Tomáš Varčok

DISCOURAGING FACTORS IN KNOWLEDGE SHARING IN SOFTWARE DEVELOPMENT TEAMS

UNIVERSITY OF JYVÄSKYLÄ
FACULTY OF INFORMATION TECHNOLOGY
2020

ABSTRACT

Varčok, Tomáš
Discouraging factors in knowledge sharing in software development teams
Jyväskylä: University of Jyväskylä, 2020, 84 p.
Information Systems, Master's Thesis
Supervisors: Seppänen, Ville; Kovalainen, Mikko

Knowledge sharing has a crucial role in intellectual work such as software development. Intellectual capital in form of specialized knowledge, skills, and expertise belong to the most valuable resources that software development organizations possess. The ability to manage and apply knowledge correctly can result in an important competitive advantage. There appears to be a wide agreement on the high importance of managing and sharing knowledge within teams among practitioners. Interestingly, this importance does not seem to be reflected in practicing knowledge sharing in real environments, which often leads to major problems such as lower work efficiency, weaker cooperation, failing to utilize the full potential of available knowledge, and losing valuable knowledge exclusively owned by leaving individuals.

This thesis argues that it is important to identify the reasons behind this gap. The aim is to identify the discouraging factors – the issues that obstruct knowledge sharing and lower its quantity and quality. Additionally, the thesis seeks to identify how the influence of these discouraging factors could be reduced or eliminated.

Due to the explorative nature of the study, the selected research approach is an interpretive case study. The qualitative data were collected during semi-structured interviews with ten participants holding different professional roles in three different teams within one Nordic software development organization.

The premise of the study was confirmed as the data show that there truly is a gap between how important practitioners see knowledge sharing, and how much it is reflected in their regular work. The reasons for this gap were identified in form of multiple discouraging factors, the most noticeable ones being the Perceived difficulty, Lack of attention, Insufficient or incorrect encouragement, and Missing systematic approach. Practitioners can benefit from this study by understanding what is hindering knowledge sharing in their teams. The thesis offers several suggestions on how practitioners can overcome these obstacles. The main recommendations are to pay more attention to knowledge sharing, lower its perceived difficulty, and integrate it into the existing processes and practices in a concrete executable form.

Keywords: knowledge sharing, software development, discouraging factors

FIGURES

TABLES

TABLE OF CONTENTS

1 INTRODUCTION

I think there is no one who would say that knowledge sharing is not important. Everyone agrees with that but then nobody is doing it.
(a participant of the study)

Already a long time ago, organizations were failing to deliver IT projects on-time, on-budget, and according to specification (The Standish Group, 1995). Despite wide research efforts and various software development methodologies focusing on these issues, there are still many IT projects that face challenges or even fail (The Standish Group, 2015). This suggests that there is still a need for improvements, and research efforts addressing these problems should not fade out.

Knowledge has a crucial role in software development because it is a knowledge-intensive intellectual activity, and knowledge affects the IT project's success and the team's performance (Ryan & O'connor, 2009, 2013). Expertise, specialized skills, and knowledge are the most important and most valuable resources that software development organizations have (Faraj & Sproull, 2000; Rus and Lindvall, 2002). To leverage the true potential of possessed knowledge, it is necessary to not only collect it but also coordinate and apply it (Alavi & Tiwana, 2002; Faraj & Sproull, 2000; Rus & Lindvall, 2002).

The idea for this research originated primarily from practical observations and signals from practitioners of different nationalities and IT positions (developer, manager, …) from multiple organizations. Most of these practitioners agreed on one interesting phenomenon – despite the wide and clear agreement that knowledge sharing is very important in software development teams and organizations, this perceived importance is not reflected in practice. Various explanations were offered in form of guesses, but it was noticeable that more sophisticated inquiry into the topic would be beneficial.

The focus of this thesis is placed on knowledge sharing within a single software development team, not between multiple teams in an organization. Within one team, the knowledge is more relevant to all its members and it is directly applicable in such scope (Aurum et al., 2008). Transferring knowledge from one team/project to another is a very different scenario (Szulanski, 1996). Despite the

concepts of knowledge sharing and knowledge management being interconnected, the thesis primarily focuses on knowledge sharing as a broader concept. The interest is largely on an individual´s point of view, motivation, feelings, concerns, and challenges. The type of knowledge that this thesis is interested in is tacit knowledge, skills, and expertise. It seems that sharing of status and project knowledge has much more space and methods in use, such as status meetings, daily scrum meetings, kick-off and retrospective sessions, etc.

The prior research has primarily investigated best practices of knowledge sharing and knowledge management and documented existing practices in organizations. However, it seems that little attention has been paid to understanding why existing good practices are not used by practitioners. What challenges and obstacles do they face? This thesis argues that to improve the current situation, it is not enough to only specify the correct approaches, but it is important to identify, describe, and understand the discouraging factors that keep individuals and teams from effectively managing and sharing their knowledge. It seems that software development companies and their employees recognize the importance and benefits of knowledge sharing and knowledge management, but their efforts in these areas are often inconsistent, ad-hoc, and very diverse in terms of quality (Aurum et al., 2008; Dingsøyr et al., 2009; Prikladnicki et al., 2003). Therefore, this thesis suggests that practitioners could benefit from identifying and understanding the reasons that lead to such a situation. Additionally, that kind of results is believed to be valuable for other researchers as well, because those could allow more precise targeting of future research efforts that would aim at improving knowledge sharing in the same context.

The thesis attempts to address the discovered gap by gaining insights into the topic. The aim is to discover and understand what are the factors that discourage or prevent the adoption of knowledge sharing practices in the context of software development teams. Furthermore, it attempts to provide a set of recommendations on how the effects of discouraging factors could be lowered or eliminated. The selected methodology is the interpretive case study and primary data are collected by conducting semi-structured interviews with participants from three different teams within one software development organization.

The research questions the thesis seeks to answer are:

1. What are the challenges and practices of knowledge sharing and knowledge management in software development teams?
2. What discourages or prevents the adoption of knowledge sharing and knowledge management practices in software development teams?
3. How could the quality and quantity of knowledge sharing be improved?

The role of the first research question is primarily to support and guide the empirical research. The second question seeks to understand why the declared importance of knowledge sharing is not reflected in practice. After the possible obstacles are identified, the third question aims at collecting suggestions on how the current situation could be improved.

The thesis is structured as follows: chapter 2 familiarizes the reader with basic concepts of knowledge, knowledge sharing in software development teams, distributed software development, and known challenges and practices in the area. Next, chapter 3 presents the research goals, research questions, selected methodology, case description, and how empirical data were collected and analyzed. Chapter 4 introduces the collected empirical data and those are further discussed in chapter 5. Lastly, chapter 6 concludes the study, outlines the limitations, and suggests possible topics for future research.

2 THEORETICAL BACKGROUND

This chapter presents concepts that are relevant to this research and specifies the targeted context. At first, knowledge, its types, and collective knowledge are discussed, followed by introducing ways of coordinating knowledge within a team. Later, the focus is brought to distributed software development and how it affects team knowledge. At the end of this chapter, the attention moves to known challenges and existing practices in the areas of knowledge sharing and knowledge management in the software development field.

Relevant literature was searched within databases of Google Scholar and the AIS eLibrary of the Association for Information Systems. The primary keyword *knowledge* was combined with one or more keywords such as *sharing, management, coordination, integration, software development, challenges, practices, motivation, teams,* etc. After identifying relevant articles, their promising references were also followed and analyzed. The quality of each source was carefully evaluated using available metrics.

2.1 Knowledge

Cambridge dictionary defines knowledge in business English as: *"skill in, understanding of, or information about something, which a person gets by experience or study"* (Dictionary.cambridge.org, 2019). Knowledge is the central concept of this thesis. At first, this chapter introduces different types of knowledge. Then the focus moves through memory types onto the level of collective knowledge as a property of a group.

2.1.1 **Types of knowledge**

This section provides a basic overview of knowledge types that are mentioned later in this text. The thesis works with two types of knowledge – tacit and explicit – which are both important concepts in knowledge sharing and knowledge management in software development teams.

The concept of tacit knowledge was introduced by Polanyi (1966) as the knowledge that cannot be articulated. It is based on the assumption that we can know more than we can tell. Even if we can describe something, some part stays unspoken. An example of tacit knowledge can be driving a car – it cannot be just told; it is learned by experience. (Polanyi, 1966) Explicit or codified knowledge is the knowledge that can be transmitted in formal language and it can be more general, standing further away from a specific context (Nonaka & Takeuchi, 1995).

Based on Sternberg et al. (2000), who extensively focused on properties of tacit knowledge, it is typically acquired through personal experience with little environment support and it is about knowing how, rather than knowing what. It

is closely related to action; therefore, tacit knowledge is practically useful. We can find tacit knowledge behind the common term *"learning by doing"* when people learn by performing normal activities, while they might not be consciously aware of what they are learning. Tacit knowledge is an aspect of practical intelligence, which focuses on one's ability to learn from experience and apply the acquired knowledge in practice. Tacit knowledge acquired by experience can be a source of competitive advantage because it might often be rare. Sternberg et al. (2000) have argued that the possession of tacit knowledge has a positive effect on success in practical matters. (Sternberg et al., 2000)

Based on long-term research efforts, Sternberg et al. (2000) conclude that tacit knowledge is different from job knowledge and general intelligence. Tacit knowledge and job knowledge are overlapping concepts, but not synonyms. Some tacit knowledge can be unrelated to work, and job knowledge can be both tacit and explicit. General intelligence is measured by the ability to solve academic or abstract problems, which are different from practical real-world tasks that the tacit knowledge focuses on. (Sternberg et al., 2000)

There seem to be different opinions on whether tacit knowledge can be articulated. Some researchers (Busch, et al., 2003; Nonaka & Takeuchi, 1995; Ryan & O'connor, 2009, 2013; Sternberg et al., 2000) have believed that some part of tacit knowledge can be articulated. Others have held the original definition by Polanyi (1966) that tacit knowledge cannot be articulated. Instead, they have recognized the existence of a middle ground between tacit and explicit knowledge, called *implicit knowledge*, which can be articulated (Ryan & O'connor, 2009, 2013). This thesis adopts the view that some part of tacit knowledge can be articulated and talking about sharing tacit knowledge refers to this articulable part.

In his *Dynamic theory of organizational knowledge creation*, Nonaka (1994) proposed that: *"organizational knowledge is created through a continuous dialogue between tacit and explicit knowledge"* (page 1). This dialogue or interaction is also called "knowledge conversion" and consists of four modes: socialization, externalization, combination, and internalization (Nonaka, 1994; Nonaka & Takeuchi, 1995) as displayed in Figure 1. Nonaka and Takeuchi (1995) hence suggested that it is possible to transform tacit knowledge into explicit (externalization) and explicit to tacit (internalization).

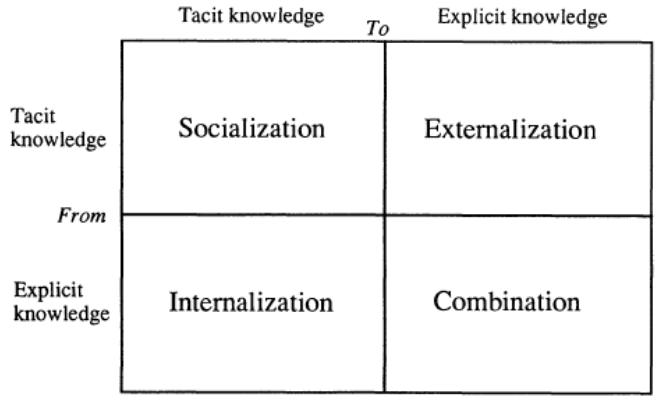

Figure 1: Modes of the Knowledge Creation (Nonaka, 1994)

The term *expertise* can be often heard in the context of knowledge workers, workers whose main capital is knowledge and who know how to use knowledge effectively (Drucker, 1993; Nonaka & Takeuchi, 1995), such as software developers. Faraj and Sproul (2000) described expertise as specialized skills and knowledge. One conception of expertise is that it is not only about possessing knowledge but also about having the ability to apply such knowledge (Sternberg et al., 2000). Alavi and Tiwana (2002) supported the importance of tacit knowledge by saying that the most valuable knowledge has the form of know-how and expertise, and these are mainly tacit or unspoken.

After introducing knowledge in this part, upcoming sections focus on where and how knowledge is stored and what implications it has.

2.1.2 **Collective knowledge**

There are different types of memory for storing knowledge. The ones that are relevant to this research are individual, external, and transactive. These are further explained to the reader in the following paragraphs that are based on Wegner (1987) and Wegner et al. (1985).

The individual memory is the internal human memory. External memory storage can be represented for example by books, notes, online knowledge repository systems, or calendar. Retrieving information from external memory usually requires physically locating the storage and using the appropriate reading method, which can make the retrieval operation slightly slower. Transactive memory system (TMS) is a term established by Wegner (1987) as an arrangement where members of a group cooperate on storing and retrieving important knowledge from various domains. The TMS consists of individual memory systems of the group's members and the communication processes that allow knowledge retrieval and sharing within the group. If an individual does not possess the information, he/she can retrieve it by knowing whom to ask. This memory type is the most complex out of the presented ones as it involves social interaction in the information retrieval stage. (Wegner, 1987). The concept of TMS was identified as relevant for this research because it seems to correspond to the contemporary state of storing and managing knowledge in software development teams. (Wegner, 1987)

The transactive memory systems can vary in a degree of overlap of storing the same information in multiple individual memory systems. When each member of the group needs to be capable of performing the same activities independently, the same information is held by multiple individuals and it is referred to as the *integrated transactive memory*. The opposite case is *differentiated transactive memory*, which is used when there is a higher emphasis on a specialization of members because different memory items are stored in different individual memories. This offers efficient use of memory capacity (information is not duplicated); however, the retrieval of information can be slower due to a need to locate the storage location and conduct the communication process. (Wegner, 1987; Wegner et al., 1985)

Transactive memory is beneficial for individuals because they expand their expertise and they also gain access to someone else's expert knowledge. Smoothly functioning transactive memory can increase the effectiveness of the group in achieving its goals. However, transactive memory increases the complexity of memory and information storage. When responsibilities are not clear, important information might be lost. (Wegner, 1987)

In the knowledge-intensive fields with a lot of diverse knowledge, such as IT, domain experts often emerge within teams. One individual gets the responsibility of concentrating all the information related to his/her domain of expertise in the individual memory system (can utilize both individual and external memory). Recognizing the domain expert and knowing that he/she is aware of that responsibility can be problematic in the early stages of a group's existence, but after longer cooperation, discussions, and sharing past experiences between individual members, it becomes smoother. (Wegner, 1987)

With established transactive memory, the emphasis can move from the tacit knowledge possessed by individual team members to the summary of knowledge that is available in the team overall and what effects does it have on the team's performance. Ryan and O'connor (2009) defined the concept of *team tacit knowledge* as: *"The aggregation of articulable tacit, individual, goal-driven expert knowledge to the team-level where different members of the team possess different aspects of tacit knowledge."* (page 2).

Team tacit knowledge (TTK) is an important factor predicting the effectiveness of the software development team, but not its efficiency (Ryan & O'connor, 2009, 2013). Effectiveness is about meeting project goals and quality, but it is not concerned with speed and budget. TTK predicts the effectiveness and is therefore important for the performance of software development teams (Ryan & O'connor, 2009). One difference that can be identified between high-performing and low-performing teams in terms of effectiveness is that members of high-performing teams have developed a (better) TMS and wider TTK and they are able to share the tacit knowledge and apply it to solve complex tasks (Ryan & O'Connor, 2013).

This chapter briefly introduced important concepts like knowledge types, different kinds of memory, and collective knowledge. Especially the concepts of tacit and explicit knowledge, transactive memory, and team tacit knowledge are very important as this thesis focuses on the aggregation of all tacit knowledge available within the team and how it is distributed and shared among individual team members.

2.2 Coordinating knowledge

The mere existence of knowledge in an environment might not bring major benefits if the knowledge is not coordinated to be shared and applied appropriately. This chapter discusses the current understanding of effective coordination of knowledge in software development teams; however, only on a more general

level, because concrete practices are discussed in a later chapter. The motivation of individuals to share their knowledge is also examined here.

2.2.1 **Importance and need for coordination**

Tacit knowledge is one of the most valuable resources that software development organizations have. It originates from experience and is very practical, closely related to action (Sternberg et al., 2000). Tacit knowledge is valuable and can have a positive effect on success (Ryan & O'connor, 2009; Sternberg et al., 2000). Similarly, Rus and Lindvall (2002) claimed that intellectual capital is the main asset of software development organizations.

Organizations are trying to get professionals possessing expertise to their teams; however, to ensure the high quality of work, the mere presence of expertise in the team may not be enough. To leverage its potential, it is necessary that team members can also coordinate their knowledge. (Faraj & Sproull, 2000) They learn with every project and every task but if this created knowledge stays with them, the organization misses a possibility to benefit much more from that learning (Rus & Lindvall, 2002). As members of software development teams are knowledge workers working with intangible products and processes, their expertise requires coordination (Ryan & O'connor, 2009), which then increases the team's performance (Faraj & Sproull, 2000). As Faraj and Sproull (2000) pointed out, to be effective in accomplishing complex intellectual tasks, the team needs to realize where the expertise is located (knowing skills, knowledge, and experiences of each other) and where it is needed. Knowledge management efforts should also focus on encouraging knowledge application instead of just piling or gathering content (Alavi & Tiwana, 2002). Alavi and Tiwana (2002) stated: *"In the long run, organizations cannot be differentiated by how much they know but by how well they use what they know."* (page 8).

Additionally, sharing and coordinating knowledge between colleagues is important because knowledge from them (from inside the team) has a higher value than knowledge acquired from elsewhere. Aurum et al. (2008) found that other team members were considered the most valuable knowledge source because knowledge from colleagues is usually well applicable to the project's environment and it is relatively easily obtainable.

2.2.2 **Knowledge sharing**

As suggested earlier, it is challenging to transfer tacit knowledge without significant information loss. Ryan and O'connor (2009) concluded that: *"Tacit knowledge is acquired and shared directly, through good quality social interactions."* (page 10). They argued that acquisition and sharing of tacit knowledge requires a transactive memory system (TMS), in the role of team's collective mind, and quality social interactions. The more developed TMS and/or higher quality of social interactions, the higher level of team tacit knowledge. The quality of social

interactions was found to have a stronger effect in that relation. (Ryan & O'Connor, 2013)

Managers should support the development of relationships and social interactions in their teams because they are important for locating and sharing knowledge (Bock et al., 2005). Unless an explicit person-skill index exists in the team or organization, personal networks can be the key for locating knowledge possessed by other individuals (Aurum et al., 2008). It is therefore beneficial if a worker's personal network is wide. After the knowledge source is located, social interactions are a way to share tacit knowledge among team members (Ryan & O'connor, 2009). As Faraj and Sproull (2000) suggested, to be effective and efficient in coordinating the expertise, the team members must know each other's skills, specialized knowledge, and experiences. Therefore, it is important to support getting to know each other more closely especially within newly established teams or after the arrival of a new team member (Faraj & Sproull, 2000). Additionally, people in teams or groups have higher trust in information and knowledge from the peers who they know than from the ones they do not know (Desouza et al., 2006) or do not perceive as reliable and trustworthy (Szulanski, 1996).

Despite organizations seeing the importance of knowledge sharing and knowledge management, it appears that efforts in this area are rather inconsistent and ad-hoc (Aurum et al., 2008). Software development is a highly competitive global environment, and Aurum et al. (2008) argued that to stay competitive in such an environment, organizations should adopt a more systematic approach for managing their knowledge.

Knowledge management focuses on capturing, storing, distributing, and applying knowledge in organizations (Aurum et al., 2008; Davenport & Prusak, 1998). It is a complex area that attempts to maximize the value originating from existing knowledge (Davenport & Prusak, 1998). This thesis primarily focuses on how knowledge is shared between peers; however, the concepts of knowledge sharing and knowledge management are closely related to each other.

