Knowledge sharing in and between agile software development teams using knowledge practices
– An interpretive case study at a medium-sized medical IT company

Kunskapsdelning inom och mellan agila utvecklingsteam med hjälp av kunskapsinitiativ
– En fallstudie av ett medelstort IT-företag i medicinska sektorn

Adrian Sidenvall

Supervisor : Carl-Johan Petri
Examiner : Alf Westelius
**Upphovsrätt**


**Copyright**

The publishers will keep this document online on the Internet – or its possible replacement – for a period of 25 years starting from the date of publication barring exceptional circumstances. The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/her own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/.

© Adrian Sidenvall
Abstract

Agile methods for software development have become popular, especially since the agile manifesto was written in 2001. Many positive effects have been found in organizations using agile methods, but also several dangers. Communication and collaboration in teams is often mentioned as something that works well with the methods, but interactions between teams are often lacking. Since agile teams are cross-functional and focused on products rather than specializations, knowledge becomes spread out in the organization. Within teams, different members often have different deep knowledge, but instead have a lot of knowledge about their products in common. This allows them to discuss knowledge related to the product well, but limits their possibility to discuss advanced topics and experiences regarding their deep knowledge or specialization within their team. These issues are important to consider when applying agile methods in organizations, and the research about the issues is quite thin.

In this research, I have taken an interpretive approach and carried out a case study at the development department of a medium-sized IT company providing large software systems for the healthcare industry. Three different teams have been studied through interviews with all members as well as observations of agile practices like daily meetings. Further, I have studied three different knowledge practices that were used to complement the creation and sharing of knowledge that happens in teams. These three complementary knowledge practices had different forms and handled knowledge in different ways. One was closely related to what literature often calls communities of practice, which are groups where members share an interest and interact to deepen their knowledge. In this practice, meetings were used to discuss experiences and knowledge about topics within specific areas. Another was more focused on one-way communication through presentations and reading tips, spreading more basic knowledge to a wider audience. The third complementary knowledge practice let employees use 12 work hours every sixth week to do whatever they wanted that related to their knowledge, allowing them to e.g. explore new technologies and be creative or simply read up on some interesting topic.

My results show that agile teams support some sharing and creation of knowledge, especially through having members work closely to each other and share experiences, and through practising their skills in daily work, with help from each other when necessary. Like other research has shown, a lack of practices for interactions between teams was found in the agile methods. Such interactions were considered crucial since I found teams to be comparable to theories about communities of knowing, where teams create strong perspectives, the sharing of which is important for utilization and creation of knowledge. The complementary knowledge practice that related to communities of practice was shown to be good for connecting employees with similar specializations, who would normally be separated in different teams. This allowed for creation and sharing of knowledge as individuals needed to explain their experiences and could combine knowledge from different members of the community. The other two complementary knowledge practices were shown to be good for increasing motivation to create and share knowledge, and showing that the organization valued the knowledge of individuals. I argue that combining the use of agile teams with a few complementary knowledge practices like those studied can create an environment that supports creation and sharing of knowledge about what the teams are working with as well as specializations, and creates motivation and inspiration for doing so.
Acknowledgments

I want to thank the development department where this case study and thesis work has been carried out. The department and the entire organization has been welcoming and helpful for the entire duration of the study, and I thoroughly enjoyed being there. I want to give a special thanks to the department manager who supported me by offering his time for discussions and interviews, and by giving me a lot of freedom in doing my research. I also want to give a special thanks to everyone else that I interviewed, without the help and openness of whom, this study would not have been possible. I further want to thank Alf Westelius and Carl-Johan Petri for their great support through weekly discussions that were immensely helpful, and their thoroughness in reading and giving feedback on this report. My opponent, Fredrik Löfberg, also contributed with further feedback on the report, for which I am very thankful.
## Contents

Abstract ................................................................. iii
Acknowledgments ....................................................... iv
Contents ................................................................. v
List of Figures .......................................................... vii
List of Tables ............................................................ viii

1 Introduction ............................................................. 1
   1.1 Background ..................................................... 1
   1.2 Purpose ......................................................... 2
   1.3 Scope ............................................................ 2

2 Method ................................................................. 4
   2.1 Approach ......................................................... 4
   2.2 Before entering the field ...................................... 5
   2.3 Entering the field .............................................. 7
   2.4 Data collection ................................................ 9
   2.5 Analysis ........................................................ 12

3 Frame of reference .................................................. 16
   3.1 Agile Software Development background ................... 16
   3.2 Agile methods in practice ..................................... 18
   3.3 Knowledge Management background ........................ 19
   3.4 Designing knowledge practices .............................. 22

4 Creation and sharing of knowledge at MedTech ................... 24
   4.1 Case company background .................................... 24
   4.2 Interview and observation results ........................... 29

5 Analysis ............................................................... 37
   5.1 Teams ............................................................ 37
   5.2 Tribes and inter-team interactions ........................... 40
   5.3 Guilds ............................................................ 41
   5.4 Explore & Share ................................................. 43
   5.5 RAD Friday ..................................................... 43
   5.6 How the forums work together ............................... 44

6 Conclusion ............................................................. 47
   6.1 Limitations ...................................................... 48

7 Discussion ............................................................. 50
List of Figures

4.1 The structure of the development department with tribes, teams, and guilds. Teams are represented by the coloured circles. 26
List of Tables

4.1 Team members of team A, the distributed team. . . . . . . . . . . . . . . . . . 29
4.2 Team members of team B. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
4.3 Team members of team C. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
In this chapter, the background and purpose of this thesis work are presented, which leads to my research questions. At the end of the chapter, the scope of the study is defined.

1.1 Background

Agile methods for software development have become very popular thanks to their ability to respond to the turbulent software industry. Agile methods were originally developed from what used to be called iterative or incremental development (Larman and Basili 2003), and also took inspiration from Japanese approaches with focus on knowledge management (Schwaber and Beedle 2002; Takeuchi and Nonaka 1986). In contrast to traditional methods where detailed plans are made at the beginning of a project, agile methods let developers create the software piece by piece and get feedback from customers often (Schwaber and Beedle 2002). They also avoid unnecessary documents and instead focus on interactions (Fowler and Highsmith 2001).

When using agile methods, developers work in cross-functional teams and are supposed to be co-located and situated in the same room without barriers between them (Schwaber and Beedle 2002). The team is given joint responsibility for their products and work with collective ownership, which requires and builds trust and communication in the team (Chau, Maurer, and Melnik 2003). This teamwork provides support for the social aspects of knowledge sharing, as team members know the strengths and weaknesses of one another well and want to help each other do well since it helps the team (Chau et al. 2003). However, what has been identified as more difficult is to manage knowledge between different teams in an organization of several different teams working with different products or projects (McDermott 1999).

The field of knowledge management discusses how organizations can manage knowledge to make it easier for individuals and groups to create, share and understand knowledge (Davenport, De Long, and Beers 1998). There has been plenty of research done in the field during the last 25 years, where many different theories have emerged. Knowledge can be defined in several different ways depending on where
one believes it resides, such as in objects, in processes, or in the minds of individuals or groups (Alavi and Leidner 2001).

The research about agile methods is still in a quite early stage and the main focus is still to become convinced that agile methods actually work and increase effectiveness. Some studies dig deeper into how agile methods affect and handle knowledge management focusing on different aspects like documents (Melnik and Maurer 2004), communication (Pikkarainen, Haikara, Salo, Abrahamsson, and Still 2008), learning (McAvoy and Butler 2007), and trust (McHugh, Conboy, and Lang 2012). Some have also started to investigate issues with barriers to knowledge sharing with agile methods (Ghobadi and Mathiassen 2015; Heeager and Nielsen 2013), and others investigate how inter-team knowledge sharing can be achieved (Santos, Goldman, and Souza 2014; Kuusinen, Gregory, Taylor, Sharp, and Barroca 2016). Research is signalling that complementary practices for sharing and creation of knowledge between teams are necessary, as there is not enough built-in support for this in the agile methods (McDermott 1999). I will call such practices complementary knowledge practices, referring to that they complement the agile methods where they lack support for creation and sharing of knowledge. Common ideas for such practices are what can be defined as communities of practice (Wenger, McDermott, and Snyder 2002; Brown and Duguid 1991), where individuals working with similar things can discuss their experiences, as well as different types of presentations and seminars with focus on one-way knowledge sharing.

1.2 Purpose

The purpose of this study is to investigate how knowledge is shared in and between teams in an organization with several teams using agile methods and complementary knowledge practices. Data are presented from a case study at a medium-sized company providing large IT-products with biannual releases for the medical sector who have been using agile methods for several years and have three major complementary knowledge practices that are used by most employees. The case study is focused on the knowledge environment within teams as well as between teams through the complementary knowledge practices, and efforts are made to better understand how teams and knowledge practices complement each other. The purpose is also to provide general analysis as to what is important for knowledge sharing in agile teams and knowledge practices in agile environments, some of which can also be applied in non-agile teams and environments. The conclusion presents the answers that were found to the following questions:

- How does use of agile methods and cross-functional teams support creation and sharing of knowledge?
- What support can be provided through complementary knowledge practices to improve knowledge sharing in an organization with several cross-functional teams?

1.3 Scope

This research has taken on a mainly qualitative approach based on a case study at a single organization. Data has been collected mainly through interviews and observations, which has then been analysed in relation to literature that I have found to be relevant and interesting. This approach limits the scope to what has been available to study at the case organization as well as the literature that I have found and read.
I chose to focus on qualitative aspects in my study and only did some basic quantitative analysis, which limited my results on some of the studied aspects that would have needed more quantitative data collection. Instead, I focused on social aspects within the fields of agile development and knowledge management, investigating perceptions of individuals working in agile teams in an organization with a clear focus on knowledge.
For the main part of this study I was situated at the case company, MedTech, where I collected my data. There were several steps involved, both before I got access to MedTech and during my study there. These steps are described in this chapter and explanations of my choices are made. The chapter is organized in chronological order where applicable, starting with the choice of focus and company and a pre-study, and going on to what I did when entering the field and my methods for collecting and analysing data. First however, my general approach is described.

2.1 Approach

Since I had no personal experience from working in agile software development teams before this study, I found that I needed to understand the experiences and perceptions of others to make any conclusions about them. I chose to do a case study of a few teams at only one case company to allow myself to spend the time that was necessary to understand how those teams worked and the perceptions of all the team members. Even though I only got data from one company in this way, I could still compare it to existing literature describing other cases. I preferred qualitative data collection methods, like observations and interviews, over quantitative ones, like surveys and database analyses, because I wanted to understand how things were perceived rather than simply what was being done. Some quantitative analysis was still done by comparing opinions between interviewees and grouping them to see which opinions and perceptions were most common in the group that I interviewed. More quantitative data collection could have been done to get a more complete picture of some studied aspects, as well as to allow for more triangulation.

When investigating the perceptions of others, interpretations are difficult to avoid, and thus also subjectivity. Like several other researchers in similar fields (e.g. Pikkarainen et al. (2008) and Heeager and Nielsen (2013)), I found Klein and Myers’s (1999) principles for interpretive field studies to provide excellent support to handle this subjectivity and help me consider necessary perspectives, like how my interactions with subjects could affect results. Since their work is well-cited and matched
my needs and personal thoughts well, I did not investigate alternative theories much. However, I also used the more hands-on descriptions for doing interpretive research by Walsham (2006), which helped me put Klein and Myers’s principles in practice. Furthermore, I tried to combine different theories with each other as well as with my own data, for which I used Eisenhardt’s (1989) framework for building theories from cases. Her work is also well-cited and I found it to be solid and straightforward.

2.2 Before entering the field

Before I got access to the case company, or even found a case company, I needed to do some work and thinking. This is described here.

2.2.1 Choosing a main focus and case company

First of all, the main focus and case company of the thesis work needed to be decided. I chose the main focus – agile methods and knowledge management – because I found both areas to be interesting and that the integration of them seemed relevant at the time. I discussed this main focus with a professor at Linköping University, Alf Westelius, who has courses that discuss knowledge management, and he supported the idea and contributed with feedback on my thoughts. I also did some primary searches for literature regarding the combination of agile methods and knowledge management and found that there seemed to be a need for more studies in the area. For example, researchers expressed needs for studies where knowledge sharing and factors like trust is investigated in more and less agile teams (Ryan and O’Connor 2013), investigations of inter-team knowledge sharing in enabling contexts (Santos et al. 2014), and the development of strategies or frameworks for inter-team knowledge sharing in agile software development (Heeager and Nielsen 2013).

Next, I needed to find a case company where I could carry out the field work. My main criteria for this choice was that the company should have been using agile methods for a reasonably long time, meaning that they would have adapted the methods for their situation and that there were teams that had been working together for some time. I also wanted a company that had put some thought into knowledge management and had procedures for this that I could investigate. As I wanted to contribute to the existing literature with theories about the combination of agile methods and knowledge management, a case company with interesting procedures to investigate was preferred over any kind of statistical representation of the situation in most companies. This could be compared to theoretical sampling where cases are “selected to replicate or extend emergent theory” as described by Eisenhardt (1989, p. 537). Though the emergent theory at this stage was more like ideas than actual theory, I wanted a case company where I thought I could find interesting things to investigate.

During the years prior to the study I had developed an interest for a company that I will call MedTech, which is a medium-sized medical IT-company. I knew that they had been using agile methods for several years and had stated goals regarding knowledge management. I contacted them with a proposition about my thesis work and had a meeting with the manager of the department that developed all their products for the medical sector. During the meeting, I found out about how they had introduced and developed their use of agile methods, and three different complementary knowledge practices that were active and worked reasonably well, according to the manager. The manager also expressed an interest in innovation and requested me to add that as a focus for the thesis work. I was offered to do the study at MedTech and be positioned there with good access to employees for interviews and observations. We later decided that I would focus on 3 teams of which I would interview all members. The thesis work
was agreed upon about 5 months before the field study would commence, which left plenty of time for a pre-study.

It should be noted that for the entire report, when MedTech is mentioned it is an anonymized name for the development department of the medium-sized IT-company, not the entire organization. More details about MedTech will be described in chapter 4.

2.2.2 Pre-study

The pre-study consisted of reading relevant literature and taking notes of what I found to be interesting. It was done part-time during the autumn and winter of 2016 and full-time for 2 weeks at the beginning of January 2017, during which time I did not yet have access to MedTech. The purpose of the pre-study was to get a better understanding of the fields that I was going to study and how they are connected, as well as methods for doing such studies. The pre-study was limited to reading and analysing literature as no empirical work could be done yet. The pre-study was necessary because I needed to get some ideas for what could be the most interesting and relevant areas to study, as recommended by Eisenhardt (1989), and how these areas best could be studied. I was new to the fields and to doing case studies, which meant that plenty of reading was necessary to get a broad understanding of the available literature and see similarities and conflicts in different findings. Having a broad base of literature later made it possible to tie my emergent theory to existing literature to increase internal validity, generalizability, and theoretical level, as suggested by Eisenhardt (1989).

