Exploring Knowing in Practice
An Ethnographic Study of Teams in the Agile Setting
Abstract

Studies related to knowledge often fall under the traditional knowledge management discourse which views knowledge as a static and transferable entity. However, the predominant knowledge in agile software development is tacit and resides in the heads of individuals with the expertise. The study acknowledges the importance of human agency and moves away from the view of knowledge as a static entity to the perspective of “knowing” which is constituted in our daily actions and practices. The focus is to examine how “knowing” or knowledgeability is enacted through the practices of teams in agile projects.

A conceptual model was developed from current literature and used as a guide for data collection and analysis. The study subscribed to phenomenological hermeneutics as the philosophical tradition to interpret meaning and the worldviews of the participants. Ethnographic research was conducted in two companies which work with software development using agile methodologies. The findings indicated several practices which were enacted by the two organizations, of which many of them appeared to be interrelated. The “knowing” which was constituted in these practices were categorized as “knowing how to develop competencies/ capabilities” and “knowing how to coordinate effort”.

The practices found in the study were similar to some of the practices commonly cited in current knowledge management literature. Notwithstanding this, new knowledge was not created from the conversion of tacit to explicit knowledge and vice versa. On the contrary, “knowing how” was generated and sustained through the ongoing actions and everyday practices of the individuals in the agile teams.

Key Words: Knowledge, Knowledgeability, Knowing, Knowing in Practice, Agile Software Development
Acknowledgements

This thesis has been written from the dark days in winter to the abundance of sunshine in spring. Its completion would not have been possible without the following people.

First and foremost, I would like to thank my thesis supervisor, Jeff Winter, for his guidance, support and constructive feedback. The Skype chats have been very valuable and are deeply appreciated.

I would like to express my gratitude to my examiner, Christina Mörtberg, for her advice and insightful comments throughout the study. Thank you for your suggestions and helping me to find a clear sense of direction in times of uncertainty.

I would like to extend my appreciation to Surikat and to the Net Entertainment office in Gothenburg for participating in this study. The participants have been very helpful and forthcoming in the research and it has truly been a pleasure conducting the research in these two organizations.

Last but not least, I would like to thank my family, friends and classmates who have been very encouraging and supportive.
# Table of Contents

1. Introduction ................................................................................................................................. 1
   1.1 Background and Problem Area ................................................................................................. 1
   1.2 Topic Justification ..................................................................................................................... 2
   1.3 Purpose and Research Question ............................................................................................... 3
   1.4 Significance and Contribution of the Study .............................................................................. 3
   1.5 Scope and Limitations ............................................................................................................. 4
   1.6 Disposition of the Thesis ......................................................................................................... 5
2. Theoretical Framework .................................................................................................................. 6
   2.1 Perspectives on Knowledge ...................................................................................................... 6
      2.1.1 Data, Information and Knowledge ..................................................................................... 7
      2.1.2 Tacit Knowledge .............................................................................................................. 8
      2.1.3 Explicit Knowledge ......................................................................................................... 10
   2.2 A Perspective on Knowing in Practice ...................................................................................... 11
   2.3 Knowledge Management ........................................................................................................ 12
      2.3.1 Knowledge Creation – SECI Model .................................................................................. 13
   2.4 Agile Software Development ................................................................................................... 16
      2.4.1 Scrum ............................................................................................................................... 17
      2.4.2 Pair Programming ............................................................................................................. 18
   2.5 Knowledgeability in Everyday Practices .................................................................................. 18
      2.5.1 Sharing “knowing how” .................................................................................................. 19
      2.5.2 Engaging in Communities of Practice ............................................................................. 19
      2.5.3 Observing and Learning by doing .................................................................................... 21
      2.5.4 Verbal communication .................................................................................................... 21
      2.5.5 Using metaphors and analogies ...................................................................................... 22
      2.5.6 Storytelling ..................................................................................................................... 23
   2.6 Conceptual Model .................................................................................................................... 23
3. Methodology .................................................................................................................................. 25
   3.1 Philosophical Tradition ............................................................................................................. 25
   3.2 Frame of Reference .................................................................................................................. 26
   3.3 Methodological Approach ....................................................................................................... 27
   3.4 Research Setting ...................................................................................................................... 28
3.4.1 Surikat AB ................................................................. 29
3.4.2 Net Entertainment ..................................................... 29
3.5 Profile of the Research Participants .................................. 30
3.6 Method for Data Collection and Analysis .............................. 31
   3.6.1 Literature Review .................................................. 31
   3.6.2 Observations ......................................................... 31
   3.6.3 Informal Interviews ................................................ 34
   3.6.4 Semi-structured Interviews ..................................... 34
   3.6.5 Documents .......................................................... 35
3.7 Role of the Researcher .................................................. 35
3.8 Data Analysis ............................................................. 36
3.9 Trustworthiness .......................................................... 38
3.10 Ethical Considerations ................................................ 41
4. Analysis and Results ....................................................... 43
   4.1 Sharing “knowing how” .............................................. 43
   4.2 Engaging in Communities of Practice ............................... 45
   4.3 Learning by observing ................................................. 46
   4.4 Learning by doing ..................................................... 47
   4.5 Verbal communication ............................................... 49
   4.6 Using metaphors and analogies .................................... 51
   4.7 Storytelling ............................................................. 52
   4.8 Other findings and insights ......................................... 54
   4.9 Summary of the findings ............................................. 57
5. Discussion ........................................................................ 59
   5.1 Generating and sustaining knowledgeability in agile teams ........ 59
   5.2 Implications for organizations ....................................... 60
   5.3 Implications for knowledge management: Can knowledge be managed? ... 61
6. Conclusion ........................................................................ 63
   6.1 Reflections .................................................................. 63
   6.2 Suggestions for future work ......................................... 64
7. References ......................................................................... 65
Appendix A - Interview Questions ....................................... 73
List of Figures

Figure 2.1 The Reversed Hierarchy (Adapted from Tuomi, 1999, p.9).........................8
Figure 2.2 The Tacit/ Explicit Dimension (Adapted from Grant, 2007, p.177)...............9
Figure 2.3 Information and Knowledge Conversions (Adapted from Stenmark, 2001, p.7)....10
Figure 2.4 SECI Model (Adapted from Nonaka and Takeuchi, 1995, p.45)......................13
Figure 2.5 The Duality of Participation and Reification (Adapted from Wenger, 1998, p.63)..................................................................................................................20
Figure 2.6 Conceptual Model..........................................................................................24
Figure 3.1 Hermeneutics Circle of Understanding Applied in the Study......................38
(Adapted from Butler, 1998 and Kurti, 2011)

List of Tables

Table 2.1 Traditional and Agile Perspective of Software Development.......................17
(Cited in Nerur and Balijepally, 2007, p.82)
Table 3.1 List of Observations and Informal Interviews.............................................33
Table 4.1 Overview of Practices, Activities and Knowing...........................................58
(Formulation of findings inspired by Orlikowski, 2002)
1. Introduction

This chapter presents the topic of the research. The background and problem area are introduced. This is followed by the justification for the topic, the purpose of the study and the presentation of the research question. The contributions of the study and its scope and limitations are further discussed.

1.1 Background and Problem Area

The global economy has seen a significant transition from an industrial economy to a knowledge-based economy where production and services are primarily driven by knowledge intensive activities (Powell and Snellman, 2004). The ability to manage and mobilize knowledge has taken on a greater focus and become all the more crucial for companies to remain competitive in today’s knowledge-driven economy (Dalkir, 2005; Temple, 1999; Haldin-Herrgard, 2000).

Discussions on knowledge are often subsumed under the knowledge management literature. Knowledge management is a fairly new concept, it is nebulous, complex and has potential overlaps with information management; yet it is something which is commonly practiced in many organizations (Bouthilier and Shearer, 2002; McInerney, 2002). Knowledge management is often seen as a silver bullet for firms to achieve competitive advantage. Knowledge-driven industries such as the software sector are no exception. To help teams to learn from past and existing projects, solve problems and enhance work productivity, the importance of reliable and effective knowledge management has come into the fore (Bari and Ahamad, 2011; Dingsoyr, 2002).

To the extent that knowledge management literature examines knowledge, it has emphasized on the creation, codification, storage, transfer and distribution of knowledge through information technologies (Stenmark, 2000; Alvesson and Kärreman, 2001; Mingers, 2001). However, the focus on managing knowledge by making it explicit rests on the presumption that knowledge is something static and that it can easily be transferred or distributed. Such a perspective negates the importance of human activity in exhibiting knowledgeable work.

Software development is knowledge intensive – it is a collaborative process that needs to combine domain expertise with technological skills and process knowledge (Chau and Maurer, 2004). The knowledge found in such a practice is tacit and as McInerney (2002) describes, dynamic and actively acquired through sensory stimulation, listening to and observing others, reading, being aware of feelings, life experiences and learning. Since knowledge is constantly in flux and changing through action, it raises the fundamental question if it is appropriate to continue viewing or treating knowledge as an entity which is stable and can be managed. This study proposes a move away from the focus on knowledge towards exploring the concept of “knowing” which is found in our actions and daily practices.
1.2 Topic Justification

The tacit aspect of “knowing” is well-established in the Scandinavian tradition. This is represented in the field of informatics such as in the areas of computer design, systems design and participatory design. Scandinavian authors such as Ehn (1988) and Stolterman (1992) discuss tacit knowing in terms of skills and “knowing how” to perform tasks by understanding situational elements. Moreover, the focus on the participants’ knowing and how to include the knowers’ expertise in everyday activities have been discussed in the works of Elovaara, Igira and Mörtberg (2006) and Elovaara and Mörtberg (2010).

However, in other contexts than the Nordics, much of the focus on tacit knowledge is discussed under the area of knowledge management (Baumard, 1999; Busch, 2008). Existing perspectives on knowledge in knowledge management – even though contribute to our understanding of knowledge as an organizational asset – prioritize the constructs of knowledge-as-object or knowledge-as-disposition over knowing-as-doing (Orlikowski, 2002). Knowledge is seen as something which people possess rather than the “knowing” which is found in individual and group practices (Cook and Brown, 1999). It is important to note that although the Scandinavian tradition may be rooted in some organizations, there are companies in Scandinavia which have not been affected by this tradition.

Polanyi (1958) and Ryle (1949) speak of “knowing” as it is an essential constituent of knowledge. In the same vein, Schön (1983) and Suchman (2007) suggest that our knowing is found in our actions. Building upon the notion of “knowing”, Orlikowski (2002) further discusses the concept of knowing in practice. Knowing in practice can be defined as knowledgeability that is constantly generated through ongoing action (Orlikowski, 2002; Feldman and Orlikowski, 2011). Orlikowski (2002) emphasizes the centrality of action in knowledgeability in terms of “knowing how” and knowing in practice, and uses these terms interchangeably.

To understand the perspective of “knowing” that is constituted in our action requires a shift in the lens away from the concept of knowledge as discussed in the traditional knowledge management discourse towards the concept of how our “knowing” is put into practice. A large part of this perspective is influenced by the works of Orlikowski (2002) and Feldman and Orlikowski (2011) who emphasize on understanding daily practices and the role of human agency in performing knowledge work. As Orlikowski (2002) asserts, “knowing” evolves from the ongoing and situated practices of people as they interact with the world. Contrary to the traditional view which differentiates knowledge from practice, this view of “knowing” guides us to understand that knowledge and practice are mutually constitutive. It is hence worthy to pursue the perspective that concentrates on knowledgeability or knowing in practice rather than knowledge.

This study chooses to examine knowing in practice through the agile setting of software development for several reasons. Lawton (2001) highlights the presence of knowledge management in the software industry as early as the 1980s and that the focus on managing, storing and converting knowledge has become all the more significant since the 1990s. The
nature of software development has similarly evolved, from the traditional aspects of control and predictability to a heightened value on adaptation, flexibility and human interaction found in agile methodologies (Nerur and Balijepally, 2007).

As Ryan and O’Connor (2013) argue, the increasing popularity of agile methods warrants a closer look at its underlying principles. Much of the knowledge in agile development is tacit (Nerur, Mahapatra and Mangalara, 2005; Ryan and O’Connor, 2013). Studies which examine the concept of knowledge in the agile setting (Holz and Maurer, 2003; Chau, Maurer and Melnik, 2003; Maalej and Happel, 2007) generally discuss the creation and sharing of tacit knowledge as if it were a transferable entity. Even though there are studies which discuss knowledgeability, there remain few studies related to agile practices. Understanding how knowledgeability is produced in the agile setting of software development hence becomes all the more relevant and necessary.

1.3 Purpose and Research Question

Russo and Stolterman (2000, p.315) states that “the choice of a research goal is not about what to study but also from what perspective to approach the study”. The purpose of this study is to use the lens of “knowing” to examine how knowing in practice or knowledgeability is enacted in teams in agile projects. The study draws from the hermeneutics tradition to understanding the phenomenon of interest.

To achieve the objective of this research, I posit the following research question:

1. How do teams in agile projects generate and sustain knowledgeability?

1.4 Significance and Contribution of the Study

This study acknowledges the complexities associated with the knowledge discourse. It attempts to shift the focus from the contemporary discourse of knowledge management which views knowledge as a disposition or an object towards the role of situated action which constitutes knowing in practice. This is achieved by investigating how actual teams in agile projects generate knowledgeability.

A further contribution is a closer examination of the emerging practice of software development, in particular, how systems design and development work takes place in teams in agile projects. I consider the agile setting to be particularly appropriate for the empirical study as it encapsulates how more and more teams are working today. The agile view of software development is people-centric and emphasizes on interactions between team members to sustain knowledgeability and mutual learning.
1.5 Scope and Limitations

In terms of the epistemology of knowledge, this study does not view explicit and tacit knowledge as separate constructs. In subscribing to Polanyi’s view, this study recognizes that there exists ‘tacitness’ in all knowledge, albeit in varying degrees. This perspective has similarly been influenced by Kurti (2011) who recognizes that tacit knowledge is embedded in action and cannot be detached from the knower. Fundamentally, this study adopts Orlikowski’s perspective that tacit knowledge is a form of knowing that is rooted in action and recurrently produced through the daily activities of people. Similar to Orlikowski, this study uses the terms knowledgeability, “knowing how” and “knowing in practice” interchangeably.

The rationale for choosing to examine agile methodologies is predominantly influenced by the increasing popularity and relevance of how software development is practiced today. There are several agile methodologies such as lean software development, extreme programming, test driven development and more. Nonetheless, it is not within the scope of the study to compare and assess the different agile methods used between the companies. Only Scrum and pair programming were found to be used by the organizations in this study. Notwithstanding this, these two methodologies – particularly Scrum – are widely practiced by many companies and constitute an important part of the agile movement.

The research setting spans across two small companies which work with software development in Gothenburg, Sweden. The main focus is on co-located teams. Even though one of the companies has offices in various countries, the scope of the study is limited to the Gothenburg office. Nevertheless, it is hoped that the findings from this thesis will be insightful and relevant to other studies in similar settings and contribute to the knowledge within the area of knowledgeability.
1.6 Disposition of the Thesis

The thesis comprises six chapters and it is structured as follows:

- **Chapter 1: Introduction**
  This chapter presents the background and problem area, justification for the topic and the research question. It further discusses the contributions, scope and limitations of the study.

- **Chapter 2: Theoretical Framework**
  This chapter introduces the concepts and theories which are related to the study. The conceptual model which was developed from the literature review is also presented.

- **Chapter 3: Methodology**
  This chapter provides an overview of the research design, methodological approach and philosophical underpinning of the study. It includes the measures taken to ensure the trustworthiness and ethical considerations in the study.

- **Chapter 4: Analysis and Results**
  This chapter presents the analysis of the empirical findings.

- **Chapter 5: Discussion**
  This chapter discusses the results from the study and the implications of the findings for organizations and knowledge management.

- **Chapter 6: Conclusion**
  This chapter concludes with the reflections and recommendations for future research.
2. Theoretical Framework

This chapter presents the concepts and theories which are related to the study. It begins with a discussion of the perspectives on knowledge and a perspective on “knowing in practice”. This is followed by an overview of knowledge management and agile methodologies. Lastly, a conceptual model which was constructed based on the literature review is presented.

2.1 Perspectives on Knowledge

Knowledge is a complex and multi-faceted construct with varying definitions. The classifications of knowledge have been around since the Greek era where philosophers such as Aristotle categorized knowledge into *episteme*, *techne* and *phronesis*. *Episteme* is context-independent and objective knowledge. *Techne* refers to the practical skill necessary for action whereas *phronesis* is context-sensitive practical wisdom based on the ability to decide upon and undertake the best set of actions in a specific situation (Nonaka et al., 2008).

Stenmark (2001) brings to our attention two traditions of knowledge: the commodity view and the community view. In the positivist commodity view of knowledge, knowledge is defined as an absolute and universal truth (Stenmark, 2001). Knowledge is considered to reflect and represent the world as it is (Baets, 2005). The positivist view further sees knowledge as a separate construct from its owner and as an entity which exists independent of the individual who uses and learns it (Stenmark, 2001; Baets, 2005).

Authors such as Polanyi (1962) and Berger and Luckmann (1966) refute the positivist view of knowledge and argue that knowledge cannot be understood in isolation as it is inherently socially constructed. Similarly, Nonaka and Takeuchi (1995) consider knowledge as a flow of information found in the beliefs and commitments of the owner. In the community perspective, a more constructivist approach underlies the understanding of knowledge; it recognizes that knowledge cannot be separated from the activities and interactions between individuals (Stenmark, 2001).