2.2.3 **Motivation for knowledge sharing behavior**

Sharing knowledge is an activity that some workers perform more than others. The knowledge sharing efforts can be motivated intrinsically, extrinsically, or driven by culture and established processes. The following paragraphs will outline some of the reasons why IT professionals would share their knowledge with colleagues.

In their respected paper, Bock et al. (2005), introduced three categories of motivational drivers that influence the willingness of an employee to share knowledge, based on a synthesis of prior literature and conducted interviews. These categories and drivers are Economic (Anticipated extrinsic rewards), Social-Psychological (Anticipated reciprocal relationships, Sense of self-worth), and Sociological (Fairness, Innovativeness, Affiliation). The concept of Subjective norms refers to the subjective feeling of how much others expect knowledge

sharing behavior from the individual. The results of their research are visualized in Figure 2. (Bock et al., 2005)

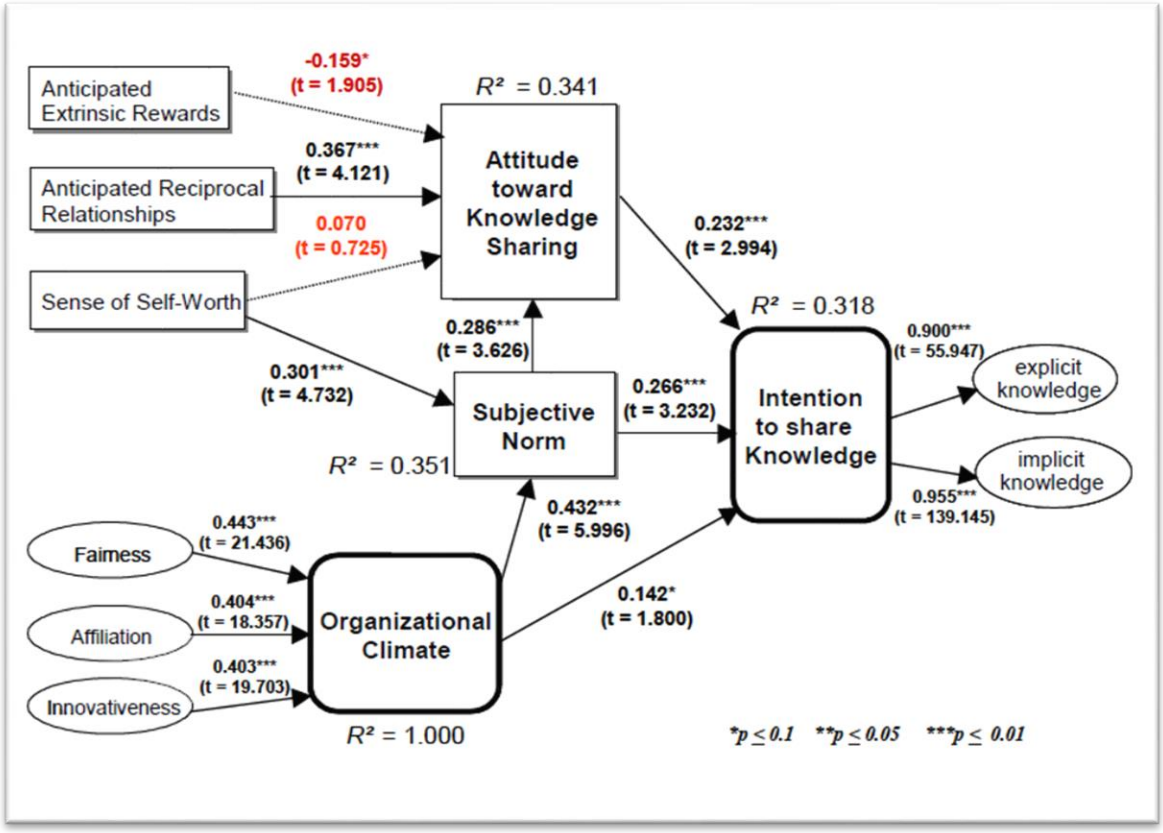

Figure 2: Knowledge sharing drivers (Bock et al., 2005)

Anticipated extrinsic rewards concern expectations of receiving rewards for one´s knowledge sharing behavior. As described in some studies, these rewards typically include monetary incentives, career progression, and their combinations (Aurum et al., 2008; Bock et al., 2005). Aurum et al. (2008) raised some doubts about the long-term effects of monetary bonuses, but they reported that career progression was identified to be a significant motivator. On the other hand, the study by Bock et al. (2005) discovered that contrary to common beliefs, extrinsic rewards might hinder positive knowledge sharing attitudes. They offered several possible explanations from other studies such as the possible negative impact of extrinsic rewards on intrinsic motivation, only temporary effects of rewards, and differences in perception of what reward is appropriate. Therefore, they concluded that extrinsic rewards should not be stressed as a primary motivational driver for taking part in knowledge sharing activities. (Bock et al., 2005) This is also supported by Szulanski (1996), whose paper's results suggested that the common practice of creating motivation through incentives seems inadequate. He recommended focusing on developing learning capacities in organizational units, building closer relationships, and communicating practices within organizations (Szulanski, 1996).

Anticipated reciprocal relations driver is about maintaining and improving relationships with others thanks to knowledge sharing behavior. This driver was found to have the most significant effect on an individual's attitude towards knowledge sharing. (Bock et al., 2005) Aurum et al. (2008) also found that gaining recognition can be a good motivator for team members. Some workers wish to help others to be efficient and to avoid frustration and might expect it to lead to a more positive working environment (Aurum et al., 2008).

The *Sense of self-worth* relates to personal feelings of being beneficial by giving value to the organization and/or colleagues through knowledge sharing activity. The sense of self-worth positively influences the subjective norm based on the idea that if one has knowledge beneficial for others, they probably expect him/her to share it. (Bock et al., 2005)

Finally, *organizational climate factors* also influence an individual´s knowledge sharing intentions. Bock et al. (2005) identified factors of *fairness* (climate of trust), *innovativeness* (creativity and changes are supported, tolerance to failure of new efforts), and *affiliation* (feeling of belonging to the organization or colleagues). Organizational climate factors were found to strongly influence subjective norm (one feels that it is supported and expected in the environment) and less strongly also the intention to share knowledge (Bock et al., 2005). Possibly related to the affiliation factor, some workers might feel the need to share their unique knowledge because they want to ensure that their eventual sudden absence (a sickness, an accident) would not have a significant negative impact on the whole team and project (Aurum et al., 2008).

Bock et al. (2005) concluded their paper by claiming that: *"Effective knowledge sharing cannot be forced or mandated."* (page 15), and organizations desiring to establish knowledge sharing should focus on empowering the facilitating factors. Before launching knowledge-sharing initiatives, their promoters should emphasize supporting the development of social relationships and interpersonal interactions between employees. They also underlined the importance of providing feedback to people as it might invoke peer pressure to become (more) active in knowledge sharing activities. (Bock et al., 2005)

Rode (2016) investigated the effects of different motivating factors on knowledge sharing in the specific context of enterprise social media platforms (ESMPs). He discovered that strong extrinsic motivational factors were *expected gains in reputation* and *anticipated reciprocal benefits*. An identified intrinsic factor was *self-efficacy in knowledge-sharing*, meaning a belief that one possesses knowledge valuable for others. Low self-efficacy in knowledge sharing decreases active participation because employees might be afraid that their contributions will be of a low value to others and in the ESMP such contributions would be highly transparent to the whole organization. Interestingly, *enjoyment in helping others* did not seem to play a role in knowledge sharing motivation in ESMPs. (Rode, 2016)

Other uncategorized motivating factors identified among practitioners by Aurum et al. (2008) included enabling delegation of work. If a unique knowledge to perform a certain specialized task is shared, others can then accomplish the

task themselves and they do not need to re-assign the task to another specialist. The primary motivational factor for sharing knowledge of practitioners from two case companies was to enable all their colleagues to perform their duties. (Aurum et al., 2008)

2.3 Distributed software development

Recent advances in technology have reduced the importance of physical collocation and made working in virtual teams a feasible arrangement; however, it has to be managed well otherwise realizing the benefits might be at risk (Desouza et al., 2006; Griffith et al., 2003).

Organizations adopt distributed working models for various reasons. Many use it to address a shortage of professional workers by accessing the global pool of professional labor (Battin et al., 2001; Herbsleb & Moitra, 2001). Connecting professionals from different cultures and educational backgrounds might result in advances in innovation or increase problem-solving capabilities (Ebert & De Neve, 2001). A substantial number of organizations aim at lowering their costs especially by hiring in markets with a lower cost of labor (Boden et al., 2009; Ebert & De Neve, 2001; Prikladnicki et al., 2003). In some cases, it might be beneficial or even necessary to be geographically or culturally closer to customers (Damian & Moitra, 2006; Ebert & De Neve, 2001). Other reasons might include follow-the-sun workflow and connected lower time-to-market, or acquisition opportunities (Herbsleb & Moitra, 2001).

Despite a rather wide scale of anticipated benefits, the constraints introduced at the same time can negatively affect achieving those benefits. The physical distance between team members introduces challenges to the accessibility of knowledge so it might be very difficult to effectively coordinate and utilize needed knowledge even though it is actually present somewhere within the team (Alavi & Tiwana, 2002). By implementing the global software development approach, the work becomes more complex and it is often needed to coordinate and integrate multiple knowledge sources (Desouza et al., 2006). Some researchers, like Ebert and De Neve (2001), have discouraged from forming virtual teams and strongly recommended establishing collocated teams with relocating experts from other countries for as long as needed.

2.3.1 **Distances**

Distributed software development introduces three distances into the cooperation between colleagues – geographical, temporal, and socio-cultural. These distances create obstacles in daily work, hinder coordination, communication, and collaboration, and make it challenging to ensure a common understanding among the dispersed software development team's members. (Carmel & Agarwal, 2001; Ågerfalk et al., 2005)

Geographical distance refers to the physical distance between coworkers (Carmel & Agarwal, 2001; Ågerfalk et al., 2005). Overcoming the geographical distance is strongly dependent on reliable ICT (Ågerfalk et al., 2005). Ågerfalk et al. (2005) pointed out an interesting thought – it can be more practical not to measure the geographical distance in kilometers but rather in how difficult it is to get from one site to another (transportation options, travel time, crossing borders, visa requirements, etc.).

Temporal distance represents the dislocation of colleagues in a matter of time due to time zone and/or work patterns differences (Carmel & Agarwal, 2001; Ågerfalk et al., 2005). Naturally, the time zone differences are well known and often significant; however, already colleagues in close time-zones (+/- 1 or 2 hours) might not have many common hours to synchronously cooperate and communicate if there are differences in working habits such as usual start and end working times and a lunch break timing (Ågerfalk et al., 2005).

Socio-cultural distance creates a separation between team members based on differences in national and organizational culture, language, values, work ethics, etc. (Carmel & Agarwal, 2001; Ågerfalk et al., 2005).

2.3.2 **Knowledge in distributed software development**

When considering that tacit knowledge is best transferred among team members via social interactions (informal interactions, direct observations, etc.) and that virtual teams have limited opportunities for social interaction, it seems obvious that their members are less likely to successfully transfer tacit knowledge between each other (Griffith et al., 2003; Ryan & O'connor, 2009). Furthermore, the preference of local communication might lead to a disbalance of accessible knowledge between remote sites (Herbsleb et al., 2001; Taweel et al., 2009).

Less virtual teams rely more on implicit and tacit knowledge and share the knowledge mostly via direct interactions and working side-by-side. More virtual teams have a higher dependency on explicit knowledge, which they share through technology-supported media. Members of more virtual teams are likely to need to transform tacit knowledge to explicit, more declarative in nature, so it could be effectively transmitted. (Griffith et al., 2003)

Virtual work hinders social interaction, so it reduces the ability to develop new tacit knowledge in the team. However, Griffith et al. (2003) suggested that making the team´s knowledge more explicit combined with the use of IT to overcome the team's distribution can have a positive side effect of creating permanent repositories of easily accessible explicit knowledge. Based on this, it might appear that even though distributed work brings many difficulties and obstacles, transforming tacit knowledge to explicit and more extensive use of technology might create some benefits (Griffith et al., 2003).

Distribution of a team represents an obstacle for creating and maintaining collective knowledge. It reduces the level of social interactions among team members, which hinders the forming of collective knowledge. Yet, the collective

knowledge in virtual teams might still be constructed in a more explicit form, and as such, it can be more easily accessible to everybody. (Griffith et al., 2003)

An interesting contrast about an approach to knowledge and its sharing might be seen between the distributed teams and teams following an agile methodology, while there are also teams belonging to both categories. While Griffith et al. (2003) put emphasis on explicit knowledge, the principles of agile development emphasize tacit knowledge and interactions (Beck et al., 2001).

2.4 Knowledge sharing challenges

Knowledge sharing and knowledge management are complex areas that involve many challenges. Those are even stronger in distributed software development because knowledge (expertise, skills, ideas, best practices) is distributed across locations (Desouza et al., 2006). Global software development has been accepted as a popular approach already a while ago, but multiple limitations and challenges have been known and existing already since then (Herbsleb et al., 2001). When utilizing the global distribution of the workforce, organizations must pay attention to knowledge-sharing challenges if they want to be successful (Wendling et al., 2013). This chapter presents the main challenges that the prior literature observed in teams and organizations.

2.4.1 **Communication challenges**

Geographical distance divides internal team communication to local and remote. Local communication can also be face-to-face but remote communication purely relies on the use of ICT (Ågerfalk et al., 2005). Even non-global distances between team members might significantly reduce communication (Herbsleb & Moitra, 2001). In remote communication, people sometimes have problems regarding knowing who to contact or reaching that person in time through available communication channels (Herbsleb et al., 2001). Herbsleb et al. (2001) discovered that members of distributed teams communicated more often with collocated colleagues than the remote ones because the local communication was perceived as more effective. Weakened or ineffective communication can be a threat to realizing the benefits of distributed software development (Herbsleb & Moitra, 2001). These findings might unveil a possible threat of creating a gap between geographically distributed parts of the team, which might be hindering the creation of social relationships and building trust.

In their study, Taweel et al. (2009) found that teams´ knowledge was negatively affected by teams' geographical distribution, especially due to the lack of informal and unplanned interactions. The type of information that is usually shared during these informal interactions was not effectively distributed across the team (Taweel et al., 2009).

2.4.2 **KMS, documentation**

Given that virtual teams are more dependent on explicit knowledge (Griffith et al., 2003), they typically use some knowledge management system (KMS) as a knowledge repository, which has an important role in facilitating the transfer of knowledge. However, Aurum et al. (2008) noticed that it might be difficult for software developers to explain how they use the knowledge they possess to solve tasks and problems. This is most probably related to the nature of tacit knowledge and the difficulty of transferring it to other people.

It is important to remember that the benefits of KMS realize only if the stored knowledge is being retrieved and applied by other team members. If knowledge is only stored but not retrieved, there is a risk that the knowledge repository becomes an information graveyard (Dingsøyr et al., 2009; Dingsøyr & Smite, 2013; Prikladnicki et al., 2003). Efficient retrieval of stored knowledge can be prevented or obstructed by the system's usability issues or design flaws. Several reviewed studies reported issues with the ineffective or missing search function of the KMS, which is crucial for finding required information in usually large content of the knowledge repository (Aurum et al., 2008; Dingsøyr & Smite, 2013; Manteli et al., 2011). Even though internal knowledge is perceived more useful because it is usually well-applicable, some people might still prefer global public internet sources over the organization's KMS if the KMS is perceived inefficient or has usability issues (Aurum et al., 2008).

Specifications, processes, implementation, or integrations evolve during the software development lifecycle. Keeping the knowledge in the repository up to date is important in distributed software development to prevent misunderstandings and incorrect assumptions (Herbsleb & Moitra, 2001). This could be achieved by setting up processes for updating and revising the repository content; however, updating the existing knowledge might not be given a high priority and can often be considered difficult (Aurum et al., 2008). This might result in lowering the quality of and trust in the KMS, which becomes an obstacle to knowledge sharing in the team.

Novice members can often be eager to use knowledge from KMS because they perceive it as safe to use and there is no need to justify why they have chosen it as a reliable source because that should not be questioned. However, they might have trouble with understanding, identifying outdated information, or be overwhelmed by the amount of knowledge present there. (Desouza et al., 2006)

2.4.3 **Social challenges**

Strong relationships between team members are important because they empower good knowledge-sharing behavior (Alavi & Tiwana, 2002; Wendling et al., 2013) and they also influence knowledge absorptive capacity on the receiver's side (Wendling et al., 2013). Infrequent interactions in the case of distributed teams lead only to weak ties between colleagues (Alavi & Tiwana, 2002), which might hinder knowledge transfer (Szulanski, 1996). On the other hand, it was

suggested by Wendling et al. (2013) that strong emphasis on relationships in knowledge management, when not accompanied by other means, can be limiting for team members without good relationships with others.

In offshoring arrangements of global software development, fear and resistance might often emerge because people might perceive their remote colleagues as a threat to their own work positions or feel a loss of control. This might occur especially towards colleagues from countries with a lower cost of labor. (Herbsleb & Moitra, 2001) "More expensive" workers can be afraid of losing their jobs while at the same time they are expected or even forced to train their offshore colleagues who represent that threat (Ebert & De Neve, 2001). Additionally, offshoring arrangements might bring cultural issues and, in the beginning, also problems with low trust in the competence of remote colleagues (Battin et al., 2001).

Some individuals might not be willing to share valuable and unique knowledge because they fear of losing the ownership, superior position in the team, and privileges related to those. Unlike the previous issue of fear, this issue that we can call *knowledge as power* can occur even at the same location among colleagues with good social relationships. Sharing the information does not have to be a threat to one's survival in the company but a threat to benefits that he/she currently enjoys. (Szulanski, 1996)

2.4.4 **Organizational, management, and procedural challenges**

While discussing the concept of transactive memory by Wegner (1987) earlier, emerging of specialized domain experts within the team was mentioned. Concentrating knowledge of a certain domain to one person is convenient for other team members, effective for teamwork, and in alignment with the idea of the transactive memory system and its benefits (Wegner, 1987). However, Desouza et al. (2006) interestingly pointed out that some people might not appreciate being labeled as domain experts because it could limit their career or intellectual growth and development only to one specific area or direction. This issue can be called *domain expert lock* because an individual's professional development is locked inside a certain domain.

An interesting point was brought up by Dingsøyr et al. (2009), who claimed that managers tend to value explicit knowledge more than tacit knowledge; however, the literature suggests that focusing only on one form of knowledge is probably not going to form a successful knowledge management strategy. Organizations should manage both tacit and explicit knowledge (Dingsøyr et al., 2009).

Planning and managing offshore arrangements is very challenging. There are plenty of options, but sometimes even small decisions can make a difference between success and failure. Two cases of offshoring reported by Boden et al. (2009) illustrated how differently cooperation with foreign colleagues can look like. The first case included intensive personal contacts, workshops, flat hierarchy, and the offshore team could come up with their own ideas and affect planning. In the second case, the environment was very formal, contextual knowledge

was not transferred to the offshore location, and feedback and expertise from the offshore team were not considered when making decisions because the offshore team´s responsibility was just development. (Boden et al., 2009)

An interesting challenge can be faced by distributed teams following an agile methodology. Agile methodologies encourage interactions over documentation (Beck et al., 2001); therefore, a strong personalization strategy is often followed in such settings. However, that can cause the documentation being outdated and knowledge being concentrated where the bigger part of the team or higher roles, like architects or specialists, are located (Manteli et al., 2011).

