, T. (2012). On using planning poker for estimating user stories. *Journal of Systems and Software*, *85*(9), 2086–2095.

Mann, C., & Maurer, F. (2005). A Case Study on the Impact of Scrum on Overtime and Customer Satisfaction. *Proceedings of the Agile Development Conference* (pp. 70–79). Washington, DC, USA: IEEE Computer Society. doi:10.1109/ADC.2005.1

Marshall, M. N. (1996). Sampling for qualitative research. *Family practice*, *13*(6), 522–526.

Maturana, H. R., & Varela, F. J. (1980). *Autopoiesis and cognition: The realization of the living* (1st ed., Vol. 42). Dordrecht: D. Reidel Publishing Company.

McAdam, R., & McCreedy, S. (1999). A critical review of knowledge management models. *Learning Organization, The*, *6*(3), 91–101.

McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2003). The impact of pair programming on student performance, perception and persistence. *Proceedings of the 25th international conference on Software engineering*. Portland, OR, USA.

McHugh, O., Conboy, K., & Lang, M. (2011). Using agile practices to influence motivation within IT project teams. *Scandinavian Journal of Information Systems*, *23*(2), 85–110.

McHugh, O., & Hogan, M. (2011). Investigating the rationale for adopting an internationally-recognised project management methodology in Ireland: the view of the project manager. *International Journal of Project Management*, *29*(5), 637–646.

Melnik, G., & Maurer, F. (2004). Direct verbal communication as a catalyst of agile knowledge sharing. *Agile Development Conference, 25-28 June* (pp. 21–31). Salt Lake City, Utah.

Meyer, S., Ward, P., Coveney, J., & Rogers, W. (2008). Trust in the health system: an analysis and extension of the social theories of Giddens and Luhmann. *Health Sociology Review*, *17*(2), 177–186.

Miller, J. and Mingins, C. (1998). Putting the Practice into Software Engineering Education. In M. P. S. Cranefield & and S. Macdowell (Eds.), *Software Engineering Education & Practice* (pp. 200–207). New Zealand.

Moe, N. B., Dingsoyr, T., & Dyba, T. (2008). Understanding self-organizing teams in agile software development. *19th Australian Software Engineering Conference (ASWEC 2008), 25-28 March*. Perth, Australia.

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.

Moe, N. B., & Šmite, D. (2007). Understanding lacking trust in global software teams: A multi-case study. *Product-Focused Software Process Improvement, July 2-4* (pp. 20–34). Riga, Latvia.

Monge, P. R., Bachman, S. G., Dillard, J. P., & Eisenberg, E. M. (1982). Communicator competence in the workplace: Model testing and scale development. *Communication yearbook*, *5*(505), 27.

Myers, M. D. (1997). Qualitative research in information systems. *Management Information Systems Quarterly*, *21*(2), 241–242.

Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and organization*, *17*(1), 2–26.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 72–78.

Nikitina, N., Kajko-Mattsson, M., & Strale, M. (2012). From scrum to scrumban: A case study of a process transition. *International Conference on Software and System Process (ICSSP)* (pp. 140–149). Zuerich, Switzerland.

Nonaka, I., & Konno, I. (1998). The concept of "Ba": building a foundation for knowledge creation. *California Management Review*, *40*(3), 40–54.

Nonaka, Ikujir, & Takeuchi, H. (1995). *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. Oxford University Press, USA.

Nonaka, Ikujiro, & Takeuchi, H. (2011). The wise leader. *Harvard Business Review*, *89*(5), 58–67.

Noor, K. B. M. (2008). Case study: a strategic research methodology. *American Journal of Applied Sciences*, *5*(11), 1602.

Oertig, M., & Buergi, T. (2006). The challenges of managing cross-cultural virtual project teams. *Team performance management*, *12*(1/2), 23–30.

Onions, P., & Orange, G. (2002). The three K's: a model for knowledge that supports ontology and epistemology. *6th World Multi-Conference on Systemics, Cybernetics and Informatics, SCI*. 14-18 July, Orlando, Florida.