Another domain of knowledge which is often thrown into contention is the epistemology of knowledge, specifically, understanding what constitutes tacit and explicit knowledge. Explicit knowledge is defined as knowledge which can be transmitted in systematic language and become codified (Nonaka, 1994). On the other hand, tacit knowledge refers to knowledge that cannot be easily articulated and it only exists in the hands and minds of people; often displayed through actions and demonstration (Nonaka 1994; Stenmark, 2000). Knowledge management literature commonly cites explicit and tacit knowledge as separate constructs while Polanyi (1958) argues that tacit and explicit knowledge are mutually constituted and should not be treated as two separate types of knowledge. The following sections will discuss the various conceptualizations of knowledge in greater detail.
2.1.1 Data, Information and Knowledge

The data-information-knowledge paradigm is often used to explain the relationships between data, information and knowledge and how they are transformed. Bouthilier and Shearer (2002), point out that there remains confusion over the concepts of information and knowledge, and the distinction between the related concepts is essential in order to discern the management of information from the management of knowledge.

According to the traditional hierarchical view of data-information-knowledge, data is seen as a pre-requisite for information to be produced and information is a prerequisite for knowledge to surface (Tuomi, 1999).

I. Data can be defined as isolated facts (Davenport and Prusak, 1998; Tuomi, 1999). It is commonly seen as structured records which are not able to explain or provide insights as to why a particular event has occurred or may occur (Davenport and Prusak, 1998).

II. Information can be understood as data endowed with meaning (Davenport and Prusak, 1998; Jashapara, 2011) and value through contextualizing, categorizing, calculating, correcting and condensing data (Davenport and Prusak, 1998).

III. Knowledge can be defined as information which is put into context (Tuomi, 1999) and subsequently used for predictions and to guide action (Nonaka and Takeuchi, 1995; Bouthilier and Shearer, 2002). According to Davenport and Prusak (1998), knowledge comprises framed experiences, values, contextual information, and expert insight which seek to evaluate information.

It is implied that data by itself is of little relevance and value and that knowledge is more valuable than data or information as it is closer to action-taking and decision-making (Davenport and Prusak, 1998).

Notwithstanding the popularity of the traditional hierarchy, Tuomi (1999) argues that raw data does not exist. This is because facts are created from a person’s knowledge and data will only surface if a meaning structure is first made present and used to represent information. Therefore, the traditional knowledge hierarchy should be reversed as knowledge is an essential prerequisite to information and data. An example given is that in order to store information in computer databases, knowledge has to be first decontextualized and structured according to predefined and independent data entities. The computer, as a tool, does not accord meaning to the content that is being processed. On the contrary, it is the computer programmers who engage in cognitive thinking and design meaning, and constantly negotiate meaning in order to make automatic processing possible (Tuomi, 1999). In order for meaning to become verbal and textual such as in the context of information, it must be articulated in a linguistic and conceptual context. Data is produced when entities which have no meaning are extracted from information and assigned for automatic processing (Tuomi, 1999).

The traditional knowledge hierarchy assumes that data becomes information when meaning is added. However, Tuomi (1999) argues that data is created when information is situated in a
predefined data structure that defines its meaning. This defies the conventional notion that data is a raw input for information; rather, data is produced by adding value to information and making it possible for it to be automatically processed. In the reversed view of the knowledge hierarchy illustrated in Figure 2.1, data only materializes after information is produced and information emerges only after knowledge is existent.

![Figure 2.1: The Reversed Hierarchy (Adapted from Tuomi, 1999, p.9)](image)

### 2.1.2 Tacit Knowledge

Tacit knowledge is an area which has not only been inadequately addressed, but to a large extent been misconstrued as well (Tsoukas, 2002; Grant, 2007). This makes it imperative to discuss the foundations of tacit knowledge and how it is perceived by different authors.

Tacit knowledge was first coined by the philosopher Michael Polanyi as a response to the overwhelming positivist literature which views knowledge as an objective construct. He rejects the notion that knowledge is objective and can always be articulated. Polanyi (1966) argues that tacit knowledge is central to knowing and that we cannot isolate explicit knowledge from an individual’s tacit knowing. In Polanyi’s view, there exists an inherent underlying ‘tacitness’ in knowledge whereby tacit and explicit knowledge are mutually constituted and must not be treated as two separate types of knowledge.

A key contribution by Polanyi (1958) is the discussion of two dimensions of awareness, namely focal awareness and subsidiary awareness. Polanyi (1958) uses the example of hammering a nail to explain how we attend to these tools in different ways. For the person handling the hammer, the main focus is on driving the nail in. The person is focally aware of his action. However, the person is at the same time as aware of the feeling in the palm of the hand while holding the hammer. This refers to subsidiary awareness. While we are driving a nail in, we are concurrently merging our focal and subsidiary awareness. We know the feelings in the palm by using them to attend to the hammer which is hitting the nail. A skillful
individual (master) is thus able to focus on the main objective while the less skilled individual (novice) has to focus on the proximate device, which in this case is the hammer.

Polanyi (1958) elaborates that if a pianist were to alter his attention from playing the piano (focal awareness) to what he is doing with his fingers (subsidiary awareness), he will be confused. Instead of focusing on the tool as the main object, we should see it as an instrument which we place our subsidiary awareness on. As we are constantly relying on our subsidiary awareness to attend to something else, we have come to know something without being able to identify it (Tsoukas, 2002). It is in this sense that we understand what Polanyi (1966, p.4) means by “we know more than we can tell”.

Another important notion underscored by Polanyi is indwelling. Polanyi (1958) emphasizes the importance of assimilating and indwelling in the tool we use – essentially, to embody the tool - in order to be able to use it competently. Indwelling opposes the traditional dichotomies which assume that there is a distinction between constructs such as the mind and body, subject and object, and the knower and the known (Nonaka and Takeuchi, 1995). Polanyi (1966) relates to the master and apprenticeship relationship and explains that meaning is only discovered when the apprentice is able to achieve the same kind of indwelling as the master. Until then, the apprentice is reliant on the meaning which rests with the master. Tacit knowledge can thus be learnt through the processes of imitation and observation.

Grant (2007) demonstrates his understanding of Polanyi’s view by illustrating a continuum of knowledge (Figure 2.2). At one end of the spectrum is the presence of little ‘tacitness’ in knowledge which subsequently progresses to a situation where knowledge is shared among experts with similar backgrounds and experiences. This then progresses to a situation where there exists a strong personal component of knowledge which makes it difficult to express. The end of the spectrum to the left is characterized by an extremely high level of ‘tacitness’ which makes it impossible to articulate the knowledge.

Figure 2.2: The Tacit/Explicit Dimension (Adapted from Grant, 2007, p.177)
Grant (2007) claims that the level of ‘tacitness’ depends on the use of language. For example, the knowledge is more explicit when there is a high degree of acceptance and specificity in the language used. In contrast, knowledge becomes more tacit when experience and complex levels of language are necessary in order to derive meaning.

There are several other authors who share Polanyi’s view of tacit knowledge. Subashini (2010) underscores the importance of tacit knowledge and argues that explicit knowledge loses its meaning when there is a lack of insight to tacit knowledge. On a similar note, Swan and Newell (2000) point out that tacit knowledge is highly situated as it is embedded in personal beliefs, values and experiences. Furthermore, Tsoukas (2002) argues that tacit knowledge loses its meaning and value when it is taken out of its context of application.

2.1.3 Explicit Knowledge

Contrary to the view that tacit knowledge can easily be converted into explicit knowledge, Tuomi (1999) argues that it is a shift in meaning which results in some parts of knowledge being made explicit (focal) with the rest remaining in the periphery as tacit knowledge. The shift in meaning occurs through the process of de-contextualization and explicit knowledge could be understood as decontextualized information. Tuomi (1999) elaborates that sense making depends on the sense maker’s level of tacit knowledge, and requires that the original articulator and the sense maker have corresponding meaning structures.

Stenmark (2001) argues that “explicit knowledge” in the form of documents, records and software code, albeit useful, is not considered as knowledge. These entities are essentially information even though they may require knowledge to create and interpret them. Although it is possible for some of our tacit knowledge to be articulated into information, tacit knowledge would have to depend on the traditions and experiences shared by individuals in order to be comprehensible. According to Alavi and Leidner (2001, p.109 cited in Stenmark, 2001), “information is converted to knowledge once it is processed in the minds of individuals and knowledge becomes information once it is articulated”. This understanding is illustrated in Figure 2.3.

![Figure 2.3: Information and Knowledge Conversions (Adapted from Stenmark, 2001, p.7)](image-url)
2.2 A Perspective on Knowing in Practice

Schön (1983) emphasizes that our knowing is in our action and suggests that human beings are reflexive and constantly aware of the social and physical contexts in which they carry out their activities. He claims that our knowing is typically tacit and that much of the professional working life depends on the tacit knowing that is in action.

By introducing “tacit knowing in action”, Schön (1983) implies that there are actions, recognitions and judgments which we are able to perform spontaneously without having to consciously think about them. He argues that people are unaware that they have learnt to do specific things; rather, they are simply able to do them. Likewise, Polanyi (1958) states that we are not always able to explain or articulate the knowing which our action reveals. In a study on system designers, Stolterman (1992) showed that the designers had difficulties explaining the concepts behind good quality in systems design. Russo and Stolterman, (2000) claim that certain aspects related to skill performance cannot be framed nor made explicit by practitioners. Although designers are able to differentiate between a good or bad result, they may not be able to account for the effect of a particular design practice on the outcome.

Suchman (2007) discusses how our knowledge and actions are related to particular circumstances where knowing and action occur. She defines situated action as a course of action which is dependent on its material and social circumstances, and describes plans as being rooted in the situated activity and surrounding circumstances. Although plans may function as input for situated action, they do not determine its course. Rather, it is the embodied skills (knowing) which come to the fore. Suchman (2007) explains that despite the amount of effort taken to plan how to run a series of rapids in a canoe, the eventual action taken to get the canoe through the water is based on the person’s embodied skills.

Ryle (1949) demonstrates the difference between “knowing how” and “knowing that”. “Knowing how” derives meaning from activity and it is related to the ability of a person to perform tasks. “Knowing that” refers to knowing what is required to put something into practice, such as knowing a set of rules in a game. Ryle (1949) describes how a boy’s “knowing how” of playing chess can be seen through his actions. Even though the boy may not be able to recite the rules of the game, he is still able to play the game through his actions.

Polanyi (1958) uses Ryle’s distinction between “knowing that” and “knowing how” and suggests that each aspect of knowing is not distinctive and is ever present with the other. He uses the example of riding a bicycle. The action in staying upright is a part of knowing how to ride the bicycle. However, people often find difficulties explaining the “knowing that” which is keeping them upright.

Orlikowski (2002) argues that we recognize the “knowing how” (the ability to play chess or ride a bike) by observing the practice (chess playing or bicycle riding). Fundamentally, the practice derives meaning from the “knowing how” that constitutes it. If we were to separate the “knowing how” of playing chess from the practice, the chess playing practice will no longer be recognizable. Therefore, “knowing how” and practice are mutually constitutive.
In essence, “knowing how” is a capacity to perform or act in particular circumstances and we recognize the “knowing how” by adopting the practice (Orlikowski, 2002; Cook and Brown, 1999). The focus on practice acknowledges that social life is continually generated through the recurring actions of individuals (Feldman and Orlikowski, 2011). As Orlikowski (2002) discusses, knowledgeability or knowing in practice is an ongoing social achievement which is constituted and reconstituted in everyday practice. Knowledgeability is hence reconstituted in different contexts and over time as people continually engage in social practices, and regenerate the knowing which is produced in those practices.

2.3 Knowledge Management

Studies related to knowledge are often subsumed in the area of knowledge management. Knowledge management - which came into prominence in the 1990s - is a relatively new theoretical field of study (Stenmark, 2001; McInerney, 2002; Bouthilier and Shearer, 2002; Grant; 2007; Grossman, 2007). Over the years, knowledge management has evolved and diffused across various disciplines (Dalkir, 2005) such as information systems, cognitive science, communication, organizational science and several others (McInerney, 2002).

Swan et al. (1999) present two dominant categories of knowledge management: the cognitive network view and the community networking view. The cognitive model focuses on converting information into actionable knowledge through the use of extensive technology (Swan et al., 1999); it has particular interest in the reproduction of knowledge (Swan and Newell, 2000). In contrast, the community view is concerned with how dialogue is created in active networking (Swan et al., 1999); it focuses on tacit knowledge and the sharing of knowledge by influencing the workplace environment (Alvesson and Kärreman, 2000).

Knowledge management has likewise been extended to organizational effectiveness where focus is placed on using information systems to manage organizational knowledge (Alavi and Leidner, 2001). Companies rely on business processes, information technologies, knowledge repositories and individuals to acquire, store and re-use knowledge (Eschenfelder et al., 1998 in Kakabadse, Kakabadse and Kouzmin, 2003). As such, they have to constantly create new knowledge and disseminate it throughout the organization in order to improve productivity and attain competitive advantage (Nonaka, 1995; Kakabadse, Kakabadse and Kouzmin, 2003).

Several authors (Choo, 1998; Alvesson and Kärreman, 2001) claim that information management focuses on recording information systematically while knowledge management is intended to place more emphasis on people management. Notwithstanding this, the majority of knowledge management studies still tend to lend a heavy emphasis on capturing, converting and disseminating knowledge through the use of IT (Swan, Scarbrough and Preston, 1999; Stenmark, 2000; Mingers, 2001; Alvesson and Kärreman, 2001).

This overall IT-laden perspective of knowledge management has been criticized by several authors (Stenmark, 2000; Swan, Scarbrough and Preston 1999; McInerney, 2002) who argue that there is too much focus on the codification of explicit knowledge through IT systems,
resulting in people related issues being sidelined. Moreover, this view assumes that knowledge workers are passive recipients of information (Fischer and Ostwald, 2001). McInerney (2002) argues that even as technology may be useful, the active nature of knowledge requires human intervention in order for knowledge management to be successful.

2.3.1 Knowledge Creation – SECI Model

While there are numerous processes and theories, the SECI (Socialization, Externalization, Combination and Internationalization) model remains one of the most frequently cited theoretical frameworks in knowledge management literature. This warrants the need to discuss the underlying underpinnings of the model and how knowledge is conceptualized.

Nonaka and Takeuchi (1995) argue that organizations do not process information from their external environment but create new information and knowledge internally to solve problems and to adjust to the changing environment. The ontological position of this model purports that while knowledge creation begins with the individual, the purpose of knowledge creation is to make personal knowledge available to the others in the company. The epistemological stance is that knowledge can be defined as tacit and explicit knowledge.

There are four knowledge conversion processes: socialization, externalization, combination, and internalization. In each of these processes, tacit knowledge is either converted to tacit or explicit knowledge and vice versa. The SECI model is illustrated in Figure 2.4.

![SECI Model](image)

**Figure 2.4:** SECI Model (Adapted from Nonaka and Takeuchi, 1995, p.45)

**Socialization** involves the conversation of tacit to tacit knowledge by the social interaction between individuals. Experience is a critical aspect in tacit knowledge acquisition as shared experience helps individuals to better understand one another’s thinking process (Nonaka, 1994). According to Nonaka and Takeuchi (1995), it is not necessary for language to be present in order for tacit-tacit knowledge conversion to take place. In the case of
apprenticeship, apprentices learn from their masters through observing and imitating their masters when they are practicing their craft.

**Externalization** converts tacit knowledge to explicit knowledge. This conversion occurs through the use of metaphors, concepts, hypotheses, diagrams and models. When articulating tacit knowledge through writing, these expressions may not necessarily suffice. However, the gap between images and expressions provides an opportunity for reflection and interaction (Nonaka and Takeuchi, 1995).

**Combination** engages social processes to combine different bodies of explicit knowledge, with the aim of making it more systematic. The social exchange of explicit knowledge takes place through documents, meetings or computer-mediated communications. This is followed by sorting, including and decontextualizing existing explicit knowledge to create new knowledge. Computer systems are usually used to support this process (Nonaka, 1994; Nonaka and Takeuchi, 1995).

**Internalization** which converts explicit knowledge into tacit knowledge is closely related to “learning by doing”. According to Nonaka and Takeuchi (1995), internalization occurs in the form of shared mental models or technical knowhow. Techniques used in this process include documentation or storytelling which helps individuals to internalize what they experience and hence enrich tacit knowledge. Documents or manuals enable the transfer of explicit knowledge to other people which in turn help them to attain direct experience of what others are experiencing. Internalization is not dependent on people re-experiencing the experiences of others. One could appreciate the realism of a success story simply by reading about it. The experience of the past may convert into a tacit mental mode and as a result become part of the organizational culture when it is embodied by members of the organization (Nonaka and Takeuchi, 1995).

### 2.3.1.1 Ba – Shared Spaces

Another key aspect of the knowledge conversion model is the presence of *ba*, defined as a shared space for the sharing of individual and collective knowledge (Nonaka and Konnon, 1998). According to Nonaka and Takeuchi (1995), information resides in media and networks while knowledge resides in *ba*. As such, knowledge becomes information when it is separated from *ba*. The successful support of knowledge creation hinges on the different *bas* which drive the continuous and cyclical knowledge conversion from tacit to explicit knowledge and vice versa (Nonaka and Konno, 1998).

The socialization phase is characterized by the *originating ba*. Face-to-face experiences are an important aspect of the conversion of tacit to tacit knowledge and they are supported by open organizational design and customer interfaces (Nonaka and Konno, 1998).