To find relevant literature I used keyword searches on Google Scholar with different combinations of keywords like: “agile”, “knowledge”, “management”, “sharing”, “creation”, “innovation”, etc. This resulted in lots of hits and I proceeded with reading the titles to create an idea of what the papers were about. For any papers that seemed like they could be relevant, I read the abstract. For the papers that still seemed relevant, I read the entire paper, or at least the conclusions and other sections that were necessary to understand the findings of the paper. For the area of knowledge management, I tried to get a broad view of different ways to view knowledge and how it can be managed. I avoided getting locked-in on any particular theories by comparing different theories to each other and analysing their conflicts and similarities. For agile methods, I read some basic literature on what agile methods are, such as Agile Software Development with Scrum by Schwaber and Beedle (2002), but I mainly focused on different studies of how they work in practice, with special focus on studies with findings that seemed relevant from a knowledge management perspective.

For all literature, I took notes of what I read, my impressions, and how it could be relevant for my study. I also made highlights in the text. The notes varied from short summaries of the relevant findings of that paper to longer reflections of how the findings could be interpreted and adapted to the kind of situation that I was going to study. I always did at least some reflection of how each article could be relevant for my study, even if I did not find the article to be interesting. This helped me in keeping an open mind for different theories, as described important by, e.g. Eisenhardt (1989). When any paper that I read had a reference to something that I found to be interesting, I read the abstract of the referenced paper and handled them in the same way as other articles. For each paper, when I read some finding that was in any way conflicting with my previous conceptions, I got a chance to question and develop my conceptions through dialogical reasoning, as suggested by Klein and Myers (1999).

As a result of the pre-study, I learned more about what would be interesting to study. This led to a research question beginning to form, which was not meant to be a final research question but rather a broad question for myself to know what I should focus on, as suggested by Eisenhardt (1989). At this stage, the research question was
something like: “How does agile methods affect knowledge management for the entire organization?” This question evolved throughout the study.

2.3 Entering the field

I got access to MedTech toward the end of January 2017 and had a desk and computer there. During the first few days I got to present myself to everyone that was available at the department and shortly describe what I was going to study and answer any questions about that. This made almost everyone aware of who I was and why I was there. I also had some meetings to discuss the organization and thesis work and listened to some presentations about the company, its history, its products, and its different processes. These were held by some different employees, including the department manager, developers, and product owners. These meetings and presentations helped in building a preliminary view of the context through a historical and social perspective, as Klein and Myers (1999) describes to be important. However, the perspectives I got here were not in any way viewed as definite truths, but were to be questioned throughout the study.

2.3.1 Style of involvement

As I was beginning to interact with employees at MedTech, I tried to keep in mind how these interactions with subjects could affect the study (Klein and Myers 1999). Walsham (2006) suggests a spectrum of styles of involvement ranging from “neutral” to “closely involved”. I would place myself quite close to the latter, as aside from the purpose of this thesis work I also tried to improve the situation at MedTech through my analysis of the current situation. However, I did not participate in the work of the subjects, but rather tried to be a silent observer as much as possible. Further, as a student and novice in the fields, I could be seen as fairly unbiased, but I was under a contract with MedTech and was given some compensation for the study which could be seen as a source of bias toward making managers at MedTech happy. It is also worth mentioning that my supervisor at MedTech was the department manager, who was also the manager of all my subjects, which meant that I could be perceived as working for him. The department manager was clear that he did not want me to review individuals in any way but only analyse the situation regarding agile methods, knowledge management, and innovation. I tried to be clear to the subjects that this was the case. The possibility that subjects perceive the researcher to have a vested interest is mentioned by Walsham (2006) as a possible disadvantage of close involvement, together with that it is time-consuming and that you can be affected by the views of the subjects. Some advantages can be gained if the subjects see the researcher as someone who is trying to help them, which can make them more open. I found that I could sometimes help subjects by providing advice and discussions, which I perceived to make them more willing to help me by being open. I kept in mind that what I said to subjects could affect what they said to me, but I considered that I said so little that it would not have any major effects on my results.

2.3.2 Taking notes

For the entire duration of the study I was careful to write down as much as possible, even things that did not seem important at the time. This is suggested by Eisenhardt (1989) who says that you should write down impressions of everything you encounter. This did not only make it easier to build the context and remember what I had seen and thought, but it also helped me in being more aware of what happened around me and what I could learn from it. By asking myself what I was learning from the notes I was
taking and how notes differ from each other, I continuously improved my note-taking while simultaneously getting a good start for the analysis (Eisenhardt 1989).

2.3.3 Selecting subjects

In selecting participants, or subjects, for the study, I used theoretical sampling (Eisenhardt 1989). This meant that I chose subjects based on which ones would give the most interesting findings, rather than randomizing the selection to gain more statistical validity. To do this, I had discussions with the department manager about which teams that would have the most differences and that would capture as many interesting features as possible. The features that were considered were what agile method the team based themselves on and their enthusiasm to the method, whether the team was co-located or distributed, and how they used tools and practices like scrum boards and retrospectives. The result was 3 teams with 4-5 members each that used agile methods in varying ways and whereof one team had its members distributed between two cities in Sweden.

2.3.4 Within-case analysis

A vital part of any interpretive field study is the within-case analysis (Eisenhardt 1989). I applied this by comparing different pieces of literature to each other and different collected data to each other, as well as comparing literature to my own data. For data from interviews, I wrote down any similarities and differences between the perceptions and opinions of different interviewees and tried to analyse what these multiple interpretations (Klein and Myers 1999) could mean. I did this using a document that I created after the first interview where I made short descriptions of perceptions and opinions that I thought could be interesting from that interview. When I conducted more interviews I simply added that interviewee’s name to an opinion or perception if they expressed similar thoughts, or described how the new interviewee’s perceptions and opinions differed or conflicted if that was the case.

The document with similarities and differences from interviews was useful when finding and analysing literature as I then had a list of things that could be interesting to find connections to. When there was an opinion or perception that came up often in interviews, I knew that was an interesting area that I should find literature that connected to. Sometimes I had already read some papers about that area, and sometimes I tried to find new ones. In any case, by considering the conceptions that I had built in interviews and taking a step back to include the experiences and conclusions from other researchers and cases helped me in building conceptions of the whole of the situation as well as the parts. This process of continuously seeing new perspectives and allowing them to change my conceptions, which builds my conception of the whole of the situation, which in turn changes my conceptions of the parts, corresponds well to what Klein and Myers (1999) describe as the Hermeneutic Circle.

Since I studied 3 different teams I could view them as different cases and thus search for cross-case patterns, as suggested by Eisenhardt (1989). Sometimes it was also relevant to make other divisions of subjects and view them as cases, e.g. testers and developers, or more and less experienced employees. I mainly applied this by building perceptions of different cases and thinking about what differences I could see between them, rather than making any direct comparisons of data between cases. Thinking about differences in this way helped me to think about different perspectives and seeing things that I would not have noticed otherwise (Eisenhardt 1989). In a few comparisons, I made more structured analysis of what e.g. members of different teams had said that created patterns, but mostly I trusted my perceptions and used the comparisons to find interesting aspects to study further.
The within-case analysis also allowed me to see emerging patterns that could be interesting to investigate further. I could then adapt my studies based on these findings by for example doing other observations, asking other questions, or interviewing other people. For example, I removed a question that I found was difficult to interpret and answer after the first interview, and I also adapted my follow-up questions to get answers that were easier to compare to answers from other interviewees as well as literature. According to Eisenhardt (1989) this process of changing the method along the course of the research is allowed and even favoured when building theory from cases. In all interviews, I had the final question: “Who do you think I should speak to who would give a perspective that is completely different from yours?” Even though I could not interview everyone that was mentioned, this gave me insight into where I could find different types of persons in the department, and also how the interviewee viewed themselves in comparison to others at the company.

2.3.5 Enfolding literature

Adding new literature in the within-case analysis is supported by Eisenhardt (1989) who says that it is perfectly valid to add literature during the study when building theory. This was important to me since I did not know what would be most interesting by the start of the study and thus did not know what literature I should read. For example, I realized during the study that the concept of communities was more important for the case than I had realized in the pre-study, so I searched for, and found, more theories and field studies about that. Klein and Myers (1999) state that connecting theory to interpretive field studies is crucial, even though some say that generalizations cannot be made outside of the context. Lee and Baskerville (2003) elaborate further on how it can be relevant to generalize different results in different ways. When enfolding literature in my within-case analysis, I used generalizations from data to description to compare the results of other studies to my findings, and generalizations from theory to description to apply and try theories of others in my context.

2.4 Data collection

The data collection was performed through two main sources: observations and interviews. The results from these were combined to create the data of this study. The choice of data sources was based on that I wanted to see how the situation was for myself and also hear impressions of employees to be able to compare my impressions to theirs. Interviews are a very common data source for interpretive case studies, which is also supported by Eisenhardt (1989) as well as Walsham (2006). Both authors also recommend having multiple data sources to be able to triangulate the results, where both mention observation as another good source. Both also state that it is favourable to combine the qualitative data with quantitative data, for example from surveys. I considered using a survey but did not find an area where I thought it would create much value for the study. I also investigated an online resource for discussions at MedTech but found it to have too little activity to be interesting to study further.

2.4.1 Observations

I performed direct observations of the agile teams that I studied as well as the complementary knowledge practices. I chose direct observations, where I was not part of the event that I observed, over participant observations, where I would have participated in the event, mainly because the events generally included discussions regarding advanced knowledge that I was unable to take part in because I did not have the proper knowledge. Because of that, I decided to only focus on what the subjects were doing
since attempts to participate probably would not have given me the same experience as that of the subjects, and thus would not have contributed to the results in a meaningful way.

In my direct observations, I tried to position myself in a way that I would not be perceived as a part of the event that I was observing, and from there I took notes in silence to disturb as little as possible. My impression was that my presence and note-taking did not affect the subjects noticeably and they seemed relaxed. I chose not to record the events that I observed because recording could potentially make the employees being observed uncomfortable, and notes were sufficient. In general, I tried to avoid interactions with the subjects during the observations, as this could make the subjects interpret what I was saying or doing and thus possibly change their behaviour (Klein and Myers 1999).

I began by observing the daily meetings in the three teams that I studied, mostly to get a feel of the atmosphere of the groups. I also observed the agile practices of Scrum of Scrum meetings and a sprint demo. I chose to make some observations before I started doing interviews because I wanted to see for myself what seemed to be interesting before I set up questions. This allowed me to avoid asking some questions that the interviewees would feel had obvious answers or that they could not answer, and instead ask a bit more thought-provoking questions. Further, observing social interactions rather than just asking about them gave me different perspectives that I could compare to build my conception of the social context that Klein and Myers (1999) suggest is important to understand in interpretive research. In some cases, interviewees had perceptions of practices that did not match the perceptions that I had gained in observations, which made me think about the reason for that mismatch, and ultimately get a more complete conception. I usually did not discuss these mismatches of perceptions during interviews because it could affect other answers by the interviewee, but in a few cases I discussed them later if I wanted to get some elaboration on their perception.

I also wanted to observe some complementary knowledge practices and investigated which ones would be most interesting to study at MedTech. Through discussions with the department manager I found three complementary knowledge practices that were central at MedTech, which were called guilds, Explore & Share, and RAD Fridays. Guilds mainly consist of meetings to discuss interesting topics within a defined area for that specific guild. Explore & Share consists of presentations and reading tips from employees. RAD Friday is an event that occurs once every six weeks where all employees get 12 work hours to do whatever they want that somehow relates to their work and/or knowledge. I chose to observe these 3 complementary knowledge practices because I found them to be used actively and because they seemed to be important for the knowledge management at MedTech. I use the names for the complementary knowledge practices that are used at MedTech to avoid confusion for readers that are familiar with these names. Guilds are for example also used at Spotify, see Kniberg and Ivarsson (2012).

When observing these practices, as well as in any other observations, I focused on how knowledge was shared, how things were discussed, if there were any unspoken social rules or hierarchies, etc., rather than what knowledge was shared. However, when I observed meetings in a guild discussing agile methods, where it was sometimes discussed how different teams used agile methods, the knowledge being shared was also interesting for my research. Still, I had my main focus on how knowledge was shared.

During my study, guilds emerged as a complementary knowledge practice that I found to be more central for knowledge sharing at MedTech than the others. Guilds were also easier to observe than Explore & Share and RAD Friday because they occurred more often and contained more interactions. Because of this, I collected more
data about guilds and analysed them more, which is evident in the rest of the report. There were also a few online-based knowledge practices that I chose to not focus on because I judged them to not be used with enough activity to be interesting to study.

2.4.2 Interviews

I did semi-structured interviews with all members of the three teams. These interviews covered all areas that I believed could be interesting to study and were meant to be scouting rather than to go into details. There were 13 team members in total and thus 13 interviews, which ranged from 45 minutes to 2 hours depending on how long answers the interviewee gave. I limited myself to doing one interview each day to get sufficient time to process that interview and avoid mixing up what different interviewees had said when thinking about follow-up questions.

The questions for the interviews were based on a combination of what seemed relevant for knowledge sharing in and between agile teams from the literature and what I had perceived to be interesting in my observations. I started each area of questions with one or two ambiguous and open questions, such as: “What do you think is the intention with having Guilds?” I did this in part to make the interviewee think a bit about what they actually thought about the event in question, and in part to get some thoughts from the interviewee before I asked more detailed questions that could affect how the interviewee thought. When I had received those first impressions about the event, I went on to ask more detailed questions that made the interviewee think about different aspects. I also adapted the questions during the interviews, skipping questions that seemed irrelevant for that interviewee or that had already been answered, and adding follow-up questions when the interviewee had interesting answers. I kept my sheet of questions identical for all interviews except the first one. After the first interview, I changed a question from focusing on implicit roles in teams to whether it was obvious who would do what in the team, because I found that it was difficult to separate implicit roles from explicit ones. I also added a question about involvement from product owners that I had not thought about beforehand but got some interesting thoughts about from the first interviewee.

I approached the interviewees through email invitations where I explained the nature and purpose of the interview, and informed them that I wanted to record the interview if they allowed it and that I would handle the results anonymously. Walsham (2006) gives some suggestions for interviews in interpretive research and stresses the importance of keeping the interviewee, as well as the organization, happy. He suggests that this can be done by being clear to the interviewee about purpose and confidentiality and to provide feedback to the organization. I tried to apply this as much as possible in the invitations and interviews by being clear and honest, and avoiding questions that would make interviewees uncomfortable. If I still found interviewees to be uncomfortable with questions, I did not pressure them for answers. I generally felt like the interviewees were comfortable and open and that they wanted to help me, possibly and partly because it could help the organization. I did not have any problems with conflicts between what would make the organization and the interviewees happy since the organization, represented by the department manager, was clear with that the employees and what they thought was most important. They also gave me freedom to do the study the way I thought best, which allowed me to adapt the study to make interviewees happy whenever possible without compromising the results. It is possible that I could have gotten some other answers if I put more pressure on the interviewees, but I found it valuable that the interviewees rather wanted to help me.

Walsham (2006) discusses recording of interviews and states that it has pros and cons, where the main pros are that you get a complete record of what was said that you can go back to and that you can concentrate on the interviewee, while the cons are
2.5 Analysis

that it might make the interviewee less open, it does not capture non-verbal elements, and making transcriptions is time-consuming. Since I was doing the interviews alone I would have to take extensive notes during the interview if I did not record. I decided that possible losses in openness from interviewees would be made up for by me being more engaged in the interview while still getting records of exactly what was said in each interview.