Ortiz, A. (2003). Preparing undergraduate students for Java certification. *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications. October 26-30* (pp. 178–183). Anaheim, CA.

Osterloh, M., & Frey, B. S. (2000). Motivation, knowledge transfer, and organizational forms. *Organization science*, *11*(5), 538–550.

Oxford Dictionary. (2013).*Oxford University Press*. Retrieved July 15, 2013, from http://oxforddictionaries.com/definition/english/knowledge

Paetsch, F., Eberlein, A., & Maurer, F. (2003). Requirements engineering and agile software development. *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.*, 308–313.

Pandey, S. C., & Dutta, A. (2013). Role of knowledge infrastructure capabilities in knowledge management. *Journal of knowledge management*, *17*(3), 435–453.

Patton, M. Q. (1990). *Qualitative evaluation and research methods*. Newbury Park, CA: Sage.

Petersen, K., & Wohlin, C. (2009). A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of Systems and Software*, *82*(9), 1479–1490.

Pressman, R. S., & Ince, D. (1992). *Software engineering: a practitioner's approach. Vol. 5. New York: McGraw-hill, 1992.* (Volume 5.). New York: McGraw-hill.

Prifling, M., Gregory, R., & Beck, R. (2008). Project Management Techniques for Managing Cross-Cultural Differences in IT Offshore Outsourcing. *Americas Conference on Information Systems (AMCIS) 2008 Proceedings, 14-17 August*. Toronto, Ontario, Canada.

Puck, J., Rygl, D., & Kittler, M. (2007). Cultural antecedents and performance consequences of open communication and knowledge transfer in multicultural process-innovation teams. *Journal of Organisational Transformation & Social Change*, *3*(2), 223–241.

Racheva, Z., Daneva, M., & Buglione, L. (2008). Complementing measurements and real options concepts to support inter-iteration decision-making in agile projects. *34th Euromicro Conference on Software Engineering and Advanced Applications (SEAA'08)* (pp. 457–464). Parma, Italy.

Remus, W. (1996). Will behavioral research on managerial decision making generalize to managers? *Managerial and Decision Economics*, *17*(1), 93–101.

Riege, A. M. (2003). Validity and reliability tests in case study research: a literature review with "hands-on" applications for each research phase. *Qualitative Market Research: An International Journal*, *6*(2), 75–86.

Rodriguez-Ulloa, R., & Paucar-Caceres, A. (2005). Soft system dynamics methodology (SSDM): combining soft systems methodology (SSM) and system dynamics (SD). *Systemic Practice and Action …*, *18*(3), 303–334.

Roode, D. (2008). The ethical researcher : festschrift : dedicated to Derrick Kourie. *South African Computer Journal*, *41*, 38–42.

Rowlands, B. H. (2005). Grounded in practice: Using interpretive research to build theory. *The Electronic Journal of Business Research Methodology*, *3*(1), 81–92.

Royce, W. W. (1970). Managing the development of large software systems. *Proceedings of IEEE WESCON* (Vol. 26). Los Angeles, USA.

Rubinstein, D. (n.d.). Standish Group Report: There's Less Development Chaos Today. *Software Development Times*. Retrieved July 15, 2013, from http://sdt.bz/30247

Ryan, R. M., & Deci, E. L. (2000a). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, *55*(1), 68.

Ryan, R. M., & Deci, E. L. (2000b). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, *25*(1), 54–67.

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.

Rygielski, C., Wang, J., & Yen, D. C. (2002). Data mining techniques for customer relationship management. *Technology in society*, *24*(4), 483–502.

Salleh, N., Mendes, E., & Grundy, J. (2011). Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review. *Software Engineering, IEEE Transactions on*, *37*(4), 509–525.

Schenk, D., & Lungu, M. (2013). Geo-locating the knowledge transfer in StackOverflow. *Proceedings of the 2013 International Workshop on Social Software Engineering, 18-26 August* (pp. 21–24). Saint Petersburg, Russian Federation.