2.4.5 **Employee turnover**

The current situation on the IT job market has been that there is a high demand for skilled professionals, which can make it challenging for companies to retain their experts. It seems natural that there is a migration of professionals between companies. However, when an employee leaves, the company often does not lose only a human resource, but also all the specialized knowledge, skills, and capabilities that he or she possessed (Rus & Lindvall, 2002). Leaving employees often leave a knowledge gap, which puts extra requirements on others (Taweel et al., 2009). Even if a new highly skilled professional familiar with the technology is hired as a replacement, he or she still needs to learn the domain and context knowledge, and any time spent on learning that knowledge is removed from the project delivery time (Battin et al., 2001). Experiences like these highlight the importance of knowledge management (Taweel et al., 2009).

This issue is not unique for the IT industry and has existed for quite a long time. In their respected book about knowledge and knowledge management, Davenport and Prusak (1998) illustrated this issue on some known international companies. They claimed that the issue of employees leaving together with valuable knowledge contributed to a higher interest in knowledge and knowledge management because companies often realized the value of employee´s knowledge only after it was gone and left consequences to deal with (Davenport & Prusak, 1998).

2.4.6 **Technical**

Information technology plays a crucial role in allowing collaboration among team members of distributed teams (Griffith et al., 2003; Wendling et al., 2013). In the early years of distributed software development, issues such as slow and unreliable network connections were brought up (Herbsleb & Moitra, 2001); however, major technical advancements were achieved since then. Wherever technology is involved, there are naturally some constant minor issues, but it seems that no major technical issues and challenges have been identified in connection to knowledge sharing in distributed teams nowadays. Obviously, this does not concern issues regarding how the technology is used and the limitations that come with replacing personal contact by using technology.

2.5 Knowledge sharing practices

As Szulanski (1996) reported, knowledge sharing is usually perceived as difficult if it cannot be routinely handled and requires ad-hoc solutions. The existence of effective practices for sharing knowledge within the organization might reduce the perceived difficulty and stickiness of knowledge (Szulanski, 1996). Challenges of knowledge sharing and knowledge management have existed for a long time, hence, a variety of practices to address them emerged. This chapter introduces a set of different practices for knowledge sharing and knowledge management. The goal is not to provide an extensive overview of all known practices, but just to introduce some common ones. It is reasonable to believe that not all practices are applicable in all environments given the differences between organizations, distributed work arrangements, projects' specifics, methodologies in use, and individuals in the team.

2.5.1 **Knowledge Repositories**

The concept of the knowledge management system (KMS) or some knowledge repository was already introduced in the earlier chapters together with its known challenges. Based on how often it is mentioned in different studies, some form of KMS seems to be a rather standard tool in IT organizations throughout the years (Alavi & Tiwana, 2002; Aurum et al., 2008; Dingsøyr & Smite, 2013; Dorairaj et al., 2012; Prikladnicki et al., 2003; Taweel et al., 2009).

Organizations should invest resources into good design and maintenance of their knowledge repositories. Studies emphasize the importance of keeping them up to date and relevant (Aurum et al., 2008) and paying attention to good design to avoid known obstacles and issues that would prevent efficient knowledge sharing or maybe even cause the repository not being used at all and turn into an information graveyard (Dingsøyr & Smite, 2013).

Especially "*wiki-based*" KMS seem quite popular among IT organizations as they allow storing knowledge in an organized way with versioning and efficient search functionality (Dorairaj et al., 2012; Taweel et al., 2009).

2.5.2 **Informal sessions, meetings**

Aurum et al. (2008) observed that some organizations organize expert info-sessions where experts share their knowledge about interesting topics. These sessions facilitate knowledge sharing, support informal communication, and uncover knowledge network (who knows what) (Aurum et al., 2008). Identical practice for sharing own technical expertise with the team was identified by Dorairaj et al. (2012), where an agile coach emphasized that it was very helpful to also store the presentation materials in the KMS for revisiting it later.

Alongside informal sessions, formal training is another option of arranging knowledge transfer; however, formal training programs might be seen as sources

of irrelevant knowledge unless it can be immediately applied in the individual's work (Aurum et al., 2008).

Team meetings can be seen as an important opportunity for presenting ideas, giving advice, assisting with problem-solving, and agreeing on future progress, which makes them ideal for knowledge acquisition from other team members (Aurum et al., 2008). Dorairaj et al. (2012) documented the practice of bi-weekly technical sessions, where the primary focus was on an informal open discussion, rather than presentations, on technical topics.

Based on the discussions of tacit knowledge by Polanyi (1966) and Sternberg et al. (2000), it is closely related to experience. Given the difficulty of transferring tacit knowledge, performing daily activities (so-called "learning by doing"), is one of the best ways to acquire it (Sternberg et al., 2000). Practitioners also identify learning by doing among the most common sources of their knowledge (Aurum et al., 2008). Some practices to stimulate learning during normal work include mentoring by an experienced colleague, assigning or selecting specific types of assignments, encouraging to seek novel tasks, and job rotation (Sternberg et al., 2000). Some practitioners also consider code reviews useful for acquiring very specific knowledge (Aurum et al., 2008) because it originates in the team and is directly applicable there.

2.5.3 **Social aspects**

Kotlarsky and Oshri (2005) attempted to figure out whether social ties and knowledge sharing support successful collaboration in global software development teams. At the time of their study, not enough attention was paid to human and social aspects involved in the globally distributed cooperation and their effects on coordination and collaboration success. Existing solutions were mainly technical. They claimed that social ties (trust and rapport) and knowledge sharing (transactive memory, collective knowledge) improve collaboration in distributed software development teams. Therefore, organizations should pay attention to and support the creation of social ties between members of globally distributed teams to ensure successful collaboration. Resources should be dedicated to addressing the human aspects of inter-team collaboration. (Kotlarsky & Oshri, 2005)

Cultural issues and lack of trust in remote colleagues´ professional qualities might be addressed especially by an increased volume of interactions, for example in a form of common team meetings (Battin et al., 2001). Another possible option to build mutual understanding of different cultures might be building international teams or rotating management across sites (Ebert & De Neve, 2001). Battin et al. (2001) highlighted the important role of liaisons from the remote site, identifying them as the key success factor of the case project. These engineers relocated for few months to the main site to learn the system and participate in planning activities, but importantly, they also established good relationships with the central part of the team (Battin et al., 2001).

Generally, frequent communication and collaboration are crucial. The study by Dorairaj et al. (2012) found that active daily collaboration between members of distributed teams supports knowledge creation. Moreover, short physical visits support more efficient knowledge sharing through face-to-face communication and getting to know each other (Dorairaj et al., 2012). While some companies use short visits to remote sites (Boden et al., 2009; Dorairaj et al., 2012), it was reported that some companies can be careful with travelling because of high cost and being time-consuming (Battin et al., 2001).

2.5.4 **Locating the knowledge**

Keeping track of knowledge location within a team can be challenging in a distributed setting. One way to avoid possible awareness and coordination issues can be an index of knowledge, which explicitly maps who knows what (Dingsøyr & Smite, 2013). Davenport and Prusak (1998) refer to this mapping as the "Yellow Pages".

Some companies use Enterprise Social Media Platforms (ESMPs) for supporting communication and collaboration. To fulfill its benefits, active participation of employees is naturally needed; however, the study by Rode (2016) reported that only 10% of registered users were actively participating in the ESMP.

The existence of global online area-specific communities was reported as a way to provide quick and focused support (Aurum et al., 2008). Similarly, communities of practice were reported by Dorairaj et al. (2012) as a platform to discuss and suggest possible solutions to problems in different projects.

2.5.5 **Organizational, management, and procedural practices**

Preceding chapters explained why knowledge management is important in the context of distributed software development teams. However, from the practitioners' point of view, it might not be given too high priority (Aurum et al., 2008; Dingsøyr et al., 2009).

The organizational environment is an important aspect because it influences the initiation, implementation, and results of knowledge transfer (Szulanski, 1996). Moreover, the cultural practice of promoting sharing knowledge is considered very important for software developers (Aurum et al., 2008). Prikladnicki et al. (2003) reported that investments to knowledge management in form of tools and practices encouraging knowledge sharing reduced many obstacles in global software development.

Szulanski (1996) pointed out the importance of routines and guidance for sharing knowledge because their presence can reduce the perceived difficulty of that process. Based on Aurum et al. (2008), managers should evaluate if suitable knowledge management tools are offered and knowledge management activities should be included in the project plans and schedules. That way, they will not be only a side activity done voluntarily by a few active team members. Additionally, it could increase the motivation of other team members to be active in knowledge

management activities as well. Moreover, knowledge management activities should be performed during the project lifecycle and not only after the project is completed. (Aurum et al., 2008)

Needed knowledge management (KM) activities differ based on the project's nature and specifics, so it would be difficult to have a generic KM process model. Each project or team should have its own KM process model constructed. (Aurum et al., 2008). When forming the KM model, teams should at first identify KM needs, challenges, and barriers and afterward define what knowledge should be shared where and how (Dingsøyr & Smite, 2013). It should be clear who is responsible for what part of KM. There may be a specific role responsible for KM or it can be distributed to all the individuals in the team to manage their own knowledge. (Aurum et al., 2008)

The current situation regarding KM models and processes in organizations does not seem satisfactory. Prikladnicki et al. (2003) reported that both studied organizations were missing a formal and consistent KM process, which represented a major obstacle for sharing knowledge. A survey by Dingsøyr et al. (2009) comparing the current situation (importance) of KM in organizations and the desired future situation, showed that practitioners see quite a large potential for improvement in KM processes in their organizations.

2.5.6 **Onboarding, training new employees**

Employee turnover was previously mentioned as a serious challenge that organizations currently face and need to cope with. When experienced people leave, they are often replaced by less experienced ones, who have a lot of knowledge to absorb especially in the beginning (Rus & Lindvall, 2002). Onboarding is a process *"in which new employees gain the knowledge and skills they need to become effective members of an organization"* (Dictionary.cambridge.org, 2019).

Mentoring is a practice, where experts assist and provide support to less experienced colleagues, primarily via sharing their knowledge. In their effort to improve an existing mentoring program in the case company, Bjørnson and Dingsøyr (2005) discovered that mentoring is viewed very positively. Employees had a very positive attitude towards the mentoring program, and it was considered important – good for transferring competence, consulting solutions to problems, creating relationships, etc. Employees thought that mentors should accept the role voluntarily and that a new employee should not have to ask for a mentor but be automatically offered one in the beginning. The mentor should have a certain time allocated for the mentoring activities. The original form of the mentoring program before the study was focusing mostly on assistance with practical issues, where discussion and reflection were missing. Authors believed that to boost the learning effect, mentors should not only provide direct answers but instead, they should take the role of a discussion partner and ask open questions to encourage student´s own thinking. Finally, the mentor should be proactive even when no questions are asked by a student. (Bjørnson & Dingsøyr, 2005)

Pair programming of a team of a junior and a senior developer is one of the practices that could help to transfer experience in form of best practices and understanding performance impact, which even highly technically knowledgeable junior developers are often missing (Dorairaj et al., 2012).

Codified knowledge, usually in some KMS, is considered useful for new team members because they can find a lot of information there, connected by links to related topics (Dorairaj et al., 2012; Taweel et al., 2009) and they often consider it as a reliable source of information (Desouza et al., 2006).

2.5.7 **Agile methodologies and knowledge sharing**

Adopting agile methodologies can have a positive effect on knowledge sharing. *The Manifesto for Agile Software Development* emphasized individuals and interactions (Beck et al., 2001) and social interactions were proven a good way of sharing and growing tacit knowledge, which then may result in a higher team's performance (Ryan & O'connor, 2009). A commonly used argument that agile methodologies refuse documentation is not completely correct. Agile teams indeed put less emphasis on documentation; however, they do not leave it out totally, they only choose when it is suitable and efficient to create and maintain it. (Dorairaj et al., 2012)

2.5.8 **Communication and tools**

One of the most used communication tools in distributed teams appears to be an instant messaging (IM) tool, also known as "chat" (Manteli et al., 2011; Niinimaki et al., 2010; Wendling et al., 2013). Its popularity can be attributed to multiple factors. Practitioners consider it an efficient tool because of its informality (compared for instance to an e-mail) and lively interaction since the informality decreases the overhead of communication such as thinking about grammar, spelling, etc. It is also perceived as a lightweight tool for quick simple questions. (Niinimaki et al., 2010) Manteli et al. (2011) reported that the recent introduction of an IM tool in the case environment was perceived as a significant improvement for communication because communication over chat is much faster and the tool allows to see when remote colleagues are online and available.

E-mail and mailing lists are standard and widely used communication channels, which have a more formal nature, and which might also serve as permanent storage of communication and documents (Manteli et al., 2011; Niinimaki et al., 2010).

A very common synchronous communication channel is audio/video conferencing primarily used for daily or weekly team meetings, coordinating work, and discussing ideas. One-to-one calls might be perceived as intrusive and disturbing since topics discussed via this tool are often complex and difficult to answer, but in urgent cases, such calls can be very efficient. People seem to prefer textual communication for simple matters and audio for more complex matters and discussing ideas. Desktop sharing has also proven useful for presentations

and demonstrations to customers, trainings, and problem-solving. (Niinimaki et al., 2010)

In their study to understand communication tools and related practices for overcoming distances in global software development, Niinimaki et al. (2010) emphasized that it is important that communication processes and tools suit the team and project both technically and socially. They suggested that mutual agreement about the use of different tools should be made – how tools are used, for what purposes, the usual response time to an e-mail, being logged in to an IM tool whenever working and available for communication, where to store information permanently, and informing the rest of the team about important private conversations (Niinimaki et al., 2010).

2.6 Summary of the theoretical background

This chapter introduced concepts important for this study and prepared the background for empirical research. It was established what types of knowledge will be considered and what is meant by collective knowledge. The study aims primarily at the tacit knowledge that is available in the team. Further, it was explained why coordinating knowledge is important and what motivates individuals to share their knowledge. This is essential for understanding how important knowledge sharing is and how and why knowledge is or is not shared between colleagues. Later, the context was further specified to distributed software development and its complexity was outlined. This context was selected as it has become a very usual arrangement in many organizations and teams and it is a very challenging environment for knowledge sharing, so it deserves the attention of researchers. Finally, challenges and known practices reported in prior literature were introduced to gain an overview of the situation in organizations in recent history.

3 METHODOLOGY

This chapter presents what are the objectives of the study and what approach was selected to achieve them. At first, the objectives and research questions are introduced. Then, the chosen methods and strategies are discussed, followed by describing the case environment, data collection process, and how the data were analyzed. In the end, the ethical limitations are explained.

3.1 Objectives and research questions

Considerable attention has been paid by researchers to knowledge sharing in software development organizations and there are known practices on how to share knowledge effectively and efficiently. However, signals from practitioners often suggest that the quality of sharing and managing knowledge is unsatisfactory. They often state *"people don´t want to do it"* as a reason. This research tries to understand what are the factors that discourage or demotivate practitioners in software development teams from effectively sharing and managing their knowledge. In simple words, the question is: *"Why they don´t do it?"*

The study aims to answer the following research questions:

1. What are the challenges and practices of knowledge sharing and knowledge management in software development teams?
2. What discourages or prevents the adoption of knowledge sharing and knowledge management practices in software development teams?
3. How could the quality and quantity of knowledge sharing be improved?

The focus is on the knowledge inside one software development team and how it is shared and managed there. Sharing knowledge between different teams in an organization is out of the scope of this research. Such a scenario involves additional factors, like for example applicability of knowledge outside of its original context (Szulanski, 1996), and it could divert the attention from the main objectives here.

3.2 Selected methodology

In order to achieve its objectives, this study employs a qualitative research approach with an interpretive case study method. The unit of analysis is a software development team and data were collected via semi-structured interviews. The description and justification of selected approaches and methods follow in the subsequent paragraphs.

The main streams of research in the information systems (IS) field can be divided into quantitative and qualitative. Quantitative research primarily works with numbers and often aims to generalize or validate theories. Qualitative research, on the other hand, intends to understand human perceptions, understanding, and experiences within a specific social and cultural context. (Myers & Avison, 2002; Stake, 2010) The importance of qualitative methods in the IS field increased as the focus shifted from technological topics to organizational and human-technology interaction issues (Myers, 1997). The sample size in qualitative research is typically smaller and traditional data collection methods include interviews, observations, questionnaires, document analysis, etc. (Myers & Avison, 2002; Stake, 2010). This study aims to understand the topic of knowledge sharing from the practitioners´ point of view and describe and analyze their opinions, experiences, and attitudes. Specifically, it seeks to identify the discouraging factors, i.e. why people would not be active in knowledge sharing activities. Therefore, qualitative research is an appropriate approach.

The idea for this research has been primarily driven by practical needs after an interesting phenomenon was noticed among practitioners. Despite seeing the importance of sharing knowledge with their colleagues, they are somewhat reluctant to be active in that process. Given the objectives, the low coverage of this issue by existing literature, and the exploratory nature of this study, there is a high emphasis on collecting rich data from practitioners´ point of view with very few or no preconceived theoretical assumptions or hypotheses. Therefore, the interpretive research approach was selected for this study because it corresponds to the needs described above. Interpretive research does not aim to generalize but instead attempts to understand the studied phenomena more deeply through understanding the assigned meanings, beliefs, and intentions of people in a certain environment (Klein & Myers, 1999; Orlikowski & Baroudi, 2002). The ideal case would be that the researcher keeps an open mind when interpreting the data; however, it is recognized that in reality, the researcher´s beliefs, values, and assumptions inevitably have a certain effect on the investigation and interpretation of the data (Orlikowski & Baroudi, 2002; Walsham, 2002).

The case study is the most common method of qualitative research in the field of IS and it aims at investigating a phenomenon in its natural context (Eisenhardt, 1989; Myers & Avison, 2002). Benbasat, et al. (2002) summarized three reasons why case study is a feasible research strategy in IS: (A) It allows to study a phenomenon in its natural context, observe the current practices, and generate theories from practice; (B) It allows to answer questions of "Why?" and "How?"; (C) It is a suitable strategy for areas where little research has been conducted, which allows coping with the fast pace of changes in this field. Other commonly used methods in the IS research (laboratory experiments, field experiments, field studies) involve the researcher manipulating or measuring defined variables. In case studies researchers do not practice experimental control or manipulation and instead take the role of an observer or an investigator. (Benbasat, et al., 2002) The case study is considered the best approach for this study for multiple reasons. The knowledge used in software development teams, working style and culture,

and motivation to share knowledge are tied to its natural environment. The ultimate aim of this research is to understand *why* practitioners do not share their knowledge and *how* it could be improved, and those are two types of questions (why, how) that the case study approach is particularly suitable to answer. Finally, there is a research gap regarding this area and perspective. The unit of analysis is a software development team because the focus of this study is on knowledge existing in the team and how it is shared between the team members.

The qualitative interviews belong to the most important data collection methods in qualitative research (Myers & Newman, 2007; Walsham, 2002), and as Walsham (2002) argued, they can give the researcher very good access to the interviewee´s interpretations and views. There are several different types of qualitative interviews, such as structured, semi-structured, or theme interviews. These differ in how much is planned and prepared in advance and how much improvisation is allowed. The semi-structured interview was chosen for this study because it has an incomplete script with space for improvisation (Myers & Newman, 2007), which is suitable for the exploratory interpretive research at hand. It gives the researcher the flexibility to ask further questions in case clarification is needed or to encourage interviewees to elaborate on their interesting thoughts (Myers & Newman, 2007).

3.3 Case description

Empirical data were collected in one IT company by carrying out semi-structured interviews with its employees. The main goal was to understand different points of view on knowledge sharing and knowledge management activities, and obstacles and factors that prevent, discourage, or demotivate from adopting such activities into practitioners´ work.

The case organization is a medium-sized Nordic IT company, which is open to modern and agile approaches and seems to support many improvement opportunities. The company representatives were contacted during autumn 2019 to scan for possible interest in participating in this research effort. A meeting with 3 team managers was organized and all of them showed a high interest in the topic of this research and immediately agreed to participate with their teams. Their main motivation for participation was a wish to develop their processes and practices and improve the current situation with knowledge sharing in their teams.