When doing semi-structured interviews, Klein and Myers’ (1999) principle of interaction between the researcher and the subjects becomes central. Since I wanted to build and maintain trust from the interviewees, I tried to be fairly open about my thoughts if interviewees asked for them. This of course resulted in interviewees interpreting and analysing what I said, which could lead to them changing their opinions or saying other things than what they would have said otherwise. However, not answering questions about my thoughts could possibly also affect the answers from interviewees, where they might guess what I thought and adapt to that or choose not to be open about their thoughts because I was not open with mine.

After the interviews, I listened to the recording as soon as possible to have fresh memories of my impressions of how the interviewee said things, what body language was used, and such, which minimized the con with recordings not capturing non-verbal elements mentioned by Walsham (2006). This usually meant that I started listening to the recording within 30 minutes after the interview, but in a few cases, it had to wait until the following morning. I chose not to transcribe the interviews because of time constraints, and instead I took notes, that still were quite extensive, of the opinions, thoughts, and examples that the interviewee expressed. After listening to the entire recording, I read through the notes and made a new document of notes of the most interesting results, with some reflections, of that interview. This process created a source of subjectivity, but as Walsham (2006) suggests, the mind of the researcher must be used as a tool for analysis in interpretive research. I did however discuss my findings from the interviews with my mentor at the university, who could reinforce or challenge my interpretations. Since my mentor had not listened to the interviews, he relied on my explanations of what I found to be interesting, and I used the discussions to get some input on whether he thought the same things were interesting and to hear his interpretations to see if they differed from mine. If our interpretations differed we discussed further, which allowed me to see the cause of the differences, possibly changing my interpretation or seeing another perspective.

After the analysis of a particular interview, I compared the findings to the findings from other interviews and created a document with similarities and differences as described in section 2.3.4 about within-case analysis above. I did this to try and find patterns and conflicts, as well as multiple interpretations, as suggested by Klein and Myers (1999). Doing these comparisons brought two major advantages. First, seeing these patterns and conflicts helped in building the context as well as seeing what would be most interesting to study further, sometimes making me adapt follow-up questions for coming interviews. Second, reflecting on different interpretations among interviewees and confronting and studying these further made me view things from different perspectives, which led to changes to my conceptions of different situations and the whole, similarly to dialogical reasoning and the hermeneutic circle described by Klein and Myers (1999).

2.5 Analysis

Below, my method for analysis of literature and my own data are described, as well as how I decided when to stop adding literature and data.
2.5. Analysis

2.5.1 Literature

As I described in the section about my pre-study above, I found literature through key-word searches and from references in the papers I read. I tried to keep a broad range of literature, with papers that had similarities as well as differences to each other as well as to my own research, which is described as highly important by both Eisenhardt (1989) and Walsham (2006). This allowed me to analyse what caused these similarities and differences, and compare them to my own data to see which pieces of literature I found it to be most similar to. This process of putting different theories and results, my own and those of others, against each other helped building the internal validity, generalizability, and theoretical level of my work, as supported by Eisenhardt (1989). For example, if I found my subjects to have a negative opinion about some practice, I would create my own conception of what caused that opinion. If I then read about opinions about the same or a similar practice in someone else’s research, I could compare my conceptions to those described by the other researcher and try to understand their perspective, possibly changing my conception.

Klein and Myers (1999) discuss the basics of interpretive research and that interpretive researchers do not try to apply laws to human behaviour, but that this does not mean that the results cannot be generalized. Conclusions from one case can describe ideas that sometimes can be transferred to other cases, even though care must be taken for differences in the context between the cases, and especially culture. Lee and Baskerville (2003) take the issue with generalizability further and describe a framework with four different types of generalizations that can be valid in different situations. This framework was also referred to by Walsham (2006) as working well for interpretive case studies. I found all four types of generalizations to be useful for my research, both when generalizing from literature and when considering the generalizability of my own research.

I used generalizations from one set of empirical statements to another, called Type EE by Lee and Baskerville, frequently to compare my situation and results to results that I read about in the literature. When I did such generalizations, I always tried to consider the differences in context between the situations. I thought about things like effects of culture differences between situations, different types and sizes of companies and products, and possible bias by the other researcher. I found that it often felt relevant to compare my case to other cases, either because there were similarities or differences and conflicts. However, I always applied critical thinking when doing these generalizations because I assumed that there are no laws or absolute truths when describing human behaviour, as described by Klein and Myers (1999).

Even though I do not consider laws about human behaviour to work in all cases, there are still many theories trying to describe common ways for humans to think and act. Generalizations from such theories to my empirical statements, called Type TE generalizations, were useful for me to test whether the theories seemed to hold in my case and to help make better descriptions of the case. For example, I used theories about different types of knowledge and modes of knowledge creation to describe what knowledge was created and how it was done in the case. I mostly focused on well-cited theories that had been confirmed to work well by many other researchers, which made them more likely to work for me. However, for theories about human behaviour there are no absolute truths, and I always questioned the theories and compared them to each other as well as to my own results.

To build my theory, I needed to be able to say that the results of my case study can be relevant in other settings, and thus generalize from empirical statements to theory called Type ET generalizations. To make such generalizations meaningful and reliable, I needed to make detailed descriptions of the context of my empirical statements to allow readers to understand how that context could have affected my results. Any
researchers or practitioners who make use of my theories should compare their con-
text to mine and consider differences and effects of these, similarly to how I considered
differences when doing Type EE generalizations from other cases.

Finally, I made generalizations from theories of others to build my own theories,
called Type TT generalizations. My study was not very big or deep, for example being
limited to just a few teams in a single organization, which limits the generalizability
of my resulting theories. By building my theories on previous theories that have been
recognized by other researchers, I could make extensions rather than completely new
theories. That way my theories could become more reliable and easier to generalize
from.

When choosing literature, there will always be a source of subjectivity, as I cannot
read all existing literature in a field and will naturally not weigh each publication
equally. I rejected many papers based solely on their title, and some on their abstract,
because I did not see how they would be relevant for my study. They might however
have been relevant from someone else’s perspective. Walsham (2006) defends such
sources of subjectivity and says that the researcher must choose literature that inspires
them. This is something that has happened both naturally and consciously for me
where I have chosen to read more about theories and concepts that I found a personal
interest in. Therefore, there is subjectivity in my choices of literature, but I consider
it unavoidable and I have tried to judge literature fairly and to be honest with what
literature I have based myself on.

2.5.2 Data

When analysing the data I got from interviews, I focused on the general perceptions
about different events and practices and how these differed between subjects. I tried
to view the events and practices from their perspective and in that way change my
conceptions through dialogical reasoning, as described by Klein and Myers (1999).
For example, if a subject had an idea about what guilds were for and how they should
work that differed from mine, I tried to understand why they wanted them that way
and developed my understanding about what guilds were and could be.

When doing my interviews, I also found that for some events and practices, the per-
ceptions varied a lot between subjects. I found that such situations were an excellent
opportunity for me to understand the context and different opinions of subjects better
through considering what Klein and Myers (1999) call multiple interpretations. If, for
example, two subjects had completely different interpretations of what the agile prac-
tice of working in iterations was good for, I tried to understand how the context had
created that conflict. I could reflect on whether one of the subjects had been working
for the organization much longer and had experienced what it was like to work there
before they introduced agile methods, possibly giving him/her a different perspective.

Sometimes, there were perspectives and interpretations that varied depending on
which team the subjects belonged to, which gave me opportunity to consider if there
were differences in context between teams, e.g. team-specific culture. Since I inter-
viewed all members of three teams that were generally quite different, I could view
them as different cases and search for cross-case patterns, as suggested by Eisenhardt
(1989) when investigating multiple cases. If several members of one team had similar
perspectives about something while several members of another team had a different
perspective, I could try and understand how the contexts of the teams had influenced
those differences. The same method could also be applied to investigate how per-
spectives of those who had been working at MedTech before they started using agile
methods differed from those of newer employees.

When a researcher does interviews and observations, some thoughts and opinions
might not reach the researcher, possibly because the subjects intentionally hide them or
2.5. Analysis

because they are something that is so obvious to the subjects that they do not mention it. There could for example be unspoken rules and power structures that affect the subjects. To handle such hidden aspects, Klein and Myers (1999) recommend the researcher to think critically and to be suspicious of the subjects, trying to find signals that tell a different story than what is being said. I tried to apply this suspiciousness when doing and analysing interviews and observations, but generally found it to be difficult, possibly because of my lacking experience from doing case studies and being in a setting with several teams developing software.

Using the framework with four types of generalizability by Lee and Baskerville (2003), my research can be said to be generalizable to descriptions as well as theories of others. That is also supported by Klein and Myers (1999) who argue that cases can be related to or create ideas and concepts that can be applicable to other situations. To make it easier, more relevant, and more reliable to generalize my empirical statements, I tried to make detailed descriptions of the context where I gathered them. In this way, other researchers can draw their own conclusions about my empirical statements and decide how relevant they are in respect to their own research.

2.5.3 Reaching closure

When considering when to stop doing more observations and interviews, and when to stop finding more literature to iterate back to my data, I aimed for theoretical saturation, as recommended by Eisenhardt (1989). Theoretical saturation means that continued adding of data will not contribute to any noticeable learning, because the data is similar to data that has already been observed. For example, I applied this when observing daily meetings in the beginning of the case study. After a few weeks of observing these meetings most mornings in all three teams, I found that I was not seeing anything new, so I stopped observing those meetings. However, theoretical saturation is an optimal case that I could not reach for all sources of learning in my study. I could have learned more by doing more interviews with my subjects, studying more teams, observing more guild meetings, reading other literature, etc., but this could not be done due to time constraints. In these cases, I had to consider what would bring the most value in relation to the time it would take. Even though I would have liked to do more interviews, I decided that I needed more time to analyse the results of those I had already done and finding relevant literature to compare my findings to.
3 Frame of reference

In this chapter, I present relevant theories and results of studies regarding agile methods and knowledge management. The chapter is divided in sections about first agile methods and then knowledge management. Each of these sections begins with some background and basic concepts of the research area in question, and then goes on to studies about particular aspects of the areas, and especially topics that connect the areas.

3.1 Agile Software Development background

Agile software development is a collection of methods that all adhere to the agile manifesto that was created by the Agile Alliance in 2001 (Fowler and Highsmith 2001). The agile manifesto contains a set of values:

“We value:

• Individuals and interactions over processes and tools.
• Working software over comprehensive documentation.
• Customer collaboration over contract negotiation.
• Responding to change over following a plan.”

(Fowler and Highsmith 2001, p.2)

These values set the foundation for agile methods by setting some general prioritizations, each bullet point stating what is valued over something else. It is important to note that something being less valued than something else does not mean that it should not be done, i.e. that working software is valued over documentation does not mean that no documentation should be made. In this way, agile methods are “light” – there are few rules to follow.

The intention with agile methods is that teams using them should respond well to changes, and thrive in the turbulent and fast-paced software industry. This is achieved in different ways by different agile methods, but the core is to focus on developing
software by splitting it up in smaller pieces that the customer can provide feedback on, which the team can respond to quickly. In this study, I will focus on the agile methods called scrum and kanban because those are the ones being used at MedTech.

### 3.1 Scrum

Scrum is one of the most commonly used agile methods and is described in more detail by Schwaber and Beedle (2002). The word “scrum” comes from rugby and was first used to describe product development methods by Takeuchi and Nonaka (1986), who laid the grounds for what would become the method now called scrum. A rugby term was used because the development method is similar to a game of rugby in that “the ball gets passed within the team as it moves as a unit up the field.” (Takeuchi and Nonaka 1986, p.137). The ball being a metaphor for the product, the movement of the ball is reflected in that new functionality is added and improvements are made continuously, rather than through a rigorous process with different stages, like in Tayloristic methods such as the Waterfall model.

In Scrum, employees work in cross-functional teams with a recommended size of seven plus minus 2 (Schwaber and Beedle 2002). In the team, work is divided in iterations called sprints. Sprints are short periods of time, usually 2-4 weeks, where the team focuses on some functionality which has been decided in sprint planning at the start of the sprint. Functionality is taken from a list of things the final product is expected to do, called the product backlog, and placed in the sprint backlog. The sprint backlog is shown on a visual artefact called the scrum board. The scrum board contains all tasks sorted by priority and is divided in columns for stages of development, e.g. To Do, In Progress and Testing. The scrum board is originally intended to be physical, e.g. using a whiteboard with post-it notes, but it is becoming common to have a virtual online-based scrum board to allow it to be viewed and changed from multiple locations. During each sprint, the intention is that the team should not be disturbed by any tasks that do not contribute to the goal of the sprint, which is to finish the sprint backlog. By the end of the sprint, a sprint demo is held to show what progress has been achieved in the sprint to stakeholders and get feedback.

To support this process, there are several scrum practices. Short daily meetings are held in the team where team members tell each other what they have done since the last meeting, what they will do until next meeting and if there are any problems. The meetings are sometimes called daily stand-ups as they are held standing up to promote keeping them short. They are recommended to be time-boxed to 15 minutes. The team has a scrum master who is a member of the team who is tasked with protecting the team from disturbances and removing hurdles, in general making the team as effective as possible in meeting its sprint goal. The team also has a retrospective at the end of each sprint to discuss what has gone well in the process and what can be improved for the next sprint. It is common that scrum teams apply pair programming, which is when two team members work on the same screen and keyboard, where one writes the code while the other reviews it as it is being written and provides feedback through questions and comments. The pair then switches roles with regular intervals, often something like every 30 minutes. Finally, and maybe most important, are the practices called open office space and whole team. Open office space means that the entire team is located in one office without walls between team members, enabling communication. Whole team is a practice that urges all team members to take part in all team activities, such as daily meetings and sprint planning, making sure that the whole team is in sync. When there are several scrum teams in the organization, Schwaber and Beedle suggest holding scrum of scrum meetings where the scrum masters of the different teams share the status of their teams in a meeting similar to daily meetings.
3.1.2 Kanban

Kanban is another agile method which is less prescriptive than scrum, urging its followers to use fewer practices. Kniberg and Skarin (2010) describe how scrum and kanban relate to, and differ from, each other. In kanban, teams focus on visualizing the workflow, similarly to how it is done in scrum with the scrum board. What you do differently in kanban is that you put limits on amount of Work In Progress (WIP) in each workflow state, usually visualized as columns on the board. For example, if you have one column for development with a WIP limit of 3 and one for testing with a WIP limit of 4, you cannot move a new task to development if there are already 3 tasks in the column, and you cannot move a task from development to testing if there are already 4 tasks in that column. This requires the team to find what column creates a bottleneck and solve that issue to be able to decrease the time it takes for tasks to move across the board, which is a performance measurement for teams following kanban.

3.2 Agile methods in practice

In practice, agile methods are rarely used exactly as the original creators designed them, and neither are they meant to (Schwaber and Beedle 2002). Agile methods generally do not have a long list of rules, but rather a set of principles and practices that are suggested. Each team can then choose what works best for them. However, it is generally advised to follow the suggestions to begin with and then add and remove whatever the team thinks best once they understand the methods and practices better. Agile methods have received lots of positive responses from both customers, managers and developers, for example Laanti, Salo, and Abrahamsson (2011) report of good results at Nokia, even though there were difficulties in adapting the methods. There are also several critics of agile methods, often advising caution that agile methods do not work for every team and project (Turner and Boehm 2003; Coram and Bohner 2005). Depending on how agile methods are used, they can have a lot of different effects on individual teams as well as on an organization with several teams.