Schramm, W. (1971). The nature of communication between humans. In W. Roberts & S. D.F. (Eds.), *the Process and Effects of Mass Communication* (Revised., pp. 3–53). Champaign, Illinois: University of Illinois Press.

Schwaber, K. (1995). Scrum development process. *OOPSLA'95 Workshop Proceedings, 16 October*. Austin, Texas.

Sense, A. J. (2008). Conceptions of learning and managing the flow of knowledge in the project-based environment. *International Journal of Managing Projects in Business*, *1*(1), 33–48.

Shannon, C. E., & Weaver, W. (1948). The Mathematical Theory of Communication. *The Bell System Technical Journal*, *27*, 379–423.

Shenton, A. K. (2004). Strategies for ensuring trustworthiness in qualitative research projects. *Education for information*, *22*(2), 63–75.

Siemsen, E., Balasubramanian, S., & Roth, A. V. (2007). Incentives that induce task-related effort, helping, and knowledge sharing in workgroups. *Management Science*, *53*(10), 1533–1550.

Stanca, L., Bruni, L., & Corazzini, L. (2009). Testing theories of reciprocity: Do motivations matter? *Journal of economic behavior & organization*, *71*(2), 233–245.

Sutherland, J., Schoonheim, G., Rustenburg, E., & Rijk, M. (2008). Fully distributed scrum: The secret sauce for hyperproductive offshored development teams. *Agile, 2008. AGILE'08. Conference from August 4-8* (pp. 339–344). Toronto,Canada.

Sutherland, J., Viktorov, A., Blount, J., & Puntikov, N. (2007). Distributed scrum: Agile project management with outsourced development teams. *Hawaii International Conference on System Sciences, 3-6 January*. Hawaii, USA.

Sveiby, K.-E. (2001). A knowledge-based theory of the firm to guide in strategy formulation. *Journal of intellectual capital*, *2*(4), 344–358.

Svensson, H., & Höst, M. (2005). Views from an organization on how agile development affects its collaboration with a software development team. *Product Focused Software Process Improvement, June 13-15* (pp. 487–501). Oulu, Finland.

Svensson, R. B., Parker, P., & Regnell, B. (2011). A prototype tool for QUPER to support release planning of quality requirements. *The 5th International Workshop on Software Product Management, August 30* (pp. 57–66). Trento, Italy.

Szulanski, G. (1995). Unpacking stickiness: an empirical investigation of the barriers to transfer best practice inside the firm. *Academy of Management Best Papers Proceedings* (Vol. 1995, pp. 437–441).

Szulanski, G. (1996). Exploring internal stickiness: Impediments to the transfer of best practice within the firm. *Strategic management journal*, *17*(Special Issue), 27–43.

Szulanski, G., Cappetta, R., & Jensen, R. J. (2004). When and how trustworthiness matters: Knowledge transfer and the moderating effect of causal ambiguity. *Organization Science*, *15*(5), 600–613.

Takeuchi, H., & Nonaka, I. (1986). The new new product development game. *Harvard business review*, *64*(1), 137–146.

Taylor, S., & Todd, P. A. (1995). Understanding information technology usage: A test of competing models. *Information systems research*, *6*(2), 144–176.

Tellis, W. (1997). Application of a case study methodology. *The qualitative report*, *3*(3), 1–17.

Tesch, D., Sobol, M. G., Klein, G., & Jiang, J. J. (2009). User and developer common knowledge: Effect on the success of information system development projects. *International Journal of Project Management*, *27*(7), 657–664.

Thompson, C. (1999). If you could just provide me with a sample: examining sampling in qualitative and quantitative research papers. *Evidence Based Nursing*, *2*(3), 68–70.

Timbrell, G., Delaney, P., Chan, T., Yue, W., & Gable, G. (2005). A structurationist review of knowledge management theories. *Proceedings Twenty- Sixth International Conference on Information Systems* (pp. 247–259). Las Vegas, Nevada, USA. Retrieved from http://eprints.qut.edu.au/10104

Timbrell, G. T., Delaney, P., Chan, T., Yue, W. A., & Gable, G. (2005). A structurationist review of knowledge management theories.