*Dialoguing ba* or *interacting ba* as termed by Nonaka and Konno (1998) supports tacit-explicit knowledge conversion in the externalization phase. It places heavy emphasis on dialogue and the use of metaphors to create joint meaning. In this aspect, dialogue helps to
convert individuals’ skills and mental models into common terms and concepts. Individuals not only share mental models, but actively reflect on their own mental models (Nonaka and Konno, 1998).

*Systematizing ba or cyber ba* is the place where combination happens (Nonaka and Takeuchi, 1995; Nonaka and Konno, 1998). It is a virtual world rather than a space transcending real time. In this *ba*, information systems are heavily engaged to support the combination of explicit knowledge. Examples include the use of online networks, groupware, documents and databases (Nonaka and Konno, 1998).

Internalization takes place in the *exercising ba* and it is achieved through learning by self-refinement, on the job training, peripheral and active participation. The conversion of explicit to tacit knowledge is constantly heightened through the use of explicit knowledge in real life or simulated situations (Nonaka and Konno, 1998).

### 2.3.1.2 Review of the SECI Model

This section highlights the key arguments of the SECI model and presents the other spectrum which highlights why the assumptions underlying the SECI model may be flawed.

A central tenet of the knowledge creation model is the continuous dialogue between tacit and explicit knowledge, and the “mobilization and conversion of tacit knowledge” (Nonaka and Takeuchi, 1995, p.56). Nonaka (1994) highlights that failing to sustain a dialogue between tacit and explicit knowledge could become problematic as combination and socialization alone could limit the boundaries of knowledge creation. Nonaka and Takeuchi (1995) stress that unless shared knowledge becomes explicit, it will have little advantage to the wider organization. In addition, Nonaka (1994) argues that while tacit knowledge held by individuals may be the core of knowledge creation, the benefits of tacit knowledge are realized through externalization and the dynamic interactions between the four modes of knowledge conversion.

The assumptions underlying knowledge creation and conversion through the SECI model is a stark contrast to Polanyi’s view of tacit knowledge in two significant ways. Firstly, it assumes that tacit knowledge can easily be made explicit. Secondly, it deliberately seeks to remove the ‘tacitness’ in knowledge, which consequently dilutes the social aspect of knowledge.

Tsoukas (2002) points out that it is flawed to assume that tacit knowledge can be captured, translated or converted as it is essentially displayed and manifested in our actions. According to Tsoukas (2001, cited in Tsoukas 2002, p.16), “new knowledge comes about not when the tacit becomes explicit, but when our skilled performance – our praxis - is punctuated in new ways through social interaction”.

Swan and Newell (2000) argue that since tacit knowledge is personal, highly situated and difficult or impossible to articulate, removing tacit knowledge from its context will deplete it of its intrinsic meaning and value. The intentional detachment of tacit knowledge by codifying it into explicit knowledge as propagated by the SECI model fundamentally
contradicts Polanyi’s view which emphasizes there is an inherent ‘tacitness’ in all knowledge. Polanyi (1966, p.20) states that if we suppose that tacit thought forms an integral aspect of all knowledge, “eliminating all personal elements of knowledge would, in effect, aim at the destruction of all knowledge”.

2.4 Agile Software Development

As software development continues to evolve rapidly, agile methods have come to the fore as alternatives to develop software more efficiently and effectively (Lindstrom and Jeffries, 2004). Nerur and Balijepally (2007) point out that emerging practices such as agile development run counter to traditional design practices which assume that changes can be controlled by systematic and rigid procedures. The focus has shifted from that of optimization to flexibility and responsiveness.

Agile methods are people-focused and recognize the value that people and their relationships bring to software development (Nerur and Balijepally, 2007; Lindstrom and Jeffries, 2004). Communication and collaboration between team members are crucial for agile methodologies to be successful. Programmers no longer find themselves working with a homogenous team but are exposed to aspects of shared learning, pair programming, reflection workshops and collaborative decision making (Nerur, Mahapatra and Mangalara, 2005). Instead of big up front designs, agile methods are characterized by short iterative cycles (of planning, action and reflection), intense collaboration, self-organizing teams, and a high degree of developer discretion (Nerur and Balijepally, 2007).

In essence, the Agile Manifesto (Agile Alliance, 2014) values:

i. Individuals and interactions over processes and tools
ii. Working software over comprehensive documentation
iii. Customer collaboration over contract negotiation
iv. Responding to change over following a plan

The agile perspective or emergent metaphor of design as highlighted by Nerur and Balijepally (2007) acknowledges the importance of learning through experimentation. Furthermore, it recognizes that knowing and doing are interwoven, and identifies the need to reconcile multiple worldviews. Table 2.1 below summarizes the difference between the traditional and agile perspective of software development.

Traditional software development approaches take the form of the Tayloristic model which organizes knowledge sharing according to specific roles. This involves the creation of documents to sign off from one stage of development to the other. However, documents are mainly written for the people who are working on the next stage of software development in mind and the information often gets lost in the long transfer chain (Chau and Maurer, 2004).

The reliance on heavy documentation is replaced by lean thinking and heightened face-to-face interaction in agile software development (Ryan and O’Connor, 2013; Chau, Maurer and
Melnik, 2003; Nerur, Mahapatra and Mangalara, 2005). Sharp and Robinson (2010, p.62) state that “communication in agile development is both crucial but also tacit, informal and predominantly verbal”. Much of the knowledge in agile development can be said to be tacit, in which expert knowledge is likely to reside in the heads of the development team members (Nerur, Mahapatra and Mangalara, 2005; Ryan and O’Connor, 2013).

To ensure continuous collaboration and knowledge sharing, agile methodologies rely on informal communication, task overlapping and interdisciplinary teams (Maalej and Happel, 2007). Daily Scrum meetings, pair programming or test driven development are some of the common practices in agile software development. Teams informally share experiences such as the problems encountered and argumentation behind specific decisions. This allows team members to coordinate their work and learn from each other (Maalej and Happel, 2007).

Table 2.1: Traditional and Agile Perspective of Software Development (Cited in Nerur and Balijepally, 2007, p.82)

<table>
<thead>
<tr>
<th></th>
<th>Traditional View of Design</th>
<th>Emergent Metaphor of Design</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Design Process</strong></td>
<td>Deliberate and formal, linear sequence of steps, separate formulation and implementation, rule-driven</td>
<td>Emergent, iterative and exploratory, knowing and action inseperable, beyond formal rules</td>
</tr>
<tr>
<td><strong>Goal</strong></td>
<td>Optimization</td>
<td>Adaptation, flexibility, responsiveness</td>
</tr>
<tr>
<td><strong>Problem-solving</strong></td>
<td>Selection of best means to accomplish a given end through well-planned, formalized activities</td>
<td>Learning through experimentation and introspection, constantly reframing the problem and its solution</td>
</tr>
<tr>
<td><strong>View of the</strong></td>
<td>Stable, predictable</td>
<td>Turbulent, difficult to predict</td>
</tr>
<tr>
<td><strong>environment</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Type of learning</strong></td>
<td>Single-loop/ adaptive</td>
<td>Double-loop/ generative</td>
</tr>
<tr>
<td><strong>Key characteristics</strong></td>
<td>Control and direction&lt;br&gt; Avoids conflict&lt;br&gt; Formalizes innovation&lt;br&gt; Manager is controller&lt;br&gt; Design precedes implementation</td>
<td>Collaboration and communication – integrates weltanschauungs, or worldviews&lt;br&gt; Embraces conflict and dialectics&lt;br&gt; Encourages exploration and creativity and is opportunistic&lt;br&gt; Manager is facilitator&lt;br&gt; Design and implementation are inseparable and evolve iteratively</td>
</tr>
<tr>
<td><strong>Rationality</strong></td>
<td>Technical/functional</td>
<td>Substantial</td>
</tr>
<tr>
<td><strong>Theoretical and/or philosophical roots</strong></td>
<td>Logical positivism, scientific method</td>
<td>Action learning theory, Dewey’s pragmatism, phenomenology</td>
</tr>
</tbody>
</table>

2.4.1 Scrum

In discussing about agile methodologies, it is important to give attention to one of the most popular agile methodologies used today, Scrum. Some of the key tenets of Scrum include
teamwork, frequent interaction and continuous improvement (Scrum Alliance, 2014). Scrum projects do not have an up-front design phase and depend on repeated cycles of sprints to develop software, and to deliver products iteratively and incrementally. Key meetings include sprint planning, daily scrum meetings, sprint reviews, and sprint retrospectives (Scrum Alliance, 2014).

During a daily Scrum meeting, team members provide an update on what they have done since the last meeting, what they plan to do for the day and the problems which they have encountered. Apart from creating frequent team communication, the daily meetings provide clarity on the progress of work done by each member and who the members should speak to when they are working on parts of the system which require other team members’ expertise (Chau and Maurer, 2004). Continuous learning is provided in the form of retrospective meetings, where team members gather to discuss the success factors and obstacles which they face during the development process (Chau and Maurer, 2004).

### 2.4.2 Pair Programming

Another agile practice is pair programming which involves two developers sitting side by side designing, coding and testing the software together. It is a social process where informal and spontaneous communication takes place between two developers (Chau and Maurer, 2004). The objective is to produce better code and quality by having another pair of eyes to review the production code (Lindstrom and Jeffries, 2004). According to Chau and Maurer (2004), some explicit but mostly tacit knowledge in the form of task-related knowledge, contextual knowledge and social resources is shared between the pair during pair programming.

Task-related knowledge refers to system knowledge, coding convention, design practices, technology and tool usage tricks. Contextual knowledge, on the other hand, is knowledge by which facts are interpreted and used. They include knowing from past experience or knowing for example, the appropriateness of using particular design patterns in different coding scenarios. Social resources comprise contacts and referrals. Chau and Maurer (2004) add that such knowledge is often not documented. Instead, it is uncovered through informal and casual conversations. In this sense, pair programming helps to create and strengthen the dynamics, trust and reciprocity in the team.

### 2.5 Knowledgeability in Everyday Practices

To understand how knowledgeability is generated and sustained, it is essential to examine the practices which are enacted. Practices can be defined as the actions taken to get work done (Orlikowski, 2002; Brown and Duguid, 2001). This section discusses some of the possible practices which have been informed by current literature. While several of the practices stem from the knowledge management literature, they are not assumed to be applied in the same way as in examining knowledgeability in organizations.
2.5.1 Sharing “knowing how”

Several authors (Haldin-Herrgard, 2000; Choo, 2000; Busch, 2008) assert that as tacit knowledge is experiential and contextual, it cannot be taught, trained or educated; rather, it can only be learnt. Orlikowski (2002) argues that the “knowing how” that is found in practice cannot be understood as being “stuck” in or to that practice. She argues that as “knowing how” and practices are mutually constitutive, sharing “knowing how” cannot be compared to the problem of knowledge transfer. According to Orlikowski (2002, p.271), “sharing “knowing how” can be seen as a process of enabling others to learn the practice that entails the “knowing how”. It is a process of helping others develop the ability to enact in a variety of contexts and conditions – the knowing in practice”.

The sharing of “knowing how” could occur in the form of informal teaching or mentorship where the mentor acts as a role model, provides feedback and encourages the novice to learn through hands-on experiences (Swap et al., 2001). Experimenting is encouraged by helping to provide the environment for experimentation to occur (Swap et al., 2001). Experience not only provides the historical background for understanding, it also influences how one perceives a new situation. Experience provides the familiarity and ability for one to be able to make connections between past events and current situations (Davenport and Prusak, 1998).

Individuals who share experiences are more likely to be in tuned with one another’s thinking process (Nonaka, 1994). Sveiby (1997) implies that tacit knowledge is contextual and produced through years of experience. Nonetheless, Wagner and Stenberg (1985) assert that one does not automatically acquire tacit knowledge through the years. More importantly, it is that which is learnt from the experience and how tacit knowledge is applied which is of value. Drawing from the master-apprentice relationship, Sveiby (1997) points out that the apprentice is unlikely to attain tacit knowledge if he or she does not understand the subtle skills being passed on from the master. Busch (2008) cites an example whereby the description of underwater swimming as a weightless gliding sensation will not be relevant to a person who has neither experienced flight nor swimming.

According to Swap et al. (2001), a novice often lacks the knowledge and experience to assimilate the mentor’s instructions because the ability to attain knowledge from information depends on the shared contexts and meaning between the learner and the mentor. Initially, the novice will appear out of the “zone of proximal development” (Vygotsky, 1978 cited in Swap et al., 2001). As the novice gradually gains experience through the mentor’s guidance, the “zone of proximal development” will expand and the novice will be able to take in more advanced instructions over time. In this instance, the novice can only acquire the tacit component if he or she starts to share similar experiences as the mentor.

2.5.2 Engaging in Communities of Practice

The notion of “community of practice” has in more recent times provided useful perspectives on knowing and learning. According to Wenger and Snyder (2000), communities of practice are groups of people with shared expertise or who are bound by a passion for something they
People in communities of practice share information, their experiences and knowledge in creative ways and in turn learn how to do things better as they interact regularly. Wenger (1998) explains that this form of learning is social as active participants engage in the practices of the communities and construct identities in relation to the communities. Such participation not only shapes what we do but who we are and how we interpret what we do. Wenger and Snyder (2000) suggest that communities of practice, formal work groups, project teams and informal networks are useful in complementary ways.

Wenger (1998) elaborates that meaning is created through participation and active involvement in some practice. Practices pertain to active involvement in social communities while reification means making complex and messy practices concise so that they can be easily shared with the community. Reification is achieved from manifesting the experiences of the participants and could take the form of procedures, instruments and language. Reification and practice cannot be understood in isolation. The duality of participation and reification is illustrated in Figure 2.5.

![Figure 2.5: The Duality of Participation and Reification (Adapted from Wenger, 1998, p.63)](image)

According to Lave and Wenger (1991), learning is situated in practice. Meaningful learning is entrenched in the relevant social and physical contexts and involves the novice observing the community of practice (Brown, Collins and Duguid, 1989). As Lave and Wenger (1991) explain, members who are new to the community would often begin with the role of peripheral participants and perform activities under the guidance of more experienced members. This is known as observation from the boundary or legitimate peripheral participation. However, as the peripheral members gradually gain experience and become more involved in the culture of the group, they progress from the role of an observer to a fully functioning participant. Situated learning takes place as members constantly adjust to each other’s needs. As a result, tacit knowledge is created and transferred as members work and interact with one another (Robey, Khoo and Powers, 2000).
The essence of learning in legitimate peripheral participation is to become an “insider” (Brown and Duguid, 1991) where the learner understands the culture of the group and what being a member truly entails (Robey, Khoo and Powers 2000). Instead of focusing on formal and explicit knowledge, learners embed themselves in the community’s worldview and learn to communicate in the community’s language (Brown and Duguid, 1991).

2.5.3 Observing and Learning by doing

Tacit knowledge is learnt through the processes of experiencing and experimenting, where the individual develops the intuition to make the required judgments (Choo, 2000) and obtains learned behaviors and procedures informally (Howells, 2002). Stenmark (2001) states that expertise which is found as a quality in tacit knowledge is often observed and recognized through its resulting actions.

Furthermore, learning could take place by observing expert behavior. Dalkir (2005) sees expertise as a demonstration of the application of knowledge. He suggests that there are two types of expertise: skill or motor-based (example of riding a bicycle) or cognitive based (example of making a medical diagnosis). While it is not possible to observe a person’s knowledge, one could observe the behavior of the expert. Learning takes place through observing how the expert or mentor applies the knowledge in daily situations (Dalkir, 2005).

Busch (2008) who defines tacit knowledge as expertise, skill or know-how, asserts that tacit knowledge can only be transferred through “indwelling” or the individual learning the skill through “learning by doing” and “learning by using”. Experimenting can be seen as a form of learning by doing. Learning takes place when small groups brainstorm together, try out new ideas and evaluate the results (Dalkir, 2005).

2.5.4 Verbal communication

People interact and communicate with each other through language, and as a result, meaning is created during interaction (Renzl, 2007). Language plays an essential role for understanding the reality of daily life (Berger and Luckmann, 1966). Renzl (2007) argues that meaning is an integral aspect of knowing and it evolves through language. For that reason, it is through language that we are able to draw distinctions. For example, we have to first learn the organizational language to be able to understand organizational activities (Renzl, 2007).

Jashapara (2011) argues that meaning derived from language can be ambiguous as knowledge is context-dependent and influenced by changes in social activities in diverse environments. Suchman (2007) views language as a form of situated action. Yet, it is the ambiguous, metaphorical, indexical and context-dependent aspects of language which are essential for the construction of meaning (Alvesson and Kärreman, 2000; Suchman, 2007). Renzl (2007) claims that language is not only context; it provides context as well as a way to re-contextualize content. Language is essential for presenting the context which we are able to know and the words expressed through language and meaning influence how we perceive and
interpret the world around us. As such, words themselves do not represent reality but obtain meaning through their use.

Suchman (2007) further discusses the role of language in helping us to understand situations. The term “indexical” is used to explain the situated qualities of language and it is defined by expressions which depend on their situation for significance. Suchman (2007) points out that apart from more explicit expressions containing first- and second-person pronouns such as “here” and “now”, there exist broader expressions such as adjectives, for example, “nice”, which are situated and dependent on the context of use and shared understanding in order to be comprehensible. Language is hence a form of situated action where “expression and interpretation involve an active process of pointing to and searching the situation of talk” (Suchman, 2007, p.78).