For this study, the first value in the agile manifesto, “Individuals and interactions over processes and tools” (Fowler and Highsmith 2001, p.2), is highly relevant. Cockburn and Highsmith (2001), two of the creators of the agile alliance and manifesto, elaborates on how and why individuals are important and state that “agility depends on trusting individuals to apply their competency in effective ways.” (Cockburn and Highsmith 2001, p.133). Whitworth and Biddle (2007b) found in their study of socio-psychological experiences in agile environments that the methods gave developers the feeling that they, as individuals, developed the product. This was attributed to agile practices helping developers feel that they had support and accountability, which made them more prone to use and develop their competence.

The use of team rooms has been found to be important for agile methods to work well. Levy and Hazzan (2009) suggest that team rooms encourage face-to-face communication within the team, which is supported by Melnik and Maurer (2004) who discuss further how important short communication chains and face-to-face communication is for teamwork. Team rooms, and the practice of whole team that encourages all team members to join in all team activities, also increase participation (Levy and Hazzan 2009). Since team members are situated in a team room, they will overhear conversations taking place in the room, such as if a member from another team or a customer representative comes in to talk to a team member. Ryan and O’Connor (2013) concluded that frequent high quality interactions, such as face-to-face discussions, improve the amount of shared tacit knowledge in the team. They mean that team members develop shared mental models which makes it easier for them to communicate, increasing effectiveness in the team. Whitworth and Biddle (2007b) does
however give a warning that while good and frequent communication can improve transparency and teamwork, it might also lead to stress and social exhaustion. Pikkarainen et al.’s (2008) results support that team rooms generally affect communication, and especially informal communication, in the team positively, together with other practices like sprint planning and daily meetings. They also found that this improved informal communication decreases the need for formal communication as well as documentation, which agrees well with the first value in the agile manifesto (Fowler and Highsmith 2001). However, they also found some problems with these practices, one which adds to Whitworth and Biddle’s (2007) findings of problems with stress and social exhaustion. Pikkarainen et al. (2008) found that team rooms could create disturbances for developers – especially when meetings where not all team members participated were held in the team room. Turner and Boehm (2003) discuss that the increased focus on informal communication in agile methods makes teams rely heavily on knowledge residing in its members rather than formal documents. While this can increase effectiveness, it also creates risks when the communication does not work well enough or members leave the team, taking important knowledge with them.

McHugh et al. (2012) found that the open and frequent informal communication in agile teams helps team members develop trust in each other. The increased transparency, mostly coming from daily meetings, was found to make team members’ and other stakeholders’ perceptions of results come closer to reality. McHugh et al. also found agile practices to increase accountability and collective responsibility, which also increased trust through better perceptions of what team members know and can do. Increased trust and transparency in agile methods is supported by Laanti et al. (2011), and Levy and Hazzan (2009) who also mentions daily meetings as an important factor. However, Laanti et al. found that while transparency regarding progress and problems increased, teams that were negative toward agile methods in their study at Nokia found that transparency toward the big picture of the project decreased.

Whitworth and Biddle (2007a) describe how agile methods increase motivation and cohesion in the teams by supporting a collective team culture. They attribute the development of a collective team culture to that the team has a clear common objective – to deliver the most possible value for customer in a sprint – which helps the team have good interactions without friction. Whitworth and Biddle also suggest that having sprints increases cohesion and energy in the team as they push together for the sprint goal. Reaching sprint goals is suggested to increase feelings of pride among team members, which encourages them to uphold team standards. Finally, having a collective team culture is suggested to increase feelings of personal security and control.

Conboy, Coyle, and Wang (2011) found evidence in 17 companies that agile methods removed specializations and instead gave the impression that developers needed to be “masters of all trades”. This resulted in several problems, including recruitment, training, and job satisfaction. Conboy et al. suggest finding a balance by having broad knowledge about software development while also specializing in certain areas.

### 3.3 Knowledge Management background

There are several different ways to view what knowledge is and how it can be managed, some of these are described by Alavi and Leidner (2001). However, making any detailed reviews of different perspectives on knowledge is beyond the scope of this thesis and only the perspectives that were seen as fitting best for the study are described. I decided early in my thesis work that I would base myself on the theories by Nonaka (1994) because I found them to be interesting and well defined, and that they fit well with agile methods. For these theories, knowledge is defined as a justified
3.3. Knowledge Management background

belief held by an individual or group. The belief is justified by information, which is seen as a flow of messages.

3.3.1 Knowledge creation with SECI

For Nonaka’s (1994) theories it is important to separate explicit and tacit knowledge, first described by Polanyi (1958). Knowledge is explicit if it can be expressed, for example by writing it down or explaining it in words. Tacit knowledge is based on what you do and is usually tied to a context, it is personal and difficult to put in words. Nonaka means that knowledge creation occurs through a dialogue between tacit and explicit knowledge through Socialization, Externalization, Combination, and Internalization (SECI).

Socialization

Socialization is when individuals interact and gain tacit knowledge by sharing experiences. For example this is evident in sports when a coach shows a technique and the athlete tries to replicate it until they get the right feeling and the coach becomes satisfied. The gained tacit knowledge remains tacit in the athlete as they would not be able to explain in words how they do it, but can only teach someone else the technique by showing them how to do it and sharing experiences.

Externalization

Externalization is when an individual finds a way to express their tacit knowledge in words and thus make it explicit. This could be when a musician has created a melody by feeling and then creates sheet music from it. This makes the knowledge easy to transfer as another musician can simply read the music and replicate it.

Combination

Combination is when two or more individuals combine their explicit knowledge to create new explicit knowledge. This can for example happen in an online discussion forum where different individuals contribute their knowledge and discuss it to create new knowledge. As the new knowledge is created through combination of different sources of explicit knowledge, without requiring individuals to experience and learn, the resulting knowledge is explicit.

Internalization

Internalization is when explicit knowledge is made tacit by an individual. This is similar to the traditional view of “learning”. When you learn something through internalization you personalize the knowledge to make it tacit, so you do not simply learn to recite what someone told you, but you actually understand what they meant.

Nonaka (1994) suggests that knowledge is created through a spiral through these four modes of knowledge creation. For every shift of mode, the knowledge level becomes higher, creating deeper knowledge.

3.3.2 Communities

A common way to manage knowledge is through using different kinds of communities, often referred to as communities of practice. There are several slightly different definitions of what a community of practice is, but for the purpose of this study I will base myself on the definition by Wenger et al. (2002, p.4):
“Communities of practice are groups of people who share a concern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in this area by interacting on an ongoing basis.”

This definition is broad as “interacting” could mean anything from writing in an online forum to meeting face-to-face, to working together. A “concern”, “set of problems”, or “topic”, is also broad and could be virtually anything. The important thing however is that it is a group of people who share an interest and who deepen their knowledge through interactions, not through simply reading or listening. For this definition, it does not matter if the group views themselves as a community or what the intention of their interactions is, they can still be called a community of practice. Wenger et al. (2002) describe three crucial characteristics for a community: a shared commitment to the domain of the community, interactions between its members, and that the members are practitioners in the domain.

Brown and Duguid (1991) also discuss communities of practice and stress that for them to work well in, and for, an organization, the communities need to be freed from too much structure, or canonical ways of working. The communities should instead be autonomous and be allowed to shape themselves after reality. Brown and Duguid mean that the communities then can bridge the gap between what the organization perceives and reality, creating good opportunities for learning and innovation.

Boland and Tenkasi (1995) provide a different view of communities as they focus on the making and taking of perspectives in what they call communities of knowing. Their theory about communities of knowing is different from that of communities of practice in that a community of knowing is built up of people with the same specialization or deep knowledge and puts more focus on the interaction between different communities. Boland and Tenkasi mean that what creates a community of knowing is that it is a group of people who share some knowledge and communicate about this knowledge in their own way, creating their own language and making a perspective. When the perspective has become strong enough and differentiated from the perspectives of others, the community can share their perspective with people outside of the community, allowing them to take the perspective. Boland and Tenkasi describe the process of taking a perspective as difficult and demanding, and one that can often break down. They state that the process involves judgment and bias to get a picture of what others know, often resulting in overestimates of how much of one’s own knowledge is shared by others. However, the process of taking a perspective is valuable, though difficult, as Boland and Tenkasi describe it as central in utilizing and creating knowledge in an organization.

3.3.3 Approaching knowledge differences

When individuals or communities from different practices or with different perspectives meet, there are often barriers to knowledge sharing because of the large variations in knowledge between them. Carlile (2002) discusses different levels of such variations in knowledge and suggests the use of boundary objects as a good way to highlight the variations, or boundaries. Boundary objects were first introduced by Star and Griesemer (1989) who suggested four different types of objects, ranging from repositories with detailed descriptions of matters, to standardized ways to communicate about matters. Within Star and Griesemer’s definition, boundary objects have a broad scope and can take the form of many different physical, virtual, or imaginary objects that in some way aid the interactions between individuals or groups with diverse knowledge. Wenger et al. (2002) suggests boundary objects to be good for creating similar interpretations between different communities of practice, but I consider that they can also be used within communities.
In interactions between testers and developers, regardless of whether they are just one of each or large groups, a working piece of software is an example of a boundary object. Through using names of different elements and artefacts in the software, the developers can explain what new functionality they have added without having to explain details of how they implemented it that might be difficult to understand for the testers. The working piece of software becomes an object that visualizes what knowledge the different groups have in common and where the boundaries of their knowledge are – a boundary object. But boundary objects can also be used within communities, for example in a discussion between community members, one member might have some new knowledge that he or she wants to share with the community. To make the new knowledge easier to access for the others, he or she uses a made-up example to explain the new knowledge using terms that the rest of the community understands, highlighting the new knowledge and how it differs and contributes to the knowledge that the community already had. In creating this example, the community member might not even need to create any physical representation, but can keep it entirely imaginary. It is however still a boundary object, though an imaginary one.

3.4 Designing knowledge practices

There are several ways to deliberately design knowledge practices, even though some practices work better if they exist organically (Wenger et al. 2002). Since the concept of communities of practice is wide, most knowledge practices can be gathered under that name, even though they are different in both shape and ambition. Mestad, Myrdal, Dingsøyr, and Dybå (2007) describe how the use of communities of practice evolved through three phases in an organization, going from seminars with one-way information, to interest groups with planned meetings to discuss a topic, and finally skill circles that are created and discontinued organically with few barriers to their creation. For example, Mestad et al. argue that all three phases are communities of practice, even though each phase became increasingly similar to Wenger et al.’s definition. No one type of knowledge practice is best for all situations, but the practice must be adapted to its environment and purpose.

McDermott (1999) discusses some common issues in organizations using cross-functional teams. He suggests that teams can easily become isolated, which can lead to narrow thinking as team members think more alike and are not exposed to different opinions and people with other experiences. He suggests combining the use of teams with use of communities of practice where employees meet people outside of their own team and discuss experiences and knowledge. In this way, teams become groups with common knowledge about a product, while communities become groups with common knowledge about a domain or specialization. The teams and communities can weave together different sources and applications of knowledge to help the organization access and use all its knowledge more effectively. McDermott calls this a “double-knit” organization.

Wenger et al. (2002) provide seven principles for designing communities of practice:

- **Design for evolution** – The organic feature of the community should be embraced to allow for natural evolution. As the size and environment of the community changes, the functions of the community should be capable of adapting to the changes. Wenger et al. suggest that the design with things like set meeting times, coordinators, and meeting rooms, needs to be adapted to the nature and stage of development of the community, but that the primary role of the design is to catalyse the evolution of the community.
3.4. Designing knowledge practices

- **Open a dialogue between inside and outside** – Only the members of the community know what challenges they face and how the different members can contribute to solving them. However, it is often difficult for members to see the possibilities to develop and handle knowledge in the community. This might require an outside perspective, which can be provided by a person who is not a member of the community, and maybe not even the same organization.

- **Invite different levels of participation** – Not all community members have the same need and interest for the community, and therefore they should not be expected to participate equally. It is good to have some “core” members who participate at most times, but other levels of participation are also good for the community. In the design of the community, Wenger et al. suggest to make it easy to participate in the community without contributing, and that individuals who participate but do not contribute should still feel like members of the community.

- **Develop both public and private community spaces** – Public events, like meetings and online discussions, that are open to all members are necessary for members to feel like a part of the community. However, Wenger et al. state that the public events are often over-emphasized and that more private aspects, like one-on-one discussions between community members, are important for the knowledge sharing in the community. To promote the private aspects, the design should allow for them to happen and community coordinators can try to start them.

- **Focus on value** – What makes community members want to participate in the community is that it brings some kind of value for them, even if that value is brought through that the community brings value to the organization or a team. The value of a community is often not evident, and therefore an important aspect in designing a community is to make sure that the value of the community is shown to its members. This can be done through feedback from members to the community about how the community has affected them.

- **Combine familiarity and excitement** – It is normal for communities to settle into a cycle of meetings and other interactions that does not change much over time and becomes familiar to its members. Familiarity is good for creating a comfort level that invites members to have relaxed discussions and be comfortable with sharing opinions and asking for advice, because members know how the community works and what is expected from members. However, having some activities that break the familiar cycle of the community can help infuse new and interesting ideas into the community to attract new members and keep existing ones engaged. Wenger et al. suggest to value familiarity in communities but also add some excitement, for example by having some irregular or spontaneous activities, like workshops, and participation or presentations by controversial individuals.

- **Create a rhythm for the community** – The events in a community create a rhythm, for example through familiar weekly meetings creating soft beats. There can also be stronger beats from more exciting events, and all beats do not necessarily fit into the rhythm, making it faster or slower. Wenger et al. suggest that beats created by community events ripple out to other activity among community members, like face-to-face or online discussions, where stronger beats create more ripples. Depending on what members expect of the community, different speeds and strengths of the rhythm are desirable. Wenger et al. stress the importance of finding the right rhythm for the community and maintaining it through having combinations of familiar and exciting events.
In this chapter, I present the findings I made during my study to provide grounds for my analysis. The chapter begins with some background information about the case company and its complementary knowledge practices. After that, the perceptions of employees, as well as my own perceptions about the situation regarding agile methods and knowledge management are presented as results from interviews and observations.

4.1 Case company background

MedTech is a pseudonym for the software development department of a medium-sized IT-company providing large software systems for the healthcare industry. They have a wide customer base with hospitals globally using their products in different ways. Due to a high importance of reliability and security of the products at the hospitals, MedTech launches new releases biannually, with strict procedures for testing as well as development to get necessary certifications. The department is distributed between 3 offices in Sweden and 2 in the UK, whereof I visited two of the ones in Sweden. MedTech has several large competitors and aim to compete through superior quality and innovation, putting a lot of responsibility on their employees.

4.1.1 Transition to agile methods

After having used plan-based methods with large tasks for individual employees since the company was founded, MedTech started using agile methods around the year 2010, 6-7 years before the study. The company took help from an experienced consultant to make the transition to agile methods as smooth and successful as possible. The consultant had experience from Spotify who were used as a role model for the methods and practices. MedTech started by having one team convert to agile methods, followed by one more and then the rest. There was some resistance toward the agile methods, mainly since they meant that the developers would move from their private offices to team rooms that they would share with 3-7 colleagues. From the start, all teams used scrum and were told to follow the set-up practices and rules. When teams were
beginning to become more comfortable with the methods, they could adapt them as they felt best. This has led to each team using agile methods in different ways, some more similar to kanban with WIP limits rather than sprints, but most still working with scrum and sprints.