Trauth, E. M. (2010). Knowledge Transfer Challenges for Universities and SMEs in the USA. *Americas Conference on Information Systems (AMCIS) 2010 Proceedings, August 12-15*. Lima, Peru.

Tsai, W., & Ghoshal, S. (1998). Social capital and value creation: The role of intrafirm networks. *Academy of management Journal*, *41*(4), 464–476.

Van Lange, P. A. M. (2008). Does empathy trigger only altruistic motivation? How about selflessness or justice? *Emotion*, *8*(6), 766.

Venzin, M., von Krogh, G., & Roos, J. (2000). Future research into knowledge management. In J. R. & D. K. G.vonKrogh (Ed.), *Knowing in firms: Understanding, managing, and measuring knowledge*. Thousand Oaks , CA : Sage Publications.

VersionOne. (2013). State of Agile Development. Retrieved October 15, 2013, from http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf

Voss, C., Tsikriktsis, N., & Frohlich, M. (2002). Case research in operations management. *International journal of operations & production management*, *22*(2), 195–219.

Walsham, G. (1995). Interpretive case studies in IS research: nature and method. *European Journal of information systems*, *4*(2), 74–81.

Walsham, G. (2006). Doing interpretive research. *European Journal of Information Systems*, *15*(3), 320–330.

Walsham, G., & Sahay, S. (1999). GIS for district-level administration in India: problems and opportunities. *MIS quarterly*, 39–65.

Wang, S., & Noe, R. (2010). Knowledge sharing: A review and directions for future research. *Human Resource Management Review*, *20*(2), 115–131.

Wendling, M., Oliveira, M., & Maçada, A. C. G. (2013). Challenges and solutions in distributed software development project management: A systematic literature review. *Journal of Systems and Information Technology*, *15*(3).

West, D., Grant, T., Gerush, M., & D'Silva, D. (2010). Agile development: Mainstream adoption has changed agility. *Forrester Research*.

Wiig, K. M. (1997). Knowledge management: where did it come from and where will it go? *Expert systems with applications*, *13*(1), 1–14.

Yin, R. K. (1994). *Case Study Research: Design and Methods (Applied Social Research Methods)*. Sage Thousand Oaks, CA.

Zack, M. H. (1999). Managing codified knowledge. *Sloan management review*, *40*(4), 45–58.

Zand, D. E. (1972). Trust and managerial problem solving. *Administrative science quarterly*, *17*, 229–239.

Zarraga, C., & Bonache, J. (2005). The impact of team atmosphere on knowledge outcomes in self-managed teams. *Organization Studies*, *26*(5), 661–681.

Zellmer-Bruhn, M. E. (2003). Interruptive events and team knowledge acquisition. *Management Science*, *49*(4), 514–528.

Zhou, Q. (2014). That usability course: what technical communication programs get wrong about usability and how to fix it. *Communication Design Quarterly Review*, *2*(3), 25–27.

Appendix A: Semi Structured Interview Protocol

The following is an outline of the interview questions. These will not be strictly adhered to, as new questions can arise based on the flow of the interview, since this is semi-structured (Louise Barriball and While, 1994). They are mostly inspired from Joshi et al (2007) questionnaire and the motivation questions are adapted from Ko et al (2005).

Opening Common Questions

- What are your thoughts of Scrum development methodology? Compared to previous methodologies you have used?
- Please describe the Scrum project you were on?
- What was your role in that project?
- How long have you been in this role?
- What is your understanding of technical skills in software development projects? (i.e. programming, testing, designing, database and requirements gathering knowledge)
- What is your understanding of project planning skills in software development and Scrum projects? (i.e. project scope, estimating, scheduling, task allocation, decision making, and team building)
- What is your understanding of inter-personal skills? (i.e. providing direction, communicating, assisting with problem solving, and dealing effectively with people without having authority.)

Content Questions

Perceived knowledge transferred

- Is there a person whom you have acquired skills from during the past Scrum project?
- What are your own technical skills? To what extent do you feel that you've acquired new technical skills from X? Why?
- What are your own project planning skills? To what extent do you feel that you've acquired new project planning skills from X? Why?

