

Jere Huumonen

COMMUNICATION CHALLENGES IN DIS- TRIBUTED STUDENT PROJECTS

Faculty of Information Technology and Communication Sciences
M. Sc. Thesis
February 2022

ABSTRACT

Jere Huumonen: Communication challenges in distributed student projects
M.Sc. Thesis
Tampere University
Master's Degree Programme in Software Development
February 2022

Distributed software development has become more common in recent years when the possibilities for working over distances have improved and many developers have been forced to work from home due to the recent COVID-19 pandemic. Teams and team members in distributed environments face challenges due to distance factors that separate collaborators from each other. Communication has been considered the most common challenge in such environments. Instead of face-to-face communication, collaborators must rely on communication tools to communicate with each other over distances, which can naturally cause difficulties. Various solutions have been suggested for different challenges. For instance, the use of agile practices has been proven to improve communication. Communication challenges can negatively affect project success if left unsolved, which makes it important for practitioners to understand communication challenges and strategies to mitigate and solve them. However, new research is needed for identifying all the possible challenges and their solutions.

This study investigated communication in software development projects that involved university students who had to collaborate in a distributed environment with limited face-to-face communication possibilities due to the COVID-19 pandemic. The main objective of this study was to identify communication-related challenges that hindered communication between the participants. Secondly, this study identified how the teams managed to overcome the challenges, and whether the agile practices utilized by the teams helped solve the challenges.

For this purpose, a case study was conducted. Teams of students that participated in a software project work course at Tampere University, during the 2020 fall semester, provided the data for the study in the form of documentation, questionnaires, and interviews. The data were analyzed, which resulted in a list of communication challenges and their solutions.

Many communication challenges and solutions were identified. Communication was not the most significant challenge for the student teams that participated in the study. For this reason, most of the teams had no major difficulties with communication. However, the practices that were used would have most likely caused more significant problems in real-world projects.

Further research could be able to identify different types of challenges and their solutions from similar projects.

Keywords and terms: Communication challenges, virtual teams, distributed software development, student projects, case study.

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

Contents

1. Introduction	1
2. Distributed software development.....	3
2.1. What is distributed software development?	3
2.2. Teams in DSD	4
2.2.1. Loosely coupled teams	4
2.2.2. Virtual teams	4
2.3. Benefits of DSD	5
2.4. Challenges of DSD	6
2.5. Distance factors	7
3. Communication in virtual teams	10
3.1. What is communication?	10
3.2. Media richness theory	11
3.3. Media synchronicity theory	12
3.4. Modern communication tools	14
3.5. Communication challenges in virtual teams	16
3.5.1. Geographical and temporal distance	16
3.5.2. Team configuration	19
3.5.3. Customer communication	19
3.5.4. Project characteristics	20
3.5.5. Organizational factors	20
3.5.6. Human factors	21
4. Agile software development.....	22
4.1. Scrum	22
4.1.1. Scrum teams	23
4.1.2. Scrum events	23
4.2. Communication and Agile	24
4.3. Distributed Agile Software Development	25
4.4. Agile practices and their benefits	26
5. Case study	27
5.1. Context	27
5.2. Data collection and analysis	28
5.3. Communication in the project teams	29
5.3.1. Communication tools	31

5.3.2. Communication practices	33
5.4. Agile practices	35
5.5. Most significant challenges	36
5.6. Communication challenges	38
5.6.1. Project A	39
5.6.2. Project B	40
5.6.3. Project C	43
5.6.4. Project D	45
5.6.5. Project E	45
5.6.6. Project F	47
5.6.7. Project G	49
5.6.8. Project H	50
5.6.9. Project I	51
5.6.10. Project J	52
6. Results.....	55
6.1. Communication challenges related to distance factors	55
6.2. Communication challenges related to communication tools	57
6.3. Communication challenges related to client communication	58
6.4. Communication challenges related to team communication	58
6.5. Communication challenges related to human factors	60
6.6. Answers to research questions	61
6.6.1. Answer to RQ1	61
6.6.2. Answer to RQ2	62
6.6.3. Answer to RQ3	62
7. Conclusion.....	64
References	66