### 4.1.2 Organizational structure

At the time of the study, MedTech had about 120 employees working with development and testing, including product owners but not managers. About 80 of these were situated at the office that the study was mainly done at and about 10 at the other office that I visited. The department is divided in what they call tribes that are responsible for different product categories, each tribe with some management personnel of its own. The name tribe is taken from Spotify, see Kniberg and Ivarsson (2012). The tribes are of different sizes with different amounts of developers and testers, but all developers and testers are working in teams, even when a tribe only has one team. The basic structure is illustrated in figure 4.1. The figure does not show the actual number of teams and tribes at MedTech, and in reality, there are 15 teams with an average of 6 members each, divided in 5 tribes. Tribes also have product owners who work closely with the teams and who have responsibility for prioritizing their backlogs to help their work contribute as much value to customer as possible. For everyday work, teams work autonomously and make their own decisions about their sprints and development work in cooperation with their product owner. It is avoided and uncommon that management steps in to decide what teams should work on or how features should be developed. Teams and their product owners have freedom to decide things like if more time is needed for development of a feature, how things should work and look, and what should be done first, without involving management. The department manager states that they trust teams to do what they think is best, and that they are the most knowledgeable persons to take such decisions.

All teams are cross-functional and include developers as well as testers, with some of the developers having different specializations. Specializations are not stated in the role of the employee, but some developers have unique experience with certain aspects of the development, making them more effective at solving some tasks. Each team also has a scrum master or agile coach, who is responsible for the team’s use of the agile method they have chosen, or any adaptation of methods. Scrum masters meet in biweekly scrum of scrums for the entire department. Management at MedTech generally tries to find the right team for each employee, where they fit in socially and get to work with things they are good at and enjoy. When a team has found the right number of employees who work well together, management avoids moving members in and out of the team. MedTech however also values that employees try different things and there is some movement of employees between teams, from teams to other roles, and of new employees into teams.

MedTech has stated goals to handle knowledge and innovation well, which is reflected in the resources they use for it. There is a seniority program where employees who have and share a lot of knowledge receive some extra compensation and possibility to go to events and take some working time to develop their knowledge. There are also some complementary knowledge practices where a lot of resources are spent through a high number of man-hours. There are three main complementary knowledge practices at MedTech: Guilds, Explore & Share, and RAD Fridays. All these practices are voluntary to take part in and count as paid working hours for all employees. Now follows descriptions of the backgrounds of the practices, and later in the chapter perceptions by employees are described.
4.1.3 Guilds

Guilds were introduced at about the same time as the agile methods by suggestion from the agile consultant, and the concept came from Spotify. Guilds are forums where employees from different teams and tribes who have similar competences or are interested in a certain field meet to discuss interesting topics. Some examples of guilds at MedTech are the agile guild, the programming guild, and the testing guild. The department manager described the guilds as a horizontal function to share knowledge between the vertical and customer-oriented structures in the tribes, meaning that there were similar roles in different teams and tribes who normally would not meet and discuss their expertise, so guilds were supposed to be a place to do so. An example of how guilds can be built up with employees from different teams and tribes is illustrated in figure 4.1. Each guild decides its own structure, but most guilds have meetings that are time-boxed to one hour and hold them biweekly. The requirements from the department management are that each meeting should have an agenda that is communicated to members in advance, each guild should have a stated purpose, and there should be measurable goals (which can be as simple as the number of attendees at meetings). The meetings are held in video conference rooms and participation via Skype is possible. The agenda for meetings differs both between and within guilds, where some meetings are dominated by a presentation and others are purely discussions.

At the time of the study there were 5 guilds that could be considered active, which seemed to be a normal amount. There had been occasions with both slightly more and slightly fewer guilds. The different guilds handled different fields, each with one or two employees who were responsible for planning meetings, often with help from the rest of the guild. One of the active guilds at the time of study had been started recently before the study, showing that new ones are still being started. Some guilds and topics are more popular than others and the attendance at meetings varies from about 4 to 20,
4.1. Case company background

including those participating via skype. When interest for a guild drops, it can skip
having meetings for some time, which was observed in my study.

When guilds were introduced at MedTech, there were some suggested areas for
different guilds, but since then they have become organic, which means that they are
supposed to be created and removed when need comes and goes. The department
manager stresses the importance of keeping guilds organic and driven by interest and
passion from their members. He encourages participation in guilds but stresses that it
is voluntary, and I observed that there are several employees who do not participate
regularly in any guild. He also says that the guideline is that you should not visit
more than one guild regularly, but that it can be allowed to do so in some cases, which
was observed for a few employees. My perception was that the department manager
considered that going to more guilds would make employees lose too much time in
their sprints and most employees only had one guild that was truly relevant for their
work.

The department manager says that he does not expect explicit results from guilds
but that he thinks there are positive effects in the long term. The idea is that discussions
between employees in similar roles will improve how that role works. To make that
possible, the department manager thinks that guilds should have mandate to make
decisions for changes in the department on a reasonable level and that they should also
be able to elevate bigger decisions to be discussed by management. He also stresses
that guilds should not be something that you must go to not to miss something, but
that you should only go if you are interested. He does not see it as a requirement for
employees to visit guilds, but he wants knowledgeable people to share their knowledge
and promotes guilds as a good place to do so. He does not visit guild meetings himself
because he thinks it could affect the dynamic of the meetings negatively since other
participants could see him as being in charge of the meeting which could make it
difficult for others to take initiative. He also asks other managers to not participate
in guild meetings for the same reason. There are no rewards tied directly to guilds,
but sharing knowledge is an important part of employee reviews and can result in
compensation through seniority programs.

4.1.4 Explore & Share

Explore & Share is a more recent practice for knowledge sharing at MedTech, intro-
duced about 2-3 years prior to the study by suggestion from an employee who had had
a similar practice at a previous workplace. This employee is now responsible for coor-
dination of the practice and will be referred to as “the Explore & Share coordinator” or
simply “the coordinator”. Explore & Share is divided into two parts: reading tips and
presentations. The reading tips are e-mails with articles or other texts which can be
interesting for employees. These are sent out biweekly by the coordinator and can be
about anything related to the organization in some way, e.g. new versions of software
or programming languages, uses of different technologies, news about some new kind
of IT security threat, or similar. The e-mails usually contain links to websites where
the text is found, always with a short description of the content. The coordinator usu-
ally asks the guilds if they have any ideas for reading tips, but anyone can come with
suggestions. If there are no suggestions, the coordinator has a list of reading tips to
take from, which has not become empty so far.

The other part of Explore & Share is presentations held by employees to anyone
that is interested, hopefully about a topic that is relevant to at least some other em-
ployees. These are also held biweekly and an invitation is sent by the coordinator
to all employees a week in advance, containing a few sentences explaining what the
presentation will be about and who the presenter is. The presentations are time-boxed
to 30 minutes and are held during the end of the one hour lunch. They are also video
recorded and live participation via Skype is possible, as well as watching the video later. Presentations can be held by anyone about anything, but the coordinator asks guilds if they have any interesting subjects for presentations and anyone who wants to present. There are also some employees who hold presentations quite regularly since they have much relevant knowledge, often because they have a role that handles things that are relevant to many, such as software architects. Presenting at Explore & Share is voluntary, but can be something that an employee and his/her manager agrees upon in employee reviews if the employee holds some knowledge that might be interesting for others to hear a presentation about. Presenters can take up to 3 hours of paid working time to prepare for their presentation.

According to the coordinator, the purpose of Explore & Share is to inspire employees and create interests. The idea is that if an employee is inspired or becomes interested in a topic, they will try it out or learn more about it themselves, which is often relatively simple in software development. The coordinator thinks that Explore & Share contributes to creating and sustaining an image of working with the latest technology among employees, while also spreading some knowledge and ideas. Explore & Share is different from guilds in that the former is focused on one-way information where guilds are more of a dialog, and Explore & Share handles a wider spread of knowledge compared to the more focused guilds.

4.1.5 RAD Friday

The third complementary knowledge practice at MedTech is called RAD Friday and was introduced about 5 years prior to the study with inspiration taken from Google Friday, which is a similar event at Google. RAD Friday occurs every 6 weeks and is one and a half work day (12 work hours) where employees get to choose what they want to do, as long as it can be seen as contributing to their work or to their competence in any way. For example, the time can be used for reading up on and experimenting with an unfamiliar technology, developing an interesting feature for a product, or writing some scripts to make some part of daily work easier. Employees are free to work together or alone however they like.

Sometimes, a theme is decided for a RAD Friday. This could be a technology, e.g. machine learning, which employees then are encouraged to learn more about and try. If there is a theme, some presentations are usually held in advance of the RAD Friday to spread knowledge and inspiration about the theme to employees. It is however not mandatory to do something related to the theme. At the end of each RAD Friday, a presentation is held where anyone can demonstrate what they have done. Usually, several such demonstrations are held, but far from all individuals and groups choose to demonstrate. A makeshift jury selects a winner who is judged to have had the most impressive or innovative results and receives a symbolic prize. It is fairly common that ideas from RAD Fridays are further developed to become part of a product. The department manager states that there have been dozens of ideas from RAD Fridays that has been implemented without having been in backlogs previously, and several more that have been shown to customers for feedback.

The department manager mentions several purposes with having RAD Fridays. He says that it affects the organizational culture and boosts morale since employees are given the trust to do whatever they want during paid working time. It is a chance to get to know and work with colleagues that are normally in another team or tribe, promoting unity in the organization. It promotes creativity and can generate ideas for new features and other innovations that sometimes end up in products. Finally, it is a chance for employees to get a break from their sprints and do something else for a while, which can give new perspectives and insights. Themes can give extra value if there is an area where there is a need for MedTech to explore, but the department
manager stresses that they do not force anyone to follow the theme – the freedom to choose what you want to do is most important. The department manager has the perception that RAD Friday is something that most employees look forward to, but that some get stressed or anxious because they do not know what they should do. He also thinks it has a positive effect on knowledge and creativity as many employees try technologies that are new to them.

4.2 Interview and observation results

In this section, the results from interviews and observations are presented. The interviewees are presented in tables 4.1-4.3 below, with information about if they are “pre-agile”, meaning if they worked at MedTech before agile methods were introduced there, and if they had any previous experience of working with software development before coming to MedTech and, if so, if agile methods were used at that previous workplace. I made this distinction because I found in interviews that there were patterns that separated pre-agile employees from post-agile ones and it might be relevant to know if they had any agile experience before coming to MedTech since that could affect how they thought about the methods. In the following text, when I write something that can be traced back to what one or several interviewees have said, I refer to them using the key in the leftmost column of the tables. In some cases, no one is referred if I found it to be irrelevant, possibly damaging to the interviewee, or too revealing of the identity of the interviewee. Some results come directly from observations, in which cases this is stated, but much of the other results have been affected by the impressions I got from doing my observations, which is sometimes not mentioned.

<table>
<thead>
<tr>
<th>Interviewee</th>
<th>Pre-agile</th>
<th>Previous experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1</td>
<td>Yes</td>
<td>Yes, non-agile</td>
</tr>
<tr>
<td>A.2</td>
<td>Yes</td>
<td>Yes, non-agile</td>
</tr>
<tr>
<td>A.3</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>A.4</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>A.5</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 4.1: Team members of team A, the distributed team.

<table>
<thead>
<tr>
<th>Interviewee</th>
<th>Pre-agile</th>
<th>Previous experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1</td>
<td>Yes</td>
<td>Yes, non-agile</td>
</tr>
<tr>
<td>B.2</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>B.3</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>B.4</td>
<td>No</td>
<td>Yes, agile</td>
</tr>
</tbody>
</table>

Table 4.2: Team members of team B.
4.2. Interview and observation results

Table 4.3: Team members of team C.

<table>
<thead>
<tr>
<th>Interviewee</th>
<th>Pre-agile</th>
<th>Previous experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1</td>
<td>Yes</td>
<td>Yes, non-agile</td>
</tr>
<tr>
<td>C.2</td>
<td>Yes</td>
<td>Yes, non-agile</td>
</tr>
<tr>
<td>C.3</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>C.4</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

4.2.1 Perceptions of methods and practices

In this section, the perceptions and opinions about the way employees at MedTech work in teams, and related practices that are connected to agile methods, are described. The section is divided in different areas that emerged as interesting during the study.

General impressions of the methods

I found that most interviewees were happy with how agile methods affected their work, although there were notable differences between pre-agile and post-agile employees. Even though pre-agile employees were positive toward agile methods, they viewed them far more critically than most post-agile employees did. The pre-agile employees could easily compare their experiences of working with agile methods to how they remembered it was like at MedTech before. This meant that some of them missed certain aspects of how things were before, but also found some things to have improved with agile methods. All pre-agile employees agreed that they preferred the way they worked now over how they worked before MedTech adopted agile methods. All post-agile employees were positive toward the methods, even though two of them expressed concerns about that only the development department, which is what I call MedTech, and not the entire organization are working in an agile way (A.4, C.3). “Product owners and people higher up [in the organization] have limitations that are not very agile, like having fixed scope, fixed time, and fixed resources.” (A.4)

Regarding the work being done in the teams and their efficiency, some pre-agile employees questioned the methods. One pre-agile interviewee mentioned that he sometimes missed the large tasks that were common before agile methods, sometimes stretching several months, which he could take on by himself (A.1). Two other pre-agile interviewees said that they think individuals are less efficient in their software development with agile methods (B.2, C.2). They both mentioned that it is difficult to focus in the team rooms and that there are too many people taking part in decisions.

Team rooms

All interviewees mentioned that having the entire team in the same room played an important part for how they shared knowledge and worked as a team, one interviewee describing it as “[…] fundamental – it would not be the same if we had separate rooms.” (B.1) All interviewees also said that questions in the team room, both directed at the entire team or to a specific person, were common. One interviewee also said that most problems in the team were solved by simply saying in the team room that there was a problem, which then resulted in collaboration to solve the problem (B.3). Even though there were negative aspects, interviewees seemed to find that team rooms had a positive effect. One pre-agile interviewee said: “In general, it is more difficult to get [flow] when working agile with team rooms, but it is probably better for the
4.2. Interview and observation results

working environment because there are people to ask things to and discuss with who are working on the same things.” (C.2)

Another pre-agile interviewee mentioned that he first had a hard time with moving from his private office to a team room because he thought it would be difficult to focus. The same interviewee also said “[…] it is more difficult to get [flow] now than what it was [with private offices]”, which he attributes to more disturbances and noise (B.2). However, he also sees positive sides with the disturbances and mentions that it is good that when some people talk about something in the team room, the others hear and can easily join in, possibly providing or gaining important information (B.2). It varied how easily disturbed interviewees found themselves to be, and observations in the offices in general showed that it was common to use headphones while working. In one case, an employee even had his own room with a door directly into the team room as he had difficulty focusing otherwise (C.1). However, he said that he still is easy to reach for the team as they can just go over to his room. The rest of the team did not have problems with that but said that they would not disturb him if he had his door closed or was set to “busy” on skype (C.3, C.4).