The teams differed from each other in multiple factors. One team was collocated, while the other two distributed. One team focused primarily on development, another on development and support, and the last one was largely supporting other teams and projects. Further details about the participating teams cannot be disclosed. Table 1 shows the number of participants from a single team, average professional experience within the team, and the list of years of experience of individual team members. Please note that the order (A, B, C) is random and does not correspond to any listing in this chapter.

Table 1: Study participants - team distribution

Team	Members	Avg. experience
A	4	4 (6; 5; 3; 2)
B	2	12 (20; 4)
C	4	19 (20; 20; 20; 15)

The initial selection approach was to identify practitioners, who would be interested in participation combined with careful consideration of how suitable these potential participants are and what characteristics might be missing in the sample. The team leaders were very helpful in this stage, as they promoted the research to their team members with the help of presentation material provided by the researcher. The search produced 11 interested professionals, out of who 10 took part in the interviews. The evaluation of interested participants showed very good coverage in terms of experience levels (junior/senior), professional roles, and time spent in the company. The only insufficiency was seen in distribution among teams in a ratio of 5:2:4. The ideal minimum number of participants from the same team was considered to be 3; therefore, an attempt to acquire more participants from the second team was made to increase the data validity. Unfortunately, it did not succeed. No rewards were promised so the primary motivation most probably came from an interest in the topic and recommendations of the team leaders.

Ten participants are considered a good number suiting well for the purpose and the intended scale of this research. As Table 2 shows, the sample covers a variety of different specializations that can be found in a software development organization – 4 software developers, 1 infrastructure specialist, 2 application management (support) specialists, and 3 managers. All managers had a very technical background and they were still actively engaged in technical work, each in one of the three aforementioned areas (software development, infrastructure, support). This was considered a benefit because they could offer a combined perspective of managers together with extensive experience in technical roles.

Table 2: Study participants - roles

Position	Participants
Developer	4
Infrastructure	1
Support	2
Manager	3

To avoid an elite bias, it was important to gain insights from people with different levels of experience in the field and this seems to be achieved well. When asked about their perception, no participant considered himself as a junior in the IT field anymore; however, multiple were still at the relative beginning of their

careers. Based on self-evaluation, participants were equally distributed among middle-experienced (5) and senior specialists (5). The differences in the years of experience between these two groups were significant with middle-experienced having 2-5,5 years and senior ones 15-20 years (Table 3). For practical reasons, the thesis divides the participants into two groups – *juniors* (<6 years of experience) and *seniors* (>6 years of experience). Juniors might more often be in the role of knowledge recipients, while seniors might more often hold the role of knowledge holders. This division results in groups of the same size (5) with a significant gap between them.

Table 3: Study participants - experience

Professional experience	**Participants**
2-6 years ("juniors")	5
15-20 years ("seniors")	5

Regarding the time spent in the case company, 3 participants were new (< 3 months), 3 spent a considerable time there (>4 years), and 4 were in between (1-3 years). Half of the participants were Finnish and half were from Central European countries; however, all the participants were accustomed to the Finnish working culture.

Unfortunately, all 10 participants were men because no women were interested or able to participate in the full interviews. However, several women were consulted in form of shorter informal discussions about the topic of this research and their opinions and experiences. For the aforementioned reason, only masculine form (he/his/him) is used when talking about the interview participants. The age distribution of participants is considered good as Table 4 illustrates.

Table 4: Study participants - age

Age	**Participants**
20-29	3
30-39	3
40-49	4

3.4 Data collection

Interviews were conducted in a semi-structured format. Areas of interest were prepared in advance in the interview guide; however, a more free discussion was possible to allow further clarifications and questions related to what the participant said. The order of topics mostly followed the interview guide but sometimes the order changed, for example when a participant already covered a certain

topic earlier. The interview structure is attached as Appendix 1. The interview themes were constructed based on the prior information from practitioners (their observations, problems, needs, habits, etc.) and the conducted literature review.

To make the participants fully informed, each interview started with introduction information where the researcher introduced himself and described what is the research about, what is it aiming to achieve, and how the interview will proceed. Processing, analysis, and use of the recorded data and privacy protection mechanisms were also explained. All the participants found this information clear and had no additional questions.

After this introduction, basic data about the participant were collected. Then, three interview themes were discussed. The first theme focused on understanding participant´s experiences and opinions about knowledge sharing and knowledge management, and motivation to share their knowledge. The second theme investigated the state of the team environment, how knowledge is managed there, and what effects do the environmental factors have on knowledge sharing. The last theme turned the focus more closely to the specific practices because discussing the real use-cases and experiences could have uncovered interesting insights.

Interviews were conducted at the end of the year 2019. Interviews mainly took place in the company premises to make disturbance of participants´ work as short as possible. Most of the interviews (8) were conducted personally in 3 locations in 2 different countries. This was the preferred way as it was believed that meeting personally can create a better atmosphere and relation between the researcher and the interviewee. Despite the best effort, for two participants meeting personally was not an option and they preferred using online video-conferencing software.

The face-to-face interviews were recorded using two separate devices to ensure that data could not be lost due to a technical malfunction. An unexpected benefit was that some inaudible parts were more clear on the second recording of the same interview. Naturally, participants´ permission for recording was always asked first. Interviews via online conferencing software were recorded using the built-in recording function.

The expected length of the interview was 45 minutes, which was considered sufficient to cover all the topics at a good level. In reality, the interview length ranged from 41 to 66 minutes with an average duration of 55 minutes. The whole time was used efficiently, and participants were not deviating from the topic, so the longer length was caused purely by participants being interested in the topic and describing their points of view and professional experiences more in detail.

Some secondary data were collected as well. These came from the initial discussions with the company representatives before agreeing to participate in the research and brief unstructured and unscheduled talks with other employees, who did not take part in the interviews. These data were collected only in a form of notes and they were used for getting a better picture of the current situation in the case teams and the whole company. Additionally, they served as wider support for and a better understanding of some claims of interview participants. A

small number of documents were also made accessible for examination by the researcher.

3.5 Data analysis

The recorded interview data were transcribed using text processing software. The chosen transcription style was *clean verbatim*, which uses less strict rules and allows for example removing false starts, non-speech sounds, etc. This style allowed faster transcription producing text that is easier to read while still maintaining a high precision of the transcription.

The transcription process resulted in almost 160 pages of text. The text documents were then imported into ATLAS.ti qualitative data analysis software. The software made the process of coding the data much easier and more structured and reliable. Friese (2019) advocated that using computer-assisted qualitative data analysis software improves the research process and enables the researcher to gain insights that might otherwise stay hidden.

This research is motivated primarily by practical issues and needs. It attempts to understand and describe a phenomenon, whose coverage by existing literature is limited. In such cases, the inductive approach seems ideal. Specifically, the *Conventional content analysis* was chosen for analyzing the collected qualitative data. In this approach, the codes and code categories emerge from the data, and the researcher is not restricted to any previously formed categories. The benefit of this approach is that the researcher keeps an open mind when analyzing the data and it reduces the possibility of the researcher forcing some preconceived categories or results. (Hsieh & Shannon, 2005)

At first, all the transcripts were carefully read to gain overall impressions and awareness. Then, the coding cycles started and initial codes were emerging. During the process, codes went through phases of development as they were merged or renamed. Related codes were then linked to appropriate themes. The coding process led to 115 codes, which were categorized into 13 themes. These themes represented the identified discouraging factors in knowledge sharing in software development teams that the study was aiming to identify.

3.6 Research ethics

As the research focuses on discouraging factors, so why people do not do something that might be expected or even required, the topic is viewed as sensitive and it was expected that participants might hesitate to fully admit their behavior or motives. A no-judging policy was emphasized and interviewees were asked to be open and honest because that is the only way how the real problems might be identified. The overall impression is that the level of honesty was very high and many statements even about more sensitive or awkward topics occurred.

This might have possibly been affected by the Nordic working culture, where honesty and open communication are expected and encouraged even about things that did not go very well.

As a high level of privacy was promised to allow participants to be open and honest, certain measures had to be taken in reporting the data and results. Information about participants is primarily provided in a grouped form. Despite the best effort to keep the quotations only general, there is a minor possibility that somebody very familiar with the environment could guess the identity of the author. Not using any form of participant ID prevents confirming such a guess based on other linked information and potentially tracing other quotes from the same author, which might contain more sensitive statements (for example about one´s colleagues). It was recognized that some information about the participant can be important for evaluating the importance or relevance of his opinion and for highlighting certain differences in points of view based on the level of experience or professional role. Therefore, an indicator of one´s professional experience is disclosed on the quotation level using the values of junior/senior. If the professional role of the author of the quotation is important, it can be disclosed on a case-by-case basis.

Statements containing potentially sensitive or confidential company information were not used in direct quotations or such parts were removed or replaced by generic terms.

Interview participants were given an opportunity to read the final text and raise any concerns, objections, or comments. The final text was also read and approved by a representative of the case company.

4 RESULTS

This chapter presents the results divided according to identified factors (themes from the coding process) that might influence knowledge sharing in software development teams. Some factors are negative by nature, therefore, lowering the likelihood of individual team members being active in knowledge sharing activities. Other factors have a positive nature and their negative effects on knowledge sharing can be caused by their insufficient presence in the team context. Multiple factors were discovered to pose none or only little problems in the specific case context; however, it is believed that it is important to report such findings, demonstrate that they were considered, and point out that they might be significant in other environments.

Table 5 summarizes the factors identified in the data during the process of coding. These factors are discussed in detail further in this chapter.

Table 5: Factors affecting knowledge sharing behavior

ID	Factor	Questions / Issues
1	Importance	Is it needed? Does it make sense?
2	Motivation and benefits	Why would I do it?
3	Willingness	Do I want to share/give up my knowledge?
4	Perceived difficulty	What are the costs? Do I know how to do it efficiently and effectively?
5	Lack of attention, initiative	Not paying attention to knowledge sharing.
6	Lack of time	Being busy. Focusing on primary activities.
7	Discomfort and fear	A concern of being wrong or asking "stupid questions".
8	Nature	Being considered boring.
9	Knowledge to be shared	There seems to be no knowledge to be shared.
10	Support, Allowance	Are the efforts supported and spending time allowed?
11	Encouragement	Is it actively encouraged?
12	Missing systematic approach	Is there a model, a process, a structure?
13	Appropriate tools	Are the tools efficient and effective?

4.1 Importance

At the beginning of the interviews, participants were asked about whether they consider knowledge sharing in software development teams important. Not considering the process important was expected to have a negative impact on performing it. This question was intentionally asked before discussing benefits, obstacles, and any other aspects of knowledge sharing to capture the participant´s opinion before he started to think about the topic more.

All the participants clearly expressed that knowledge sharing is very important in the context of software development teams. However, few participants recognized that the importance of knowledge sharing depends on the context, size of the project, and the team. Some answers even gave the impression that it is pointless to ask whether sharing knowledge is important because the answer is obvious. One participant also immediately pointed out an interesting contrast between the general agreement on the importance and the current situation based on his observations:

> I think there is no one who would say that knowledge sharing is not important. Everyone agrees with that but then nobody is doing it. (junior)

Another participant expressed his opinion that knowledge sharing is important, but it is not receiving as much attention as it would deserve:

> In general, it should be kind of more important than it is at the moment. (senior)

Overall, there seemed to be a high level of agreement that knowledge sharing is very important in the context of software development teams.

4.2 Motivation and benefits

It was expected that if individuals would not know why they should invest their time and energy in knowledge sharing activities, it might significantly reduce the quantity and quality of knowledge sharing in the team. When asked about their motivation or what benefits they see in knowledge sharing, interviewees were generally having no problems to come up with ideas and they covered a wide range of motivation factors and benefits, which can be grouped into three categories: Helping team colleagues, Helping the team, Substitutability and delegation of work.

The first motivating factor was the desire to help team colleagues. Several participants stated that knowledge sharing can help people to do their job, make it easier, remove obstacles, and avoid frustration. There were also signs of anticipated reciprocal relationships as one participant explained:

> I think it´s in the interest of all the team members so if someone needs help from me, from my side, I always try to find a timeslot to have a call or […] just answer some questions and on the other side I´m also, let´s say, expecting that someone can help me if has a bigger knowledge, bigger experience than me. (senior)

Most of the participants connected their motivation and possible benefits to helping the team as a whole. They believed that integrating knowledge of people of different backgrounds allows team members to learn, which results in making the work easier, the team stronger, and overall improving the team´s performance. Hearing the opinions of others also allows for producing better solutions.

Emphasized benefit for the whole team was increasing efficiency by reducing the time required for finishing tasks by possessing better and wider knowledge and eliminating spending time on new solutions to problems that have already been solved by another teammate. Higher efficiency means saving money and delivering results faster.

> Mainly because if we would share more information, then we don´t need to spend that much to investigate on our own and it would basically save money from the company. […] But currently, it is that we are just investigating every single time the case even someone basically has already investigated it back then. (junior)

At the same time, two participants pointed out a possible negative side of this increased efficiency. Some people in IT have the desire to discover and "play" with solutions to problems at hand. Good knowledge sharing could weaken the joy from work by eliminating or shortening this problem-solving part:

> I remember […] I was angry if I resolved [a task] too quickly. If it was too easy […], I didn´t even enjoy it. So it can be that people want to actually investigate and they don´t want that everything is just served there fully that you just search the wiki, the documentation, and do according to it. (junior)

Two software developers expressed the importance of knowledge sharing between the development and support sides of the team. According to them, well-working knowledge sharing allows support specialists to gain and keep awareness about the whole system, how it works, what are its parts, and what are the recent updates made to those. At the time of the study, there were some problems in this area and support specialists sometimes lacked information about changes and new features because those were not documented anywhere. That introduced obstacles and slowed down the work.

Logically, knowledge sharing allows easier substitutability of workers. Several participants perceived it as a good thing if they can be substituted. One participant said that he does not want to be the only one capable of performing certain activities, because a sudden absence of one team member should not have a significant crisis effect on the team and its capabilities. Talking about examples like "breaking a leg" or "being hit by a bus" seems to belong to an IT professionals´ folklore when talking about too much critical knowledge being concentrated only in one person´s mind.

> […] everybody should keep in mind that the documentation should be on that level, so if I break my leg and be away for a month, so somebody is able to take my responsibilities. (senior)

> At least the factor if bus drives over that person, […] then it would be pretty disaster for us because one man basically knows too much. And it is only in his head. (junior)

Many participants seemed to have a lot of work and they would be happy to delegate some tasks if somebody else would be capable of completing them. In case of an expert moving to a different project, good knowledge sharing can

prevent the need to still support the old teammates, because they suddenly have a hole in their team knowledge.

One participant enjoyed the variety of his tasks and preferred avoiding routine. While doing the same kind of task repeatedly, one does not learn anymore and there is no professional development.

> My thinking was that when I handle some incident or something complicated, I don´t want to do it again! I want that someone else knows how to do it and I don´t have to do it ever again. And it is why I usually want to document it, the process. (junior)

Both senior and junior participants pointed out that well-working knowledge sharing has a positive impact on the onboarding process of a new team member. It makes it easier to start in the team and allows faster progress on the way to profitable work. If it is done well, it also reduces the number of questions asked.

Overall, it seemed that participants had a clear idea about why it would be good to share knowledge in software development teams and what benefits it can bring.

4.3 Willingness

A major issue for knowledge sharing can be if people are not willing to share their knowledge. Knowledge is an important resource of a company, but also of an individual, and its holder can enjoy certain benefits from its exclusive possession. It was believed that people could hesitate to admit unwillingness to share their knowledge; therefore, interviewees were asked about their impressions and observations about the willingness of their colleagues to share knowledge.

Many interviewees did not see an unwillingness to share knowledge as a problem in their teams. Only two participants expressed speculations that some people might withhold certain information in a minor scale. One had the impression that some people are a bit cryptic about their knowledge because when asked a question, they provide only brief or vague information. The second person pointed out an interesting idea that people might not be proactive in sharing their knowledge, because they want others to contact them and ask questions. After all, it gives a good feeling of being a valuable member of the team.

> And maybe, it might be that you want to have that professional feeling that "I know stuff!" And people would ask for your help. (junior)

Asking questions seemed to be the primary practice of knowledge sharing within all the participating teams. No participant had seen that somebody would not be willing to answer their questions in their teams; however, certain challenges in form of delayed answering (after a few hours or the next day) were admitted.

No signs were reported about people withholding their knowledge from their new or offshore colleagues as a protection mechanism to keep one´s job

position. One senior participant actually described the opposite situation when irreplaceability can be a burden for that person preventing him/her from following new career opportunities:

> I´ve seen in that way that if I have documented my work and somebody else is able to do my tasks, I have the possibility to change my role at some point. If I don´t, I will be stuck in the same position for the rest of my career. (senior)

His thoughts were confirmed also by the cases of 3 different domain experts (not participants of the interviews), who in the past belonged to one of the case teams. Each of them was the only expert on their specific domain. They were in the situation introduced earlier as the *domain expert lock*. In a relatively short period, all of them independently decided that they need a major change of domains (technology, tools) and moved to different projects or companies to seek new challenges and career development. The researcher confirmed the domain expert lock being the reason with one of the experts and it was most likely the reason also for the other two based on their statements when leaving. Knowledge sharing from these experts was not on a very good level. They were willing to share, but knowledge sharing was not happening, because the domain was exclusively their responsibility. Their departure left the team with a big problem to solve. As one participant remembered, it was possible to solve it, but the costs were rather high:

> [Domain expert 1] knew a lot and when he left, we thought it is a disaster. And [domain expert 2] left and we thought that: "Uff, this is a super disaster now!" But always it´s just that someone else will step up and learn all the stuff. But from the company point of view, it is just a waste of resources, I would say. (junior)

A possible issue with the willingness to share knowledge was brought up by two participants in connection with potentially introducing practices to improve knowledge sharing in the case teams in the future. It could be called *"formality resistance"* and it highlights the importance of deciding the future steps carefully.

> But I think that´s really, like setting this as some kind of rule just makes that everyone hate it because it´s way too formal. (junior)

> He said that we need these guys that they can do this job. And then I planned and handled the rest of that. And it was pretty nice! And if we give too many goals already to the mentor itself that this person needs to handle these goals, then the mentor is like: "This is stupid, I don´t know if I even want to do this anymore!" Because freedom is also one nice part of it. (junior)

4.4 Perceived difficulty

There are many challenges that a person faces when conducting knowledge sharing. There is a possibility that if the difficulty is seen as high already before the knowledge sharing starts, some people might decide to completely avoid it or be

much less active in the process. Signs of perceived difficulty and obstacles were captured throughout the interviews in connection to different topics.

One aspect of the case environment that is good to mention at this point is that interviewees had a noticeably strong association between the concepts of *knowledge sharing* and *documentation* (written knowledge in their knowledge management system), sometimes using these terms interchangeably almost as synonyms. This is further discussed in the last section of the Results chapter.

Level of quality

Some practitioners might have too high expectations about the required level of quality and extent of documentation that they rather avoid contributing to it and just continue with another task in line.

> And maybe you start and write something and: "This is too much work and I don´t know how to do it that it looks professional!" Then you just discard the whole draft and: "I will just keep my own notes from the task!" (junior)

Another participant suggested at least a partial solution to this problem. His philosophy was that something is better than nothing, so the documented knowledge does not have to be very detailed, because already brief notes can be beneficial for other team members.

> […] the best and easiest way how to share some knowledge is just create some page and even not go to detail but create at least some starting point […] (junior)

Regarding the possible difficulty of transferring the knowledge to explicit form, the same participant suggested that it is not so important that the documented knowledge is in perfect form. Instead, he recommended promoting documenting knowledge in any form that is suitable for the knowledge holder, as long as it captures and communicates the important and helpful knowledge. Once the chosen form starts to bother people, it can be changed.

> Any way is a correct way, better than to not do it! So if someone wants to draw it on paper and then make a phone picture and upload it, that is also all right. […] they can record a voice message if it´s better for them […] (junior)

One way how some people in the studied environment approached high perceived difficulty of writing knowledge into the team´s knowledge management system was keeping written knowledge only private. In such a case, no requirements or expectations on quality, form, or structure had to be considered, because nobody except the author had access to it.