1. Introduction

The recent COVID-19 pandemic has forced many software development teams and developers to remote work instead of working in the traditional office environment. For many software development teams, this sudden shift has changed how team members communicate their work [Miller et al., 2021]. Before the pandemic, face-to-face communication was often the preferred communication channel in collocated teams due to its efficiency. Once the pandemic restrictions started to take place, many software development teams had to establish more effective online communication channels. However, relying on software communication tools is not a new phenomenon in software development. For decades, organizations have used geographically distributed teams to reap the benefits of globalization [Mockus and Herbsleb, 2001]. In distributed software development, software products are developed by teams and team members that are working from different sites [Lanubile, 2007]. Collaborators in distributed projects must often deal with the geographical distance between team members, as well as differences in culture, language, and time zones. Such characteristics give rise to challenges related to all parts of collaborative work: communication, coordination, cooperation, and management. However, communication-related challenges are the most common challenges in distributed software projects [Ghani et al., 2019].

Researching challenges that are encountered in software projects is important because they can negatively affect projects. Practitioners need to be aware of communication challenges to be able to effectively mitigate and solve them. Communication challenges specifically decrease the effectiveness and efficiency of communication, which can negatively affect project success [Alzoubi et al., 2016]. Several studies have identified communication challenges in distributed settings (e.g., Hummel and others [2013], Alzoubi and others [2016]). However, not all challenges and solutions have been identified. More research is needed on this topic.

The objective of this paper is to identify communication challenges in software projects that involve university students. The pandemic restrictions in Finland have affected universities by closing their doors and forcing students to study remotely during the 2020 – 2021 semester. Most university courses had to be organized fully online, including a software project work course at Tampere University. During the course, teams of students were tasked to create software products for real clients. Under normal circumstances, students would be able to collaborate face-to-face during the project. This time, face-to-face collaboration was limited due to the COVID-19 restrictions, and the teams had to rely on online communication tools. This provided a unique opportunity to research communication-related challenges encountered by the student teams during the 4–6-month software projects. In addition to identifying challenges, it is also necessary to identify solutions to

them to understand how student teams manage to overcome communication challenges. Furthermore, the student teams used certain agile practices that might improve communication and help the teams to overcome communication challenges. For this reason, the relation between agile practices and overcoming communication challenges is also investigated in this study. Communication challenges are studied based on the following research questions:

RQ1: What communication-related challenges can be found in student software projects?

RQ2: How were the communication challenges solved?

RQ3: How did agile practices help to solve or mitigate the challenges?

To answer these questions a case study was conducted. Data were collected from student projects that participated in a software project work course at Tampere University during the fall 2020 semester. The students provided data in the form of questionnaires, project documentation, and project manager interviews. The data was then analyzed, and communication challenges and solutions to the challenges were identified.

The remainder of the paper is structured as follows. The theoretical background of the thesis is presented in Chapters 2, 3, and 4. Chapter 2 focuses on the concepts of distributed software development and explains distributed projects, their benefits, and challenges, which is crucial for understanding the environment in which the student projects operate. Chapter 3 explains the main concepts of communication and provides background information about challenges related to communication, which is required for analyzing communication challenges in student projects. Chapter 4 explains agile software development and its relationship to distributed software development and communication. Chapter 5 contains the case study itself and explains the data collection methods, provides background information about the projects, and presents the challenges for individual teams. Chapter 6 contains the results and answers to the research questions. Finally, Chapter 7 concludes the paper and summarizes the main findings.

2. Distributed software development

In the past, software development projects were commonly worked by collocated teams that operated from a single physical site where developers were physically close to each other. In modern software development, teams and team members are often distributed across multiple sites, which is often called distributed software development. Virtual teams are one of the team types in distributed settings. They are especially common in recent times when COVID-19 has forced many software development teams and team members to collaborate in online settings. The student teams that are the subject of this paper can be considered virtual teams. In this chapter, a deeper dive into distributed software development is taken to understand the environment in which the student teams operate.

2.1. What is distributed software development?

Distributed software development (DSD) or *global software development* (GSD) can be defined as splitting the development of a software product or service among globally distributed sites [Lanubile, 2007]. In distributed settings, software projects are worked by different types of distributed teams. Projects in distributed settings are often called *distributed projects*. They can be worked by a single team whose team members are distributed across multiple locations, or multiple teams that are separated from each other [Ghani et al., 2019]. Distributed teams or team members can be distributed either locally within the same country or globally in different countries [Alzoubi et al., 2016]. In DSD literature, the most common scenario has been a distributed project that consists of multiple teams that are globally distributed [Ghani et al., 2019].