From just being in the office, I found it to be common that members of different teams went to other team rooms to ask, explain, or discuss things. I did not observe such interactions closely as I did not want to intrude, but I got the impression that they were focused on knowledge and experiences of how different implementations and tests could be done. Sometimes a member of one team went to the team room of someone else and sat by their computer for a while, talking while looking at the screen. Other times, one or a few members of one team would go to another team room and have an informal meeting with the entire team. These types of things happened daily in most teams.

The distributed team had their team room separated in two offices, about 2 hours by car away from each other. One office with 2 members and the other with 3. However, they found it to work well and that it was similar to having a single team room thanks to a virtual window with cameras and large TV screens in both offices that were on for most of every day. They also had microphones for the virtual window but those were only turned on when there were team meetings or someone wanted to discuss something that required people from both offices. The team members from the different offices also aimed to meet each other every other week, but the need for that had decreased since they got the virtual window (A.4). They also had a text based chat where they asked questions, often directed at the entire team and answered by whoever had time and knowledge to do so. The chat was also used for small talk and jokes, which was seen as important by all team members as they missed those aspects of working in a team that a co-located team gets at lunch, coffee breaks or casual talk in the team room. It was also common that questions in the chat resulted in skype calls between two or a few team members. Similar chats were used by the other studied teams, but not as actively.

Teamwork

All interviewees felt comfortable with asking for help in the team room and generally knew who in the team would be best to ask or at least felt that they would somehow get help. One interviewee described the atmosphere in the team as “open and supportive” (B.4), which I would say is consistent with what other interviewees said, as well as with what observations showed. All interviewees felt a team spirit in their respective teams, and also collective responsibility for the tasks they were working on. One interviewee said: “We do things that are uncomfortable for ourselves to help the team.” (A.1) Another interviewee said: “Sometimes you are disturbed [by questions], but it’s in the nature of the team and when I get stuck I want to get help quickly. [The work in the
4.2. Interview and observation results

Questions about whether it was a goal that all team members should be knowledgeable about all tasks that the team was working with was something that split interviewees, both between and within teams. Most interviewees felt that everyone in the team should have some basic knowledge about what everyone else in the team was doing and that it was fine to take on tasks to learn, even though it would decrease effectiveness (A.1, A.2, A.4, A.5, B.3, B.4, C.1, C.3). However, only a few thought that anyone in the team should be able to perform any, or almost any, task in the team (A.4, A.5). Others found it to be more important to get tasks done as fast as possible, meaning that the team member that would do the task most effectively should take it (A.1, C.2).

My impression from interviews and observations was that there was some difference between teams regarding how they thought tasks should be divided among team members. I found team A to have the general intention that all team members should be able to do most tasks, although there were some tasks relating to older software that they found it unnecessary for everyone to learn about since that software would not be used much longer. Some minor specializations were visible in team A. In team B, my impression was that it was up to each team member if they wanted to specialize in an area or rather wanted to be generalists, but that all members should have some basic knowledge about everything the team did. Two of the members of team B had broad specializations and avoided some tasks while they favoured other tasks. Team C was working with a quite narrow part of a product and the two pre-agile members, who were developers, had a lot of experience in the area. They had a quite clear division of tasks between those two developers, leaving more general tasks to the post-agile members.

Knowledge sharing in the teams

When asked about the most common way of sharing knowledge in the team, simply talking and asking questions openly or directed at someone in the team room was mentioned by all interviewees. Another answer to how knowledge was shared in the team was through code reviews (A.1, B.1, B.2, B.3, C.4). Interviewees felt that code reviews often were a good way to share knowledge about how to write good code, especially if the code author and reviewer had different amounts of experience (B.2). The reviews were said to give knowledge both to the author of the code and to the reviewer, as the reviewer could read and understand code that someone else had written and the author got advice on how the code could be improved (B.1).

Finally, another way of sharing knowledge in the team that was mentioned by about half of the interviewees was through pair programming (A.1, A.2, A.3, A.4, A.5, B.1, B.3). Pair programming was mentioned positively by all members in team A, but none in team C. In team C, the two post-agile interviewees questioned the efficiency of pair programming and felt uncomfortable with taking time from the more experienced developers (C.3, C.4). However, a post-agile interviewee in team B found it to be an excellent way to learn (B.3). By the proponents, pair programming was said to be an efficient way to gain knowledge, both for the person writing and for the person watching and helping from the side (A.2).

Daily meetings

Daily meetings were appreciated by all interviewees except one, who found that the team members communicated well anyways (C.2). Observations showed that they were kept short, normally between 5 and 10 minutes, and they were found to be a
4.2. Interview and observation results

good way to synch the team. Several interviewees considered it an important practice to truly work as a team and promote team spirit (A.2, A.3, A.5, B.2, B.3, C.3, C.4,). They also mentioned that an important feature of daily meetings was that it gave the team a few minutes where everyone was gathered and saw and listened to each other, which allowed team members to better understand if someone had problems, was tired, or similar. This was mentioned by members of the distributed team to be of even higher importance for them as they usually do not get chances to just talk to each other over coffee or in the team room (A.1, A.2, A.3, A.4, A.5). In team B, observations showed that the product owner was present for almost all daily meetings, while the product owners of the other two teams were there for approximately half of the meetings or less. The product owners were there to keep track of what the teams were doing and to keep them in the right direction. This was appreciated by the teams as they could get quick answers about priorities and such (A.1, A.2, A.3, A.4, B.1).

4.2.2 Perceptions of complementary knowledge practices

In this section, perceptions and opinions about the different complementary knowledge practices at MedTech are described, starting with guilds and going on with Explore & Share, and finally RAD Friday.

Guilds

What interviewees wanted from guilds varied; where some wanted a forum for deciding best practices (B.2), some wanted discussions (A.5, B.3), some wanted mainly presentations (C.1, C.2), and one wanted a combination of these (A.4). These variations were present between guilds as well as within them. The guilds themselves decided how their meetings should work, and observations showed that this usually seemed to be done in a democratic way within the guild by letting guild members vote about things like topics for upcoming meetings and ways that meetings should be held. However, in some guilds the form of meetings varied a lot, possibly due to that responsibility for planning meetings was rotated between guild members.

There was some variation in whether interviewees thought guilds were a good place to make decisions about best practices and similar things. One interviewee found it to be common that guild meetings resulted in decisions about best practices or standards and that one needed to be there to be a part of those decisions (B.2). Others however questioned why guilds would have any power to make decisions that would affect others, as it varied who visited the meetings (C.1, C.2). Some said that guilds were good for discussing alternatives before decisions, but that the actual decision should be made somewhere else (A.4, C.1, C.4). In my observations, I saw no occurrences where formal decisions were made at guilds, but it was common that good ways to do different things were discussed. This sometimes resulted in changes to how these things were done in the teams of those that participated in the meeting.

Perceptions of whether guild meetings mostly deepened knowledge or rather mostly worked to create a breadth of knowledge among members varied. Some interviewees said that no deep knowledge was discussed on meetings, either because they were too short (A.1, B.4), or because they needed to adapt to the members who were least knowledgeable about the topic (A.5). However, another interviewee stated that guild meetings could discuss deep knowledge and that not every meeting needed to be understandable for everyone (C.1). Another interviewee said that some meetings discussed deeper knowledge, but that those meetings were not very well-visited (A.4). Yet another interviewee said that guild meetings mostly broadened his knowledge which was good for increasing his understanding of different areas and that he visited them to feel more needed, as he then could help others (C.3). My impression
was that the depth of knowledge varied depending on the topic. When a meeting discussed a topic that all participants were familiar with, it was natural that it touched deeper knowledge. There were also meetings that discussed topics that none or few of the participants had a lot of knowledge about, resulting in broadening of knowledge.

Almost all interviewees said that they felt comfortable with asking questions at guild meetings if they did not understand something, although most stated that there was rarely a need to ask anything. An exception was an interviewee that was a member of a guild about an area that he was not knowledgeable in but wanted to learn more about, who felt that it would take too much time from the meeting if he asked about everything, and instead found answers by himself later (A.3). Observations showed that there were lots of questions at most guild meetings, even if the meeting was focused on a presentation. Some guild meetings were almost entirely driven by questions. For example, if one member showed a demonstration of how to use some framework, the other members asked questions that decided what the demonstrator would show and explain. However, while active participation by those that went to a meeting was important, I observed several meetings where some participants were silent for the entire meeting. No pressure was put on silent participants to be more active.

One interviewee stated that he often chose to not go to guild meetings mainly because he found them to be ineffective (C.2). He referred to that the meetings often lacked structure and resulted in “a general discussion about complaints about [some tool, framework or similar] and when they left after an hour they had not accomplished anything.” This is something that I recognized in some guild meetings that I observed, while others were completely different. Some meetings could contain a short presentation about a subject for discussion, followed by about 45 minutes of discussions that quickly got stuck on a small part of what the presenter had planned to be discussed. This could for example be because members had conflicting opinions or perceptions and had difficulty understanding each other, or because the presentation was unclear and there were misunderstandings. Other meetings could for example be focused on learning about some tool through having guild members use it live on a large screen, or on discussions about experiences in the teams. These meetings would often have a natural flow that gave a feeling of effective learning. However, there were also meetings that were mostly discussions that still had a good flow, with different members contributing with experiences and asking questions that kept the discussion going.

Most interviewees said that they sometimes shared knowledge and experiences gained in the team at guild meetings, while only a few said that they shared knowledge from the guild with the team. There were two main explanations to why they did not share knowledge with the team. First, they felt that if the team was interested they would have been at the meeting (A.4). Second, the entire team usually went to the same meetings (B.3, C.1). In the latter case, some interviewees said that they sometimes discussed the meeting with team members afterwards (B.3). Observations showed that sharing experiences from the team at guild meetings happened naturally when the meeting was about something that members had experience with from their team. I found guild members to be open about their experiences at guild meetings, even about things that were not working well in their team. I observed few occasions where guild members talked about meetings, though I did not search for such interactions and they could have happened without my knowledge. Guild meetings were in a few cases mentioned briefly at lunch or coffee breaks, but I observed no bigger conversations about them.

During my observations, I only saw one occurrence where a guild did something out of the ordinary as they took a full day of discussions at a location away from the offices. All other activity was focused on guild meetings, often held biweekly. Though different topics attracted different participants, there was normally a core group that came to most meetings and less active members that came to some meetings. It some-
4.2. Interview and observation results

times happened that people from other departments within the organization visited guild meetings.

**Explore & Share**

About half of the interviewees said that they physically go to Explore & Share presentations regularly, i.e. about 25-50% of the occasions (A.5, B.1, B.2, B.4, C.1, C.2, C.3, C.4). A few said that they rarely saw the presentations live but regularly watched the recordings (A.2, A.3, A.4), and one sometimes watched them live via Skype (A.1). Some interviewees said that they did not go to some or all presentations because they wanted to do other things on their lunch (A.3, A.4, B.3), and one said that he sometimes did not go because he did not want to lose working time (C.3). About half of the ones that watched presentations regularly said that they aimed to go whenever the topic sounded relevant to them (A.3, A.4, A.5, B.2, B.4, C.1, C.3), while some seemed to disregard how relevant the topic was for them (A.1, A.2, B.1, C.2, C.4). Most of those who watched presentations said that they at least sometimes found them to be inspiring (A.1, A.2, A.3, A.4, A.5, C.1, C.2, C.4). Regarding the reading tips part of Explore & Share, only 5 of the interviewees mentioned them. Of those, 2 said that they found it to be too time-consuming to read them (A.5, C.3), 1 said that he reads the descriptions in the e-mail, but rarely the article (B.4), and the final 2 said that they regularly read the reading tips (A.1, A.2).

Observations showed that the presentations were quite well-visited, often with about 15-30 listeners. I found the presentations to be well-prepared and kept on a level where I could understand most of what they said even though I do not have experience with working with the products at MedTech. My impression was that the presentations were generally appreciated by those who listened to them and questions and comments during, or at the end of, presentations were common. My method limited how well I could study the use and effects of reading tips, where for example surveys could have been a good addition. My impression was however that the reading tips were not as important for the sharing and creation of knowledge at MetTech as the presentations or the other studied practices.

**RAD Friday**

All interviewees said that they generally enjoyed RAD Fridays, even though a few said that it could sometimes be difficult to come up with something to do which could become stressful (B.2, B.4, C.1). Most interviewees also said that RAD Fridays often were inspiring, mentioning the demonstrations at the end as well as the entire event with the time for development or learning (A.1, A.2, A.4, A.5, B.1, B.3, B.4, C.3). Most interviewees described the purpose of RAD Fridays as mainly focused on getting some time to develop or try out things that they did not have time for otherwise (A.1, A.2, A.3, A.4, A.5, B.1, B.3, B.4, C.1, C.2). The rest found the purpose to rather be focused on promotion of creativity, competence development, and generating new ideas (B.2, C.3, C.4).

Some interviewees mentioned that RAD Friday was a practice where there was no pressure to accomplish anything. They felt that they could try things that they knew would be difficult to succeed with and be innovative, because there were no expectations from anyone (A.5, B.1, C.2, C.3). A few interviewees said that they sometimes worked with others to be able to achieve more during the limited time of the practice, often with people from other teams and tribes (A.3, A.4). Some also said that they felt like RAD Fridays increased their overall feeling of happiness and motivation at work, and not just during the practice (A.1, A.2, A.4, B.1). Some said that they had RAD Fridays in the backs of their heads at many times, thinking about
4.2. Interview and observation results

ideas that they could develop there (A.4, B.2, B.4, C.4). One interviewee said: “When you come up with something [i.e. a problem or an idea] you know there will be an occasion to do something about it.” (B.1)

Since RAD Fridays only happened once every 6 weeks and were mostly focused on doing things like programming and reading by one’s own computer, or with a few colleagues, it was difficult to study the practice. I was only present at MedTech for two RAD Fridays, during which I tried to get a feel of the practice from observing different presentations and talking to employees at lunch and coffee breaks. My impressions were in line with what interviewees said as I found most employees to have a good time experimenting with new technologies or fixing things they had not found time for in sprints. I also saw occurrences where employees seemed a bit stressed about coming up with a good idea about what to do, and some trying to find someone that they could team up with. At the demonstration at the end of the practice, I got the impression that at least some of the employees demonstrating their results were proud to do so, and that many of the listeners were interested and impressed. In general, I found that most employees seemed to participate in the practice and that there was a slight shift in atmosphere as people talked about their ideas with some excitement and curiosity.
In this chapter, I analyse the data from my study in relation to my frame of reference from literature. The structure of the chapter is that I begin by analysing teams, followed by tribes and inter-team interactions. I then go on with sections for the three complementary knowledge practices at MedTech, followed by a section about how these five different forums work together.

5.1 Teams

Teams are the producing units at MedTech and are built up by developers and testers. They are supposed to be cross-functional and contain all the skills that are needed to solve any of the tasks that can come their way. The ambition of each team is to produce as much value to customers as possible in every sprint. How this is done is up to the teams themselves, but there will always be a need to manage knowledge in an effective way. Sometimes, the required knowledge will be present in the team member that is doing a task, and then it is easy to apply the knowledge to complete the task. More often, a task will require knowledge that is spread out among team members, requiring knowledge sharing and teamwork to happen. In some cases, the required knowledge might not even exist within the team, and knowledge will need to be found outside of the team or created. All this knowledge management can be made easier by several different characteristics in teams. Below, I first analyse the environment and characteristics of teams, and then go on to analyse how knowledge is created and shared within this environment.