- What are your own inter-personal skills? To what extent do you feel that you've acquired new inter-personal skills from X? Why?

Capability of Source

Technical capability

- According to you, what were their technical expertise? How would you rate these skills? Do you feel that this also influenced your ability to learn from them?

Managerial capability

- According to you, what were their project planning skills? How would you rate these skills? Do you feel that this also influenced your ability to learn from them?

Motivation of Source

Intrinsic motivation

- Did you feel they enjoyed working in Scrum project?
- Did you feel they enjoyed learning new things about Scrum project?
- How does they handle challenges?

Extrinsic motivation

- Did you feel that financial rewards motivated their work in the Scrum project?
- Did you feel that peer-recognition motivated their work in the Scrum project?
- Overall, do you feel they were more naturally or externally motivated?
- Do you feel that their motivation influenced your ability to learn from them?

Credibility of source

- What would you say about the trustworthiness of this team member? For example, do you ever doubt that they will finish a task that you have assigned them?
- What would you say about the competence of this team member?
- Do you feel that their motivation influenced your ability to learn from them?

Communication extent of source

- How do you find the communication skills of the other team mates?
- Do you feel that their motivation influenced your ability to learn from them?

Concluding Common Questions

- What characteristics do you feel that good communicators should have?

Thanks very much. Do you have any questions for me?

Appendix B: Organizational Permission to Perform Research Study

Good Day,

I am Deon Takpuie, an Information Systems Masters student at the University of Cape Town conducting research on knowledge transfer within Scrum development teams.

The research objectives are to:

- Determine how motivation affects the perceived extent of knowledge transferred by a team member.
- Determine how credibility affects the perceived extent of knowledge transferred by a team member.
- Determine how the extent of communication of a team member affects the perceived extent of knowledge transferred.
- Determine what other characteristics team members associate with a person that can transfer knowledge successfully.

Subsequently, the research question has been formulated as follows: *"What is the effect of knowledge transfer skills of Scrum team members on the perceived extent of knowledge transferred to other team members".* This is the aim of the research.

This study will also be of benefit to companies using Scrum development in the following ways. It will produce a list of factors to enable managers to identify the right type of team members to initiate knowledge transfers and for human resource personnel to train others with these skills.

As a means to answer these research questions, we would like to request permission to perform a research study at your company. For the research study, we would need to interview members of your Scrum development teams. Viewing supporting project documentation and direct observation such as participating in Scrum meetings would also be appreciated. Participant in this study will remain anonymous and all the information gathered about the organisation and the Scrum teams will be treated as strictly confidential. In addition, participation to the study will be voluntary and participants can opt to withdraw from the research at any time. Each interview should last approximately 1 hour and no participants will be requested to supply any identifiable information, thus ensuring their anonymity. Ethics approval from the Commerce Faculty Ethics

in Research Committee is currently being processed, so all that is requested for now is your company's initial interest to participate in this study. We will forward you the formal ethics approval letter, when it is available.

Thanks

Kind Regards

Deon Takpuie (Researcher)

tkpdeo001@myuct.ac.za

Dr. Maureen Tanner (Supervisor)

mc.tanner@uct.ac.za

Appendix C: Individual Participation Consent Form

UNIVERSITY OF CAPE TOWN

Department of Information Systems

Leslie Commerce Building

Upper Campus

Or Private Bag, Rondebosch, 77001

Cape Town

Tel: 650-2261

Fax No: (021) 650-2280

Interview participation consent form

One of the requirements for completing a Master's degree in Information Systems (IS) in the Commerce Faculty of the University of Cape Town (UCT) is the completion of a dissertation research project. As a student pursuing the aforementioned degree, I am investigating the role of a team member in knowledge transfer within Scrum development teams. Therefore, I would like to request official permission to interview you for approximately 60 minutes. The aim of the research is to find out:

"What characteristics are associated with team members who are able to transfer tacit knowledge successfully in Scrum projects?"