The terms DSD and GSD have been used interchangeably in the literature. However, some definitions make a distinction between GSD and DSD. Globally Distributed Software Development (GDSD) or Global Software Development (GSD) is sometimes defined as software development in which the distribution of the team exceeds national or continental boundaries [Bannerman et al., 2012]. DSD on the other hand could be understood as a broader concept that includes both the local and global contexts. Šmite and others [2014a] define GSD as the “development of a software artifact across more than one location”. They argue that the word “global” refers to the globe and hence also includes the local context of sites being distributed within the same country [Šmite et al., 2014a]. To reduce misunderstanding, the term DSD is used in this paper.

The environment in which distributed projects operate is slightly different compared to collocated projects. Distributed projects involve distributed team members or teams that are located on different sites (geographical distance). They may also be in different time zones or otherwise available at different times (temporal distance). Furthermore, collaborators in the global environment may encounter more socio-cultural differences, linguistic differences, and differences in work practices and process maturity compared

to collocated projects. Also, political, and legislative differences between countries can cause problems in a global environment [Moe and Šmite, 2007].

2.2. Teams in DSD

A team can be defined as “a group of people who are interdependent with respect to information, resources, and skills and who seek to combine their efforts to achieve a common goal” [Thompson, 2000]. Traditionally teams in software development were usually collocated, meaning that all team members work from the same physical location. Advancements in information and communication technology have enabled teams and team members to communicate and collaborate over long distances, which has enabled distributed projects. Teams in distributed software development can be roughly categorized into two types of teams: loosely coupled teams and virtual teams [Moe et al., 2016].

2.2.1. Loosely coupled teams

Loosely coupled teams are collocated teams that collaborate with other teams that are located on separate sites. Each set of collocated collaborators can be considered as a single team, and each team works mostly on separate tasks [Moe et al., 2016; Vallon et al., 2018]. Even though these types of teams are collocated, they can be considered distributed because the interaction is required between different teams, and the teams work on the same project with the same goals. In the literature, these types of teams have been given many names, such as *isolated distributed teams* [Vallon et al., 2018] or *loosely coupled teams* [Moe et al., 2016].

2.2.2. Virtual teams

Virtual teams can be defined as geographically distributed collaborations that rely on technology to communicate and cooperate [Morrison-Smith and Ruiz, 2020]. Based on the definition, virtual teams can be considered as the opposite of collocated teams. In these types of teams, team members are distributed across separate sites, but they work on the same tasks as a team [Vallon et al., 2018]. In distributed projects, there can either be a single virtual team with distributed members or multiple virtual teams each with distributed members [Ghani et al., 2019].

The term virtual team seems to be the most common way to refer to the concept. However, other names for it exist in the literature. Vallon and others [2018] called them *integrated distributed teams*. Other names include distributed team, remote team, computer-based team, online team, and cross-site team, which have been used interchangeably with the term virtual team [Abarca, 2020]. In some cases, the terms have been used inconsistently and they may have subtle differences. For instance, the term distributed team can refer to both virtual teams and loosely coupled teams [Vallon et al., 2018]. A distributed team, when used in a singular form, could be understood as a team in which

the team members are distributed across multiple sites. However, in the plural form, it could mean that the teams themselves are distributed across multiple sites, not necessarily the team members inside the teams. Furthermore, there seems to be a lack of consensus about the definition of virtuality or virtual teams. Orhan [2017] compared different definitions of virtuality and found that lack of face-to-face communication was the only characteristic shared by all definitions. In this paper, virtual teams are understood based on the geographical separation of the team members, and their need for technology to communicate.

In the past, virtual teams used to be rather rare in industry settings. Even in the literature, studies about virtual teams in software development have been reported to be scarce [Šmite et al., 2014b]. For instance, the systematic literature review by Jalali and Wohlin [2010] reports only a few cases of virtual teams in the literature between 1999 and 2009. Historically, open-source projects have been considered as some of the most distributed projects because they often rely on fully virtual teams [Fagerholm, 2014]. Virtual teams can operate as a temporary or permanent structure [Šmite et al., 2014b]. The former was a common practice in the early days of virtual work, and since then the term has evolved [Orhan, 2017]. These days virtual teams are more common in the industry, which has also been seen in the literature. For instance, newer systematic literature reviews by Vallon and others [2018] and Ghani and others [2019] showcase studies about virtual teams being reported more often. The trend of virtual teams in software development will most likely continue to increase due to the pandemic.