Guidelines, knowing how to

Members of software development teams might often be given freedom in many areas and they are trusted to work as they see the best based on their knowledge and experience. However, in the case of knowledge sharing this might not be the best approach, because the result might be that then many people avoid it.

> There is no specific form or method or, no clear way how to do it; therefore, it´s not happening. Or it´s limited or it´s not systematic. It´s based on an individual´s approach. (junior)

> […] actually, I can´t even remember that I would have seen any kind of training or any sessions on how to actually plan and implement the documentation well and what should be the structure or… I can´t remember any training lessons or any brownbags or anything related to that. It is just that we do how we see the best… […] It is just that kind of free area what we should do but we don´t. (junior)

There was a wide agreement that the existence of some simple guidelines on how to share and document knowledge would be helpful. Even though some guidelines existed in the case teams regarding development or ways of how things should be done, no guidelines specifically regarding sharing and documenting knowledge were discovered.

> Yeah, absolutely! It would be good because when you know some guidelines on how you should do it, it will always ease up the action that you will do it. […] Yeah, I would think so that it would be good to remind everyone that why to document and how to actually do it. At least maybe not the correct way, but the effective way. (junior)

Each team had a different environment and style of work; therefore, company-level guidelines could not be detailed enough, or they could be too restricting. An agreement created within the team could provide specific useful guidelines that would take team specifics into account.

> I think if we just find the time for the session and some explanation and we put some agreements to [KMS], let´s say, I think people should follow it. We are intelligent people, so I think people know that it´s just good to have something like that. (senior)

Distributed teams

Two out of three participating teams had team members distributed across multiple locations and countries. Sharing knowledge in distributed software development teams is a challenging activity and it can be a factor that might cause people to be less active or totally avoid sharing knowledge with remote colleagues. Members of distributed teams were well aware that such arrangements hinder communication and knowledge sharing.

> So if you think about remote locations, so knowledge sharing happens only when you contact someone and you talk to him or send a message. That´s really not so many hours in a day that you actually chat with people. (senior)

Many interviewees were emphasizing the importance and high frequency of local-only communication (only at one location of a distributed team) as a means of sharing knowledge with colleagues. One junior participant described the important ways of sharing knowledge like coffee talks, asking questions to colleagues from the next table, etc. but he admitted that sharing knowledge between all the locations is more challenging. That is in agreement with how another

participant (senior) talked about constant knowledge sharing in his daily work while immediately adding that there are also team members in another country and that it is a bit different with them. He said that there is a lot of verbal communication and not everything gets written down, because it is easier and faster that way.

Some signs of a preference for sharing knowledge in one´s mother tongue were noticed as it can sometimes be more effective and efficient. Many interviewees were mentioning differences in personality regarding relation to and activity in knowledge sharing activities; however, none brought up noticing issues connected to cultural distance. On the contrary, one manager mentioned:

> There could have been more like obstacles from that perspective that people from different countries have different kind of practices or different ways of work. But in this case, I haven´t seen any obstacles. Just wanted to bring that up. (senior)

Overall, knowledge sharing in distributed teams by means of online communication brings many obstacles. As one participant pointed out, this should be recognized and lead to higher attention and activity in calls and messages.

Onboarding and temporary collocation

A team´s distribution can be a problem in the process of onboarding a new team member. In some cases, a suitable local mentor was not available for the new person, which made the mentoring more demanding and challenging. The reason might have been for instance that there was nobody experienced enough or nobody at all from the team at the location, where the new team member started. Three cases of problems with onboarding in a distant location with no or few mentors available were described by participants. At least partial solution introduced by participants would be a temporary collocation of the new team member with the main part of the team, where he or she could receive intensive introduction training and establish good personal relationships with remote colleagues.

> So if it´s here, 8 hours a day, next to you, then it´s really kind of much easier to discuss and share the knowledge since things might pop up to your mind and you might say to him like: "Yeah, by the way, this and this." But you don't necessarily write it to the chat that: "Hey, just remember this." (senior)

Several members of the team where a temporary collocation happened only after a long time agreed that meeting remote team colleagues face-to-face made communication, cooperation, and knowledge sharing much easier and more efficient. Additionally, it allowed a much better mapping of knowledge location (who knows what within the team). There was a mutual agreement on the benefits of temporary collocation between the new team members and the old ones.

> But in my opinion, the best way to start the knowledge sharing is to meet the people, kind of right away face-to-face. Then it´s so much easier to chat or send mails, or whatever. You know what kind of people are there. (senior)

4.5 Lack of attention, initiative

All the participants saw knowledge sharing in software development teams as very important; however, the question is if it was reflected and received enough attention during their regular work. Every participant was asked whether knowledge sharing should be an integral part of their job and if it is currently the case. It was suspected that people might focus only on the core activities as programming or finishing given tasks, and knowledge sharing could be seen only as a side activity to be done when there is not much else to do. There was a clear strong agreement that knowledge sharing should be considered an integral part of the job, which is generally a good sign. However, when asked whether it is really the case, almost everybody admitted that it is not.

> It should be, but I know it´s not. It should be! (senior)

> No. […] But it should be, yeah, definitely should be part of the job! (senior)

Two interviewees thought that possible issues are that it is not required, or at least not specified as a task, whose completion could be measurable. Others pointed to a lack of some concrete form or a way how to do it. Additionally, there seemed to be an imbalance between the activity of different people – some shared knowledge a lot, and some not really.

> […] it should be something that you can measure, so there should be a task or subtask for documenting the actual work […] (senior)

Attention

The overall impression from the interviews, observations, and unstructured discussions with employees was that knowledge sharing did not receive a lot of attention. It was not required in any way and it was not brought up during team meetings often. An English proverb "*Out of sight, out of mind*" seemed perfect for describing the situation. Knowledge sharing was easily forgotten while people focused on the so-called "normal work" (e.g. writing code, solving service requests, completing other tasks).

According to one manager, critical and often repeated issues usually received enough attention and were documented. However, there were cases when even that was not true. Even such an important process as onboarding of a new team colleague had some glitches in form of forgetting the need for some guidance and mentoring of the new person. After those mistakes were noticed a few months later, there was an agreement that nothing like that should happen and it left the knowledge holders and managers with bad feelings. The possible reasons were that everybody was busy, and nobody noticed the situation.

> I just realized that this happened and basically the worst case is that you are there alone without any mentoring and everyone is just doing work and you are just: "What am I supposed to do?" (manager)

However, at the time of the study, there was a case of onboarding that was pretty unique. A lot of attention was given to the new team member and all colleagues were involved in intensively mentoring him. The high quality of this process might have been partially driven by learning from past mistakes or the personal proactivity of a person who took the role of coordinating this onboarding.

A certain suspicion raised that good ideas, efforts, or initiatives in the area of knowledge sharing died because people focused on different things and they did not receive that much attention even though they would be greatly welcomed and beneficial. As one participant explained, knowledge sharing requires putting some attention to it, but not only on the knowledge sharing itself but also on its planning and management.

One manager pointed out that this research itself might have a positive influence on knowledge sharing when, after the interview, he stated that it was pretty useful and nice to think and talk about these things because normally people just do their work and do not have time for this.

Initiative individuals

Few practitioners stated that the approach to knowledge sharing is highly individual and depends on one´s personality. Some people are very good and active and some not really. Several cases of very initiative people were observed in the participating teams in recent history. These individuals often came with some good ideas in the area of storing or sharing knowledge that had the potential to significantly help the team in their daily work. Examples of such ideas could be a good onboarding process, adopting knowledge sharing as part of some working process, or introducing new knowledge-sharing practices. They acted without any external encouragement, purely because they saw a problem, came up with an idea and were willing to take care of its implementation. The rest of the team often welcomed it but was not eager to get extensively involved as they considered it to be the author´s mission to lead.

In some teams, it seemed that these initiative individuals represented the only driving force in the process improvement efforts. The question is whether it is good to rely purely on individuals who are proactive in knowledge sharing because the quality can then vary significantly depending on if such an individual is present within the team or not. Moreover, these effects usually lasted only during the person´s presence in the team.

In the IT field, people might often migrate between different teams, projects, or companies. Naturally, this also applies to the aforementioned initiative workers and the recent history had several examples when such a person left the team. Despite the team members seeing the benefits of efforts he/she initiated and ran, nobody usually stepped up to ensure their continuation. Good ideas and efforts died. What was left in the team was only the shared praising of their former colleague for these ideas and efforts.

> […] and it was kind of his mission to complete. And then when he left, it was there. Nothing happened after that basically with that task. (junior)

Seniors and Juniors

Evidence started to form that there was a contrast between how the initiation of knowledge sharing is seen by senior and junior-level employees. The more experienced employees, seniors, are often in the role of knowledge holder. Less experienced people, juniors (and/or middle-experienced) are often in the role of knowledge recipients. The approach to knowledge sharing is, of course, highly individual, so the following generalization might not be always correct; however, it might point out an often-overlooked issue.

Overall, it seems that seniors were more likely to expect receiving questions and sharing knowledge primarily upon request. Juniors, on the other hand, expected seniors to be proactive in sharing their knowledge. They saw them as being busy and, therefore, might feel bad about regularly disturbing them with questions or might be afraid of asking stupid questions. Here, we can see a missing intersection that might significantly reduce knowledge sharing between knowledge holders and their less knowledgeable and experienced colleagues.

Senior employees or experts also seemed to hold slightly more positive opinions about the current situation regarding knowledge sharing practices and the situation in their team. Sometimes it was in contrast with more critical opinions of their junior team colleagues. A possible explanation for this might be that issues resulting from poor knowledge sharing are more likely to be felt by less experienced and knowledgeable workers than their expert teammates.

One senior participant stated that he does not see the possible discomfort of juniors in asking questions and possibly bothering people as a problem. However, this discomfort and unfulfilled expectations of knowledge holders actively sharing their knowledge were actually present in the same team according to a junior team colleague. This suggests that the issue might be unnoticed by team colleagues even for a longer time.

Despite it being a common practice in the company and the participating teams, there were two cases in the past few years when a new junior team member did not have an appointed mentor. One of them described it as a rough start. He considered the project to be in a hurry and all the team colleagues very busy, so he felt that he cannot disturb them much and primarily tried to work on his own, which went slowly.

> Well, basically you could say that I was thrown into the deep end of the pool when I came here the first time. Because the thing is that the project and the few developers which we had here were really really busy. They didn´t have any time to instruct me, ways of working, or anything like that, so it was quite rough during the start. But yeah, little by little you learn something and it goes forward. There were no real instructions or mentoring. (junior)

Interestingly, in both cases of juniors without a mentor, it seemed that their team colleagues would not mind helping them and would find at least some time for that. There might be two main reasons why they did not. Firstly, they expected that if the new colleague needs help, he will be asking questions or in any way pointing out that he has troubles and would need assistance. Secondly, they

expected that somebody is responsible for taking care of the new person, mentoring him, guiding him, etc. so they did not see a need to be initiative and get involved in that process. Unfortunately, they did not know that no such process was ongoing.

Based on the answers of multiple participants of different levels of experience, a possible picture of ideal cooperation between senior and junior colleagues emerged. It would combine activity on both sides in a series of interactions and it could reduce the threshold and show that not that much attention is needed after all. The interactions would come in a form of checkpoints during independent work. The senior colleague would assist at the beginning with picking a suitable task and consulting the planned solution. Then the junior employee would work independently while exercising the approach of learning by doing and discovering what and how works. Questions could be asked in the meantime as needed. The final solution would then be checked by the mentor and the feedback provided. This model would correspond to the shared responsibility that was suggested by one senior participant and almost identical to what another senior participant from a different team said.

> Well, the trainee or student who is coming needs to be active, needs to ask questions if he doesn't know. But also the tutor or mentor needs to be active […]. But the responsible one must be the mentor, but student or newcomer needs to be active also. (senior)

4.6 Lack of time, being busy

Time has been widely mentioned as one of the main limiting factors because practitioners often feel busy and they find knowledge sharing activities such as writing or updating documentation to be time-consuming. It seems to be a common problem that practitioners decide to postpone documenting knowledge for various reasons and they never come back to it. According to two participants, the difficulty of writing documentation increases as time passes from the task´s completion, and the same probably applies to the likelihood of actually never doing it.

> [Knowledge sharing] is not difficult in terms of difficulty, but it´s difficult to get the time to do the documentation after you have finished something […] Because we are always in a rush. Just have to take the time to make the notes for the next person who would do the same stuff. (senior)

Delays in getting answers to questions seemed to be a common situation. The main reason was considered to be people being busy at the moment when a question was asked. If we take into consideration that asking questions is the most used practice of knowledge sharing in the participating teams, some problems can arise.

Being busy also seemed to be limiting creativity and coming up with new ideas on how to make knowledge sharing work smoothly and efficiently. People

focused on their normal work and did not have time to think about the big picture.

> But we are all busy guys so not so many ideas or development ideas come. (senior)

Style of work and time demands are different between development and support work. In development projects, there might be busy times before deadlines. In support teams, keeping production environments up and running smoothly is the top priority. It is understandable that in such peak times of time demands, knowledge sharing priority was low. As one participant from the support team pointed out, the workload was very dynamic for them and there might be very busy times followed by rather silent periods. Therefore, some process should be in place to ensure sharing knowledge after the busy time is over. An example could be to write a "To document later" task list, which would prevent forgetting what was not possible to document because there was no time.

> Of course, we are working in an operational level on production environments, so if there is panic in some live environments we don't document then. Just fix the problems and we need to update the information later on then. (senior)

Multiple participants found it very useful to learn from highly skilled experts that are available in the teams or the company and who possess a lot of valuable knowledge. Unfortunately, it seemed that these experts were so occupied with regular work, that they did not have time to share their valuable knowledge within their teams or the company. One of the participants who saw it as a problem suggested that this topic should be discussed with the HR department or the higher management.

> Maybe internally there aren't that many people available to do trainings. The smart people are always so busy. That´s a big problem I think in this company. So those people definitely should have time to do the sessions and share the knowledge and not just do the daily coding or whatever they are doing. (senior)

Experts might be caught in a vicious circle. They do not have time to share their valuable and unique knowledge, because they need to focus on tasks that perhaps only they can do. But because they do not share their knowledge, they will always stay the only ones who are capable to handle those tasks and, as time goes by, they might get overloaded with such work.

> They are occupied with some tasks which other team members cannot resolve because of a lack of knowledge for example. (junior)

On multiple occasions, the rate and style of how participants were blaming lack of time or being busy as an obstacle for knowledge sharing raised a certain suspicion that in some cases it might be used as an excuse or a "classic reason". Some formulations gave the impression that time might be the factor one can always blame even though the real reason might be different. This should definitely not

be interpreted as suspecting the participants to lie, but as something to be recognized while attempting to identify the real discouraging factors in other environments or studies.

4.7 Discomfort and fear

Some thoughts regarding discomfort and fear connected to knowledge sharing activities were brought up. Participants generally agreed that the most efficient method of sharing knowledge is face-to-face. Human interactions, however, might make some people feel uncomfortable or afraid.

When one junior-level participant was starting in a new team, he expected that his more experienced colleagues would at least sometimes call him to come to observe how they are working on something important or interesting. This has not been the case and he suspected discomfort to be the reason.

> For me, it´s uncomfortable to go to ask, and for him, it´s uncomfortable randomly to show me what he is doing, because nobody likes it when somebody is staring over his shoulder. So, it led kind of nowhere. (junior)

A possible solution suggested to this problem was that the two people would set a deal and give a timeframe for the shadowing. The mentor would show what he wanted to show and would make it clear to the student that "the public part is over". The student would then leave back to his desk so the mentor could read his e-mails and chat messages without anybody looking over his shoulder.

Another sign of discomfort was noticed when one highly experienced participant was asked if he would be willing to teach other people, and he hesitated. He was fine with teaching one person or giving some guidelines; however, he seemed uncomfortable about doing training for a group of people. He had no previous experience with that, so he had doubts about his potential performance.

Human interaction might be more difficult for some than for others. According to one senior practitioner, personality aspects are the biggest difficulties and obstacles for knowledge sharing.

> I would say the biggest issue is the personality and being shy, let´s say, or afraid of failing or saying something badly that no one would like it. (senior)

He also recognized the possible fear or unwillingness of junior colleagues of asking questions that could be viewed as too basic or stupid. A different senior participant stated that he sees the mentor as responsible for reducing or eliminating the discomfort and fear from asking questions.

> The discomfort needs to be removed somehow and that´s the responsibility of the mentor. He is saying enough times that you can always come to ask or ask the guys around. So there shouldn´t be the discomfort for interrupting the job or something. There needs to be time allocated so that there is time for questions. (senior)

One of the suspected reasons why people would not share their knowledge is that they might think that it is not worth to be shared, underestimating the value of what they know. Some knowledge can seem like basic knowledge, which probably everybody possesses. In other cases, knowledge holders might think that the knowledge would be of no use to their colleagues.

> I think they think what they want to say is not enough deep or enough wide or… Like it´s not enough for being shared. (junior)

Especially in one of the teams, a highly common practice of maintaining private knowledge repositories was discovered. Practitioners were writing very simple notes sometimes with as simple tools as a TXT file and a basic text editor. Contained knowledge served for assisting the author if a similar task/issue would come up in the future, or for assisting other team members if they ask. The interesting aspect is that if these notes and hints would be public for teammates, it might save the time needed for asking, answering, and generally for solving the issues and obstacles. So why people keep them private?

One discovered explanation might be that if nobody can see the notes, nobody can find any mistakes there. Also, there are no requirements for the form – they can be written in any form, any language, and any structure. Writing private notes is therefore seen as a safe and easy/fast option. The solution to this problem could be lowering the expected difficulty and fear from storing the knowledge publicly via providing guidelines and making team agreements.

4.8 Nature of knowledge sharing activities

Based on what all the participants described, it seems that the two most common practices of sharing knowledge were *Questions & Answers* and *Documentation*. It was recognized by some that writing more knowledge to the knowledge management system (KMS) would reduce the number of often similar questions being asked from them and disturbing their work. Unfortunately, there seems to be a significant factor that often leads people to avoid sharing their knowledge via the KMS. Writing knowledge down is considered time demanding and, most importantly, boring.

> It takes extra time and, well, it´s more fun just to do real things than to document the old ones. (junior)

> Some people, generally IT people like to solve problems, not to, you know, be a redactor. (junior)

It was obvious that the participating teams strongly associated knowledge sharing with documentation (knowledge in KMS), hence, it was viewed as very important. However, the perceived high importance of documentation might not have practical effects due to people´s avoidance to contribute there.

Two participants (senior and junior) mentioned that they see most of the knowledge sharing is at the theoretical level, which might not be that useful. They put emphasis on practical learning by doing, with some form of assistance available if needed. This might also be something that should be taken into account while designing solutions for knowledge sharing.

> I think that the worst part of knowledge sharing is that most of the knowledge is theoretical. So sometimes someone tells me: "OK if there will be this kind of issue, you can do this and this." But it´s only theoretical! It´s great when I can see how it looks in practice and do on my own with someone next to me or something like that. So, I think that the practical way is best for me. (junior)

4.9 Knowledge to be shared

Some knowledge is shared during daily work (asking questions, writing documentation), while other practices such as onboarding and mentoring, informal expert sessions, more formal trainings, etc. usually require reserving time and some planning. Scheduled and planned practices were used in the case environments earlier, but their frequency dropped. One of the issues was the lack of interesting and important topics to discuss there. At first, scheduled time windows were replaced by organizing sessions only when there was something to cover. Possible topics were collected in the KMS to identify interesting and important topics; however, it seemed that switching to an on-demand model led to the sessions quickly disappearing.

One interviewee suggested that topics for such sessions do not necessarily have to be novel and it is beneficial to also refresh and sort existing knowledge.