Nonetheless, Polanyi also points out that we often know how to do things without either knowing or being able to explain to others why what we do works (Grant, 2007). Polanyi (1958) specifies that if we are unable to articulate all that we know then we can never quite know what is implied in what we say. Similarly, Nonaka and Takeuchi (1995) argue that language does not play an essential role in tacit-tacit knowledge conversion as apprentices often learn through observing and imitating how their masters practice their craft.

2.5.5 Using metaphors and analogies

Guzman and Wilson (2005, cited in Busch, 2008) assert that language itself is inadequate in capturing and transmitting knowledge. Busch (2008) argues that knowledge differs from information in the sense that knowledge contains a tacit component whereas information can be articulated in words. The use of metaphors and analogies hence become all the more imperative. Hey et al. (2008, p.283) define metaphor as “a figurative expression which interprets a thing or action through an implied comparison with something else; a symbol” and analogy as “illustration of an idea by means of another familiar idea that is similar or parallel to it in some significant features”.

The importance of metaphors and analogies can be seen in several instances. Nonaka and Takeuchi (1995) emphasize that writings are not sufficient for the externalization of tacit knowledge and further suggest the use of metaphors, concepts, hypotheses, diagrams and models. Hey et al. (2008) describe that metaphors can be used to frame and define a design problem by mapping the users’ understanding and reactions to a product; and eventually create meaning. For example, a shower may be perceived as a reset which is associated with washing away and starting anew. On the other hand, analogies can be used for concept generation and selection. In this case, a team seeking to design a device to fold laundry would create analogies which are associated to folding, such as paper folding or metal folding.

Busch (2008) highlights that metaphors are commonly used in the IS domain and familiarity and experience are crucial for the decoding of metaphors. He goes on to cite several examples. ‘Debugging’ is commonly used even though it does not literally mean the removal
of insects from a program. Similarly, the term ‘rebooting’ does not imply that one has to kick the machine in order for it to function. As Busch (2008) notes, a novice who does not have experience with computers is unlikely to understand the concept of debugging or be able to perform the task at hand. This underscores the importance of the role of the receiver in tacit knowledge communication.

2.5.6 Storytelling

Organizations which have traditionally been more interested in the explicit forms of knowledge are increasingly turning to narratives and storytelling to share knowledge and experiences, and to convey complex and multidimensional ideas (Sole and Wilson, 2002). Storytelling and narratives help to embed tacit knowledge socially through the connections of stories to ideas, lessons and best practices within a community setting (Jashapara, 2011). As stories provide a rich context, they help to retain information in the memory for a longer time and help to transmit valuable tacit knowledge (Dalkir, 2005). Moreover, it facilitates the exchange of highly contextual knowledge in problem solving (Sole and Wilson, 2002).

According to Sole and Wilson (2002), the purpose of a knowledge-sharing story is two-fold: to provide a simple yet focused experience so that the key points are understood, and to offer a real world account to the listener. For effective knowledge sharing to take place, Denning (2006) suggests that stories should focus on mistakes and provide details on the actions taken to rectify the situation. He is of the view that success stories often lack the details in explaining how things were performed and is as such less effective in communicating knowledge.

The use of storytelling in knowledge sharing, however, depends on the depth of the knowledge. While storytelling may be more appropriate for sharing simple tacit ideas, it may not be as effective in situations which require profound skills-based knowledge. Apprenticeship, mentoring and training would be more appropriate if one were to try to attain mastery (Swap et al., 2001; Sole and Wilson, 2002). Sole and Wilson (2002) suggest that a new surgeon is more likely to attain new skills by spending time working closely with experienced colleagues instead of listening to stories of success and mistakes.

2.6 Conceptual Model

This study subscribes to the perspective that knowledgeability or “knowing how” is constituted in the everyday practices of individuals. The conceptual model, as illustrated in Figure 2.6, comprises possible practices which have been informed by current literature in Chapter 2.5. As mentioned, several of the practices are commonly cited in the knowledge management literature. However, we do not expect them to be enacted in the same manner as discussed in knowledge management.

From the literature review, some of the practices appear to be interrelated. As such, the practices should not be viewed as completely exhaustive or exclusive categories. Central to
the conceptual model is the notion that “knowing” is ongoing and continually constituted and reconstituted as individuals engage themselves in the different practices.
3. Methodology

This chapter provides an overview of the research design and approach. The philosophical tradition underpinning the study is first presented. This is followed by a discussion on the methodological approach, a description of the procedures, the role of the researcher and how the data was analyzed. The chapter concludes with a discussion on how measures were taken to ensure the trustworthiness and ethical considerations in the study.

3.1 Philosophical Tradition

Questions of epistemology (the nature of knowledge), ontology (the nature of reality), and methodology (finding out what can be known) are critical aspects which shape and guide the course of any research (Creswell, 2009).

This study draws on the interpretivist approach which acknowledges that reality is socially constructed and therefore incapable of being understood independently of the social actors who construct and make sense of that reality (Orlikowski and Baroudi, 1991). Interpretivism is primarily hermeneutics and phenomenology, as interpretive studies seek to understand phenomena by examining the meanings which people have assigned to them (Boland, 1985).

The main philosophical approach for this study is hermeneutics. Hermeneutics is concerned with the interpretation of meaning (Gadamer, 1976); specifically how we come to understand a text and how we give meaning to the unfamiliar (Boland, 1991). Nonetheless, hermeneutics is not limited to the interpretation of written texts (discourses). It can be used to understand meaningful action with various degrees of complexities (practice) by interpreting actions as texts (Alvesson and Sköldberg, 2000). In this regard, hermeneutics complements the aim of the study as it seeks to attain a deeper understanding of the topic by interpreting the meanings of the participants’ perspectives, opinions and actions.

There exist several philosophical underpinnings of hermeneutics. According to Butler (1998), the conservative view seeks to disclose meanings of an action-text as intended by the author. The pragmatic or constructivist perspective assumes that meaning lies in the historical contexts of the interpreter and the interpreted while the critical perspective seeks to make interpretation emancipatory.

This study subscribes to the constructivist perspective, specifically to phenomenological hermeneutics conjoined by the philosophies of Gadamer and Heidegger. Heidegger provides an ontological description of ‘being’, in which he expounds on how humans experience and act on the world as part of ‘being-in-the world’ (Cohen and Omery, 1994; Butler, 1998). As humans interact with and relate to each other in different contexts and activities, a key part of being in the world is being with others (Mingers, 2001).
Butler (1998) acknowledges Heidegger’s view that the meaning of phenomenological description is rooted in interpretation, and hermeneutics provides the ontological insight into human interpretation and understanding. Phenomenology can be defined as “a way of study that respects the intentionality of actors, the symbolic nature of language, and the universal hermeneutic problem of understanding” (Boland, 1985, p.200).

Hermeneutics not only develops a process for understanding, but clarifies the conditions which the understanding occurs (Laverty, 2003). Gadamer (1976) states that understanding is episodic (every event is a moment itself) and trans-subjective (a constant connection and alteration between the past and present). A key tenet of hermeneutics, as pointed out by Gadamer (1975), is the ‘circle of understanding’ which focuses on the relations between the parts (component parts) and the whole (as constructed by the component phenomenon). Understanding is achieved through a cyclical process of relating each individual part to the whole where the parts and the whole cannot be understood independent of each other.

Gadamer extends the notion of the hermeneutic circle to explain how interpretation occurs as a result of a ‘fusion of horizons’ (Butler, 1998). Butler (1998) explains that the interpreter tends to possess a preconceived (prejudiced) idea of a phenomenon. When the interpreter seeks to understand each component phenomenon in relation to the whole, the interpreter is constantly going through the circle of understanding. This will gradually lead to the emergence of different perspectives and an eventual fusion of horizons between the interpreter and the phenomenon of interest. Therefore as researchers, we are constantly re-examining our pre-conceived ideas and allowing ourselves to be influenced by what we learn. A discussion of how the hermeneutic circle has been applied in this study can be found in Chapter 3.8. An illustration is presented in Figure 3.1 in the same chapter.

Gadamer (1976) asserts that there exists a dialectical interaction or co-constitutionality between the expectation of the interpreter and the meaning of the text. He adds that understanding as a fusion of horizons is a linguistic process in which language and the understanding of meaning are not isolated processes but essentially one and the same.

3.2 Frame of Reference

According to Walsham (1995; 2006), the interpretivist researcher is one who is personally and subjectively involved in the study. The researcher first enters the worldview of the participants, filters the interpretation according to the conceptual predispositions and then proceeds to send back a version of the events to those involved in the study.

Russo and Stolterman (2000) assert that all humans work from a particular point of reference or worldview. Researchers should be self-reflective and aware of how their preconceived views and prejudices could affect the scope and nature of the research. Husserlian hermeneutics advocates suspending judgments about our perceptions of the world by putting the world “in brackets” (Mingers, 2001, p.105). ‘Bracketing’ involves setting aside prior assumptions, with the aim of understanding the participants’ accounts with an open mind.
This study opposes the notion of ‘bracketing’ and subscribes to the views of Gadamer and Heidegger who argue that a researcher cannot separate description from the researcher’s interpretation (Dowling and Cooney, 2012). Heidegger states that our natural attitude or being-in-the-world cannot be expressed or consist of conscious beliefs, ideas and intension; rather, it is the sub-conscious attitude that has been socialized into us (Mingers, 2001). As prejudices are a result of how we orientate ourselves to a phenomenon, it is critical that we acknowledge our own biases so that the text can state its own truth against the prejudices (Gadamer, 1975).

Since this study draws on the hermeneutic tradition, it is important to discuss my background, experience and perspectives before I embarked on the study. I am currently an employee at Surikat, one of the companies in this study. I have been working in the IT sector for about three years after joining Surikat in 2011. My experience lies mainly in the areas of project management and process improvements for the software development team. As such, I am familiar with the general processes in software development projects and the agile environment. My interest in how software development teams work with knowledge grew at the time when I was doing a course in knowledge management at Linnaeus University. At the same time, Surikat had recently started using the Confluence wiki and was looking into ways to improve the structure of information to enhance “knowledge sharing”.

My preconceived ideas of knowledge management and knowledge were very much geared towards the notion that knowledge had to be made explicit in order for it to be useful. This preconception consequently started to change when I was writing my thesis proposal. My discussions with several lecturers brought me to literature which discussed other perspectives of knowledge other than the tacit-explicit knowledge dichotomy commonly found in current knowledge management literature. Kurti’s (2011) study on the factors which enable working with tacit knowledge effectively and her approach towards the topic was an inspiration for taking a closer examination of the tacit aspect of knowledge. As I started reading more about tacit knowledge and the concept of knowing in action, I learned to see knowledge as something which is fluid, contextual and embedded in the daily practices of people.

Choosing to examine how “knowing” is performed in the practices of teams working in agile projects subsequently became the primary interest of my thesis. The practices in Chapter 2.5 describe the preconceived ideas I had in relation to the study. The conceptual model points out the possible practices which I assumed could be enacted by teams working on agile projects. A further discussion on my prejudices is described in Chapter 3.7.

3.3 Methodological Approach

Quantitative and qualitative researches differ in their ontological and epistemological considerations. Bryman (2012) explains that the quantitative paradigm is positivist and assumes that all phenomena can be reduced to empirical indicators. Ontologically, the quantitative paradigm is objectivist which implies that reality exists independent of human perception. Epistemologically, the researcher and the participants are seen as separate entities
(Sale, Lohfeld and Brazil, 2002). In qualitative research, the epistemological position is interpretive as the researcher seeks to understand the world through investigating the worldview of the participants. Ontologically, the position is constructionist as social entities are seen as outcomes of interactions between individuals (Bryman, 2012).

The approach taken for this study is qualitative and in the form of an ethnographic research. The principal aim of ethnographic research is to understand the research setting in its natural context instead of in an experimental condition. Ethnography is interpretive. It is as Walsham (1995; 2006) describes to involve the researcher entering the worldview of the participants, making explanations and presenting the interpretations to the participants after.

According to Blomberg and Karasti (2012), understanding the world requires immersing oneself in the research setting and collecting information where activities concerning the topic of interest are occurring. Ethnography studies the daily settings of the participants as they encounter it and ethnographic research usually occurs over an extended period of time (Hughes et al., 1995). Other key principles of ethnography include holism which seeks to understand activities in relation to the larger setting and chain of events, in-depth descriptions of occurrences without assigning judgment to people’s activities, and understanding the participants’ practices through their points of view (Blomberg and Karasti, 2012).

Myers (1999) discusses the pros and cons of ethnographic research. Firstly, being at the research setting for an extended period of time allows the researcher to gain an in-depth understanding of the environment and the participants. Secondly, being in the field allows the researcher to gain insights and challenge current assumptions which are often taken for granted. The main drawback of ethnography is that it is time consuming. Moreover, critiques have suggested that this form of research lacks breath as the findings are concentrated in a particular setting. Myers (1999) argues that as more ethnographic studies are completed over time, the findings may reveal general models of meaningful contexts.

Relating specifically to the software development industry, Hughes et al. (1995) note that the complexities of the business and activities warrant a more facilitative approach to conduct ethnographic studies. As such, the authors introduced a ‘quick and dirty’ approach to ethnography. This form of ethnographic research is characterized by short focused studies to gain a quicker general picture of the setting, and taking strategic decisions to select work situations which could provide significant information. I chose to adopt the ‘quick and dirty’ ethnographic approach as it was suitable for a study in a complex and dynamic agile environment. While this approach provides quick and broad understanding, the main disadvantage is that it is difficult to gain a deeper insight of the research setting.

3.4 Research Setting

The study was conducted in two small enterprises based in Gothenburg, Sweden. Both companies are in the software development industry and work with agile methodologies. The
language commonly used at Surikat is Swedish while the official working language at Net Entertainment is English.

### 3.4.1 Surikat AB

Surikat AB is an IT company which develops internet and mobile based solutions for global logistics clients. The company was established in 2003 and it has today 16 employees of which 10 of them are in the software delivery team. About half of the employees have been with the company since its establishment. The office has an open-spaced concept where the developers are seated close to one other. The company works with in-house solutions and as such, most of the developers can be found in the office except when they are engaged in external meetings. The majority of the employees are Swedish.

The company has been using agile methodologies for a few years now. There are several projects employing the Scrum methodology. Scrum planning is conducted via Jira, an online project planning tool. Apart from that, pair programming is also heavily practiced in the organization. The key tenets of Scrum and pair programming were introduced in Chapter 2.4. Three years ago, the company started using the Confluence Wiki (a web-based collaborative repository). According to the Chief Operating Officer, the aim was to create a “knowledge space” where relevant information about the various projects and domains could be stored and structured. Before that, there was heavy reliance on the internal server where documents were uploaded and stored in unstructured folders.

### 3.4.2 Net Entertainment

Net Entertainment is an online casino games production company which was started in 1996 in Stockholm. The company spans across several countries with games being developed in offices in Stockholm, Gothenburg and Kiev to name a few. The research took place in the Gothenburg office which was established in 2011. There are a total of 14 employees in the games production team. The team comprises software developers, testers, designers, a sound consultant and the Game Lead. The office has employees from Sweden, Norway, Bulgaria and the UK. English is the official working language and all employees are encouraged to speak English so as to maintain transparency and understanding of what is going on.

The office has an open concept where employees are seated close-by and can easily interact with one another. In terms of agile methodology, the company is an active user of Scrum. Daily stand-up meetings are held at 9 in the morning where the team gathers to discuss their progress and what they will be doing for the day. Team members first gather in the kitchen where the Game Lead asks each member to give a number from 1 to 10 in response to the following three questions: “How are you feeling today?”, “How happy are you to be here?” and “How confident are you in your work today?” All the numbers are written on a glass window with a white board marker. The team then moves over to the Scrum board (whiteboard) where they pass a small ball around and take turns to give an update on the tasks they are working on. Post-It notes are stuck on the board with the stickers of the team members next to every task. Further coordination and sprint planning are conducted via Jira.
3.5 Profile of the Research Participants

Purposive sampling was used to select the participants. Purposive sampling is designed to provide greater depth of information by drawing from a smaller number of carefully selected participants. Purposive sampling frames are informal and based on the judgment of the researcher (Teddlie and Tashakkori, 2009).

The participants in Surikat comprised the people who were currently working in agile projects; a total of 5 software developers. As for Net Entertainment, the participants were the Game Lead, the software development and testing team; a total of 6 participants. The following includes the list of participants involved in the study. All the participants are known by their first names. Consent was obtained and the companies and participants were open to sharing their identities. The ethical considerations which have been taken for this study are discussed in Chapter 3.10.

1. **Andreas, Chief Operating Officer (COO)**
   Andreas has 14 years of experience as a software developer. He has been working at Surikat since 2003. He is involved in software development for large projects.

2. **Christian, Junior Software Developer**
   Christian has been working at Surikat for more than a year.

3. **Emil, Senior Software Developer**
   Emil has been working as a software developer at Surikat for 7 years.

4. **Joakim, Senior Software Developer**
   Joakim has been working at Surikat since its establishment. He has 17 years of experience working as a software developer.

5. **Jacob, Senior Software Developer and Project Leader**
   Jacob has more than six years of experience as a software developer. He joined Surikat two years ago.

6. **Kristine, Game Lead**
   Kristine joined Net Entertainment in 2012. She is responsible for how the company works with the Scrum methodology and oversees the running of projects.

7. **Henrik, Senior Software Developer**
   Henrik has 20 years of experience as a software developer. He has been working at Net Entertainment since the Gothenburg office was established in 2011.

8. **Marlene, Software Developer**
   Marlene joined Net Entertainment at the beginning of the year. She has worked as a software developer for 15 years.
9. **Andreas, Tester**  
Andreas has about 9 years of work experience. He has been at Net Entertainment for two and a half years.