2.3. Benefits of DSD

There are many reasons why organizations would prefer distributed teams or team members instead of them all being collocated in the same work environment. Although, in some cases, they might be forced to do so, such as during the recent pandemic.

Having teams or team members working from multiple locations can have many economic benefits, especially in the global context. Access to the global talent pool and lower costs are some of the reasons why organizations started experimenting with distributed projects in the first place [Herbsleb and Moitra, 2001]. Software projects often require multi-disciplinary skills, which can complicate the process of hiring skillful team members from the same area [Čavrak et al., 2012]. With distributed projects, developers can be hired directly from where the required talent is available. However, one major disadvantage of hiring developers from different countries and areas is the increased cultural differences that can lead to communication problems [Conchúir, 2009].

The ability to move parts of the development work to other locations also enables organizations to cut development costs by moving development work to areas with cheaper developer wages [Conchúir, 2009]. However, the true costs of distributed projects may be affected by challenges that are unique to distributed settings. Challenges

related to areas such as communication, coordination, and management can increase the development costs [Conchúir, 2009].

Proximity to the market can be an important factor as well. Teams or developers that are close to the market can have better knowledge of the local customers and business conditions, which might be a considerable business advantage [Herbsleb and Moitra, 2001].

Teams or team members that are globally distributed can utilize time-zone differences to enable round-the-clock development [Herbsleb and Moitra, 2001]. This form of DSD is often called “follow the sun” (FTS) development. In this approach, at the end of each work shift, the work of a single development site is handed off to developers in a different time zone in which the local time can be several hours behind the first site [Carmel et al., 2010]. The idea is that the work of one site can be continued on another site. FTS can increase the development speed as it enables constant progress, which can ultimately reduce the time-to-market. On the other hand, FTS can be difficult in practice due to challenges such as work coordination. However, FTS could be beneficial when used on a smaller scale, for instance, in a specific development phase such as testing [Carmel et al., 2010].

In addition to economic benefits, the distributed working environment can positively affect a team’s performance. Team members in globally distributed teams have often diverse backgrounds, which can provide several benefits. Diversity of team members can increase the number of different perspectives, which can improve teams’ creativity and ability to solve problems. Diverse teams often have access to a diverse set of resources, skills, and knowledge, which can make team members complement each other. Diverse teams can also be more prepared for dealing with challenges [Jimenez et al., 2017].

With virtual teams, team members are not necessarily tied to a specific physical location, which can offer more flexible working conditions and increase employee satisfaction. When all communication happens through online channels, team members might no longer need to spend time and resources commuting to the office.

2.4. Challenges of DSD

In addition to benefits, there are many challenges related to distributed projects. Challenges of DSD have been a subject of research for several decades [Mockus and Herbsleb, 2001]. Many of the challenges have remained the same to this day, even though the times have changed, and new technologies and trends have made distributed software development easier. Change often introduces new challenges. For instance, the recent pandemic has caused significant trouble for software teams [Miller et al., 2021]. Some new trends, such as agile software development have brought their benefits but also new challenges to distributed settings [Ramesh et al., 2006]. Case studies have been a common approach

to report challenges encountered in distributed software projects. Many systematic literature reviews and mapping studies have identified challenges over the years (e.g, [Jiménez et al., 2009], [Morrison-Smith and Ruiz, 2020]). As a result, a wide variety of known challenges exist, and they have been categorized differently by different authors.

Distance is the main cause for challenges in distributed settings as it causes either directly or indirectly most of the challenges. Some challenges are unique to distributed settings. For instance, challenges caused by time zone differences between collaborators only exist in distributed settings. Certain challenges can be encountered in both collocated and distributed environments. For instance, linguistic or socio-cultural differences can cause problems in both environments. Distance factors are addressed further in the following section. As distance factors are the cause for most challenges, some papers have categorized challenges mostly based on them.