> [E]ven if I know something, I like to hear the beginning to the end. It´s making this more ordered in your head. And to refresh your knowledge. (junior)

In the times when a highly skilled expert was leaving the team, the knowledge transfer was usually done in a rush. After the "*I am leaving*" announcement, there might be as short as 2-3 weeks to maximize the amount of knowledge that will stay in the team, so the time should be used well. One of the teams was facing such a situation at the time of the data collection. Initially, the knowledge to be transferred was not clearly identified and the chosen method of knowledge transfer was just shadowing the current tasks. Without identifying and prioritizing, important knowledge could have been lost. However, it was about to change soon, and the new idea was to identify the crucial knowledge of the leaving expert, prioritize what is important, and then conduct the knowledge transfer accordingly.

4.10 Support

It is obvious that knowledge-sharing activities consume time. And as a known aphorism says, time is money. Therefore, it is a question of whether organizations allow their employees to spend time on knowledge sharing activities because those might not produce immediate results.

Internal support
The overall impression from the collected data is mostly positive. Most of the participants perceived that if there were good ideas on how to share and store knowledge, they usually received support from management, and people were allowed to spend some time on executing them. However, some doubts or obstacles were also described.

Participants were able to describe cases when certain ideas and efforts were highly supported by the management. These included for example allocated time for mentors for onboarding of new colleagues and various types and forms of informal team knowledge sharing sessions. One participant claimed that currently probably all good ideas were welcomed and supported because there were not so many knowledge-sharing initiatives overall. Described small obstacles were usually logical such as a need to present some time estimate or being allowed to carry out the idea only with some delay due to the current workload and deadlines in the project.

Regarding knowledge sharing activities that are included in the regular work, a senior worker considered it helpful when it is clearly communicated that knowledge sharing is supported and it is allowed to spend time on it. One manager expressed a concern that maybe his team members currently do not know very well that they can and should invest some of their time also to that kind of activities and he thought that the instructions should be clarified.

Customer support
Even if knowledge sharing activities receive support internally, customers might have different opinions and priorities. Who should cover the costs of knowledge sharing is an additional question.

Some customers seemed to be solely focused on performance and delivering results. Maybe they did not find knowledge sharing as important or they simply did not pay much attention to it. Even when a good idea was formed and presented, the reaction might have been that they think it is not a good idea, or that it is a good idea, but they are not willing to pay for its implementation. In arrangements, where the customer had a strong role in managing the team, incorporating knowledge sharing into regular work was sometimes more difficult or even blocked.

On the other hand, some customers were found to have a supportive attitude towards knowledge sharing efforts. Factors that were believed to have a strong influence on this were good long-term relations and trust that what is suggested is a good idea that will bring benefits. A thought was presented that

customers who are interested in or adopted an agile way of working might more likely support knowledge sharing as well, because of seeing benefits also in a work that does not produce immediate direct results.

4.11 Encouragement

Lack of attention was previously described as one of the major issues. A possible solution could be encouraging and reminding people to remember sharing their knowledge during their regular work. Is there enough encouragement or reminders? Does it have a correct and effective form?

There were only little signs of effective encouragement that would prompt employees to pay attention to knowledge sharing. It seemed that there was a high level of reliance that people would perform knowledge sharing activities themselves and that they would be initiative in coming up with new ideas on how to share and maintain team knowledge. As two participants suggested, this might be related to the company´s working culture, where employees are, to a large extent, trusted to do what they see best instead of ordering them what and how exactly to do.

> I think it´s not kind of our way of working at least at the moment, so there is no one saying that: "Hey, let´s do that, let´s do this!" (senior)

Some encouragement has been coming from initiative individuals who cared about sharing knowledge and tried to promote its benefits to their colleagues. This leading by example seemed to slightly increase the knowledge sharing efforts of their colleagues on the operational level. However, it did not lead to anybody following and becoming highly initiative as well. In the cases when the initiative individual left the team, the effect mostly disappeared.

Lack of encouragement might be the reason behind some earlier expressed doubts about whether it is even desirable for employees to spend time by sharing their knowledge and writing documentation, therefore, possibly reducing the perceived support for such activities. Two managers agreed that there was no encouragement at that moment, but that it would for sure help.

> But I think that at least currently the situation is that team basically does not even know that we should and we can make the documentation better. (manager)

One of the participants stated that even simply reminding more often to think about sharing knowledge and writing documentation would be helpful. According to two interviewees, the encouragement should be local, coming from within the team. Company-level encouragement might not be effective, because the needs and processes of teams are very different.

> I believe the team level is the way could work because the needs are so different. So, if that is pushed from the top, that wouldn´t work for everybody. (senior)

Encouraging knowledge sharing activity should take into consideration the personality aspect. Some methods or practices might suit one person, but not another. Encouraging something that a person does not feel well about might have zero or even negative effects.

> It depends on the people I think because for example, I don´t like to make a presentation where I talk. I would prefer to write down some article for example. (junior)

Unspecific targeting

When searching for somebody who would handle some knowledge sharing task or become active in a new practice, the worst approach seemed to be asking: *"Could somebody do it?"*. While this approach worked mostly very well with normal tasks, where team members often volunteered, it did not seem to be the case in the knowledge sharing domain. According to multiple participants, the unspecific targeting (*"somebody"*) made them feel that there are other people who could do it, so they did not feel any urge to volunteer. One participant who was emphasizing this issue, which was clearly present in his team, was asked what would work for him to receive the message better. In his opinion, the encouragement or request should be targeted personally. Instead of asking *"Could somebody present us something interesting next week?"*, the question should be: *"Could John present us something interesting next week?"* The person could still say no; however, he/she would be forced to consider it at least briefly. This would give the feeling that the question really concerns the targeted person, hence it would not be so easy to ignore anymore.

> You know, it´s a bit psychological that if you talk to the team that "maybe someone could", no one wants. But if you have some schedule, or some coordinator or manager points at you: "Maybe you can?" You feel you should maybe do this and just do the presentation. Or just say: "I don´t have any interesting issue" and maybe someone else. But maybe it would be good to point on personally that it´s your turn, "Do you have something?" (junior)

One manager described that he had a good experience with getting the whole team involved. It is important to explain the situation and needs and then let the team try to design a suitable solution.

Abstract form

Participants were talking rather critically about a *"general management talk"* referring to statements about processes, values, what should be done, etc. that stay only at the abstract level. It seemed that these speeches might be sometimes basically ignored by practitioners. They might have a very good relationship with the manager and agree with what he/she says; however, they might consider those statements obvious when they are just abstract, or they refer to facts that are generally accepted by the majority of the IT community. It might result in a reaction such as: "Yeah, … sure!" and be very easily forgotten.

The abstractness of knowledge sharing tasks or activities might be a significant impediment. The regular work of IT professionals often consists of concrete

tasks that they have to complete. Request for paying attention also to sharing and documenting their knowledge might not be received very strongly if it stays only at the abstract level.

> Because, it is always like this that if you just say or mention that: "Please do the documentation", it does not affect anyhow. (junior)

A possible solution could be to transform the concept of knowledge sharing into a concrete form that practitioners are familiar with and used to work with – tasks, sub-tasks, goals, etc.

> Yeah, I think in general if it would be a task that you have to finish, then it would be kind of easier to justify at least for yourself, at least for me. […] But if someone says in general that: "Hey, remember to share the knowledge." Then not, not so easy. (senior)

> If it´s something that you can measure and finish and do. Then it´s easier to kind of keep in mind and finish it. (senior)

Compulsoriness

As was mentioned earlier, the case company working culture usually did not use strict orders and rules. Multiple participants mentioned *"not being mandatory"* as a factor that made it easier for people to avoid writing documentation or sharing knowledge, so they brought up the option of making knowledge sharing somewhat compulsory. One junior participant connected this factor to the idea of knowledge sharing being considered boring and believed that people will avoid it if it is not mandatory.

> They are not required to do it. I think that´s the main reason. It takes extra time and you might, well, it´s more fun just to do real things than to document the old ones. It´s that simple. And if it´s not required, you probably won´t do it. (junior)

This participant would like to see writing documentation to be at least partially required. One outlined solution would be to make the contribution to KMS a mandatory part of each task, or at least tasks that take over X hours to complete.

Two managers held a similar opinion. One believed that people do not want to contribute to the documentation if they are not told that it is mandatory, because it is the boring part of the work. The second manager also stated the opinion that people do not want to do it. His solution proposal was to discuss their *Definition of done* agreement and agree on what is the correct level of documentation. Defining *"good enough"* might be a good starting point and the team agreement would make it everybody´s responsibility to check the documentation and update it.

4.12 Missing systematic approach

The data suggest that no significant systematic approach to knowledge sharing or knowledge management was present. No advanced process or model was identified on the team or the organization level. Efforts of managing knowledge appeared to be mostly ad-hoc and the level of quality significantly varied between different teams and big differences were noticed also within a single team throughout the time.

> [T]here is no specific form or method or, no clear way how to do it; therefore, it´s not happening. Or it´s limited or it´s not systematic. It´s based on an individual´s approach. (junior)

According to one manager, some uncertainty might have been present among team members regarding how and where exactly a certain type of knowledge should be stored. Responsibilities over knowledge sharing activities might also have not been clear to everybody.

Regarding possible options on how to improve the current situation, the *Definition of done* artifact was mentioned by one manager as something that should be revisited and updated after mutual agreement between the team members. He believed that it would help to clarify what are the requirements regarding writing and updating documentation and that it is everybody´s responsibility. The correct required level should also be discussed and agreed upon among colleagues. Having such an agreement included in the *Definition of done* artifact could help to maintain a similar level of quality even when the project will be going through busy times.

A common process or model on the organization level did not exist, and some thoughts suggested that, given the significant differences between teams in the company, it would not be useful anyway. Compared to development teams, the focus of the technical support was primarily on quickly responding to incoming issues in production environments. Their ability to plan and schedule was limited. Some teams were co-located and others were distributed between several physical locations. In some, all team members worked in the same domain, while a more strict division of domains was present in others. The level of customer involvement also varied considerably. All of these and even more differences between teams suggested that one solution on the company level could not fit all teams.

The lack of a systematic approach was noticeable and can be demonstrated also in the process of onboarding, i.e. sharing knowledge with a new colleague. Moreover, the company representatives suggested a need and desire to improve in this area already during this study´s negotiation phase. Collected data identified major differences in onboarding experiences especially due to the lack of organized or systematic efforts.

The onboarding process was always carried out ad-hoc with an appointed mentor, a more experienced team member, having the main role in planning and

conducting the onboarding process as he/she saw best. It was visible that the quality of onboarding can be largely traced to the mentor´s performance and suitability for the role, both in a good and a bad direction. There was evidence that one of the main criteria in selecting the mentor was time availability. On several occasions, it was not recognized that some people, while being friendly skilled professionals, are not capable of or very good at teaching other people. In two cases, no mentor was assigned at all, because it was simply forgotten, and it resulted in the new colleagues going through difficult times.

A very common issue in the eyes of junior employees was the passivity of the mentor, which did not meet their expectations and caused the mentoring to not be as effective and efficient as it could have been. Interestingly, among the mentors mentioned in the study, the younger, less experienced ones, seemed to be more active and successful in cases of mentoring a new junior colleague. This was very likely connected to the different approaches and expectations between senior and junior professionals that were already described in chapter 4.5 about attention and initiative. Additionally, more experienced team members tended to be busier, which might have contributed to the reported passivity in the eyes of the mentored junior individuals.

Little evidence was found regarding learning from past onboarding experiences and reusing practices and artifacts from those. Previous own experience of a mentor was probably the only more significant example.

Especially in the case of junior software developers, there was a phenomenon that could be called *"jack of all trades, master of none"*. There was often no plan and the main emphasis was put on learning by doing, which was not organized into any logically connected path. The result was learning only single pieces of a puzzle from different areas of a very complex product. One option to improve this would be to have a plan for getting familiar with the product in a more organized way – start with area A for a week, learn about it, do few tasks there, and then move to area B. One participant considered this very beneficial:

> Well, you would get a much bigger picture of the whole project like that if you would have some kind of structured way that would go basically through [the whole product]. That would be much better than going a little piece from [A] and a little piece from [B] […] if there would be like some kind of rule, like during the first month you would learn things related to this and this, that would be a really good thing! (junior)

Some guidelines for mentors could be beneficial as well. One very skilled mentor thought that some mentors that he knows would welcome having guidelines or goals, while he also emphasized that there should not be strict rules that would restrict the creative and active mentors.

One onboarding was ongoing during the data collection phase and it seemed very different from other cases in the company and the way it was conducted seemed to bring very good results. The incoming new team member ("a student"), was not assigned one single mentor but instead, multiple team colleagues shared that responsibility. Each had a certain set of topics to teach and it was handled as almost a full-time mentoring within one week, after which it was

the next mentor´s turn. The whole process was coordinated, and topics were carefully planned so that the student learns the most important things in a suitable order. At the end of some of the one-week sessions, the student had to present his understanding and answer questions from colleagues. The whole process was considered very good and much better than anything before in terms of both the quality and the speed. One of the benefits of having multiple mentors was gaining a better understanding due to a certain overlap of topics and receiving different viewpoints on those. Moreover, it seemed to help to quickly integrate the new member into the team.

4.13 Tools

A wide variety of tools can be used to support sharing knowledge between colleagues and knowledge management in general. The case teams shared some knowledge through standard communication tools such as instant messaging (IM), e-mail, mailing lists, and audio/video online conferencing tools. However, the primary tool identified in the case teams was a knowledge management system (KMS), which was used for storing codified knowledge and information. Practitioners in participating teams used the term "documentation" for codified knowledge in their KMS and the KMS itself, hence the same term is also used here. The rest of this section focuses on problems connected to KMS.

Emphasis on documentation
All the participants seemed to have a very strong association between the concepts of *knowledge sharing* and *documentation*. When asking participants about knowledge sharing in their team, they were often using the term *documentation* as it would be a synonym. Observations and short discussions suggest that this might be the case in the whole company. Alongside answering questions, documentation appeared to be the primary conscious way of sharing knowledge.

The quality of the documentation did not seem to correspond to the given strong emphasis. This is an interesting contrast that might have important consequences. As one manager admitted, the quality of documentation was not on a very good level in the company in general, probably due to people´s attitude:

> And the level in this company of that documentation, it´s not that good. But mainly I think it´s because of the people, they don´t want to write that documentation. (senior)

> But anyway, we have a good set of tools but not using those that well. […] So from a technical perspective we don´t have any obstacles, so we could do it better. (senior)

One participant pointed out that documentation is an important method in his team because of being available for all the colleagues across multiple physical locations; however, at the same time he also admitted that documentation does

not receive enough attention because it is not required, it takes extra time, and it is boring to contribute to it.

Structure and design

Participating teams were mainly using wiki-based knowledge management systems. Opinions about their structure and design slightly differed but interviewees were mostly satisfied with them. The most significant problem could be the size of the structure that made it too complex.

> But the thing is the document structure is so big at the moment and there are so many pages. (junior)

One presented issue was that documentation can be scattered across multiple locations and systems according to the convenience of single individuals. A suggested solution to how writing documentation could be made easier for the developer or specialist could be to provide the location for the future documentation already in the task´s description.

The main approach to finding knowledge was via the search function. When asked about what could be improved in their KMS, few participants also called for some index page (sitemap), which would summarize links to the most important or most frequently used knowledge that is available within the KMS.

> Maybe there could be something like a sitemap, something like that. With the main knowledge etc. that we don't need to only rely on searching but we have some most important topics on plate, let´s say. (senior)

A remotely similar practice was used in few recent onboarding cases when the newcomer to the team received a set of recommended pages in the KMS to familiarize with. Considering the often-extensive size of the KMS, this was highly appreciated by the new colleagues. However, this did not appear to be standard practice.

Customer involvement in the development process can have a different scale and in two cases this also included using the customer´s KMS. In one of the projects, the chosen KMS presented an issue that was noticeably limiting the effectiveness and efficiency of storing and retrieving knowledge. The main problem was that the KMS was document-based, using a structure of folders and files with no easy options of linking between pieces of knowledge in different files. The hypothesis was that it was chosen because it was used previously and customer´s business-oriented representatives were familiar with it. However, as a document-based platform, it was considered unsuitable for the technical professionals, who work with pieces of knowledge and not documents.

> So, I would say the document structure on the customer side is pretty horrible. It doesn´t work at all. The only way to really find something is through the search and even that is a bit messy. To put it short, it´s not good. (junior)

Outdated knowledge
Things change during the development process and parts of written documentation become outdated. If not addressed, it can result in at least two identified problems: (a) the KMS grows in size and becomes less organized as mentioned earlier; (b) outdated instructions and information can be misleading for less experienced team members.

> Hmm, for an experienced team member it´s not a problem because he knows what is wrong or outdated but for the new team member, it can be a big problem. (junior)

There is also a possible threat that it could negatively affect the trust in the KMS and the likelihood to contribute there. The situation in every team was that parts of the KMS were outdated and people generally got used to it. The timestamp of the page´s last modification was often checked as an indicator that directly influenced the level of trust in the page´s content.

> It depends on the article but in general, there are many articles that are not updated at all. There are some legacies there from the previous implementations. (junior)

No ideal solution on how to solve this issue completely has been found. An effort to check the whole KMS to remove outdated information seemed unrealistic due to high time demands and being a boring task. The only feasible option appeared to be fighting this issue during regular use of the KMS – when making modifications to the KMS, members of all teams were expected to check and eventually modify (update or remove) also surrounding information and instructions. Unfortunately, some people appeared not to be aware of it, not remembering it, or avoiding it.

A possible overlooked opportunity might be using new team members to improve the quality and structure of the KMS. Due to their learning needs, less experienced people might be accessing KMS more often. It was admitted that they might not be able to identify all incorrect or outdated information and might be afraid of making any changes. A suggested solution was marking information with a flag such as: “Possibly outdated”, which would serve as a warning for other colleagues and an experienced person could later find all these flags and make appropriate changes to the content (update or remove).

4.14 Summary of results

This chapter presented the collected data categorized based on factors that might discourage or prevent practitioners from sharing their expert knowledge with their team colleagues. The following chapter will discuss the findings and compare them with information from prior literature.

5 DISCUSSION

After the collected data were described, this chapter summarizes them, compares the findings with reviewed literature, and draws implications for theory and practice. The following sub-sections will attempt to answer the three research questions of this study:

1. What are the challenges and practices of knowledge sharing and knowledge management in software development teams?
2. What discourages or prevents the adoption of knowledge sharing and knowledge management practices in software development teams?
3. How could the quality and quantity of knowledge sharing be improved?

5.1 Challenges and practices

1. What are the challenges and practices of knowledge sharing and knowledge management in software development teams?

The role of this research question was primarily supportive to allow building awareness of current challenges and practices in software development teams as a preparation for the data collection. The question is considered answered by the conducted literature review in chapters 2.4 and 2.5. This section will only briefly summarize practices identified in the literature review and whether, and to what extent, they were noticed in the case environment. It is important to emphasize that identifying all used practices of sharing knowledge was not the primary goal of the data collection, hence, the purpose of Table 6 is just to give a reader a rough overview of the case environment.

Table 6: Knowledge sharing practices

Category	**Literature review**	**Seen in the case environment?**
KMS	KMS for keeping knowledge	Yes, very important
	Attention to design and maintenance (up to date)	To some extent
Knowledge transfer	Expert info sessions	Earlier yes, not currently
	Formal training	Yes
	Team meetings (ideas, advice, problem-solving)	Yes
	Learning by doing	Yes
	Code reviews	Not identified
Social	Supporting social ties (team meetings, liaison)	Not identified
	Temporary collocation	Sometimes
Organizational	Environment encouragement	Yes, but ineffective
	Processes, guidance, routines, KM model	Nothing advanced, mostly ad-hoc and depending on individual activity
	Motivation by incentives	No
	Knowledge index	No
	ESMPs for knowledge sharing	No
	Global online communities	No
Onboarding	Adjusting KMS for newbies	To some extent
	Mentoring	Yes, quality varied
	Working in pairs	No

5.2 Discouraging factors

2. What discourages or prevents the adoption of knowledge sharing and knowledge management practices in software development teams?