The research objectives of this study are to:

- Determine how motivation affects the perceived extent of knowledge transferred by a team member.
- Determine how credibility affects the perceived extent of knowledge transferred by a team member.
- Determine how the extent of communication of a team member affects the perceived extent of knowledge transferred.
- Determine what other characteristics are associated with team members who are able to transfer tacit knowledge successfully.

Research ethics is an issue of utmost importance to me, the Information Systems department, the faculty of Commerce and the University of Cape Town. Indeed, this study has been approved by the Commerce Faculty Ethics in Research Committee.

Consequently, I guarantee confidentiality and anonymity of the information provided for this study. All comments and details will be treated in strict confidence and will be used strictly for the sole purpose of the aforementioned dissertation research project. The results of the research will also be made available to you.

Your participation in this study is entirely voluntary. You will not be requested to supply any identifiable information, ensuring anonymity of your responses. You may opt out of the study at any point in time without any consequences. If you opt to participate in this research project, please sign the consent form below.

Participant Consent Form Agreement

By signing this Participant Consent Form, you are agreeing to participate in a research project entitled, ***"Investigating the characteristics that a team member should have in order to transfer knowledge within Scrum development teams"***, conducted by Deon Takpuie as one of the requirements for the completion of an Master's degree in IS. The researcher guarantees confidentiality and anonymity of the details and comments you will provide in this study. All comments and details will be treated in strict confidence and will be used strictly for the sole purpose of the aforementioned dissertation research project.

Signature_____Date_____

If you have any questions regarding this research, please feel free to contact:

Deon Takpuie (Researcher) OR **Dr. Maureen Tanner** (Supervisor)

tkpdeo001@myuct.ac.za mc.tanner@uct.ac.za

0728307968 021 650 4860

Deon Takpuie (Researcher) **Dr. Maureen Tanner** (Supervisor)

tkpdeo001@myuct.ac.za mc.tanner@uct.ac.za

0728307968 021 650 4860

Appendix D: Codebook

Name	Gro...	De...	Author	Created	Modified
1 Motivation	0	0	Super	27/03/20...	27/03/20...
1.1 Intrinsic Motivation	0	0	Super	27/03/20...	27/03/20...
1.1.1 Interest	0	0	Super	01/04/20...	01/04/20...
1.1.2 Enjoyment#A1	26	1	Super	01/04/20...	27/08/20...
1.1.2.1 Enjoy Job in General	5	0	Super	02/04/20...	27/08/20...
1.1.2.2 Enjoy Working in Scrum	15	0	Super	02/04/20...	03/04/20...
1.1.2.3 Enjoy Learning Scrum	6	0	Super	02/04/20...	27/08/20...
1.1.2.4 Enjoy Learning	4	0	Super	02/04/20...	27/08/20...
1.1.3 Passion#A2	4	1	Super	01/04/20...	27/08/20...
1.1.4 Motivation to Teach	0	0	Super	02/04/20...	02/04/20...
1.1.5 Embraces Challenges#B	4	0	Super	02/04/20...	27/08/20...
1.1.6 Motivation Validitity#C	7	0	Super	02/04/20...	09/04/20...
1.2 Extrinsic Motivation	0	0	Super	27/03/20...	27/03/20...
1.2.1 Financially Motivated	11	0	Super	01/04/20...	03/04/20...
1.2.2 Goal Driven	0	0	Super	01/04/20...	01/04/20...
1.2.3 Peer Recognition	2	0	Super	01/04/20...	03/04/20...
2 Capability	6	0	Super	02/04/20...	07/04/20...
2.1 Technical Capability	24	0	Super	03/04/20...	10/04/20...
2.1.1 Experience	4	0	Super	04/04/20...	10/04/20...
2.1.1.1 Specialized Experience/Skills	2	0	Super	04/04/20...	10/04/20...
2.1.2 Software Development Skills	14	0	Super	04/04/20...	10/04/20...
2.1.3 Technical Capability Validity	5	0	Super	07/04/20...	09/04/20...
2.2 Managerial Capability	18	0	Super	03/04/20...	22/08/20...
2.2.1 Project Management	4	0	Super	04/04/20...	07/04/20...
2.2.2 Agile Methodology Knowledge	10	0	Super	04/04/20...	22/08/20...
2.2.3 Leadership#Scrum	11	0	Super	04/04/20...	22/08/20...
2.2.4 Managerial Capability Validity	4	0	Super	07/04/20...	07/04/20...
3 Credibility	0	0	Super	02/04/20...	02/04/20...
3.1 Trust	10	3	Super	02/04/20...	10/04/20...
3.1.1 Is Trustworthy	11	1	Super	09/04/20...	10/04/20...