Challenges can be present in all aspects of collaborative work. Collaboration has three main dimensions that are defined by the 3C's collaboration model: Communication, coordination, and cooperation. These dimensions were first introduced by Ellis and others [1991], and the model was later extended by Fuks and others [2007]. Communication takes place when people share information and negotiate with each other to make decisions. Coordination refers to the activities related to the management of people and their activities and resources to achieve common objectives. Cooperation refers to the execution of shared tasks as a group while sharing the same space and using shared artifacts [Steinmacher, 2013]. Collaboration requires awareness, which enables collaborators to be aware of each other's work and adjust their work accordingly [Steinmacher, 2013]. In software development, collaboration is often also affected by project management activities that aim to control the collaboration and mitigate or solve any risks or challenges that might cause problems. These controlling activities can also be seen as one dimension of collaboration. Some papers have used these high levels of collaboration to address challenges in distributed settings. In this paper, the focus is on the communication dimension.

2.5. Distance factors

Distance between collaborators is the most notable factor that separates traditional collocated teams from teams that work in distributed projects. It is also the factor that makes distant collaborations more difficult compared to their collocated counterparts because it is the root cause for many challenges encountered in distant collaborations. The concept of distance, in this case, does not only refer to distance that can be measured in meters, but many types of factors can separate team members from each other in distributed projects. The most common distances discussed in the DSD literature are geographical, tem-

poral, and socio-cultural distances. Other distance concepts exist, such as perceived distance, but they have received much less attention in the literature [Wilson et al., 2008]. These distance concepts are further discussed in this section.

In distributed software development, development teams or team members are geographically distributed, meaning that there is a *geographical distance* between them. This distance can be measured in meters. However, it might be better to measure geographical distance based on the amount of effort that is required to visit another distant collaborator, because sometimes shorter distances can require more effort depending on transport infrastructure and opportunities to travel. Two locations can be considered geographically close if there is a good transport infrastructure between them (e.g., regular direct flights) [Ågerfalk et al., 2005]. *Physical distance* might be a better option than geographical when accurate distances, measured in meters, are needed. This is because the physical distance is a broader definition and can much accurately describe smaller distance differences, such as being in the different parts of a building [Ghani et al., 2019].

Temporal distance is another important distance concept that needs to be considered in distant collaborations. Temporal distance is the distance in time. It defines the amount of time that separates collaborators from each other. It can normally be a result of differences in time zones or work shifts [Ågerfalk et al., 2005]. The temporal distance can be measured in the number of hours or time zones that separate the collaborators. However, a higher number of hours or time zones does not necessarily imply higher temporal distance. Differences in time zones and work shifts can either increase or decrease the temporal distance. For instance, in some cases, a one-hour difference can lead to high temporal distance due to differences in workday routines, while a six-hour difference in time might lead to low temporal distance due to work shifts that are compatible over different time zones [Ågerfalk et al., 2005].

Socio-cultural distance is a concept that is often discussed together with temporal and geographical distances. It defines an individual's understanding of another individual's values and practices. Socio-cultural distance includes dimensions such as language, organizational culture, national culture, political views, individual motivations, and work ethics [Ågerfalk et al., 2005]. Individuals who have similarities in those dimensions are socio-culturally closer to each other. For instance, two collaborators who share the same language and culture might be socio-culturally closer than collaborators that come from different cultures or have different native languages.

In addition to geographical, temporal, and socio-cultural distances, there are also other types of distances that can be important in the context of distributed collaborations. Wilson et al. (2008) explored the concept of *perceived distance*, which defines an individual's subjective perception of distance. This perceived distance has cognitive and affective dimensions. The cognitive dimension defines how the distance between team

members seems, while the affective dimension defines how the distance feels. Collaborators who are far away from each other do not necessarily feel distant from each other, while in some situations geographically close collaborators might feel very distant from each other [Wilson et al., 2008].

3. Communication in virtual teams

Software development is highly collaborative work. To succeed, enough communication is required. Lack of communication or poor communication practices can be a significant factor leading to project failure. Efficient communication practices become even more important in distributed settings, in which face-to-face communication is limited. This chapter takes a closer look into communication in distributed settings.

3.1. What is communication?

It is important to first understand what is meant by communication. Many definitions exist for it. Oxford English Dictionary [OED Online, 2021] defines communication as “the transmission or exchange of information, knowledge, or ideas, by means of speech, writing, mechanical or electronic media, etc.”, which emphasizes the information-sharing aspect of communication. West and Turner [2010, p. 5] emphasize the social side of communication by defining it as “a social process in which individuals employ symbols to establish and interpret meaning in their environment”. Rogers [2003, p. 5] on the other hand defines communication as “a process in which participants create and share information with one another in order to reach a mutual understanding”. It might be a more suitable definition for communication in project working environments, in which human participants exchange information, and achieving a mutual understanding is often important.