10. **Iva, Tester**  
Iva has 4 years of experience as a tester. She has been working at Net Entertainment for a year.

11. **Stine, Tester and Requirements Analyst**  
Stine has been working as a tester at Net Entertainment for 2 years.

### 3.6 Method for Data Collection and Analysis

This section introduces the methods which have been employed for the study. As a large part of ethnographic research involves the researcher immersing in the natural setting, observations were the primary mode of data collection. This was complemented by interviews and other documentary evidence.

#### 3.6.1 Literature Review

Literature review was a key aspect of the research process. When the topic of the thesis was decided, an extensive search for research publications and student theses on similar studies was done via the DIVA portal and Google Scholar. The search included studies in knowledge management, knowledge management in agile teams and tacit knowledge management. Initially, the literature review helped to shape the focus and scope of the study. Current gaps in studies and problem areas were identified through the process of reviewing past literature.

The next stage involved the searching of relevant theories and concepts from books, academic articles and journals from scientific databases. The reference lists from past studies were helpful in providing direction on which articles or books to read. At this stage, the keywords and themes extended to perspective of knowledge, knowing in action, learning and agile methodologies. It further encompassed literature on philosophical traditions of research, research methodologies and ethics. The search was conducted through Google Scholar and by visiting Linnaeus University’s library and databases. The selection of the literature was based on the relevance of the content and the quality of the literature.

#### 3.6.2 Observations

In contrast to observations which are conducted in an experimental setting, ethnographic observation takes place in the natural context where the social activities are occurring (Angrosino, 2007). Citing Gold (1958), Angrosino (2007) discusses some of the roles which could be performed by the researcher. The complete observer role finds the researcher being as detached from the research setting as possible. To maintain objectivity, the observer is neither seen nor noticed by the participants. This role has been criticized for being potentially
deceptive and ethically questionable. In the observer-as-participant role, the researcher conducts observations for brief periods, possibly to lay down the context for interviews or other types of research. The researcher is made known to the participants but relates to them only as a researcher. This is opposed to researchers who take an active and engaged membership role, and become directly involved in the daily activities of the participants in the research setting.

The observer-as-participant role was chosen for this study to complement the ‘quick and dirty’ ethnographic approach. The objective of the study was to observe and understand instead of taking an active role in influencing the practices of the participants and the outcomes of the study.

Since the main language at Surikat is Swedish, the communications between participants were predominantly in Swedish. At Net Entertainment, all activities and communication which I observed occurred in English. None of the observations were video recorded as it was considered to be intrusive. Some of the participants indicated that they were not comfortable being filmed or audio recorded as they were working. Moreover, there were concerns that client interfaces and other proprietary information would be recorded in the process.

Some of the areas of interest in the observations include:

i. The practices of the participants and the way they work
ii. The processes and tools used by the participants to carry out their work
iii. The contexts in which the participants communicate with one another
iv. How the participants communicate with one another
v. How senior and junior team members interact with one another
vi. How team members share their expertise and experiences with other team members

Field notes and questions were taken down during the observations. Apart from writing notes, I drew simple pictures to indicate seating positions of the participants during meetings and key activities which took place during the observations. This technique helped me to re-experience the observation sessions as I was going through the notes during data analysis. Questions which I had identified during the observations were subsequently asked during the informal interviews that followed the observations.

In total, 11 observations were conducted over a period of 6 weeks. The observation sessions were selected according to the activities and settings which could provide important insight. The purpose was to observe a number of different activities and contexts where the participants were interacting in their respective organizations. Several of the observations at Surikat were spontaneous. I would ask the participants if I could carry out an observation whenever I noticed that a problem had arisen or if it was a setting which I believed would provide insight to the study. This accounted for the couple of problem solving and pair programming sessions which I was able to observe. On the other hand, all my sessions at Net Entertainment were planned. As such, most of the observations were conducted in meetings.
Table 3.1 indicates the list of observations and informal interviews which were carried out at Surikat and Net Entertainment over the course of the research.

Table 3.1: List of Observations and Informal Interviews

<table>
<thead>
<tr>
<th>Location</th>
<th>Date</th>
<th>Duration</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surikat</td>
<td>21 March 2014</td>
<td>2 hours</td>
<td>Workgroup discussion between Jacob, Emil and Christian. This was followed by an informal interview with all three members.</td>
</tr>
<tr>
<td>Net Entertainment</td>
<td>1 April 2014</td>
<td>1 hour</td>
<td>Scrum stand-up meeting (15 minutes). This was followed by an introduction to the organization.*</td>
</tr>
<tr>
<td>Surikat</td>
<td>1 April 2014</td>
<td>1 hour</td>
<td>Pair Programming between Andreas and Joakim. This was followed by an informal interview with Andreas and Joakim.</td>
</tr>
<tr>
<td>Surikat</td>
<td>2 April 2014</td>
<td>1 hour</td>
<td>Discussion between Jacob and Andreas on prototypes for an application</td>
</tr>
<tr>
<td>Surikat</td>
<td>2 April 2014</td>
<td>45 minutes</td>
<td>Preparation for deployment between Joakim and Christian</td>
</tr>
<tr>
<td>Surikat</td>
<td>2 April 2014</td>
<td>45 minutes</td>
<td>Problem solving discussion between Emil and Andreas</td>
</tr>
</tbody>
</table>
| Net Entertainment  | 4 April 2014    | 1 hour and 15 minutes | ● Scrum stand-up meeting (15 minutes)  
● Problem discussion between three testers (Andreas, Iva and Stine). This was followed by an informal interview with all three testers (1 hour). |
| Net Entertainment  | 15 April 2014   | 1 hour and 30 minutes | Meeting between the software developers (Henrik and Marlene) and the testers (Andreas and Iva) to discuss how they should enhance their collaboration. An informal interview with Andreas and Iva took place after the meeting. |
| Surikat            | 16 April 2014   | 45 minutes       | Discussion session between Joakim and Christian about creating new testing environments |
| Surikat            | 22 April 2014   | 45 minutes       | Pair programming between Christian and Andreas. This was followed by a short informal interview with Christian. |
| Net Entertainment  | 5 May 2014      | 1 hour           | Sprint review and planning meeting for the game development team |
This was an introductory meeting to familiarize myself with the research setting at Net Entertainment. Participating in the morning meeting proved to be an excellent ice breaker as it allowed me to introduce myself and how I would be conducting my study. I subsequently spent the rest of the time speaking to the employees, asking questions about the organization and other practices and processes which were currently used. Most of my time was spent with the Game Lead, Kristine, who showed me around the office, how the company works with the wiki and the core values of the company. I was also able to have a couple of short conversations with some of the participants who I would be observing and interviewing.

3.6.3 Informal Interviews

Informal interviews were conducted after the observations, in cases where there were questions which needed to be addressed. The informal interviews which were conducted are listed in Table 3.1 in Chapter 3.6.2. The purpose was to seek clarification on unfamiliar terms and for the participants to elaborate on certain opinions and concepts which emerged during the observation sessions. I would often take a couple of minutes to go through the notes and questions which I had written prior to having the informal interviews. The intention was to have the informal interviews shortly after the observations so that the context was still familiar to the participants. The informal interviews most often took place in group settings and were in the form of a conversation. I would pose specific questions which came up from the observations but at the same time, ask new questions as they came to mind. The informal interviews were audio recorded while I took notes at the same time. The interviews were later transcribed verbatim.

3.6.4 Semi-structured Interviews

Interviews in the semi-structured format were conducted to complement the findings. This was used to verify the insights which were gained from the observation sessions and informal interviews, and to ask about the practices which have been discussed in the theoretical framework. According to Creswell (2009), semi-structured interview is designed to cover a set of concepts during the course of the interview, while giving the respondent scope to answer the questions based their opinions, experiences and perceptions. This mode of interview was appropriate for the study as I was able to build upon the responses of the participants and add on or remove questions whenever appropriate. The interview questions can be found in Appendix A.

The pool of participants comprised the software developers. The semi-structured interviews were staggered with one semi-structured interview planned to be conducted a few days after each observation session. This gave the researcher time to analyze and reflect on the findings. All the interviews (informal and semi-structured) were conducted in English. It did not appear to be a deterrent to the participants as English is widely spoken in both organizations. The semi-structured interviews were conducted face-to-face as it was considered the most personal and best way to convey a sense of trust and ease with the participants. A total of 6 interviews were conducted. All the interviews were audio recorded and transcribed verbatim. Notes were taken concurrently during the interviews. The list of interviews is as follows:
1. Jacob – 1 April 2014 at Surikat (25 minutes)

2. Henrik – 4 April 2014 at Net Entertainment (20 minutes)

3. Emil – 9 April 2014 at Surikat (15 minutes)

4. Marlene – 10 April 2014 at Net Entertainment (15 minutes)

5. Christian – 16 April 2014 at Surikat (20 minutes)

6. Andreas – 22 April 2014 at Surikat (20 minutes)

3.6.5 Documents

Both Surikat and Net Entertainment use Confluence Wiki to share, find and collaborate on information, and Atlassian Jira to plan and track their project tasks. During the course of the research, I had full access to the information and all attached documents on Surikat’s wiki and Jira page. At Net Entertainment, I was given an overview of the structure of the wiki and its contents during the introductory meeting and I was further exposed to the contents of specific pages during my observation sessions.

Surikat’s wiki is intended to facilitate information sharing and to make it easier for the employees to find relevant information when required. The wiki is structured according to several “spaces”. There is a space allocated for all important matters regarding the organization. The other main page often visited by the developers is the space for the delivery team. The information on this space includes information such as guidelines, project templates, standard operating procedures and notes of meetings. In addition, there are spaces created for individual projects and domains which the developers work with.

At Net Entertainment, Confluence wiki is used as a repository for corporate information, notes of meetings from internal meetings and with the other offices. The wiki is used across the different offices in Europe. The Gothenburg office has its own space and the employees are given the autonomy to structure their wiki space. There are online tutorials which have been created by the employees. Examples of some of the tutorials include information on how to perform tests and how to program certain features. The wiki is frequently updated with information, graphics and even caricatures. There is even a page for the team to present their ideas for new games. The ideas are reviewed by the Stockholm office quarterly.

3.7 Role of the Researcher

According to Blomberg and Karasti (2012), reflexivity in ethnographic research suggests that the observer is always attached to the subject of his or her research. Reflexivity in the participant observer role entails both involvement and detachment. Jordan (1996, cited in Blomberg and Karasti, 2012) likens the role of the observer to that of a novice who attempts
to be a part of the community but who has to concurrently keep a distance to be able to reflect and make sense of the situation which is being observed.

Blomberg and Karasti (2012) add that sustaining an appropriate relationship between the observer and the participants requires constant reflection of the role taken by the observer and the degree to which he or she is influencing the research setting. Reflexivity necessitates that researchers recognize how aspects such as individual histories, relations with the study’s participants and other socio-cultural factors could affect the research outcomes. As Russo and Stolterman (2000) imply, it is when researchers are self-reflective that they are made aware of the pre-conceived assumptions and prejudices which underlie the research.

Since I am an employee at Surikat, it is important to discuss the role that I have taken as a researcher vis-à-vis that of an employee. My job role involves close interaction with the development team in process-related matters. At Surikat, I am exposed to a working environment where informal communication and frequent interaction between developers are a part and parcel of the daily working life. As a researcher, I have established a rapport with the participants prior to the study. This differs from the setting at Net Entertainment in which I had to get to know the participants and familiarize myself with the organization’s culture and setting. Nonetheless, participating in the daily stand-up meetings where I was also asked how I was feeling, how happy I was to be there and how confident I was in my work helped to provide a sense of belonging and familiarity with the community at Net Entertainment.

Working at Surikat and being familiar with the agile environment has brought about certain preconceived ideas regarding the practices which may be taken by agile teams. As mentioned earlier, these initial assumptions were reflected in the conceptual model. Despite this, I have sought to apply the same approach in my observations at Surikat and Net Entertainment. I recorded the contexts and the communication which transpired between the participants. Moreover, I did not assume all my interpretations of what I had observed to be reflective of the viewpoints of the participants. Aspects of the observations which were unclear were followed up by informal interviews where the participants were asked to clarify their opinions and why certain actions and decisions were taken.

My personal assumptions were constantly checked against the findings which ensued from the observations and the interviews. In short, I was aware of my prejudices while at the same time alert to the insights and concepts which emerged during the study. The steps which I have taken to assure the trustworthiness of the study are discussed in detail in Chapter 3.9.

### 3.8 Data Analysis

Apart from subscribing to hermeneutics as the philosophical approach, hermeneutics was used as the method for data analysis in this study. This was largely inspired by Kurti’s (2011) use of hermeneutics to analyze the data in her empirical study. Figure 3.1 below illustrates how the hermeneutic circle of understanding was applied in the study. I began by writing down my preconceived ideas (prejudices). New perspectives which emerged during the literature review
led to the formulation of the research question and the scope of the study. Furthermore, the conceptual model which was influenced by literature review was developed at this stage.

Data was analyzed by transcribing the field notes from the observations and interviews to be interpreted as texts. A field journal was used to record my observations, the context of the observations as well as my thoughts and reflections. The informal interviews which took place after the observation sessions were subsequently transcribed. As Angrosino (2007) discusses, noticeable patterns will emerge during the observations as the research progresses. This will subsequently lead to additional questions which should be pursued. The empirical research for this study continued until new findings were found to replicate the earlier ones. This was done within the time planned for data collection (a period between 6 to 7 weeks).

As the semi-structured interviews were conducted over the same period as the observations and informal interviews, I was able to constantly compare the field notes with the interview transcripts. The patterns which emerged were compared with the categories of practices in the conceptual model and other key concepts which were discussed in the theoretical framework. Other insights which were not discussed in the conceptual model and had surfaced during the empirical study were also analyzed. These new insights were frequently referred to after each observation and interview session. Reoccurring findings and insights which were raised by several participants were consequently grouped into new categories. The eventual results and explanations which were analyzed from the findings can be understood as an outcome of the fusion of my horizon and the participants’ horizons.
3.9 Trustworthiness

Qualitative researches do not attend to the constructs of validity and reliability in the same manner as quantitative researches (Shenton, 2004). In the qualitative paradigm, reliability and validity are conceptualized as trustworthiness, rigor and quality (Golafshani, 2003). Stenbacka (2001) states that a quantitative research is evaluated by how well it has fulfilled its purpose of explaining whereas a qualitative research is evaluated by how well it has met its purpose of generating understanding. Golafshani (2003) claims that although it is commonly argued that validity is not applicable to qualitative research, it is still imperative to have a qualifying measure for qualitative researches.

In seeking trustworthiness, qualitative researches commonly focus on the aspects of research credibility, transferability, dependability, and confirmability as introduced by Lincoln and Guba (1985). Shenton (2004) and several other authors describe how these constructs are used.
in comparison to quantitative (positivist) researches. In addition, this section will discuss the strategies which have been taken to ensure the overall trustworthiness of this study.

**Credibility**

In quantitative researches, internal validity is used to test that a study measures what it intends to. In contrast, credibility is used to establish trustworthiness in qualitative studies (Lincoln and Guba, 1985). The focus of qualitative research is to represent the realities revealed by the participants as adequately as possible (Shenton, 2004). According to Krefting (1991), it is important for the researcher to reflect on his or her position in relation to the research setting and to identify the factors which could influence the gathering and analysis of the data. My role as a researcher was discussed earlier in Chapter 3.7. Furthermore, a study’s credibility is threatened if the research subjects respond according to the preferred social responses instead of relating to their personal experiences (Kirk and Miller, 1998).

The following approaches, as suggested by Shenton (2004), were taken to ensure the credibility of the study:

a. *Early familiarity with the culture of the participating organizations before embarking on data collection* – Working at Surikat has given me the familiarity required for the research. In the case of Net Entertainment, I participated in the organizational routines such as informal morning meetings, daily stand-ups and had several informal chats with the participants. These initial engagements not only provided adequate understanding of the research setting before data collection, it further allowed the building of rapport between the researcher and the participants.

b. *Ensuring honesty in informants* – The participants were assured before the start of any interview or conversation that there were no wrong or right answers and that their personal perspectives were valued. The interviews were open and the participants were constantly encouraged to give examples and narrate their experiences when relating to the questions.

c. *Reflective commentary* – I constantly reviewed my perceptions and prejudices and how they have been shaped as the study progressed. A field journal was used to jot down my thoughts and reflections after each data collection session. This helped to record emerging patterns from the findings and other interesting insights.

d. *Triangulation* – The study employed several data collection methods such as observations, informal interviews in the form of conversations after the observations, and semi-structured interviews. Furthermore, the data was collected in two different organizations.

e. *Thick description of the phenomenon being studied* – The study sought to give detailed descriptions of the research setting and the surrounding contexts.
f. **Iterative questioning** – Questions which were raised in previous informal interviews or observation were noted down and used for subsequent questioning. This gave me time to reflect on what had been said previously and to confirm the opinions of the participants.

g. **Member checks** – Quotations and opinions which have been conveyed during the research were sent to the participants for checking prior to finalizing them. The participants were also offered the opportunity to read through the entire transcript of the interviews if they wished to. Another aspect of member checking was the involvement of the participants in verifying some of the concepts and themes which ensued from the interviews and observation sessions.

**Transferability**

In the positivist paradigm, importance is placed on ensuring that the results of a study are applicable to a larger population. However, this generalization of findings is irrelevant to qualitative studies which take place in specific settings and focus on a small number of participants (Shenton, 2004). Transferability is concerned with the extent to which the results from the study could be transferred to similar contexts. It is therefore imperative to provide thick descriptions of the phenomenon being studied so that the readers are able to gain enough contextual information and compare the phenomenon to similar settings (Shenton, 2004).