3.1.2 Explained Trust	18	1	Super	09/04/20...	10/04/20...
3.1.3 Personal Trust	2	1	Super	09/04/20...	21/06/20...
3.2 Reputation	5	0	Super	02/04/20...	09/04/20...
3.2.1 Is Reputable	0	0	Super	09/04/20...	09/04/20...
3.2.2 Highly Competent	13	0	Super	09/04/20...	10/04/20...
3.2.2.1 Competency Explained	2	0	Super	09/04/20...	09/04/20...
3.3 Low Competence	0	0	Super	09/04/20...	09/04/20...
3.4 Approachability???	1	0	Super	02/04/20...	09/04/20...
3.5 Is Credibility A Factor	1	0	Super	09/04/20...	09/04/20...
4 Communication Extent of Source	1	0	Super	04/04/20...	09/04/20...
4.1 Frequent	11	0	Super	04/04/20...	22/08/20...
4.1.1 Frequent and Happy #Happy	1	0	Super	09/04/20...	09/04/20...
4.2 Non-Frequent	0	0	Super	04/04/20...	09/04/20...
4.2.1 Medium Frequency	2	0	Super	09/04/20...	09/04/20...
4.3 Preferred Frequency Amount	11	0	Super	09/04/20...	22/06/20...
4.3.1 Any	0	0	Super	09/04/20...	09/04/20...
4.3.2 Low	0	0	Super	09/04/20...	09/04/20...
4.3.3 Balance	8	0	Super	09/04/20...	22/06/20...
4.3.4 High	5	0	Super	09/04/20...	22/06/20...
4.4 Various Mediums	2	0	Super	09/04/20...	09/04/20...
4.5 Various Types	1	0	Super	09/04/20...	09/04/20...
4.6 Communication and KT #NB	5	0	Super	09/04/20...	09/04/20...

5 Other Characteristics	0	0	Super	22/03/20...	04/04/20...
5.1.1 Empathetic#A	8	2	Super	22/03/20...	22/06/20...
5.1.10 Think Before Communicating	2	0	Super	22/03/20...	22/06/20...
5.1.11 Allow Feedback	1	0	Super	22/03/20...	04/04/20...
5.1.12 Non-dictative	2	0	Super	22/03/20...	04/04/20...
5.1.13 Facilitation	1	0	Super	22/03/20...	04/04/20...
5.1.14 Listening Skills	4	0	Super	22/03/20...	22/06/20...
5.1.2 Anticipation#A	2	1	Super	22/03/20...	22/06/20...
5.1.3 Approachability	1	0	Super	22/03/20...	04/04/20...
5.1.4 Communication Medium	7	0	Super	22/03/20...	22/06/20...
5.1.4.1 Varying communication medium	1	0	Super	22/03/20...	04/04/20...
5.1.4.2 Best Communication Medium	9	0	Super	22/03/20...	22/06/20...
5.1.5 Authenticity#A	1	1	Super	22/03/20...	04/04/20...
5.1.6 Concise#B	2	1	Super	22/03/20...	04/04/20...
5.1.7 Share Information	3	0	Super	22/03/20...	22/08/20...
5.1.7.1 Enable Information Sharing	1	0	Super	22/03/20...	04/04/20...
5.1.8 Patience	4	0	Super	22/03/20...	22/06/20...
5.1.9 Articulate#B	7	1	Super	22/03/20...	05/07/20...