The definitions in the previous paragraph described communication as a process. This communication process contains eight key components: source, message, channel, receiver, feedback, environment, context, and interference [University of Minnesota Libraries Publishing, 2017, p. 7-10]. The *source* is what initiates the communication process by sending a message. The *receiver* is whoever receives the message. The *message* contains some information the source wants the receivers to receive. The *channel* is the way the message is transmitted to the receiver. A channel can be, for instance, an email or a face-to-face conversation. *Feedback* is the response that the receiver sends back to the source after receiving the message. The *environment* defines the physical or psychological place where messages are transmitted. It can be, for instance, a busy office full of people or a quiet library corridor. The environment produces cues that define the expected behavior for different contexts. The *context* of communication defines what people should expect from each other in a certain environment or situation, and it is often affected by the environment. The context can, for instance, define who is allowed to speak and when in a formal business meeting. *Interference* or *noise* is anything that can block messages or change the original meaning of the messages when they are being transmitted between sources and receivers [University of Minnesota Libraries Publishing, 2017, p. 7-8].

Next, a closer look into communication channels and their differences is taken to understand the difference between communication in collocated settings in which face-

to-face communication is possible, and distributed settings in which communication tools must be used.

3.2. Media richness theory

Communication channels are one of the key components of the communication process, as mentioned in the previous section. Many different communication channels exist. In software projects, some of the more common communication channels are face-to-face communication, video conferencing, online chat, email, voice calls, and documentation [Ahmad et al., 2018]. Different communication channels have different capabilities that make certain channels better than others in certain tasks or contexts. Several theories have been suggested to address the differences between communication channels. Most notably, *media richness theory* and *media synchronicity theory*. These theories are important because they can reveal how communication capabilities change when moving from face-to-face communication to communication tools that are typically used in virtual teams. They can also reveal how communication tools should be selected for certain communication tasks or contexts.

Communication channels vary in their ability to transfer information. Media richness (or information richness) theory, first introduced by Daft and Lengel [1986], is often used to define the ability of communication channels to transfer information. This can be very important when comparing different communication channels and deciding which communication tools should be selected for specific purposes. Appropriate communication media can be selected by matching media richness of a communication channel to the equivocality, or ambiguity, of the task [Daft and Lengel, 1986]

According to media richness theory, different types of media differ in richness. Media richness can be defined as “the ability of information to change understanding within a timer interval”. Richness differences are caused by multiple factors, such as the immediacy of communication feedback, the number of cues and channels utilized, personalization, and variety of language [Daft and Lengel, 1986].

According to media richness theory, communication effectiveness improves in complex communication tasks with high amounts of ambiguity when rich media is utilized for the task. Face-to-face communication can be considered the richest medium due to its capabilities of providing immediate feedback and multiple cues through voice and body language. Video conferencing is another rich medium, but it is not as rich as face-to-face communication because it is missing some of the cues of face-to-face communication. Audio calls on the other hand are less rich than video calls because participants are missing all visual cues. Rich media is often more suitable for complex and equivocal communication tasks with multiple interpretations of information. Availability of social cues and the possibility for quick feedback enable participants to communicate through rich media to reach mutual understanding more easily compared to less rich media. For instance,

explaining complex technical details might be more effective through a rich medium [Daft and Lengel, 1986].

Leaner media, on the other hand, provide fewer cues and feedback, which makes it less appropriate for equivocal communication. Lean media can still be effective for simple communication tasks and less unequivocal messages [Daft and Lengel, 1986]. Written documents, letters, and emails can be considered leaner media.

Media richness theory has been used for selecting appropriate media for communication tasks. Ever since media richness theory was first introduced, new factors related to how the appropriate media should be selected have been suggested. Social influence and individual experience can affect the selection of communication media. For instance, an individual might choose a less rich medium for an equivocal task due to influence from their supervisor, or experience or familiarity with the medium. Symbolism can also affect how media is selected for communication tasks. A letter might represent formality, while an email might represent urgency. Symbolism is often derived from organizational culture, which can also affect the usage of communication media. Overall, the view towards media richness and its characteristics has shifted from an objective to a more subjective view [Ishii et al., 2019].