To ensure transferability, the following measures were taken:

a. **Conveying the boundaries of the study** (Shenton, 2004) – Detailed information with regards to the research setting was provided. This included information about the organizations involved in the study, specifics on the number of participants, their profiles, the types of data collection methods, and the number of and dates of the data collection.

b. **Determine if the context and observed events are typical or atypical of the participants’ lives** (Krefting, 1991) – Member checking was conducted for inconsistencies or contradictions which may have appeared during the course of the study. Any confusing or contradicting findings were further examined and checked with the participants for further clarification.

**Dependability**

Reliability is a key factor in quantitative studies. In this regard, positivist researchers seek to demonstrate that if a study was repeated using the same context, methods and participants, it would yield the same results. Yet, this is inapplicable to qualitative researches as the phenomena under study are never static. Dependability refers to the consistency of findings and it is used in the qualitative paradigm to account for variability (Guba, 1981). As qualitative researches examine a range of experiences, it is important to include atypical occurrences in the findings and discuss the possible reasons for them (Krefting, 1991).
Lincoln and Guba (1985) mention the relations between credibility and dependability, whereby working on either one of these constructs will help to ensure the implementation of the other. In this study, dependability was mainly assured by providing:

a. *Detailed descriptions of the research methods* (Shenton, 2004) – This included a discussion on the research design, how the research was implemented, and descriptions of how data was gathered and analyzed.

**Confirmability**

Quantitative researches are concerned with the extent to which the research is objective while qualitative researches focus on the confirmability of a study. Confirmability is concerned with ensuring that the study’s findings come from the outcomes of the experiences and ideas of the research subjects instead of the subjective claims and preferences of the researcher (Shenton, 2004). Hence, care has to be taken so that the data will not be brought out of context.

Strategies to ensure confirmability, as suggested by Shenton (2004), were provided by:

a. *Specifying the rationale for the methods chosen* – The differences in approaches were pointed out and the strengths and weaknesses of the methods were discussed.

b. *Reflective commentary* – The researcher’s preconceived ideas were specified and acknowledged. Theories and concepts which were not proven by the data were discussed.

**3.10 Ethical Considerations**

Ethical issues and tensions are important considerations which warrant attention in any research. Klenke (2008) states that the most critical ethical principles in qualitative research are informed consent, voluntary participation, confidentiality, protection from harm and ensuring the well-being of the participants. This section describes how ethical considerations have been taken into account in this study.

**Informed consent**

According to Klenke (2008), informed consent must be obtained from its participants. In addition, information regarding the study, its procedures as well as the benefits and risks to the study must be conveyed to the participants.

Consent was first claimed from the research participants for their approval to take part in the research and for the use of the information provided during the research. The objectives of the study and the procedures were explained to the participants prior to the start of the research. The companies and the participants were informed that all information would solely be used for academic purposes and no sensitive data would be made available for ‘outsiders’. A non-disclosure agreement pertaining to product information was signed with Net Entertainment.
Direct quotations were taken from the transcripts and verified by all the participants. The participants were further offered the possibility to go through the findings and analysis. These steps were taken to help the researcher to verify the information prior to publishing the study.

**Voluntary participation**

Shenton (2004) and Klenke (2008) point out that the participants should not be coerced to continue with a study. The participants were duly informed prior to the start of the research that if they were to feel uncomfortable at any point in time, they were not obligated to continue with the study.

**Confidentiality**

Klenke (2008) states that confidentiality seeks to protect the privacy of the participants and the participants should be duly informed if the researcher is aware of situations where confidentiality would be compromised.

In the context of this study, both the companies and the participants indicated that they were open to sharing their identities. The main concern was not to reveal information and specific details on the products, services and clients which the companies were working with. Prior to embarking on the study, the participants were asked if they would like to remain anonymous. Moreover, the possibilities of using initials or other names were suggested to the participants. The researcher considers this to be an important option to the participants as they could be less inclined to share their experiences openly or choose to omit certain information if they were concerned with their identities being revealed. Nonetheless, all the participants indicated that they were comfortable with having their first names published.

**Protection from harm and well-being**

According to Klenke (2008), sensitive topics which may cause discomfort or negative emotions to be evoked should be safeguarded against. The topic of the study and interviews did not trigger any strong emotions from the participants. Nonetheless, ensuring the well-being of the participants included respecting the balance between exploration and the research setting as advised by Gillham (2000). A key consideration was not to create inconvenience to the research participants. This was done by being attentive to the social setting and by not unduly disrupting the participants’ work momentum and activities. Applying this further ensures that the participants are comfortable and have trust in the researcher and the methods. I consider appropriate distance to be an important factor. For example, sitting too closely to the participants during observations could make them feel self-conscious about what they were saying and doing.
4. Analysis and Results

This chapter presents the analysis of how knowledgeability or “knowing how” is generated and sustained in agile teams by examining their practices. As it will be discussed, most of the practices are interrelated and should not be regarded as exhaustive or exclusive. This chapter has, however, been structured to discuss the practices in categories for analytical convenience. Some of the categories are from the conceptual model while others are categories which have emerged as a result of the analysis of the empirical material. As mentioned in the methodology chapter, the working language at Surikat is Swedish. The conversations and quotes from the observations were hence written down verbatim in Swedish. Nonetheless, the Swedish texts are accompanied by English translations.

4.1 Sharing “knowing how”

Orlikowski (2002) claims that sharing “knowing how” facilitates the learning of practice as it helps others to develop the ability to react according to diverse contexts and conditions. The sharing of “knowing how” was present in both Surikat and Net Entertainment.

Mentorship and showing a team member how to do something were widely practiced in both organizations. Emil specified that he often helps fellow colleagues with their code. Marlene said that she often demonstrates how to do things: “Show them what I do maybe or if they ask me something, they need my help, how you are going to build this software and they don’t know how to do it. Show and talk. Show is the best way to do it.”

The sharing of “knowing how” by Joakim (senior developer) was observed during the deployment and problem solving session which he had with Christian (junior developer). The purpose of the deployment session was to test the flow of the system with the mobile application. Christian was responsible for one part of the system while Joakim was in charge of the other two major parts. As such, this deployment required close coordination. Christian was asking questions such as “what should we do?” and “how many steps does it take?” Joakim was concurrently giving him suggestions and showing him what to look out for and to think about. He often mentioned, “What’s good about doing it this way...”

In a similar observation, Joakim and Christian were discussing how to set up a test environment for two projects on the same server. Christian who sits opposite Joakim moved to his teammate’s workstation. Joakim was explaining the possible consequences which could surface if the team was to use or proceed with certain ways of approaching the problem. An excerpt from the conversation is as follows:

Christian: “Antar att du måste ta bort det... (Guess that you must take that away...)”

Joakim: “Det beror på. Det är ett sätt att göra det. (That depends. That is one way of doing it.)”
Christian continued asking questions and gave suggestions of what could be done. Joakim was listening to Christian’s suggestions while concurrently looking at the screen. He seemed to be processing the problem and analyzing how to proceed. All of a sudden he said, “Testa det och se om det fungerar. (Test that and see if it works.)”

This observation exemplifies the “knowing in action” which takes place during problem solving. In this context, the senior developer was not merely sharing a set of ‘static’ knowledge or “know how” which should be applied to the problem. Instead, Joakim was listening and improvising on his approach as the conversation progressed. When Christian highlighted a couple of factors and concerns, Joakim re-examined the problem and came up with new suggestions. At the same time, Christian was listening to Joakim and constantly refining his ideas in accordance to how the problem solving was evolving.

Using the context of a mentor and learner, Swap et al. (2001) discuss the importance of shared contexts and meaning between the two parties in order for the novice to assimilate the mentor’s instructions. Swap et al. (2001) add that as the novice progressively gains experience through the mentor’s guidance, the “zone of proximal development” will widen and the novice will be able to understand more complex instructions.

From the two observations above, it seemed that Christian’s familiarity with the domains and how the systems function had made it easier for him to keep up with Joakim’s suggestions of how to approach their work and cope with the obstacles. Christian did not show that he was grappling with the basics of the systems and their terminologies. Rather, he was able to focus on solving the problem at hand. This familiarity has been attained through more than a year of experience working with the domains and working closely with Joakim.

The concept of the “zone of proximal development” was similarly implied during the interview with Henrik when he described how he usually shares what he knows with fellow colleagues: “After finding out the level of knowledge. As with any type of skill, there’s certain types of skills. When you’re struggling with very basic concepts, your problem areas are different. When you learn to drive a car, you are struggling to get the gear shift to work as you want it to. When you rise above that level, you observe other things, you worry about other things and you look for... you become a better driver because you don’t focus on the tiny details, you focus on your environment and you don’t risk running into people as much. And I guess that’s the level most drivers are at. Then there’s expert drivers who don’t even consider their environments, like race car drivers. They observe the extra skill other drivers have on the race track, shifting in the race track, things that they process that we as normal drivers would look for constantly but they just mentally know because their system is so attuned to the task at hand. There’s various levels of being a developer. After a certain time, with your experience, you don’t struggling with the concepts of programming anymore, you struggle with concepts of solving the problem in an efficient way.”

Henrik elaborated, “First of all I have to know what level the person is at. If she is a beginner, or she is an expert with her field, I have to talk about different things. If she is a beginner, I need to talk about the clutch and I need to talk about gear shifting basically. If it is a
knowledgeable person, I don’t have to focus on that stuff. I can talk about the stuff that we as a company do. It depends on who I am talking to basically. But whoever I talk to is a mix between theoretical talking, drawing on the whiteboard, showing what classes we use or systems we use and how they stick together and practice doing, actually doing.”

Henrik’s analogy further corresponds to Polanyi’s (1958) discussion on focal awareness and subsidiary awareness in which a skillful individual can focus on the main objective while the lesser-skilled individual has to focus on the proximate entity. The findings reveal that being attentive to the level which the novice is at helps the expert to share “knowing how” and facilitate learning.

4.2 Engaging in Communities of Practice

As Wenger and Snyder (2000) discuss, Communities of Practice are characterized by the sharing of information, experiences and knowledge which help the members to learn and improve their ways of doing things as they interact. Wenger (1998) adds that meaning is generated when individuals are actively involved in some practice.

Characteristics of Communities of Practice were observed in the informal workgroups at Surikat. The concept of workgroup was introduced to encourage the developers to investigate new tools and development practices. It was also meant to be a platform for the software developers to develop their capabilities. There are three informal workgroups at Surikat comprising 3 to 4 members in each group. The developers form the groups themselves based on their interest in the topics. The workgroups decide on the areas which they should focus on and how often they should meet. The groups then take turns to hold a presentation or discussion about useful practices during monthly company meetings.

When asked about the purpose of the workgroups, Emil replied, “The workgroups are a good way for people to be motivated to learn new things as we see what we can learn from others and things which other people can do. Some of the things I am not able to do and I am inspired to learn more about them when we discuss in the workgroups.”

Andreas said, “The workgroups engage the developers and give them a creative arena to further develop and validate their ideas. When we meet, we discuss what the experiences of the members are and about thoughts and ideas which can be improved.”

As pointed out by Brown, Collins and Duguid (1989), meaningful learning takes place when it is rooted in relevant social and physical contexts, and with the novice observing the community of practice through legitimate peripheral participation. The novice has to attain experience and become more involved in the Community of Practice before progressing from an observer to a participant. In an informal interview after the workgroup discussion at Surikat, Christian spoke about being able to understand more work-related content now compared to several months ago. In response to some guidelines which have been posted on a wiki page, Christian said, “At first glance, I did not understand much... but at that time I wasn’t familiar with the frameworks nor the projects. Now that I have worked with some of
the projects and frameworks, I can’t say that it’s that hard for me to understand them.” This finding shares similarities with the discussion on the “zone of proximal development” in the previous section. As the novice progressively gains experience through the mentor’s guidance, he will be able to understand more complex information over time.

Participating in the informal workgroups can be seen as an additional platform for the members to share “knowing how”, to learn from one another and build up their competencies. Orlikowski (2002) clarifies that as practice enacts the situated and recurrent activities of individuals, a view of “knowing in practice” does not see competence as being transferable. Similarly in this context, “knowing how” or competence is not directly transferred between the group members as they interact in the Community of Practice. Rather, knowledgeability is produced as members continually share new insights and develop new ways of approaching software development work.

**4.3 Learning by observing**

Learning by observing was frequently practiced by the team members at Surikat and Net Entertainment. This was noted in the observations and from the interviews in which the participants cited that seeing what people do and how they solve things were important for learning. At Surikat, it was a common occurrence to see two developers sitting by the same screen. There are extra portable chairs in the office and the developers would often take an empty chair and roll it over to where their teammates were seated. These include spontaneous discussions and pair programming sessions between the developers. Similarly at Net Entertainment, it was not unusual to see people moving to their team member’s screens to observe what they were doing and to take a discussion. When I asked the testers if it was common for all three of them to sit together to solve a problem, I was told, “not all three but most often at least two people”.

During a pair programming session between Andreas and Joakim, Andreas had moved over to Joakim’s work station to program together. Both of them were focusing on the screen. Joakim was using the keyboard and the mouse while Andreas was pointing to the lines of code on the screen. While the developers were observing what each other were doing, they were not doing so silently. Andreas and Joakim were constantly discussing and asking questions as to why the other party would choose to program in a certain manner. Mutual learning was observed to have taken place. Pair programming is not necessarily done between an expert and a novice. Andreas and Joakim are both senior developers with 14 and 17 years of software development experience respectively.

When asked about how he learns from his partner during pair programming, Andreas said, “Usually you learn a lot as you see how someone is doing something directly. So you can learn everything from how to best utilize the tools, features of the programming language at hand and even how to better design the software. Also for a particular project or subsystem it can be of how it’s built and designed and how to properly extend its functionality.” Similarly, Joakim explained, “It can be that instead of asking questions about a thing, it is good to sit
together to just show and solve the problems together at the same time. It is also faster and more effective than sitting alone and looking at the code to understand them. It is a good way to learn.”

The pair programming session between Christian (junior developer) and Andreas (senior developer) appeared to have more of an expert and novice relationship. From my observation, Andreas was taking the role of a leader and guiding Christian while they were seated at Christian’s work station. Andreas was using the keyboard, programming and clicking on the mouse while he was speaking. To make a point, Andreas often said, “Där kan du se vad som händer, om det fungerar eller inte. (There you can see what is happening, if it is working or not)” Christian would listen and ask occasional questions as they were pair programming.

When asked how he learns from other team members during pair programming, Christian replied, “Well... most of the time I learn by listening and inspecting while the other team member modifies the code. The main reason for that is err... because some parts of the code is too complex for me to fully grasp or that the time needed for me to understand it is a lot more compared to if they show me how to use it and how it’s all connected.”

Based on the findings, learning takes place through observing how the “knowing how” of a mentor and even a peer is applied in daily work situations.

### 4.4 Learning by doing

As discussed in the previous section, observing a senior or more skilled team member is a key practice in agile teams. Nonetheless, the act of doing and putting what one has observed into practice appeared to be even more important for the participants. Learning by doing was found to be predominantly enacted through practicing and experimenting.

**Practicing**

The nature of programming is dynamic and there are myriad ways which developers could choose to approach a problem. As Emil pointed out, “The fun thing with programming is that every time is something new, a new process. It’s like Lego. You have different Lego pieces to build a house. You decide how you want to build it. It’s not the same house every time you build it. You learn as you construct the Lego house.” Marlene said something similar, “That’s what I like about the work. You can do it in many ways and nothing is wrong.”

Overall, the participants revealed that learning involves gaining experience through practice. The importance of practice was evidently expressed by the respondents across different questions during the semi-structured interviews. The responses are as follow:

When I asked Henrik for his opinion on how knowledge is achieved, he responded, “Knowledge is I believe a mixture of study, reading, learning from external sources and by doing. By using your theoretical knowledge about things and putting them into action by delivering products is the best way to gain knowledge.”
In response to my question on what actions are necessary for the respondents to build competence and skills, Andreas (Surikat) said, “Collection of information, then practice makes perfect.” In response to the same question, Emil explained, “There are two things in skills. You know how to build a telephone because you’ve done it 700 times or if you’ve never built a telephone but you have done other things, and then you are able to use your experience from doing other things to build the phone. You rely a lot on doing by experience.”

Similarly, Jacob described, “Practice makes perfect, it’s good to practice with someone else then be on yourself to really appreciate all the facets of a problem. If you to build a house, and you have an experienced carpenter with you, he can tell you what to do and so on. It may seem easy, but if you were to build the house yourself, you would realize it isn’t. Like, you build one house with him, and then you try to build your own house. Then you will realize that all the things he knew or told you is not so obvious when you come to do it yourself. You train with someone else then you do it yourself.”

When asked about how she usually solves problems, Marlene said, “First I think I Google things or I ask Henrik if it’s a problem in the framework he has. But I cannot Google for everything I do but if I have a problem. I read a lot of code because I know when I read. The best way is to have a combination. I want to sit down and do stuff to program and do things then I cannot always see code. They can explain it but I must do to understand it.”

The internalization phase of the SECI model suggests that learning by doing is closely related to the conversion of explicit to tacit knowledge. In contrast, the findings reveal that learning by doing involves putting one’s knowledgeability into action. This study acknowledges that the “knowing how” found in agile teams is tacit and rooted in the actions of the team members. From the analysis, learning takes place when individuals put into practice the “knowing how” which they have observed. “Knowing how” is hence constituted and reconstituted as the team members continually engage in practice.