5.1.1 The team environment and characteristics

For scrum, Schwaber and Beedle (2002, p.37) write: “A Scrum Team self-organizes so that everyone contributes to the outcome. Each team member applies his or her expertise to all of the problems.” At MedTech, I found that some teams and team members supported goals that anyone in the team should be able to do almost any task. Others rather supported specializations within the team, referring to higher efficiency and job satisfaction. According to the above quote, both of these opinions are valid
5.1. Teams

within the scrum method, as long as everyone in the team contributes to the outcome. However, even if team members are specialized in one area, they should use their expertise to contribute to solving all tasks in the team, even if they are not directly related to their area of expertise. Schwaber and Beedle (2002, p.37) suggest that this will “improve code quality and raise productivity.”

This is supported by Conboy et al. (2011) who found problems when team members in their study felt like they had to have detailed knowledge about all areas relevant to the team. They suggested the solution to be a balance between a specialization and broad knowledge, which means that all team members should understand all tasks well enough to be able to contribute with the perspective they have gained through their specialization. This results in teams where team members are different in their specializations, but have some common knowledge regarding the tasks the team is working on. There can of course also be several team members with the same specialization if that is necessary to increase productivity, or robustness as they can cover up for temporary and lasting losses of team members, i.e. sickness, parental leave, etc. Depending on what a team is responsible for, there will be needs for different levels of diversity. For example, a team that is responsible for maintenance and development of a certain part of an existing product, e.g. the servers of a web-based products, will only have members that are specialized within that area. A team that is responsible for developing a new product that includes different parts, like server, user interface, etc., will need more diversity. In both cases, a broad understanding of the tasks of the team among all team members is necessary for them to be able to contribute with their expertise to all tasks. This broad understanding, and to keep a balance with the specializations, is not easy to achieve in practice.

At MedTech, I found that there were differences between the teams but that most of the developers had specializations, while some were striving to be generalists. My impression was however that all team members did their best to contribute to the team’s sprint goal and efficiency, and did so using their skills in the way they found best. This could be attributed to the cohesion and strong focus provided through having a clear goal in the sprints, as suggested by Whitworth and Biddle (2007a). Depending on how a team chose to distribute the knowledge in the team, with clear specializations or more toward that everyone should be able to do anything, different characteristics were created. I got the impression that teams with more of a goal that everyone should at least have some basic knowledge about everything the team did had more effective communication. Teams with clear specializations seemed to have less knowledge in common between members which made it more difficult to understand each other.

To be able to know and understand what other team members were doing and contribute to that, high levels of communication is necessary. Like Levy and Hazan (2009) and Pikkarainen et al. (2008), I found team rooms to increase face-to-face communication in the teams. The close communication also increases transparency in the team, especially the communication that occurs at daily meetings (McHugh et al. 2012; Levy and Hazan 2009; Laanti et al. 2011). I found that the transparency and communication made team members aware of what was happening in the team, which created opportunities to contribute knowledge to what others were doing. For example, I found it common during daily meetings that when one team member spoke about what he or she was doing, others would ask questions and give feedback that could help him or her complete the task faster or with higher quality. While members of teams with clear specializations could not understand everything that other members were doing or discussed, I still found them to be interested and attempted to help each other using the knowledge they had when relevant.

High levels of close communication also had positive effects relating to the shortened communication chains and improved teamwork as suggested by Melnik and Maurer (2004), as well as a reduced need for formal communication (Pikkarainen et
McHugh et al. (2012) suggest that the awareness of the other team members that comes with the communication and transparency can also increase trust. I found that team members had no problems with letting other team members do tasks that would affect their own work, meaning that they trust each other to do what they say they will do. This seemed to be much due to that they knew that they would get daily updates of each team member’s progress. Once team members had worked together for a while, they had seen what the others could do and found it even easier to trust them with important tasks.

In summary, the environment of teams is one of diversity of knowledge and skills. There is close communication and high levels of transparency, and team members trust in each other and work cohesively to reach sprint goals.

### 5.1.2 Sharing and creating knowledge in teams

For everyone in the team to understand what everyone else is doing, a lot of knowledge must be shared almost constantly during the sprints. This includes several areas and types of knowledge, like practical knowledge about certain systems, frameworks, and tools, knowledge about what customers want and need, knowledge about who knows what internally and externally of the team, and knowledge about what is currently happening in the team. When using agile methods, most of this knowledge is kept tacit for most of the time, since there is minimal documentation. However, there is frequent communication, which requires team members to make their knowledge explicit through externalization (Nonaka 1994). Ryan and O’Connor (2013) found in their empirical study that high quality interactions, preferably face-to-face, in the team is crucial for the development of a shared set of tacit knowledge in the team, which they suggest relates to higher effectiveness.

I found at MedTech that close communication in the teams is common, both during practices like daily meetings, sprint planning, and retrospectives, and informally and spontaneously in the team rooms. These frequent and high quality interactions allow for sharing as well as creation of knowledge. This can also be related to theories about communities of practice (Wenger et al. 2002; Brown and Duguid 1991) as well as communities of knowing (Boland and Tenkasi 1995). Comparing to Wenger et al.’s definition of communities of practice, it is obvious that a team shares “a set of problems”, this being their sprint and backlog. It is also clear that team members “interact on an ongoing basis”, since they sit in the same room most of every work day and communicate frequently, with a minimum of once every day during daily meetings. I would also say that they “deepen their knowledge and expertise” through their interactions. Even though this might not be an explicit goal in the teams, the members’ knowledge and expertise is deepened through their interactions since they are directed at solving difficult problems, and to do that each member needs to have deep knowledge and expertise. Over time, deep knowledge and expertise is developed in all team members.

Brown and Duguid (1991) provide insight about the importance of keeping teams autonomous and close to the reality of customers, which is highly relevant to teams. Teams are autonomous in many aspects, and at MedTech the department manager stated that teams have the power to take many decisions about their products on their own. However, having teams or team members meet with customers often is expensive and difficult, considering the geographical spread of customers and that they are busy. Few personal interactions with customers limits the view of reality for teams, and makes them rely on the view provided by internal product owners, who have more contact with customers. As suggested in scrum (Schwaber and Beedle 2002), the product owner is responsible for the backlog and the tasks and priorities within it at
MedTech. If the product owner also needs to be a representative for a wide array of customers and understand their realities, it becomes a difficult role to manage. Since agile teams, including teams at MedTech, are supposed to be cross-functional, they do not naturally relate to communities of knowing where members are supposed to share a specialization (Boland and Tenkasi 1995). However, if the shared specialization in the teams is viewed as the product that the team is working on rather than the individual skills in the team, one might say that teams can be viewed as communities of knowledge. The team members share knowledge about the product the team is responsible for and make a perspective through the language they use when they talk about it. Considering that each team is responsible for different products, or different parts of the same product, it is possible that the teams make different perspectives because of the differences between their products or parts. For example, a team working with a user interface might make a perspective that is closely related to what customers need and know, while a team working with backend might make a perspective that is more related to e.g. performance. Though taking of perspectives between so widely separated teams is difficult, at least some success in doing so can open for new insights and innovations as teams and their members become aware of issues and solutions in areas that they would never have contact with normally.

At MedTech, there were differences between teams regarding how clear specializations team members had, resulting in different amounts of shared knowledge between team members. Teams with more specializations had less shared knowledge between members, which I found resulted in more difficulties in discussing knowledge, and making a perspective. Thus, teams with more of a goal that all members should be able to do any task relates more closely to Boland and Tenkasi’s (1995) theories and are likely to have a stronger perspective. All teams however had plenty of shared knowledge about their products and I found it difficult, or even impossible, to understand what team members were saying when they talked about details of their product at daily meetings, which hints that their language is not that simple. Considering that each team is its own community of knowing and has made their own perspective, all interactions between teams create opportunities for perspective taking. Stronger perspectives make it more difficult, but possibly also more rewarding, for people outside of the team to take the perspective. Some teams at MedTech work with things that relate closely to what other teams do, which makes their perspectives more similar, which should make it easier for them to take each other’s perspectives.

5.2 Tribes and inter-team interactions

Tribes, containing different numbers of teams that work with the same product category, create the inter-team organization at MedTech. Since the teams in each tribe work with the same products or products that are closely related, some interaction and coordination between the teams is necessary. There are however few formal interactions between members of teams in the same tribe, and the tribes rather seem to rely on informal communication and meetings. As suggested by Schwaber and Beedle (2002), scrum of scrums are held where scrum masters from all teams meet and update each other on what their teams are doing. However, since they are held for the entire department and kept rather short, there is not much time to share experiences, and it mostly serves as a status report and some minor coordination. This somewhat increases the transparency between teams, but relies on that the scrum masters communicate the information to their teams and that team members follow up on the information to provide opportunities for knowledge sharing.

If teams are viewed as communities of knowing as described above, interactions between teams are valuable. Since teams are working on different products or tasks,
they will gain different experiences and knowledge. Some of this knowledge can be reusable by other teams as they encounter similar problems, which happens for teams in the same tribe working with the same product as well as teams working with different products that might have similar parts. The informal interactions between members of different teams mentioned in section 4.2.1 can definitely handle this well, but rely on team members to know who knows what in other teams and take their own initiatives to find the relevant knowledge. More formal interactions between teams could increase transparency to make it easier to know who one can ask about what, making the informal interactions more effective.

5.3 Guilds

As described in section 4.1.3, guilds are groups where members meet to discuss topics that are of interest for them, regardless of what tribe or team they come from. The idea behind why guilds are needed and what they should do is similar to what I described in section 3.4 that McDermott (1999) found. There are different guilds that handle different knowledge areas, and all topics for meetings in a guild should be within or related to the guild area. The department manager at MedTech described that there are no expectations of explicit results to be generated from guilds, but rather that they will have implicit results in the long term through enabling discussions between people in similar roles. The topic and layout of guild meetings varies, where some are focused on sharing experiences between different teams, while at some meetings, guild members will discuss things that are new to all members and attempt to understand them. Below, I first analyse how guilds at MedTech have been designed and what characteristics can be seen, followed by analysis of sharing and creation of knowledge in guilds.

5.3.1 The design and characteristics of guilds

How guilds are designed and work at MedTech has several similarities to the design principles for communities of practice suggested by Wenger et al. (2002). The guilds are kept organic, which is evident in that the activity in guilds varies, and new ones are created. Within guilds, the topics and layouts of meetings are generally decided upon by votes or consensus among guild members, which ensures that topics are relevant and interesting to at least some of the members. However, the rhythm of the most active guilds is steady, but not very strong since the activity is concentrated on biweekly meetings where it is rare that something extraordinary or exciting happens. The occurrence where a guild took a full day of discussions at a location away from the offices shows that exciting things can happen in guilds at MedTech, even though it seems to be rare. I got the feeling that guilds were found to be familiar for meeting regulars, going to meetings every other week and discussing topics with other members. However, since it varied who visited the meetings, that variation could make the meetings less similar to each other, possibly making them more exciting. If guild members or management want guilds to help employees get more new ideas and create innovations, there should be more exciting events. But if guild members and management only want a place to discuss topics in a relaxed way, the current focus on familiar meetings is positive.

As recommended by Wenger et al. (2002), different levels of participation are allowed and exist in the guilds. Some members go to most meetings, while more go only when they find the topic to be interesting and I found that guild members did not pressure each other to contribute to the discussion at the meeting, letting some members sit quietly if they wanted to. Since participation in meetings via skype is possible, it is also easy to watch the meeting silently and leave without interrupting or being rude to the other members. Further, as suggested by Wenger et al., there is some dialogue...
between the inside and the outside of the guilds, i.e. guild regulars and others. From what I have seen, the outsiders only consist of other employees at MedTech who do not usually go to the guild in question. Even though it is good to get some input from outsiders in that way, it could be even better with someone from another organization with other experiences of communities. Such visits could possibly give a different perspective of the guilds and their possibilities.

Guilds provided a good public community space through their meetings, but I did not see signs of regular use of private spaces, as suggested important by Wenger et al. (2002). There were a few occasions where guild meetings were mentioned in private spaces, but bigger discussions between guild members were only observed at the meetings. It might be that there was more use of private spaces but that I missed those in my observations because I could not know where they would take place. Use of private spaces was however not mentioned by any of my interviewees, but neither was it something that I had thought about at the time or asked about. Considering only what use of private spaces I had seen, the minimal use of them could be a sign that guild members were not very passionate of, or excited by, guilds, as they would otherwise naturally want to talk about them more. However, I think it is likely that more use of private spaces existed without my knowledge.

5.3.2 Sharing and creating knowledge in guilds

Based on my observations and impressions, I found that guild members trusted and respected each other enough to be able to be open about their experiences in their respective teams and listen to, and learn from, the experiences of others. Without this trust and respect, members might have been reluctant to share because of fear of being criticized or disrespected. I think the design of guilds with having them completely voluntary and allowing for different levels of participation, as suggested by Wenger et al. (2002), play an important role in building this trust and respect. I think that through having all guild members make a choice to go to each meeting, most members will make that choice based on if they want to learn more about the topic. The resulting participants at each meeting have all prioritized the meeting over working on their sprint, and should thereby have a strong interest to learn. Since learning at guild meetings often relies on listening to and discussing knowledge and experiences of others, it lies in everyone’s interest to respect whoever is talking. As it becomes obvious that the ones talking are respected, more participants become comfortable with sharing, even if that might not have been their original intention when deciding to go to the meeting. The result is an atmosphere of learning. I also think that it is important that all participants at meetings feel like equals, which was supported at MedTech by having managers not going to the meetings.

When guilds work well and have a core group of members with the same or closely related expertise, they will start discussing their knowledge in a distinct way and develop a common perspective. They then become a community of knowing, as described by Boland and Tenkasi (1995). However, since the participants at meetings vary, the core group needs to have their discussions at meetings at a level which is understandable for everyone who participates. While the interactions with guild members who do not yet share the perspective of the core group can provide new ideas and insights in the making of the perspective, it can also limit the strength and depth of the perspective. Introducing more private discussions between core members can allow them to go deeper in the making of their perspective. The guild meetings can then become a place where the core group shares their perspective with other members, sharing their deep knowledge. This can allow less active members to gain deep knowledge through taking the perspective of the core group who have thought about and discussed the knowledge thoroughly.
5.4 Explore & Share

Sharing knowledge between guild members with highly different amounts of knowledge about a topic is not easy. It might result in discussions that get stuck, as I described in section 4.2.2, which can be dangerous for guilds as it might create ineffectiveness, or at least perceptions of it. As shown by one interviewee, this can result in reluctance to visit guild meetings. Star and Griesemer (1989) and Carlile (2002) suggest using boundary objects to highlight what differences there are between individuals or groups. At some guild meetings, live demos or well-explained examples were used, which I think are good examples of boundary objects. I found such meetings to generally be more effective than meetings where discussions did not have such a strong connection to any physical, virtual, or imaginary object.

At MedTech, the boundary objects were normally used for one guild meeting only, possibly creating a new boundary object for the next meeting. This is natural since it could vary a lot who visited the meetings, resulting in different knowledge boundaries. To avoid that the creation of boundary objects takes up too much time at meetings, I think they should be kept simple and be prepared beforehand when possible, for example through prepared demos and presentations. I think that more elaborate use of boundary objects at guild meetings could make the meetings more effective, providing better results and more perceived effectiveness for members.