3.3. Media synchronicity theory

Media richness theory has been mostly utilized for selecting appropriate communication media for communication tasks, even though the theory originally addressed communication performance [Dennis et al., 2008]. However, other theories have been introduced to expand the ideas of media richness. One of them is the media synchronicity theory. It was first introduced by Dennis and Valacich [1999] and later improved by Dennis and others [2008]. Media synchronicity theory addresses the importance of media to support synchronicity and uses media synchronicity to predict performance in communication tasks.

Media synchronicity is related to *synchronous* and *asynchronous* communication. In synchronous communication, messages are received and replied to in real-time, with participants engaging in communication at the same time, while in asynchronous communication the participants might not be able to reply to messages immediately after they have been received because the participants may not be interacting at the same time [Simpson, 2002]. Communication media such as face-to-face conversation or video conference can be considered as synchronous media, while communication media such as voice mail can be considered as asynchronous communication media. Some communication media can support both asynchronous and synchronous communication (e.g., text chat) [Dennis et al., 2008]. Dennis and others [2008] define media synchronicity as “the extent to which the capabilities of communication medium enable individuals to achieve synchronicity”.

Media synchronicity theory proposes that all communication tasks can be divided into two communication processes: conveyance and convergence. In the *conveyance* process, the information receiver processes new information to create or modify their mental models of the situation. In the *convergence* process, the meaning of the information is mutually agreed on, given that a common understanding is reached between all participants. Conveyance and convergence processes require varying amounts of information transmission (preparing, transmitting, and receiving information) and information processing (understanding information) that are influenced by media capabilities [Dennis et al., 2008].

Media synchronicity theory defines five media capabilities that influence information transmission and processing, and subsequently synchronicity: transmission velocity, parallelism, symbol sets, rehearsability, and reprocessability.

Transmission velocity (also known as the *immediacy of feedback*) defines how fast a message can be transmitted to the recipient. High transmission velocity enables recipients to receive messages immediately and create faster responses, which increases synchronicity [Dennis et al., 2008].

Parallelism defines how many message transmissions from multiple senders or conversations can occur at the same time over the medium. For instance, traditional telephone call often allows a single conversation, while electronic media often allows multiple parallel conversations. Parallelism increases the amount of possible simultaneous conversations. However, having multiple simultaneous conversations reduces shared focus, which consequently lowers synchronicity [Dennis et al., 2008].

Symbol sets (or symbol variety) refer to the number of ways a medium allows information to be encoded for communication (e.g., verbal- and nonverbal cues, written words, images, videos). Symbols might affect the efficiency of information transmission and processing. Some symbols are easier to encode or decode than others. For instance, an image may be easier to understand than the contents of the image as a written text. Particularly, media with natural symbols are better at supporting synchronicity than media with less natural symbol sets. Furthermore, certain symbol sets might be more suitable for certain messages. Using media that support appropriate symbol sets for a given communication task might improve information transmission and processing and provide more synchronicity capabilities [Dennis et al., 2008].

Rehearsability refers to the ability to rehearse or edit messages before sending them, which allows senders to ensure that the message contains the intended meaning. Rehearsability is important for communication tasks that involve new or complex information because it allows individuals to encode messages more accurately so that they can be understood by the recipient. However, rehearsability can cause senders to spend more time encoding messages, which may reduce synchronicity [Dennis et al., 2008].

Reprocessability refers to the ability to access and process a received message afterward. For instance, an email message can be accessed again after it was initially received and read unless the message is deleted. Reprocessability is important for conveyance processes because it supports information processing by allowing individuals to access previously received messages. However, similar to rehearsability, reprocessability can slow down communication because recipients can spend more time on processing previously received messages, and consequently reduce synchronicity [Dennis et al., 2008].

Normally individuals use both conveyance and convergence processes and require media that support both. Media synchronicity theory suggests that individuals may have different needs for conveyance and convergence processes in different contexts. Individuals highly familiar with each other, the task, and the media, have a higher need for synchronicity supporting media than individuals unfamiliar with each other, the task, or the media. These needs might, however, change over time. When individuals start first working with each other, the context is often unfamiliar to them, and there is a higher need for synchronicity supporting media. Once the context becomes more familiar, fewer convergence processes are needed and there is less need for synchronicity supporting media. Thus, the media needs might change over time when working with a team on a project [Dennis et al., 2008].