**Experimenting**

Another form of learning by doing was through experimenting. This was observed at Surikat, particularly in situations which required problem solving.

In an observation at Surikat, Sonar (a quality measurement platform) had crashed and Emil and Andreas were having a discussion about what had happened and how they should go about rectifying the problem. They started searching for information on Google. As they read through the information, they selected bits of information which they thought were useful. They then followed some of the steps suggested from the online forums and proceeded to test them. When that did not work, they discussed about the connections between the components. They were experimenting and improvising as they proceeded. My inference was that one must have the expertise in order to connect the different aspects to the problem. This was partly why Emil had approached Andreas to help him out with the problem in the first place.

Experimenting was likewise observed during the pair programming session between Andreas and Joakim. In areas where both of them were not too sure of how to proceed, they would
experiment as they went along. Andreas was constantly saying “Testa, testa och se om det fungerar (Test, test and see if it works)”.

When we discussed about pair programming during the informal interview, Andreas said, “A lot of development is usually experimenting to get everything to work the way you want. This does not change when it comes to pair programming but you have a buddy there on which you can use as a ‘bollplank’ (sounding board).”

As Nerur and Balijepally (2007) discuss, knowing and doing are interwoven in agile software development and learning takes place through experimentation. Schön, (1983) and Orlikowski (2002) similarly point out that situated practices encompass reflection and experimentation, in which our knowing is reconstituted as our thoughts and actions are altered. In this case, experimenting can be seen as a form of “knowing in action”. Knowledgeability is produced and sustained through a continuous course of action which is dependent on the social circumstances in which the experimentation takes place.

4.5 Verbal communication

The agile setting is characterized by informal and spontaneous interactions. As highlighted in the previous section, verbal communication appeared to be an integral factor which facilitated learning by observing. For instance, the person who was pointing at the screen during pair programming was usually speaking at the same time. Verbal communication was used to explain the decisions and actions which were taken whilst the pair were coding. Apart from observing the work which is performed, Jacob mentioned that he even observes how his team members speak. He said, “Yeah, it becomes like a group thing where everyone starts talking in the same way. Like there’s some new slang word”.

The importance of verbal communication was brought up during the semi-structured and informal interviews with the participants when they were asked how they share what they know with other team members and how they solve problems.

Emil: “Verbal communication. If I know a specific person who is an expert, I ask.”

Iva: “Yes, for me that’s not even a question the question is further on keeping the information somewhere but the verbal is by default. If it’s important we raise it and talk about it.”

Jacob: “Verbally. I don’t know if there’s any other way really. I had a summer job where I was part of a testing team and there was a 120 to 200 page design document. I always ask people when I encountered a problem what the correct way was and they started referring me to this design document maybe because I was asking too many questions but I was like read a 100 page document every time I encountered a problem? That was not effective.”

During the observations, I noticed how the team members were frequently asking questions and discussing ways to perform their tasks while they were observing their team mates. Verbal communication was widely used between senior and junior developers as well as in
the interactions between senior developers, albeit in varying degrees. The differences will be discussed in more detail later in this section.

Overall, the observations run counter to the argument by Nonaka and Takeuchi (1995) who claim that it is not necessary for language to be present during tacit knowledge sharing as apprentices learn from their masters primarily through observing and imitating how their masters practice their craft. The presence of language may not be as important in certain work settings such as in artistry and craft work. For example in carpentry, the novice learns from observing how his master hits a nail or performs certain techniques employed in carpentry.

The “knowing how” in agile software development is likewise tacit and lies in the head of the developers. However, the communication in agile software development is as Sharp and Robinson (2010) state to be tacit but yet at the same time to be predominantly verbal. It entails finding out how the application domain works, how the formal logic is used and what the algorithms that are used are. As observed in the pair programming sessions, language was used to explain why certain actions were taken or prioritized over others. In this case, verbal communication provides the context and facilitates the learning process.

This is in line with Renzl’s (2007) argument that language presents the context which we are able to know. Language was found to be imperative in creating meaning and providing the context for “knowing” to be continually generated in a collaborative environment. This was observed in the different meetings, workgroup discussions and problem solving sessions in both Net Entertainment and Surikat.

For example, during the meeting between the Flash developers and testers at Net Entertainment, there was a discussion about whether exploratory tests should be used to test bugs earlier instead of moving all the testing towards the end of a project. The developers were sharing their perspective that it would be a “waste of time” if the testers had done the exploratory tests and the developers would then have to make additional changes after the test. The testers’ perspective was that it would be a “waste of time” if they did not react to the problem. In the words of Andreas, “What’s a waste of time is that we do not act on the bug reports after some time.” Here, we see how words such as “waste of time” obtain meaning through their use. Language not only provides context but is re-contextualized through its use.

This finding is in line with Suchman’s (2007) claim that language is indexical, situational and its understanding is dependent on the context of use and shared understanding. The situational quality of language was similarly observed during the pair programming session between Joakim and Andreas, two senior developers who have worked together for more than 10 years. The excerpt from the observation is as follows:

Andreas: “Jag kan visa dig enkelt vad jag menar. (I can show you easily what I mean.)”

Andreas started drawing figures on the paper to make his point and show the connections. Joakim seemed to understand what Andreas was doing and agreed with his approach.

Andreas: “Bättre att göra en lista i stället. (Better to do a list instead.)”
Joakim: “Ja. (Yes.)”

Andreas and Joakim subsequently came to a point where the both of them were unsure if what they were doing would work.

Andreas: “Får se om vi pajar någonting. (Let’s see if we disrupt anything.)”

Joakim immediately responded: “Ja, det gör vi säker t (Yes, I am sure we will.)”

Nonetheless, Joakim proceeded to code according to what Andreas had suggested. When Joakim said that he was sure that they were going to disrupt something, this would be taken as a disagreement to Andreas’s suggestion. However, his actions indicated otherwise. By doing what Andreas suggested meant that he agreed to experiment. This example further demonstrates how generating and sustaining “knowing how” is very much dependent on the ability to understand the indexical and situated qualities of language shared by team members. Nevertheless, the indexical and contextual nature of language was more likely to be displayed and understood by team members who have worked together for a longer period of time rather than between a junior and senior developer.

4.6 Using metaphors and analogies

Since I am working in the IT industry and at Surikat, I found it challenging during the observations to decipher which terms were metaphors as I am used to hearing some of the concepts and terms on a daily basis. To gain a deeper insight on the use of metaphors and analogies, the opinions of the participants were sought during the semi-structured interviews.

From the semi-structured interviews, the majority of the participants agreed that metaphors were used in their daily work. However, not all respondents were able to give concrete examples of the metaphors that were used. My sense from the interviews was that some of the metaphors were so ingrained in the daily activities of the participants that it was difficult to speak about the terms and concepts which were used in their everyday work. The responses from the interviews are captured below.

Henrik: “Well I mean all programming has to be talked about in metaphors. Programming itself we use classes, we use instances of classes, we use metaphors for logical concepts at all times. And on a higher level we talk about these as well. I am trying to figure out a good way to describe the metaphors. But yeah, we do use metaphors... I am not very good at describing what types of metaphors but yeah it’s always there. Everything we talk about is basically metaphors.”

Jacob: “We have a lot of components in the system which requires some sort of initiation to understand, client/server, different services... They are not slang but component names. If you don’t know what they are or the role they play, it’ll be very difficult to understand what someone is talking about. Developers love to get technical even if you specifically tell them not to.”
Christian: “I doubt the metaphors and analogies that I use are very hard to relate to. However, there will most likely always exist terms and abbreviations within a project that no outsider will understand. I am certain that there are some that only the developers will understand and some that I have no clue of what it meant, even though it is all related to the same projects. I try not to use them if someone whom might not understand them partakes in the conversation.”

Andreas (Surikat): “Yes, also while working with the customer as technical details and problems can be hard to explain. Then you need to think of other words and analogies to help explain what you mean... I can’t think of any specific example now but they come up when I am talking to customers.”

Marlene: “Sometimes. I don’t know… Not so much I think.”

Emil: “Yeah, we definitely do.”

From the participants’ responses, it appears that metaphors are used to frame a problem or an aspect of work. They are further used to create joint meaning within the teams and even with customers. In this case, the use of metaphors can be seen as the “knowing that” which is required to put software development work into practice.

From the knowledge management perspective, Nonaka and Takeuchi (1995) discuss that metaphors could be used to articulate tacit knowledge and convert it to explicit knowledge. This view assumes that the mental models of individuals can be transferred once it is converted to a common term or concept. A fundamental aspect which Nonaka and Takeuchi (1995) neglect to take into consideration is the role of the receiver in tacit knowledge communication as brought up by Busch (2008). As can be seen from Christian’s response, he hears metaphors or new concepts which are alien to him even though they are related to the same projects which he is working on. This is in line with Busch’s (2008) argument that familiarity and experience are important for the decoding of metaphors. If the novice does not have sufficient experience and insight to understand the metaphors, he or she will not be able to perform the task.

4.7 Storytelling

Sole and Wilson (2002) claim that storytelling and narratives are increasingly used to share knowledge and experiences, and facilitate exchange of contextual information in problem solving. Jashapara (2011) and Dalkir (2005) similarly argue that storytelling can help to transmit valuable tacit knowledge. Furthermore, Nonaka and Takeuchi (1995) claim that storytelling can be used as a technique to help individuals internalize what they experience and enrich tacit knowledge.

Storytelling does not seem to be used extensively in Surikat to share “knowing how” and it is not used at Net Entertainment. As I had not observed any form of storytelling at all during the
observation sessions, I asked about the role of storytelling and if it was being practiced in the organizations during the semi-structured interviews.

Apart from Jacob, none of the participants explicitly told stories. Henrik mentioned that “It can be efficient in showing failures or shortcomings or efficiencies, but I haven’t thought of it as a separate technique or to communicate knowledge sharing. But yeah that’s good.” The other participants said that they did not encounter storytelling in their work.

During the interview, Jacob said that he would often tell a story which happened at his old work place to stress the consequences which could result from simple mistakes. “I often tell this story about my old work place where 15 people were given positive results to an infection. What happened was some clinic had 100 samples to be tested and the lab personnel had a group of answers as negative and the group answered negative to this keyboard command. One time they pushed the wrong key and they answered positive and they pushed yes yes. So 15 or 20 people got a false positive. There was a follow-up questionnaire for the people who are positive. So you really don’t want to answer positive falsely. So this was a mistake and we had to fix this so we removed the possibility to answer positive. So I tell that story whenever someone asks me about pop-ups. Pop-up is not necessarily good because you just learn to ignore them. This example is good because it shows that even two pop-ups will not be enough to stop a user from making mistakes.”

Jacob’s story coincides with Denning’s (2006) claim that good stories should encompass details of mistakes made. While lengthier stories may not be widely used, learning points from past projects were shared during meetings. When Andreas and Jacob were discussing prototypes for a mobile application, Andreas was comparing the system they were building with an earlier project. He pointed out a couple of points which they should think about based on his experience. He iterated a few times, “Det var någonting jag lärt mig från tidigare projekt. (That was something I learnt from previous projects.)”

At Net Entertainment, the meeting between the flash developers and testers showed that drawing on experiences and sharing key learning points were important for getting a point across during discussions. Andreas and Iva spoke about past experiences of having too many unresolved bugs which have not been attended to and the issue of wrongly fixed bugs to highlight the need to create exploratory tests. At one point, Henrik even pointed out, “We had issues in the past where we should work like this but it didn’t work! People were telling each other, you should have told us earlier!”

While storytelling and learning points were found to be insightful, they could be limited to the sharing of experiences and act only as a reminder of what to avoid when working with similar systems or applications. Nonetheless, the findings did not reveal how storytelling or the sharing of learning points could generate or facilitate the “knowing how” which was required to address project-related problems and skills-based issues.
4.8 Other findings and insights

The findings thus far discuss the practices drawn from the conceptual model. The analysis shows the extent to which the practices have or have not been observed in both companies and how they influence the “knowing” which is constituted in the participants’ actions and practices. During the course of the research, several other insights were gathered.

Learning by reading

As mentioned earlier, Henrik, Emil and Marlene indicated that they learn by seeing what people do and how they solve things. Nevertheless, all three developers commented that they also learn by looking at other people’s code.

Emil: “By seeing what other people do. By looking at their code. I look at how to design different solutions and how things can be done. Get more experience by looking at the code. But I use the information and do it myself. A programmer is like an architect. You see how to build and design a house.”

Henrik: “Learn by seeing what they do, how they solve things? Sometimes we talk as well but usually... I think programming is one of the tasks where after you’ve reached a certain level, the information transfer is easier when you read the code that others have produced. If that code is easy to follow, it’s quite easy to know how the code is intended to be used. So reading up on existing code or findings examples when there’s some trick you haven’t used before, this should be a good solution for this.”

Marlene: “Henrik shows me things, he writes on the white board about how the games work with our front end game and client then I look much through the code. I read most of the code myself.”

The participants’ profiles indicate that Emil, Marlene and Henrik have 7, 15 and 20 years of software development experience respectively. Marlene commented that it was easier for her to read and understand code now as compared to when she first started doing that. She said, “That’s a skill you have when you’ve worked. You know how this whole thing works. Because I done that in the beginning as well then I learn things but now I can read and learn easier than I do in the beginning.” However, she clarified that the way the code were written plays an important role, “It’s important for me that the code is written in explaining what things do instead of having so short names... Short names is much faster to write but when you read it then you don’t understand. So if you explain what a variable contains then it’s easier to read.”

According to Polanyi (1958), meaning is revealed when the apprentice is able to achieve the same kind of indwelling as the master. Before that, the apprentice is dependent on the meaning which rests with the master. Apart from looking at how novices learn from their masters, this finding sheds light on how senior developers (masters) continue to learn, gain competence and in turn generate knowledgeability. As it appears, experience is a key factor
which enables the senior developers to decipher other people’s code and how they are intended to be used. Even as Marlene is new to the organization, she possesses the experience and “knowing how” to interpret code. This is a competence which is unlikely to be seen in a novice or junior developer. Nonetheless, the code should not be regarded as static explicit knowledge in which one could simply read, follow and apply exactly what was written. In this case, “knowing how” is generated by the senior developers analyzing the code which they read, interpreting their usage and applying the “knowing” to their daily practices.

**Aligning processes**

Orlikowski’s (2002) study on a globally dispersed high-tech company, Kappa, found that the project management model, planning tools and common systems development methodology were important factors in enabling the project managers and software engineers to continuously align themselves with their product development work. By engaging in these activities, Kappa members constitute an ongoing “knowing how” to coordinate their product development work over time.

Similarly, the practices associated with the Scrum agile methodology were found to be central in coordinating software development work at Surikat and Net Entertainment. For instance, the frequent Scrum meetings provide a platform for teams in both companies to organize their work. At Net Entertainment, the work day begins with a daily stand-up meeting where the team gathers and individual members give an update of what they will be working with on that day. At Surikat, the Scrum meetings are held in smaller groups and at different times depending on the project. Other planned meetings such as the sprint planning and retrospectives meetings likewise provide a structure for how work tasks should be coordinated among the team members.

According to Andreas (Surikat), “All projects are using the same structure of the daily Scrum meetings where we pose three questions: What did you do yesterday? What will you do today? Are there any impediments in the way? By asking each team member, the team as a whole gets a very good understanding of what has been done and what work remains.”

Sprint planning meetings at Net Entertainment take place fortnightly. During a sprint planning meeting at Net Entertainment, I had observed how the team members aligned themselves with their teammates and their work. The Game Lead (Kristine) started by asking the team for their views on what the goals for the next sprint should be. The team members were seated around a rectangular table and took turns to verbalize what they thought needed to be done during the next sprint. The work dependencies between the different competencies such as graphics, flash and test were also raised. For example, the people working with graphics mentioned that they would have to involve the flash developers in their design meetings. The Game Lead emphasized that involving the testers early in the process was important as they could see the linkages which the others may not think about. The testers stressed on the importance of updating the tasks on Jira and to focus on game flow for the coming sprint. They even gave reasons as to why it was important to do so.
Sharing updates, discussing work dependencies and receiving feedback from fellow team members were found to provide a common sense of “knowing” and basis for the actions which individual team members should take on next.

Apart from the planned activities, the nature of working in an agile setting demands close interaction and collaboration between the team members. The teams generate and sustain “knowing how” to get things done by drawing on relevant resources and aligning coordination between the team members. This was observed at Net Entertainment when the team of testers (Andreas, Iva and Stine) ran into a problem. The team was referring back and forth to the information and tutorials on the wiki as they tried to solve the problem. Iva commented that the problem that they had was actually very specific. After attempting to solve the problem without any success, Andreas said that they should ask colleagues in another Net Entertainment office if they have experienced similar issues. He stated that he knew who to contact. The team decided that the next step was to get in touch with a person who could help them with their problem. According to Chau and Maurer (2004), the tacit form of social resources includes knowing who to contact and refer to. This form of knowing was found to be useful in generating and sustaining the “knowing how” to mitigate or solve problems in agile projects.

In another observation, the testing team at Net Entertainment had initiated a meeting to discuss about exploratory tests. The purpose was to improve communication with the developers by giving them feedback about issues as early as possible. Iva and Andreas explained that the main testing for a current project had shifted towards the end of the project which represented more of the waterfall process of software development. This could result in the team fixing a lot more bugs than if testing could be done earlier.