5.4 Explore & Share

As described in section 4.1.4, the ambition of the coordinator of Explore & Share is for the practice to inspire people and create interests. To do this, there are presentations and reading tips, that generally focus on quite basic knowledge and lays grounds for employees to learn more on their own. I found that the presentations are popular and appreciated, but that it seemed to be rather common to not read the reading tips. The given explanations from two interviewees that it was too time-consuming to read the reading tips is understandable, but I found that most reading tips were rather short and would take less than 30 minutes to read, making them faster than presentations. I would rather say that the issue is that it is more difficult to find time for reading than going to a presentation because you do it on your own and can easily postpone the reading to work on your sprint instead, possibly forgetting to read at all. Presentations also have a social aspect where you might want to listen because you know the person who is presenting, which is not the case for reading tips. Further, the disturbances in team rooms mentioned by some employees can make it hard to get sufficient focus to read in an effective and satisfying way. However, in general I found that Explore & Share did what it was meant to do and spread some interest and inspiration, even though probably mainly through the presentations rather than the reading tips.

5.5 RAD Friday

The department manager at MedTech said that there are several purposes with having RAD Fridays, see section 4.1.5. In short, the purposes he saw were as follows.

1. It positively affects culture and morale.
2. Employees get to work with people outside of their team.
3. It promotes creativity.
4. Employees get a break from their sprints which can help provide new perspectives.
His perception was that most employees look forward to RAD Fridays and that it has a positive effect on knowledge and creativity, also resulting in new ideas that are implemented in products, or are shown to customers for feedback.

Though RAD Fridays were somewhat difficult to study using my method, I got some impressions of what employees thought about them and how they affected the department through interviews and observations. I found that RAD Friday was an appreciated practice by most employees, who valued the time to get to do things that would not fit into their sprints, and try out ideas. My impression was that most employees enjoyed getting some time to decide completely on their own what they wanted to do, while some put pressure on themselves to produce good results, sometimes making them stressed or anxious by the practice. Many also spoke to each other about ideas, showing curiosity as well as excitement, which created a slight shift in atmosphere that I think could have a positive effect on creativity. Some also worked with colleagues during the RAD Fridays, both within and between teams and tribes. These results show that the last three purposes mentioned by the department manager are visible in the outcomes of RAD Fridays. The side effect of making some employees stressed and anxious is however not good, but I think that steps can be taken to decrease it. For example, by providing more support with coming up with ideas for those who struggle, or being more clear with the purpose of the practice, to avoid that employees feel any pressure coming from the organization.

In the long run, I think that also the first purpose mentioned by the department manager, about culture and morale, can be fulfilled by RAD Fridays. Employees get the feeling that the organization and management values their ideas and creativity, which makes them more proud and more likely to think about, and express, ideas. This was supported by that some interviewees mentioned having RAD Fridays in the backs of their heads often, thinking about ideas of what to do at the next RAD Friday. Some interviewees even mentioned that they found RAD Fridays to increase their overall work satisfaction. That it is not uncommon that ideas from RAD Fridays later become part of products shows that there are also practical results through innovations.

In summary, I found that all the department manager’s purposes of having RAD Fridays were visible in the outcomes, showing that the practice did what he considered it was supposed to do. My impression of the practice was that it had an overall positive effect, relating to the mentioned purposes. The effects are however abstract and difficult to study, and I cannot make any concrete conclusions about them from my study. Also, I did not find any literature about studies or theories about similar practices, requiring my results to stand on their own.

5.6 How the forums work together

According to my above analysis, both teams and guilds can be called communities of practice, even though the practice in teams is focused on the product that the team is responsible for while the practice in guilds is a specialization. Even though McDermott (1999) does not call cross-functional teams communities of practice, my findings relate closely to what he calls a “double-knit” organization. While teams are focused on customers and products, working to deliver the most possible value to customer by sharing knowledge and using teamwork, guilds work to share knowledge about specializations. This is also what the department manager had intended with guilds, describing them as a horizontal between the vertical tribes and teams as shown in figure 4.1. Further, Explore & Share works to spread some basic knowledge and inspiration, and RAD Fridays is a practice where employees get to use their knowledge freely, promoting a culture of creativity and innovation.
Looking at Nonaka’s (1994) SECI model, teams and guilds focus on different modes of knowledge creation. In guilds, it is natural that externalization and combination are the central modes since they are about voicing and discussing knowledge between different individuals in the community. Guild members need to consider their tacit knowledge and put it in words that the others can understand, allowing them to contribute with their own externalized tacit knowledge through combination. Since all guilds have a specified area and most members work with that area or are at least interested in it, most guild members have a lot of knowledge in common. Through discussions in the guild, knowledge is shared between members, which gives them even more knowledge in common and allows them to discuss their knowledge area in a more developed and effective way, building a strong perspective, as suggested by Boland and Tenkasi (1995).

In teams, the familiarity between team members, shown in the close communication and high levels of trust, create opportunities for socialization (Nonaka 1994). As team members know and care about each other, as well as about the products they are responsible for, they can share tacit knowledge without converting it to explicit knowledge by watching each other work. Since the knowledge that needs to be shared in teams is mostly related to developing software in the best way, a lot of the knowledge is tacit and gained through experience. Team members watch the work done and results created by other members, be it through pair programming, code reviews, or just being in the same room and working closely to each other. Through doing this, they will consciously or unconsciously share tacit knowledge through shared experiences. Teams are also a place to put explicit knowledge, gained from for example guilds, discussions in or between teams, or internet searches, to use. This allows for internalization as employees understand the knowledge through practise, making it tacit.

Together, guilds and teams create good opportunities for all four modes for knowledge creation described by Nonaka (1994). Through a spiral between the modes, Nonaka means that higher levels of knowledge are reached. I find that to be true at MedTech as employees gain experiences in their teams, gathering explicit knowledge from different sources and putting it to practice, creating tacit knowledge that they can share with their team members. When team members visit guild meetings together with members from other teams, they can share their experiences and gain more knowledge from their own experiences as well as the experiences and reflections of others. When they then go back to their own teams, the employees have gained new explicit knowledge that is based upon their previous tacit knowledge and can be internalized to a higher level of tacit knowledge.

This means that cross-functional teams together with one complementary knowledge practice that allows for the creation of communities of practice and knowing focused on specializations can create opportunities to use all modes suggested by Nonaka (1994) to be required for successful knowledge creation. What is however also necessary is that employees want to create and share knowledge, and that they are motivated to increase the level of knowledge residing in the organization. This can be, at least partly, supported by agile methods and their focus on close collaboration and collective ownership, possibly building cohesion and motivation as suggested by Whitworth and Biddle (2007a). Though the motivation they discuss is for doing a good job rather than directly for creating and sharing knowledge, the result can be motivation for the latter. My perception from observations and interviews was that employees at MedTech were motivated to effectively deliver value to customers, and had an understanding that to do that, they often needed to share knowledge. This was for example shown in that all interviewees said that discussions in the team rooms were common, and even though some found them to be distracting sometimes, they were appreciated because they found them to be valuable for the team. All intervie-
wees also felt a team spirit, which I think can relate to motivation for doing a good job as a team, and helping each other.

Though agile methods have a positive effect on motivation for knowledge sharing, it is mostly limited to within the teams. There are few agile practices that handle interactions between teams, which was also evident at MedTech, as described in section 5.2. What MedTech did to handle this was to introduce three different organization-wide complementary knowledge practices. Guilds, as described above, provide a good place for creation and sharing of knowledge. Guilds do not create a lot of motivation for that however, but rather rely on existing motivation among members, possibly coming from a strong interest in the subject of the guild or a need to learn to become more effective in their work. The other two practices – Explore & Share and RAD Friday – are more directed at creating motivation and giving employees a sense that knowledge is valued in the organization.

While an obvious function with Explore & Share presentations is to spread knowledge from one employee to a large group of listeners, the purpose stated by the coordinator to inspire and create interests might be more important. Interests and inspiration create motivation for gaining new knowledge, that could for example take place in guilds. In the same way, an obvious function with RAD Fridays is to get new ideas for products, but as the department states there are several other purposes, like affecting organizational culture and boosting morale.

From what I have found, cross-functional teams together with communities of practice and knowing provide sufficient opportunities for creation and sharing of knowledge. Therefore, it is not crucial that other complementary knowledge practices, like Explore & Share and RAD Friday, create such opportunities. Instead, they should be focused on creating motivation and making employees feel that the organization values knowledge and creativity.
6 Conclusion

I found agile methods, and in particular agile teams, to provide an environment with high levels of transparency and trust, and frequent face-to-face interactions. This environment facilitates creation and sharing of knowledge by having team members share experiences and discuss problems and solutions relating to the products they are working with. Through these shared experiences and close collaboration, knowledge is created through what Nonaka (1994) calls socialization. As teams are also the place where individuals get to put their knowledge to practice by developing and testing software, they also allow for creation of knowledge through what Nonaka calls internalization. However, since teams are cross-functional and gather employees with deep knowledge about specific products rather than about specializations, employees with similar specializations might be spread between teams. To provide opportunities for employees to discuss their deep knowledge about specializations with other employees with similar knowledge, complementary knowledge practices are useful.

To connect employees with similar interests and knowledge, MedTech uses guilds, which are communities where employees meet to discuss certain knowledge areas. I found these to be closely related to theories about communities of practice (Wenger et al. 2002; Brown and Duguid 1991) and knowing (Boland and Tenkasi 1995). If teams work vertically with knowledge, gathering employees with deep knowledge about specific products and enabling sharing of knowledge between them, guilds work horizontally, gathering people with similar specializations and expertise. This way, teams and guilds weave knowledge from different teams and specializations together and create what McDermott (1999) calls a “double-knit” organization. In guilds, knowledge is mainly created through having guild members talk about their knowledge and experiences, requiring what Nonaka (1994) calls externalization, and combining knowledge and experiences from different members in what Nonaka calls combination. Together, teams and guilds provide good support for all four modes of knowledge creation described by Nonaka to be necessary to create higher levels of knowledge.

What might however be lacking in teams and guilds is motivation to share and create knowledge. As suggested by Whitworth and Biddle (2007a), the focus on close collaboration and collective ownership in teams can build motivation for doing a good job, which is supported by feelings of team spirit at MedTech. In some cases, this might also lead to motivation to share and create knowledge because it is necessary
for doing a good job. Guilds on the other hand rather rely on pre-existing motivation among members to go to guild meetings and contribute to make them interesting. For this, the team as a source of motivation for creating and sharing knowledge might not be enough. At MedTech, I found that this could be supported through having complementary knowledge practices directed at spreading inspiration and interest to learn, as well as giving employees opportunities to use their knowledge in creative ways. Explore & Share is an example of a practice where employees can spread inspiration and interests through presentations as well as reading tips. RAD Friday is an example of how employees can be given opportunity to do what they want to use or develop their knowledge in a creative way.

6.1 Limitations

As in any case study, generalizability is an issue since my findings can only be properly applied at MedTech. As described in section 2.5.1, some generalizations can however be made if you consider differences in context. Applying an interpretive approach further brings with it some possible bias, as all findings are based on the researcher’s conceptions. Since data was mainly gathered through qualitative sources (i.e. interviews and observations) for this study, there can be considerable bias from subjects. The choice of literature is also biased as I could not weigh all relevant literature equally but had to choose which pieces of literature I would read based on what I found to be most interesting and relevant. The study lacks sufficient literature about effects of knowledge practices focusing on inspiring presentations and creativity, considering that the topic emerged as central in my conclusions. Because of this, I could not take those leads further. Adding literature about organizational culture could have allowed me to take my analysis of how knowledge practices can make employees feel that the organization values knowledge and learning to the next level.

It is possible that the results are affected by the fact that MedTech is a medium-sized company that works with long, biannual, releases for the medical industry which requires high levels of quality. Though I have not investigated any other organizations, I imagine that for example a smaller organization with products with shorter releases and closer collaboration with customers could have a different dynamic in their teams. Although most of my results are on a more general level, such details could affect the atmosphere of teams and the entire organization, for example possibly through affecting feelings of collective ownership and team spirit. This should be taken into consideration when generalizing my results to other organizations. Further, interviews were only carried out with a few employees at MedTech and during a limited time, which limited the possibility of triangulation of results. Since teams at MedTech use the methods scrum and kanban, the results are somewhat biased by the effects of those specific agile methods. For example, the popular method eXtreme Programming has more focus on pair programming, which can affect knowledge sharing in teams. However, the concept with cross-functional teams is similar for most agile methods.

Although one of the studied teams was distributed and I studied some effects of the distribution, I could not draw conclusions about those effects since I did not have time to read sufficient literature about distributed teams. My findings however point toward that the distributed team at MedTech worked well, which interviewees mainly attributed to their use of a virtual window and a text-based chat including personal and humorous conversations, as well as that the team members met each other physically at regular intervals. In general, I found few differences between distributed and co-located teams. I did however observe difficulties in having group discussions, such as in guild meetings, with most members in the same room while a few participate through video-chat.
Similarly, I did not study innovation properly even though it was considered in the research. However, innovation is an aspect of knowledge management that is important to consider for most organizations, including MedTech. Innovation at MedTech was mainly supported through RAD Fridays, as described in chapter 5. My findings however point toward that interactions between teams, considering them as communities of knowing, can also be an important source of innovation in organizations using agile methods in several teams.
Discussion

This research contributes to literature by providing a case study that shows how agile methods and cross-functional teams can be complemented by other knowledge practices. I have shown that use of complementary knowledge practices can improve sharing and creation of knowledge between teams, which is not properly supported by agile methods on their own. This is important for literature as well as for practice, and my results can be used both as an example to build more theory from, and for organizations to get advice on how agile teams can be complemented by other knowledge practices. For practical implementation, contextual differences of course need to be considered, and more research with other cases and methods would be valuable to validate and challenge my results and show other perspectives. I think that proper implementation of agile methods and complementary knowledge practices is crucial for organizations to support an environment that allows for employees to share and create knowledge effectively, which can lead to higher effectiveness and more innovations.

There are several areas within this field that needs more research. Future research should continue my investigations of how cross-functional teams can be complemented by other knowledge practices, for example by providing more cases or by doing more quantitative studies. One interesting topic that should be studied more is what types of knowledge is shared in what forums. This could help in making sure that no type of knowledge that is important for organizations is left out in their knowledge management environment. Future research should also go deeper into how knowledge is shared and created within teams, investigating strengths and weaknesses with agile methods and cross-functional teams. This could help tweak how agile methods are used to improve the knowledge environment, and highlight what needs to be achieved by complementary knowledge practices. Finally, future research should investigate how different types of complementary knowledge practices affect an organization, mapping types and effects more clearly than what has been done yet. If generally applicable maps and models for types of knowledge practices can be created, it could make it much easier to apply them in practice, as well as to know what needs to be studied further.


Kniberg, Henrik and Anders Ivarsson (2012). “Scaling Agile @ Spotify”.

Kniberg, Henrik and Mattias Skarin (2010). Kanban and Scrum-making the most of both. Lulu.


Ryan, Sharon and Rory V. O’Connor (2013). “Acquiring and Sharing tacit knowledge in software development teams: An empirical study”. In: Information and Software Technology 55.9, pp. 1614–1624.