The second research question aimed at understanding why members of software development teams do not share their knowledge. Is it that they want to keep their valuable knowledge to themselves? Perhaps they do not consider sharing it important? This section discusses the discouraging factors that were identified in the collected data and that are outlined in Table 7. The factors are introduced in two categories according to whether or not they appeared to discourage or prevent knowledge sharing in the studied environment. RQ3 will also be partially answered in this section to introduce some solutions suggested by the study´s participants together with the identified issues.

Table 7: Overview of factors discouraging knowledge sharing behavior

Factor	**Practical examples of contained issues**
Not confirmed or not significant factors	
Importance	Not considering knowledge sharing important
Motivation and benefits	Not knowing why one should share own knowledge
Willingness	Desire to keep benefits from the possession of unique knowledge
Confirmed and significant factors	
Perceived difficulty	Perceived high quality requirements, difficulty to codify tacit knowledge, missing guidance, distributed teams
Lack of attention	Seen as an extra activity, not included in normal work, different expectations between highly and less-skilled workers
Lack of time	Not having time for knowledge sharing, experts busy to share their valuable knowledge
Discomfort or Fear	Avoiding disturbing colleagues, fear of asking stupid questions, underestimating the value of possessed knowledge, keeping private knowledge repositories
Nature of the task	Seen as boring, being skipped
Knowledge to be shared	Not gathering topics, impression that there is no knowledge to be shared
Support, Allowance	Support maybe not very clearly communicated, varying customer support
Encouragement	Unspecific targeting, abstract phrases, not being required
Missing systematic approach	Lack of process or model, only ad-hoc efforts with fluctuating quality
Appropriate tools	High emphasis on documentation, outdated knowledge in KMS

5.2.1 **Not confirmed or not significant factors**

Some factors emerged from the data or the reviewed literature as potentially discouraging practitioners from sharing their knowledge; however, they were not confirmed as significant in the case environment. It is believed that such factors still need to be reported because they might present considerable issues in a different setting. Additionally, it might be important for practitioners and researchers to point out that despite possible common beliefs, some factors might have none or little effect in some settings. Three factors with no or low effect of discouraging knowledge sharing behavior are *Importance*, *Motivation and benefits*, and *Willingness*.

The study discovered a high level of agreement that knowledge sharing is very important. Most of the participants, however, recognized that this importance is not sufficiently reflected in the regular work. This finding confirmed the premise that served as a motivation to conducting this research – despite being aware of the importance, many practitioners in software development teams

are not actively sharing their knowledge in a scale corresponding to their perception of the importance. The current finding is almost identical to what Aurum et al. (2008) concluded, with the difference that their claim focused on the passivity of organizations, not individuals. Overall, practitioners in the case environment recognized that knowledge sharing is important, which corresponds to how the importance of knowledge, knowledge sharing, and knowledge management has been reported and advocated by various researchers throughout the years (Aurum et al., 2008; Faraj & Sproull, 2000; Rus and Lindvall, 2002; Ryan & O'connor, 2009, 2013).

Considering knowledge sharing important does not seem to be just an empty phrase since the participants were able to explain specific benefits that it can bring to their team. These benefits can be linked to their motivation to share knowledge. The main motivating factors were helping one´s colleagues (to do the job efficiently, remove obstacles, and avoid frustration), helping the team (integrating knowledge, increasing team´s performance, empowering support specialists), and allowing delegation of work (substitutability and avoiding routine). All these drivers are intrinsic and there were no signs of anticipated extrinsic rewards as discussed by Bock et al. (2005) or Aurum et al. (2008). No extrinsic rewards at all might be considered a space for improvement; however, at the same time, some studies found extrinsic rewards not to have an as significant effect as the intrinsic drivers (Bock et al., 2005; Szulanski, 1996).

Interviews and secondary data showed that willingness to share knowledge was generally not considered to be an issue. Some people might consider the exclusive possession of important knowledge as an assurance that they cannot be replaced. In the light of globalization in IT and offshore arrangements, people could be afraid of losing their jobs because of cutting costs and replacing them with people from markets with lower labor costs (Ebert & De Neve, 2001; Herbsleb & Moitra, 2001). However, this did not seem to occur in the case company. One of the primary ways of sharing knowledge was asking questions and those were always answered, even though it might have been with some delay. An interesting idea was that people might not proactively share their knowledge because they enjoy being asked as it makes them feel valuable. Suggestions were raised that if knowledge sharing would be made too formal, formality resistance might come in and affect people´s willingness to share their knowledge. In some cases, valuable knowledge was kept only by single individuals. This was not due to the unwillingness to share it, but it corresponded to the knowledge storage strategy of the transactive memory system and domain experts as introduced by Wegner (1987). The issue of domain expert lock brought up by Desouza et al. (2006), regarding limiting one´s area of expertise and growth only to a certain domain, was present and it affected decisions of multiple domain experts to change a team.

In summary, practitioners demonstrated that they know why they would share their knowledge by recognizing its importance and being able to present practical benefits that it can bring, and no major issues were reported regarding team members being unwilling to share their unique knowledge.

5.2.2 **Confirmed factors**

This section introduces all the other factors that were found to have a significant influence on knowledge sharing in the participating teams.

Perceived difficulty
There were signs that the perceived difficulty of knowledge sharing was rather high. Especially in the case of KMS, practitioners seemed to have a high expectation about the level of required quality and extent. In many cases, this appeared to lead to writing only personal notes instead of storing knowledge publicly. The general impression was that while the extent of contributions to KMS was quite unsatisfactory, writing private notes was a common practice.

Lack of specific guidelines and simple methods of how to share knowledge could be a significant aspect preventing such behavior by increasing the perceived difficulty. Some calls for training or providing a set of simple recommendations were raised by participants. This can be connected to claims by Szulanski (1996) that the perceived difficulty of knowledge sharing increases if it cannot be routinely handled and requires ad-hoc solutions. Similarly to the study participants, he suggested that perceived difficulty can be reduced by introducing effective and efficient practices (Szulanski, 1996).

As known from existing literature, distributed team arrangements can have negative effects on communication and knowledge sharing (Alavi & Tiwana, 2002; Ebert & De Neve, 2001; Griffith et al., 2003; Ågerfalk et al., 2005). Preference for local-only communication was noticeable because, in line with the findings of Herbsleb et al. (2001), it is considered more efficient. That also corresponds to what is known about sharing tacit knowledge in IT organizations (Griffith et al., 2003; Ryan and O'connor, 2009). Remote onboarding and mentoring of junior employees were also recognized as very difficult.

Lack of attention, initiative
All the participants claimed that knowledge sharing should be part of their job alongside the core activities such as programming; however, many immediately added that it was not the case at that time. Knowledge sharing did not receive as much attention as it would deserve. Possible reasons are that knowledge sharing was not required or mandatory, it lacked some concrete form as a task whose completion could be measurable, and no way how to do it was specified. The earlier used proverb "Out of sight, out of mind" describes the situation well. People in the case teams seemed to focus purely on the core activities and they often did not pay attention to storing and sharing knowledge. This can result in not using the full potential of knowledge available in the team (Faraj & Sproull, 2000; Rus & Lindvall, 2002) and some good practices slowly disappearing. Initiative people were the main driving force in knowledge sharing improvements. Unfortunately, their influence was valid only during their presence in the team. Teams have been failing to accept those as their own and keep them for the future as an active part of the process or at least documented.

An important finding is a difference in how senior and junior employees saw knowledge sharing between peers. On one side, seniors expected to share their knowledge upon request and might have not shown that much own proactivity. On the other side, juniors might generally be afraid to ask questions because of fearing asking stupid questions or disturbing their more experienced and busy colleagues. Juniors expected the knowledge holders to share their knowledge proactively. Unfortunately, the expectations of these two groups did not meet, and this could be unnoticed even for a longer time, hinder the knowledge sharing and slow down the professional growth of juniors. Additionally, the data gave the impression that senior experts might hold a more positive image of the current situation than their less experienced colleagues, possibly because juniors are more likely to feel issues present in knowledge sharing much stronger. When trying to identify existing issues, it might be beneficial to pay close attention to the point of view of junior team members and the struggles that they face.

Lack of time, being busy

Lack of time was often blamed as the main problem and limiting factor. However, skipping storing knowledge in documentation combined with delayed answers to colleagues´ questions might result in a lot of wasted time as well. If writing documentation is skipped, one might need to answer questions later anyway, so there might be no saved time after all. Moreover, the delay in providing an answer might cause a colleague to spend too much time on a task or block him/her from working and force to switch to another task. Temporal distance in distributed cooperation arrangements might further increase these answering delays (Carmel & Agarwal, 2001; Ågerfalk et al., 2005).

The expertise of the company´s internal experts was considered very valuable; however, these experts did not have enough time to share their knowledge because they were occupied by normal work. Sometimes they might have been the only ones capable of doing some work, which kept them busy, but that might never change if they do not share their unique knowledge. Furthermore, it happened multiple times that such experts decided to leave the team or the company, and a large portion of their unique and important knowledge left with them. Issues with disappearing knowledge are also known in the existing literature (Davenport & Prusak, 1998; Rus & Lindvall, 2002; Taweel et al., 2009).

A certain suspicion raised from the data that the lack of time could at least partially be possibly used as an excuse, as the reason one can always blame without it being questioned. While identifying the obstacles for knowledge sharing, practitioners and researchers should maybe be encouraged to go beyond the classic “lack of time” reason. Additionally, management might want to clarify what is the real situation regarding knowledge sharing being supported and whether it is allowed or even encouraged to invest time in it.

Discomfort, fear

Discomfort can play an important role in pair work, asking questions, and training others. Generally, people react and feel differently about social interactions;

however, those have been described as the best way to share tacit knowledge (Bock et al., 2005; Griffith et al., 2003; Ryan & O'Connor, 2009). Even senior employees who have a lot of valuable knowledge and are willing to share it might find some methods of teaching uncomfortable or unsuitable. This suggests that while planning knowledge sharing activities, it is important to listen to people´s feelings and preferences.

Personality aspects like being shy or being afraid of asking stupid questions might pose a significant problem if asking questions is one of the primary knowledge-sharing channels. In the case environment, this was seen as a possible problem mainly before a new junior employee became integrated into the arrangements of the transactive memory system (TMS) as defined by Wegner (1987). From the other side, it was brought up that being proactive in sharing one´s knowledge might be hindered by underestimating the value of the knowledge in question. This could be linked to the motivational drivers *sense of self-worth* described by Bock et al., (2005) and *self-efficacy in knowledge-sharing* in the research by Rode (2016), which both focus on the belief that one´s knowledge is valuable.

One coping strategy against discomfort in sharing knowledge was writing private notes. It was considered to be a safe and easy option as nobody can see and judge them and there are no quality or format requirements. This corresponds to what Rode (2016) pointed out about people being afraid of sharing knowledge in the transparent environment of enterprise social media platforms (ESMPs). Unfortunately, this might significantly reduce the amount of publicly available explicit knowledge in the team´s KMS.

Nature of the activities

The nature of knowledge sharing activities might be a strong discouraging factor on its own. Sharing knowledge, especially via the documentation, was considered boring. Unless specifically required by some rules or team agreement, practitioners preferred to start implementing or investigating a new task than documenting the old one. In addition, there appeared to be a high emphasis on practical learning, which was considered to be much more effective than sharing knowledge at a theoretical level ("If this or this happens, do this..."). These two aspects of knowledge sharing are something that should be recognized and taken into consideration when designing knowledge sharing models, processes, and practices.

Knowledge to be shared

One way how to share knowledge in the teams is through informal sessions. Those were identified both in the case environment and prior literature (Aurum et al., 2008; Dorairaj et al., 2012). One factor limiting sharing knowledge via these sessions can be the impression that there are no interesting or novel topics to discuss or share. While it might be true, there is a serious risk that topics exist, but are just not being identified and collected. There is a certain possibility of an influence of the "*out of sign, out of mind*" factor when knowledge sharing sessions are not scheduled regularly. To fight it, collecting new topics could be reminded

for instance on a regular team meeting. A structured approach should especially be used in cases when a knowledgeable colleague is leaving the team, because as Rus and Lindvall (2002) and Taweel et al. (2009) pointed out, leaving experts often create a knowledge gap. The disappearing knowledge should be precisely identified and prioritized to ensure the most important knowledge is transferred in a limited time.

Support

Internally, the support for knowledge sharing activities and ideas appeared to be on a high level. Good efforts were usually supported, and management allowed dedicating some time to those in a reasonable timeframe. This is in line with the view that knowledge, skills, and expertise are one of the most valuable resources that software development organizations have (Rus & Lindvall, 2002; Ryan & O'connor, 2009) and that to leverage its full potential, it must be coordinated (Faraj & Sproull, 2000; Ryan & O'connor, 2009). However, there were some signals that the support might not have been communicated well and not being reminded.

The more challenging area was receiving support from customers, who might be exclusively result-oriented. That sometimes hindered or even blocked more sophisticated practices of sharing knowledge within the team. From some customers´ point of view, knowledge sharing can be an activity that does not produce immediate and direct outcomes and might slow down delivering results and eventually also incur additional costs. Good relations and experience with given recommendations being beneficial were one way that seemed to increase customer´s support. The effects of customers on knowledge sharing in software development teams did not seem to be extensively covered in the reviewed literature.

Encouragement

The encouragement of knowledge sharing activity did not seem to be sufficient, did not have a good form, and was often not targeted well. Low emphasis on encouragement was probably related to the company´s working culture, where employees are given rather wide freedom and responsibility. Some encouragement was coming from initiative colleagues in the form of *leading by example*; however, this effect disappeared when such a leader left the team. There was a certain agreement that more encouragement of knowledge sharing would be beneficial as it would increase the attention paid to it, remind the desired activities, and ensure the employees that it is allowed and expected to spend some working time on it. Attention should be paid to the team´s needs, project´s specifics, and individuals´ preferences in order not to promote unsuitable practices.

The existing and occurring encouragement had two major identified issues – unspecific targeting and abstract form. Asking *"Could somebody…?"* or stating that *"someone should"* did not work well because it does not target any specific member of the team. A possible solution to this would be to ask directly a specific person, while still offering a chance to refuse. Additionally, encouragement by abstract statements such as *"Knowledge sharing is important!"* appeared to be

ineffective and should be replaced by turning the abstract concept of knowledge sharing into concrete and measurable form like for example tasks or sub-tasks.

Some calls were also made regarding making knowledge sharing compulsory at least in some form. Ideal, yet not very strict, forms could include internal team agreements or including it into the *Definition of done* artifact.

Missing systematic approach

No advanced knowledge sharing or knowledge management process or model was identified during the study and efforts were mostly nonsystematic, ad-hoc, and depending on individual activity. Similar observations can be found in prior literature (Aurum et al., 2008; Prikladnicki et al., 2003). This finding could be related to the reported uncertainty of how some types of knowledge and information should be handled and stored and unclear responsibilities. The Definition of done artifact formed by team agreement was brought up as something that could improve the situation. As such, it could be considered a good starting point of a more systematic approach. A tailored process or model on a local level (team, project) was advocated over a unified version on the organization level to allow customization based on local specifics. The need for project-specific models was also mentioned by Aurum et al. (2008).

Onboarding is a crucial process in sharing knowledge with a new team member and the lack of a systematic approach was seen there. Similarly to Bjørnson and Dingsøyr (2005), mentoring by an experienced colleague was considered as very important in the case environment. Even though it was an established practice, its quality varied significantly case by case and there were several issues and imperfections that impacted its effectiveness. These included no or only poor planning and coordination, ad-hoc approach with no established process or guidelines, not reusing past experience and artifacts, insufficient criteria in mentor selection, and no logically connected learning path. Improvements suggested by the participants included more attention on mentor´s selection, forming guidelines for mentors, and guiding learning efforts in an organized manner. A recent innovative approach to onboarding and mentoring in one of the teams was introduced in the Results chapter and could serve as a good inspiration throughout the company because it effectively and efficiently addressed some of the issues that were present in the past and in other teams.

Tools

Communication tools supporting knowledge sharing such as instant messaging, e-mail, mailing lists, and audio/video conferencing tools were discovered. These appear to be standard tools and are well-known in the prior literature (Manteli et al., 2011; Niinimaki et al., 2010; Wendling et al., 2013).

A very strong emphasis was put on the KMS and knowledge stored there, which were termed as "documentation" by the study participants. The high level of perceiving knowledge sharing and documentation (KMS, written explicit knowledge) to be almost synonyms was surprising and it was giving the impression that knowledge sharing might be seen more as an artifact than a process. Although there was a high emphasis on documentation, the quality and attention

paid to the documentation did not correspond to it. This might lead to problems such as lowering the trust in the documentation and the KMS possibly becoming an information graveyard (Dingsøyr et al., 2009; Dingsøyr & Smite, 2013; Prikladnicki et al., 2003).

The emphasis on documentation might be problematic due to challenges, problems, and attitudes to expanding and updating it. Many of them are connected to topics of earlier chapters such as being busy, high time demands, being considered a boring activity, etc. People often seemed tempted to avoid it and at the same time, it was considered to be the primary knowledge sharing channel in the team, which might affect the quality and quantity of knowledge sharing overall. This strong association between knowledge sharing and the rather unpopular documentation might hurt the reputation of the general concept of sharing knowledge with colleagues. Furthermore, it can restrict creativity in finding suitable ways and methods of sharing knowledge, because of considering it to be already covered by "having documentation".

Regarding structure, wiki-based KMSs were found suitable for storing technical and project knowledge. Existing literature also reports the popularity of this type of KMS in IT organizations especially due to keeping knowledge organized with effective search functionality (Dorairaj et al., 2012; Taweel et al., 2009). One case of document-based KMS was reported to be inappropriate and having negative effects on knowledge storing and retrieval. Additional identified issues included the extensive and complex size with too many pages and documentation sometimes being scattered across multiple locations or systems.

Improving some existing flaws of the structure could be a very time-demanding and boring task. Therefore, it could be wiser and more effective to ensure that all the important and up-to-date knowledge is easy to find by the search function. Search functionality was earlier reported as crucial at finding the needed information in the complex KMS and some issues with ineffective or completely missing search function were also brought up in the literature (Aurum et al., 2008; Dingsøyr & Smite, 2013; Manteli et al., 2011). A sitemap could be useful for orientation in the extensive and complex structure, especially for newcomers. Dispersed locations of knowledge could be avoided by giving clear instructions where some specific knowledge belongs, even on the task level.

Outdated knowledge, for example from previous implementations, was widely present and brought real problems, especially being misleading, lowering the trust to KMS, and increasing its size. Herbsleb and Moitra (2001) pointed out similar issues while advocating for the importance of keeping KMS up to date. In agreement with Aurum et al. (2008), updating knowledge in the KMS might be considered difficult and be given only low priority. The most efficient strategy to fight this appears to be a long-term process of updating, removing, or at least flagging outdated information whenever spotted during normal work. Encouraging the team´s newbies to be active in this could be very effective if the process takes into consideration the challenges that they face with using and updating the KMS (Desouza et al., 2006).

5.3 How to improve the current situation

3. How could the quality and quantity of knowledge sharing be improved?

This research question was already partially answered in the previous section by presenting some suggestions related to reducing the influence of the individual factors. Here, a more high-level perspective on possible improvements is summarized.

The good message from the data is that practitioners considered knowledge sharing very important. They were able to name various benefits and explain the motivation to share their knowledge with colleagues. No significant issues regarding the desire for exclusive possession of unique knowledge were discovered. Therefore, the study environment did not have any clear resistance to sharing knowledge. That is a good sign because *"where there is a will, there is a way"* and the emphasis can then move on identifying suitable ways of sharing knowledge.

The primary general issue appeared to be the discrepancy between the declared importance of knowledge sharing and the attention paid to it during normal work. This was the case on multiple levels – technical professionals, managers, and the company-level culture and processes. Evidence of considering knowledge sharing important was found everywhere, but it was not reflected in the normal work, management decisions or actions, or company processes and culture.