Iva commented, “We have these design meetings, usually mainly art related, rarely even flash. Then art and flash collaborate together but test often gets left out. We try to get involved this way so that we get more details earlier. In the end when we get the feature absolutely complete and we start running tests, we know this looks like that because it was a design decision. It has happened before we are not entirely sure why something looks like it looks.” In this case, contextual knowing or phronesis (context-sensitive practical wisdom) was observed to be a critical factor in improving processes and how the team collaborates. This was displayed through knowing how to apply the best set of actions in particular scenarios.

Apart from generating the “knowing how” to develop competencies in software development, the “knowing how” to coordinate activities was found to be equally important in ensuring the success of agile projects. This could take the form of following agile project methodologies or through spontaneous meetings initiated by team members.
The role of documentation

Current knowledge management literature places heavy emphasis on the codification of knowledge. The combination phase in the SECI model suggest that documentation can be used to convert explicit knowledge into more systematic explicit knowledge. Likewise, in internalization phase, documentation is stated to help convert explicit into tacit knowledge. As opposed to documenting knowledge, the findings indicate that the teams at Net Entertainment and Surikat use the wiki to upload and search for information. Furthermore, much of the understanding of how to use the information and apply it to the daily practices is dependent on the context of its use. The following captures some of the sentiments of the participants:

Emil: “It’s difficult for a system developer to follow documents and solve the problems. It is information. You need to know the terminologies to find information online. If you want to build a bridge and you don’t know what a “bropelare” (bridge pillar) is called then you cannot find it. Google gives you information but you cannot follow the information exactly. As a programmer, you discover new information all the time. You never solve a problem just by looking at the wiki.”

Andreas (Net Entertainment) commented that the instructions and guidelines on the wiki “serve as a guide to fish up new thoughts. The guidelines are more of how it looks like and not how you should use it.” He elaborated that context is important for the instructions to be useful: “The information now (referring to the wiki) is not very well structured. Context is very important especially for a remote studio to a larger company.”

The significance of context was observed during the workgroup discussion session at Surikat. Emil, Jacob and Christian were going through the guidelines for developers on a wiki page when they stumbled upon a section written by an ex-colleague. Apart from guidelines and tips, the page discusses concepts and includes diagrams as well as screen shots with code. Despite the availability of information, the common sentiment expressed by the group was that it was not easy to understand the guidelines. This was especially so since the information pertained to a complex project and none of the members were familiar with the domain. As Jacob commented, “It’s just excerpts of information.” Christian added, “He left some pages on the wiki and several zip-files containing a lot of information. I have not read everything since there’s just too much information about projects which I do not work with.”

4.9 Summary of the findings

The findings indicate that knowledgeability is generated and sustained in the agile environment through daily practices and corresponding activities. Through the analysis, two distinctive types of knowing were derived and categorized accordingly. They include “knowing how to develop competencies/ capabilities” and “knowing how to coordinate effort” (see Table 4.1 below for the summary of the findings).

In comparison to the conceptual model, storytelling was hardly used and found to be the least relevant practice. While metaphors were widely used in both organizations to create joint
meaning, their usage did not appear to generate “knowing how”. From the analysis, metaphors were found to constitute the “knowing that” which is required for software development work to be put into practice.

Two other practices further emerged during the study. The first was learning by reading and the second, the practice of aligning processes. Learning by reading was observed to be more commonly enacted by the senior (more experienced) participants. Much of the emphasis on learning is placed on the novice learning from the master. This finding turns the focus to how senior developers (masters) learn and continue to develop their competencies. Apart from the practices which are enacted to develop competencies, the practice of aligning processes was found to be displayed by team members to coordinate effort and enhance collaboration.

Table 4.1: Overview of Practices, Activities and Knowing (Formulation of findings inspired by Orlikowski, 2002)

<table>
<thead>
<tr>
<th>Practice</th>
<th>Activities comprising the practice</th>
<th>Knowing constituted in the practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sharing “knowing how”</td>
<td>• Senior providing guidance/mentoring and being attentive to the novice’s level of competence</td>
<td>Knowing how to develop competencies/capabilities</td>
</tr>
</tbody>
</table>
| Engaging in Communities of Practice (CoP) | • Sharing insights and experiences  
• Assimilating in and learning from other members in the CoP |                                                                                       |
| Learning by observing     | • Observing a mentor or a peer in daily work practice                                                                 |                                                                                       |
| Learning by doing         | • Practicing  
• Experimenting                                                                                                       |                                                                                       |
| Learning by reading       | • Senior developers reading and interpreting code, and applying the “knowing” in daily practices                      |                                                                                       |
| Verbal communication      | • Providing meaning and context to practices through everyday language                                                  | Knowing how to coordinate effort                                           |
| Aligning processes        | • Following the agile project methodology  
• Team members initiating spontaneous meetings to discuss about work and solve problems                               |                                                                                       |
5. Discussion

This chapter discusses the results from the study and the implications of the findings for organizations and knowledge management.

5.1 Generating and sustaining knowledgeability in agile teams

The findings have elucidated how tacit knowing is enacted in the daily practices and activities of team members in agile projects in the two organizations. It is important to note that most of the practices are interrelated with activities which overlap.

The findings suggest that the sharing of “knowing how” is critical in facilitating learning by observing and learning by doing. In order for the mentor to guide and share “knowing how” effectively, attention must be paid to the competency level which the novice is at.

Observing how a senior team member or a peer performs work is another fundamental aspect of learning in agile software development. Nevertheless, learning by observing alone is not sufficient. The analysis indicates that learning by observing contributes to learning by doing as it is important for the individuals to be able to put what they have observed into practice. Learning by doing is mainly enacted by practicing and experimenting. This is similarly expressed by the senior developers who shared that they learn by reading code followed by doing. However, the ability to read and understand code requires a familiarity with the context of how the code were written and intended to be used. Learning by reading is more likely to be performed by the senior developers.

Furthermore, engaging in Communities of Practice in the form of informal workgroups was found to provide opportunities for team members to share “knowing how”, to learn from one another and develop their competencies. A fundamental consideration for teams is to ensure that newer members are able to assimilate in the Community of Practice. Knowledgeability is produced as team members collectively share their experiences, insights and develop new techniques or approaches.

The knowing which is constituted in the practices mentioned thus far point to “knowing how to develop competencies/ capabilities”. Another aspect of knowing which the study found to be equally important is “knowing how to coordinate effort”. Verbal communication appears to be an important practice in providing meaning and context through everyday language. The analysis suggests that while language provides the context for knowing and action in the agile setting, it is at the same time re-contextualized through its use.

Furthermore, the teams generate and sustain “knowing how” to get things done by aligning processes and coordination between the team members. The findings indicate that contextual knowing and engaging relevant social resources help to align processes. This encompasses drawing on past experiences and knowing how to apply appropriate measures when managing
new situations. Key activities include following agile project methodologies, and initiating spontaneous meetings to discuss about work matters and to solve problems.

5.2 Implications for organizations

Orlikowski (2002) claims that “knowing” is contextual and how it is applied is dependent on the situated activities of the individuals. As emphasized in the study, knowledgeability or “knowing how” is not a static entity which is transferable. In contrast, the knowing which is enacted by the teams is an ongoing process of discovery and development which is rooted in their actions. Orlikowski (2002) reminds us that competence is continually achieved rather than given. Acknowledging this perspective has bearing on how agile teams should work with knowledgeability. Instead of focusing on how to transfer best practices among agile team members, it is worthy to place more emphasis on understanding the contexts and conditions which could facilitate or hamper the generating of knowledgeability in agile teams.

Although this study does not concur with the notion of knowledge conversation in Nonaka and Takeuchi’s (1995) SECI model, there are aspects of the model which provide useful insights. A relevant area is ba, also known as the shared space or contexts where knowledge sharing takes place. Face-to-face communication through structured or informal dialogues, mentoring and working alongside an experienced colleague are characteristic of the originating ba, dialoguing ba and exercising ba discussed in the SECI model.

The findings reveal that shared contexts and open physical work spaces are crucial in facilitating and supporting the interactions between team members in the agile setting. The proximity to which the team members are seated to each other makes it easy to communicate with a fellow teammate and to seek help and guidance. Following agile methodologies helps team members to meet frequently through daily scrum meetings and weekly or fortnightly sprint planning meetings.

The importance of face-to-face communication has likewise been underscored by Cabrera and Cabrera (2005) who suggest that this personal form of communication provides a rich medium for information exchange. Lengnick-Hall (2003, cited in Cabrera and Cabrera, 2005) discuss that having employees together physically helps to increase the occurrence of information sharing. At the same time, this increases the familiarity between employees which could lead to shared understanding and enhanced feelings of community.

In addition, agile practices are people-centric and collaborative in nature. Net Entertainment and Surikat have an open organizational culture which supports frequent dialogue and informal interactions. This encourages the teams to engage in spontaneous communications, proactive problem solving and for the teams to cooperate to get things done.

Being aware of the practices and activities helps organizations to understand how agile teams are contributing to knowledgeability in their organizations. Nonetheless, understanding the contexts and conditions in which certain practices are more or less likely to be performed
could further help organizations to create favorable circumstances to sustain knowledgeability.

5.3 Implications for knowledge management: Can knowledge be managed?

Discussions on knowledge often fall under the umbrella of knowledge management. Current literature tends to describe knowledge management as a pertinent approach for companies to attain competitive advantage and organizational effectiveness. Similarly, in the field of information systems (IS), the emphasis has shifted from information processing towards knowledge transfer, with the use of information technologies to facilitate the acquisition and codification of knowledge (Busch, 2008). Despite its growing popularity, there are fundamental assumptions in the knowledge management discourse which should be discussed.

The conceptualization of knowledge is diverse. For instance, this study subscribes to the view of tacit knowledge as knowing which is rooted in action. As knowledge is ambiguous and complex, several authors (Alvesson and Kärreman, 2001; McInerney, 2002; Bouthilier and Shearer, 2002; Kakabadse, Kakabadse and Kouzmin, 2003) question if knowledge can truly be managed. Moreover, Bouthilier and Shearer (2002) state that the distinction between knowledge management and information management remains vague and the ambiguity is compounded by the confusion surrounding the concepts of knowledge and information.

Wilson (2002) argues that while data, information and information resources may be managed, it is impossible to manage knowledge. As we often do not know what we know, our knowing may only appear when its application becomes necessary. This has similarly been raised by Orlikowski (2002) who asserts that as knowing cannot be detached from its practice and context, it cannot be transferred. At the closest, only data and information could be transferred.

The findings from the study suggest that “knowing” cannot be devoid of its context. This was observed in the practices such as learning by observing and doing, verbal communication between team members and the teams aligning processes. Wilson (2002) states that expressing what we know by conveying messages in verbal, graphical or gestural forms do not constitute knowledge. On the contrary, the messages contain information which may be comprehended and integrated by a knowing mind. Stenmark (2001) argues that messages in the form of documents, records and software code are essentially information.

Likewise, the findings suggest that documentation on the wiki was perceived to be information rather than explicit knowledge which could be systematically extracted for use. This runs counter to Nonaka and Takeuchi’s (1995) view that computer systems can be used to decontextualize existing explicit knowledge and create new knowledge. The role of the knower is taken up by Tuomi (1999) who argues for a need to take a reversed perspective to the traditional data-information-knowledge paradigm. As Tuomi (1999) asserts, the knower creates facts from his or her knowledge and data is present only if meaning is used to represent information. Hence, the computer alone does not accord meaning to the context that
is being processed. Rather, it is the computer programmers (knowers) who design and negotiate meaning. This underscores the importance of the knower in the “knowing”.

The findings indicate that even though it is a useful practice to share information and provide guidelines on the wiki, the use of information relies on the individual’s “knowing” and interpretation of how the information should be applied. The manner in which information was structured and written was found to depend largely on the knower. This is in line with Wilson’s (2002) argument that the knowledge structures for the sender and receiver cannot be assumed to be the same. Dixon (2000) explains that even though a person may be able to write down everything he or she knows, this does not mean that another person would be able to interpret the information as intended or produce the same outcome. The content only reflects what the person has done in the past and not how it can be used for future situations.

The importance of the knower is once again emphasized in the collaborations which take place during agile meetings, problem solving sessions and pair programming. In these contexts, individuals (knowers) engage in activities where their “knowing how to develop competencies/capabilities” and “knowing how to coordinate effort” are enacted. As a consequence of the collaboration and interactions, mutual learning between knowers takes place. “Knowing” is hence constituted and reconstituted in the relationship between the knowers, the activities and the knowledge which is created.

As discussed, knowledge is constituted in our knowing; it is dynamic, context-dependent and cannot be devoid of the knower. This makes the management of knowledge essentially impossible. The propensity to achieve competitive advantage and organizational effectiveness from so-called knowledge management best practices and measures remain nebulous. Perhaps knowledge management is as Wilson (2002) implies to be nothing different from the management of information or the management of work practices to enhance the sharing of information.
6. Conclusion

This chapter provides the conclusion to the study. It further presents the reflections of the researcher and suggestions for future research.

This study has exemplified the value of taking the perspective of knowing in practice to examine how knowledgeability is generated and sustained by agile teams. Such a perspective runs counter to the traditional knowledge management discourse which views knowledge as a static and transferable entity. A conceptual model was developed from current literature. Although several of the practices were associated with knowledge management, particularly in the knowledge creation and sharing processes, the study did not assume that these practices would be applied in the same manner in generating and sustaining knowledgeability in agile teams.

The study subscribed to phenomenological hermeneutics as the underlying philosophical tradition to interpret meaning and understand the participants’ worldviews. The empirical study was performed in two companies which work with agile projects. The rationale was that extending the study to two different organizations would provide broader insights to the practices employed by agile teams in different companies.

The findings revealed two critical forms of “knowing”: knowing how to develop competencies/capabilities and knowing how to coordinate effort. “Knowing how to develop competencies/capabilities” was present in sharing “knowing how”, engaging in Communities of Practice, learning by observing, leaning by doing and learning by reading code. “Knowing how to coordinate effort” was enacted through verbal communication and aligning processes. Most of the practices appeared to be interrelated and dependent on one other for knowledgeability to be generated and sustained in the organizations.

While the practices in the findings may share some similarities with those cited in the knowledge management literature, new knowledge was not created by converting it from tacit to explicit knowledge and vice versa. On the contrary, “knowing how” was produced and sustained through the ongoing actions and everyday practices of the participants. I concluded by discussing the implications of the findings for organizations and knowledge management.

6.1 Reflections

Exploring a topic on knowledge was interesting but at the same time challenging. The conceptualizations of knowledge are not only multifaceted but diverse. As highlighted, knowledge can be viewed as an object, a disposition or a form of knowing. The complexities are deepened by different interpretations of concepts of knowledge, such as tacit knowledge which was discussed earlier in the study. Choosing to approach the study through the lens of knowing in practice was valuable as it led to a deeper understanding of the tacit aspects of knowledge or “knowing” through examining the actual practices of agile teams. This
perspective accords importance to human agency and the role of the knower in generating knowledgeability.

In addition, ethnographic research proved to be a particularly suitable approach for this study as I was able to observe the practices enacted in the everyday work of agile teams. This comprised the individuals’ activities as well as how they interacted with one another and their work settings. Having a background in IT helped immensely in understanding the contexts which the observations were conducted such as the terms and concepts which were used by the participants at both Net Entertainment and Surikat. Furthermore, being able to conduct the research in two settings has provided broader insights to the study as I was exposed to the different practices in both organizations.

Nonetheless, a limitation of the ‘quick and dirty’ ethnographic approach is that the observations were limited to specific occurrences. As I am an employee in Surikat, I was constantly exposed to the research setting. However, besides knowing the project planning meetings beforehand, it was not possible to predict when a problem would occur or when the developers would be pair programming. This supports Hughes et al.’s (1995) argument that the complexities of the software development industry require a more facilitative approach to ethnographic research, such as through the ‘quick and dirty’ method which was employed for this study.

While project planning meetings and work group meetings with developers are a common part of my everyday work, I have never taken such close scrutiny of how developers communicate when they are discussing technical problems or how they behave during pair programming sessions. Even though I had several preconceived notions of the practices which may be found in the organizations, my perspective towards the research topic was shaped immensely by the actual empirical research. In the words of Gadamer, a “fusion of horizons” – a fusion between my horizon as a researcher and the participants’ horizons – had taken place throughout the course of the study.

6.2 Suggestions for future work

This study has examined knowledgeability in smaller agile settings. A suggestion for future work would be to extend the research into larger organizations and compare how knowledgeability is generated and sustained in larger enterprises. In addition, future studies could shift the focus to distributed project teams. The findings from this study show that verbal communication and face-to-face interaction are important practices engaged by both organizations. It would hence be interesting to explore how teams which are geographically dispersed interact with one another to generate “knowing how” and examine how technologies are used to facilitate this process.

Another suggestion would be to conduct the ethnographic research over a prolonged period of time, for example, over a few months. Extending the research would likely identify other practices and insights which have been overlooked by this study.
7. References


**Websites**


Appendix A - Interview Questions

1. How long have you worked as a software developer?
2. How long have you been working in this company?
3. How do you learn from a more experienced team member?
4. Is mentorship being practiced in the company? Do you/ did you have a mentor?
5. How do you share what you know with other team members?
6. Is storytelling or giving historical examples practised in the team?
7. Do you use metaphors, analogies or any specific terms which you believe can only be understood by members in the team? Could you give examples or elaborate?
8. How do you think knowledge or skills are achieved?
9. What actions do you think are necessary in order to build up your competence and skills?
10. How do you usually solve problems (do you seek help from a team member or do you turn to documents or a database)? Could you elaborate?