The main recommendation that can be derived from the collected data is to increase the attention paid to knowledge sharing. There should be more discussion and planning. Some ad-hoc processes could be turned into more organized and repeatable approaches. The support for knowledge sharing activities should be clearly stated and effective encouragement used. The expected form and level of quality should be agreed upon within the team to make clear guidelines on how to share knowledge effectively and efficiently and to make taking action easier for everyone.

Given the differences between teams even within the same organization, this section cannot provide more detailed recommendations. Readers are encouraged to read the RESULTS chapter and the section answering RQ2, which can assist them in identifying what factors pose a problem in their own environments and what could be possible ways to reduce their influence.

5.4 Implications for practice

Practitioners can benefit from this study by understanding what factors might be discouraging or preventing knowledge sharing in software development teams. They might find obstacles that were unnoticed for a long time in their teams. Only once identified, it will be possible to address them. The study also suggests

multiple practices that could lead to improving the quality and quantity of knowledge sharing.

This study was exploratory and did not aim to generalize the findings. Even though some signals from practitioners from different European software development organizations indicate a relatively high level of generalizability, there is no strong systematically collected evidence for it in this study. Readers should keep in mind that each environment is different and has different needs, so the applicability of discovered discouraging factors and possible solutions in a specific environment must be carefully evaluated. The results of this study can serve for gaining understanding at the beginning of an effort to improve knowledge sharing inside a reader´s team. Thanks to a sample that represented practitioners with different roles, amount of experience, and background, the audience that can benefit from these results is rather wide.

5.5 Implications for theory

This study was primarily motivated by practical observations and reports from different software development organizations across multiple European countries, which suggested that knowledge sharing does not work well for a wide variety of reasons that were not previously well described. Some previous research efforts focused on motivation to share knowledge or reporting knowledge sharing practices used in software development organizations. Given the mentioned reports from practitioners, this study did not try to repeat the efforts of mapping practices and motivational factors, meaning *"What works and why?"*. Instead, it looked at the matter from a different angle by asking: *"Why knowledge sharing currently does not work and what prevents it from working well?"*. The collected data and results indicate that this approach is something that has been missing and that has the potential to bring useful knowledge for both practitioners and researchers. As a pilot study, it will hopefully encourage more researchers to focus on this topic for example by constructing applicable solutions and recommendations.

The results suggest that practitioners might be well aware of the importance and benefits of sharing knowledge with their colleagues, hence, possible expectation that the lack of knowledge sharing is caused by not seeing the importance or benefits might not be correct in all settings. Researchers should be aware that simply reminding the importance might not have a significant effect. However, the importance of knowledge sharing in software development teams might still need to be emphasized and explained to customers´ representatives without IT background, who might sometimes obstruct knowledge sharing efforts and practices.

There are currently many practices for sharing knowledge. It can be argued that reporting them over and over or introducing new ones might be of a low value to practitioners unless what prevents the practices from being used is addressed first. Discouraging factors should be considered while designing new

efforts and practices to support or encourage knowledge sharing. Potentially useful information for future researchers can be that the data suggested a demand especially for simple, fast, and easy-to-implement practices for effectively sharing one´s knowledge.

6 CONCLUSIONS

The main objective of this thesis originated from the surprising discrepancy between the theoretically declared importance of knowledge sharing in software development teams and the lack of action and attention paid to it in practice. The primary goal of the thesis was to attempt to identify what is the reason for this discrepancy and how the gap between what is declared and how the reality looks like could be reduced or even eliminated. More specifically, the focus was placed on identifying the discouraging factors, which prevent or discourage software development professionals from effectively sharing their knowledge with their team colleagues.

The prior literature provided the base for the whole research effort with the primary focus being on knowledge, its coordination, and what are the known challenges and practices of managing and sharing knowledge in software development teams. Additionally, the attention was more closely paid to the distributed software development, which is currently a common working arrangement in this global industry, and where different types of distances between colleagues make sharing knowledge much more challenging.

The empirical research aimed at understanding the phenomenon more in detail and did not pursue acquiring generalizable results, therefore, the interpretive case study method was selected, and qualitative empirical data were collected during semi-structured interviews. Study participants came from three different teams in one software development organization and held multiple different roles that are typical in the field (software developer, application management specialist, infrastructure specialist, manager). Owing to this role coverage, the richness of the empirical data increased.

The results of the thesis confirm that there truly is a gap between how important knowledge sharing is considered and how this importance can or cannot be seen in reality. A set of discouraging factors in knowledge sharing was identified. Some of them did not seem to play a significant role in the case environments, but most of them were considered to be quite strong. It is believed that the significance of individual factors may vary in other environments, hence, all the factors were reported.

The discouraging factors that were found to have little effect in the case environments were: Importance, Motivation and benefits, and Willingness. Participants claimed that, in their opinion, knowledge sharing is very important in the field of software development. Furthermore, they were able to reasonably explain their motivation for sharing knowledge and describe its benefits for individuals and the whole team. There was no strong evidence that practitioners would be unwilling to share their knowledge; however, some interesting reasons why individuals could be less proactive were discovered and described.

Based on the collected data, the most significant factors seem to be Perceived difficulty, Lack of attention, Insufficient or incorrect encouragement, and Missing systematic approach. Sharing knowledge was considered to be difficult

especially when there were no recommended ways how to do it and the expected level of quality of documentation was not agreed upon. Too often practitioners seemed to focus only on the primary work (writing code, solving service requests, etc.) and forget or ignore knowledge sharing needs, because those were considered a side activity. Encouragement of sharing knowledge in the teams appeared to be insufficient, incorrectly targeted, or expressed in way too abstract statements. When not enough attention is paid to knowledge sharing, there might not be any process, model, or even simple team agreement constructed and all (if any) knowledge sharing activities and efforts are only ad-hoc. That might be a missed opportunity to learn from the past and reduce the influence of the other discouraging factors.

The main contribution of this thesis can be seen in identifying and describing the discouraging factors that hinder or prevent knowledge sharing in software development teams. Only once these factors are known and understood, it is possible to address them. Some suggestions for limiting the effects of these factors and for increasing the quantity and quality of knowledge sharing were also introduced. Other researchers can benefit from understanding the current needs of practitioners and challenges that they face in this area to better target their future research efforts.

6.1 Limitations

Like every other academic study, this one also has certain limitations that could have affected the presented results. The limitations that the researcher is aware of are shared here with the reader to allow critical evaluation of this study.

To begin with, some limitations originate from the selected methodology. By its nature, the interpretive research does not aim to generalize. Instead, it seeks a deeper understanding of the studied phenomena. As a case study, the data were collected in one software development organization. Even though the participants were from multiple different countries (both by origin, and current location), all of them were accustomed to Finnish working culture and habits.

Primary data were collected during semi-structured interviews and there are certain risks related to self-reported data because participants´ impressions and observations might not always be precise. It was emphasized at the beginning of the interview that honesty is the key to identify the discouraging factors and it appeared that interviewees understood it as they admitted many rather negative aspects as well and they seemed to talk very openly; however, some information still might have been withheld or remained unspoken.

Interviews and analysis of qualitative data always involve a risk of misunderstanding or misinterpreting. This study has a single author, who conducted the interviews and performed the data analysis alone, so there is a possibility of researcher bias.

The sample size of ten participants is not big. Participants were from three different teams and working in different roles. This was needed to fulfill the aim

of the study of gaining deeper insights and it allowed to collect rich data from different environments and perspectives. However, the need to cover more environments and roles/responsibilities resulted in distributing the total number of participants into small groups where crosschecking claims relevant to the specific team and/or role was limited.

The selection of participants was largely based on people who volunteered for it either due to the interest in the topic, or encouragement of the team leader. Especially in the beginning, there was a concern of possible elite bias with participants being individuals who are very active in knowledge sharing. Based on the interviews, this does not seem to be the case. Some participants had thought about the topic before or have been initiative in it to some extent, but on average, participants seemed to represent individuals both more and less active in knowledge sharing.

Finally, certain limitations must be admitted on the side of the researcher as well. As a master´s thesis, this can be considered the first proper research conducted by the researcher. Despite the best effort, considerable time invested, and numerous consultations, some flaws originating from inexperience might still occur. Data collection was done with sufficient preparations and leading the interviews went well. The researcher attempted to keep an open mind and not to guide the interviewees according to his assumptions; however, in some cases, the participant´s response or thoughts might have been affected by the researcher. Data analysis was conducted in an organized way using renowned computer data analysis software.

6.2 Future research

Throughout this study, some ideas what the future research could focus on came up. Furthermore, the limitations of this research could also inspire other researchers to advance efforts in the same area.

As mentioned earlier, this study focused on teams in a single software development organization. Future research could be conducted in other organizations to compare how the results differ in other settings. All participants were accustomed to the Finnish working culture and it seems that cultural aspects could affect at least some of the discouraging factors and this should be explored further.

It could be interesting to investigate how different development methodologies affect knowledge sharing within the team. How does the emphasis on documentation in waterfall-style approaches compare to highlighting of interactions in the agile methodologies as far as knowledge sharing within the team is concerned?

Findings of the study and individual discussions with practitioners suggest that looking at knowledge sharing from the selected angle of *"Why people do not do it?"* could be very important for practitioners and further research efforts with the same approach are encouraged.

It seemed that one of the main problems of utilizing knowledge sharing practices by practitioners is the difficulty of adopting them. The data suggest that there is especially the need for simple and easy-to-implement practices. Even very effective practices might be of low value to practitioners if they are difficult to adopt and never become integrated into the team´s way of working. Future research should take this into account to accurately target the existing demand and needs. Research that would plan, implement, and monitor adopting of different concrete knowledge sharing practices in software development teams could be very interesting and beneficial for both practitioners and researchers. If the practices would vary for instance in terms of effectiveness, efficiency, difficulty to adopt, and time demands, it could identify what are the success factors of adopting different knowledge sharing practices by a software development team.

One considerable issue seemed to be that customers might sometimes be obstructing knowledge sharing efforts if they are more closely involved in the development team´s daily work. It might not be clear for them why the team should focus on something that does not produce immediate results and slows down the product delivery process. Some academic support advocating the importance of knowledge sharing in software development teams targeting the business-oriented audience that lacks IT knowledge and experience could help to improve the situation.

REFERENCES

Alavi, M., & Tiwana, A. (2002). Knowledge integration in virtual teams: The potential role of KMS. *Journal of the American Society for Information Science and Technology, 53*(12), 1029-1037.

Aurum, A., Daneshgar, F., & Ward, J. (2008). Investigating Knowledge Management practices in software development organisations–An Australian experience. *Information and Software Technology, 50*(6), 511-533.

Battin, R. D., Crocker, R., Kreidler, J., & Subramanian, K. (2001). Leveraging resources in global software development. *IEEE software, 18*(2), 70-77.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. & Thomas, D. (2001). *Manifesto for Agile Software Development.* Retrieved from https://www.agilemanifesto.org

Benbasat, I., Goldstein, D. & Mead, M. (2002). The case research strategy in studies of information systems. In Myers, M. D., & Avison, D. *Introducing Qualitative Methods: Qualitative research in information systems* (pp. 78-99). London: SAGE Publications, Ltd doi: 10.4135/9781849209687

Bjørnson, F. O., & Dingsøyr, T. (2005, June). A study of a mentoring program for knowledge transfer in a small software consultancy company. In *International Conference on Product Focused Software Process Improvement* (pp. 245-256). Springer, Berlin, Heidelberg.

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

Boden, A., Nett, B., & Wulf, V. (2009). Operational and strategic learning in global software development. *IEEE software, 27*(6), 58-65.

Busch, P., Richards, D., & Dampney, C. N. (2003, January). The graphical interpretation of plausible tacit knowledge flows. In *Proceedings of the Asia-Pacific symposium on Information visualisation-Volume 24* (pp. 37-46). Australian Computer Society, Inc..

Carmel, E., & Agarwal, R. (2001). Tactical approaches for alleviating distance in global software development. *IEEE software*, *18*(2), 22-29.

Damian, D., & Moitra, D. (2006). Guest editors' introduction: Global software development: How far have we come?. *IEEE software*, *23*(5), 17-19.

Davenport, T. H. & Prusak, L. (1998). *Working knowledge: How organizations manage what they know*. Boston (Mass.): Harvard Business School Press.

Desouza, K. C., Awazu, Y., & Baloh, P. (2006). Managing knowledge in global software development efforts: Issues and practices. *IEEE software, 23*(5), 30-37.

Dictionary.cambridge.org. (2019). *KNOWLEDGE | meaning in the Cambridge English Dictionary.* [online] Available at: https://dictionary.cambridge.org/dictionary/english/knowledge [Accessed 20 Sep. 2019].

Dictionary.cambridge.org. (2020). *ONBOARDING | meaning in the Cambridge English Dictionary.* [online] Available at: https://dictionary.cambridge.org/dictionary/english/onboarding [Accessed 30 May 2020].

Dingsøyr, T., Bjørnson, F. O., & Shull, F. (2009). What do we know about knowledge management? Practical implications for software engineering. *IEEE software, 26*(3), 100-103.

Dingsøyr, T., & Smite, D. (2013). Managing knowledge in global software development projects. *IT Professional, 16*(1), 22-29.

Dorairaj, S., Noble, J., & Malik, P. (2012, August). Knowledge management in distributed agile software development. In *2012 Agile Conference* (pp. 64-73). IEEE.

Drucker, P. F. (1993). *Post-capitalist society*. Oxford: Butterworth-Heinemann.

Ebert, C., & De Neve, P. (2001). Surviving global software development. *IEEE software, 18*(2), 62-69.

Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of management review, 14*(4), 532-550.

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

Friese, S. (2019). *Qualitative data analysis with ATLAS.ti*. SAGE Publications Limited.

Griffith, T. L., Sawyer, J. E., & Neale, M. A. (2003). Virtualness and knowledge in teams: Managing the love triangle of organizations, individuals, and information technology. *MIS quarterly, 27*(2), 265-287.

Herbsleb, J. D., Mockus, A., Finholt, T. A., & Grinter, R. E. (2001, July). An empirical study of global software development: distance and speed. In *Proceedings of the 23rd international conference on software engineering* (pp. 81-90). IEEE Computer Society.

Herbsleb, J. D., & Moitra, D. (2001). Global software development. *IEEE software, 18*(2), 16-20.

Hsieh, H. F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative health research, 15*(9), 1277-1288.

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

Kotlarsky, J., & Oshri, I. (2005). Social ties, knowledge sharing and successful collaboration in globally distributed system development projects. *European Journal of Information Systems, 14*(1), 37-48.

Manteli, C., Van Den Hooff, B., Tang, A., & Van Vliet, H. (2011, August). The impact of multi-site software governance on knowledge management. In *2011 IEEE Sixth International Conference on Global Software Engineering* (pp. 40-49). IEEE.

Myers, M. D. (1997). Qualitative Research in Information Systems. *MIS Quarterly, 21*(2).

Myers, M. & Avison, D. (2002). An introduction to qualitative research in information systems. In Myers, M. D., & Avison, D. *Introducing Qualitative Methods: Qualitative research in information systems* (pp. 2-12). London: SAGE Publications, Ltd doi: 10.4135/9781849209687

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

Niinimaki, T., Piri, A., Lassenius, C., & Paasivaara, M. (2010, August). Reflecting the choice and usage of communication tools in GSD projects with media synchronicity theory. In *2010 5th IEEE International Conference on Global Software Engineering* (pp. 3-12). IEEE.

Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization science, 5*(1), 14-37.

Nonaka, I., & Takeuchi, H. (1995). *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. Oxford university press.

Orlikowski, W. & Baroudi, J. (2002). Studying information technology in organizations: research approaches and assumptions. In Myers, M. D., & Avison, D. *Introducing Qualitative Methods: Qualitative research in information systems* (pp. 50-77). London: SAGE Publications, Ltd doi: 10.4135/9781849209687

Polanyi, M. (1966). *The Tacit Dimension*. Routledge, London.

Prikladnicki, R., Nicolas Audy, J. L., & Evaristo, R. (2003). Global software development in practice lessons learned. *Software Process: Improvement and Practice, 8*(4), 267-281.

Rode, H. (2016). To share or not to share: the effects of extrinsic and intrinsic motivations on knowledge-sharing in enterprise social media platforms. *Journal of Information Technology, 31*(2), 152-165.

Rus, I., & Lindvall, M. (2002). Knowledge management in software engineering. *IEEE Software, 19*(3), 26.

Ryan, S., & O'connor, R. V. (2009). Development of a team measure for tacit knowledge in software development teams. *Journal of Systems and Software*, *82*(2), 229-240.

Ryan, S., & O'Connor, R. V. (2013). Acquiring and sharing tacit knowledge in software development teams: An empirical study. *Information and Software Technology, 55*(9), 1614-1624.

Stake, R. E. (2010). *Qualitative research: Studying how things work*. Guilford Press.

Sternberg, R. J., Forsythe, G. B., Hedlund, J., Horvath, J. A., Wagner, R. K., Williams, W. M., Snook, S. A., & Grigorenko, E. L. (2000). *Practical intelligence in everyday life*. Cambridge University Press.

Szulanski, G. (1996). Exploring internal stickiness: Impediments to the transfer of best practice within the firm. *Strategic management journal, 17*(S2), 27-43.

Taweel, A., Delaney, B., Arvanitis, T. N., & Zhao, L. (2009, July). Communication, knowledge and co-ordination management in globally distributed software development: Informed by a scientific software engineering case study. In *2009 Fourth IEEE International Conference on Global Software Engineering* (pp. 370-375). IEEE.

The Standish Group. (1995). *The CHAOS Report (1994)*. Retrieved from https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf

The Standish Group. (2015). *The CHAOS Report 2015*. Retrieved from https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf

Walsham, G. (2002). Interpretive case studies in is research: nature and method. In Myers, M. D., & Avison, D. *Introducing Qualitative Methods: Qualitative research in information systems* (pp. 100-113). London: SAGE Publications, Ltd doi: 10.4135/9781849209687

Wegner, D. M. (1987). Transactive memory: A contemporary analysis of the group mind. In Mullen, B., & Goethals, G. R. (Eds.) *Theories of group behavior* (pp. 185-208). Springer, New York, NY.

Wegner, D. M., Giuliano, T., & Hertel, P. T. (1985). Cognitive interdependence in close relationships. In W. J. Ickes (Ed.), *Compatible and incompatible relationships* (pp. 253-276. Springer, New York, NY.

Wendling, M., Oliveira, M., & Carlos Gastaud Maçada, A. (2013). Knowledge sharing barriers in global teams. *Journal of Systems and Information Technology, 15*(3), 239-253.

Ågerfalk, P. J., Fitzgerald, B., Holmstrom Olsson, H., Lings, B., Lundell, B., & Ó Conchúir, E. (2005). A framework for considering opportunities and threats in distributed software development. In *Proceeding of International workshop on distributed software development*.

APPENDIX 1 INTERVIEW STRUCTURE

Introduction

- Researcher´s introduction
- The topic of the study, scope, goals, clarifying main terms
- The interview – types of questions, recording, data processing, privacy
- Openness, no-judging policy, the importance of honesty

Participants data

- Age group, gender
- Seniority (years of experience) in the field
- Seniority in the company
- Position/role

Theme 1: Knowledge sharing and knowledge management

- Grades (1-5) to: Importance, Current state
- Importance, benefits, part of the job, difficulties/obstacles
- Own activity in this area
- Own motivation to share knowledge

Theme 2: Environment

- The situation in the team – what works, what does not, why, and what could help
- Existing knowledge management model or processes, instructions, guidelines
- Culture, openness, and willingness to share knowledge
- Support, encouragement

Theme 3: Specific practices

- Familiar practices, used practices
- Onboarding, mentoring
- Knowledge repositories (KMS, documentation)
- Informal team knowledge sharing sessions
- Formal training (internal, external)
- Skills development plan