According to media synchronicity theory, the most suitable media for communication tasks is one that provides the best capabilities given the situation. No single medium could be labeled as the most appropriate medium for a task because no single medium provides the best capabilities for both information transmission and information processing. For instance, face-to-face communication, video conferencing, and audio conferencing have fast information transmission, but low information processing capabilities. These can be considered more synchronous media. Less synchronous media, such as text documents or email have high information processing capabilities, but slow information transmission. Combining multiple different mediums can balance the advantages and disadvantages of the selected mediums, which can improve communication performance. For distant collaborations, it is important to consider the capabilities of media because communication processes in distant environments may require different media capabilities [Dennis et al., 2008].

3.4. Modern communication tools

One of the main characteristics of virtual teams is the use of electronic communication media instead of face-to-face communication which is often the main communication channel in collocated teams. There are limited possibilities for face-to-face communication in virtual teams because of the distance differences between collaborators, and in some cases, the team members might not see each other face-to-face during the entire project. Instead, virtual teams often rely on computer-mediated communication (CMC)

technology [Morrison-Smith and Ruiz, 2020]. CMC covers a variety of concepts and is commonly defined as human communication that is performed using computers [Simpson, 2002]. CMC has been available in various forms since the 1960s. However, CMC tools started becoming popular in the 1990s when computers and the Internet became widespread [Thurlow et al., 2004, p. 26]. CMC includes many tools such as text-based chats, audio and video conferencing, email, and discussion forums [Simpson, 2002].

Over the past two decades, social media has changed how individuals communicate with each other. Before the era of social media, email was the most popular way to communicate over the Internet. The dominance of email changed already in 2009 when social media platforms bypassed email as the most popular way to communicate over the Internet. However, email remains a popular tool for communication [Cardon and Marshal, 2015]. Social media has also affected the development of many communication tools. Organizations have started utilizing social media for internal communication, although this development has not been as fast for organizations as it has been for individuals [Cardon and Marshal, 2015]. This has also led to the development of enterprise social networking (ESN) platforms (e.g., Yammer), which are social networking platforms designed for organization-wide outer loop communication. In recent years there has been also an increasing number of communication platforms that are mostly designed for inner-loop communication for teams. For instance, Slack and Microsoft Teams. These platforms offer multiple tools for communication, which makes them more effective and efficient for team communication compared to traditional tools such as email [Cardon and Marshal, 2015].

In recent years there has been a rapid growth in communication and collaboration tools. Calefato and Ebert [2019] compared some of the more popular modern tools for team collaboration. The study included communication tools such as Microsoft Teams, Slack, and Skype for Business. Modern communication and collaboration tools include features that allow communication in multiple ways, such as audio and video calls, text chat channels, screen sharing, file sharing, and calendars. These tools serve as hubs of collaboration and communication, instead of as a single channel of communication which is often the case with traditional tools such as email [Calefato and Ebert, 2019]. Modern communication and collaboration tools such as Microsoft Teams and Slack are not only for improving collaboration and communication, but they can also improve awareness about co-workers, tasks, and artifacts [Calefato and Ebert, 2019]. The variety of asynchronous and synchronous channels in these tools can even make traditional tools such as email obsolete for team communication. Slack even markets itself as a replacement for email when it comes to team communication [Slack, 2021].

A collocated environment is not necessarily the best environment for a software team, even though face-to-face communication is available. With the help of modern communication tools, virtual teams can create an effective virtual work environment, in which the physical separation of collaborators is not a barrier to communication [Lous et al., 2018a].

3.5. Communication challenges in virtual teams

In this paper, the focus is on identifying communication challenges from student projects. Communication was selected to the challenges it can create in collaborative work environments, especially in distributed settings. Communication can be considered the most common challenge in agile DSD [Ghani et al., 2019]. Before going into the case study, we must first understand the communication challenges that can be expected in software projects.

Communication challenges can be understood as constraints that decrease the efficiency and effectiveness of communication, and consequently, they can negatively affect the project's success [Alzoubi et al., 2016]. Several studies have discussed communication challenges related to software projects in distributed settings. Case studies have been a common approach to reporting such challenges. There have been also various systematic literature reviews that have examined communication challenges based on several case studies and other research papers, including Hummel and others [2013], and Alzoubi and others [2016]. Alzoubi and others [2016] conducted a systematic literature review on communication challenges in globally distributed agile software development to identify communication challenges and solutions to them. They found a total of 17 communication challenges and categorized them into six distinct categories: distance differences, team configuration, project characteristics, customer communication, organizational factors, and human factors.

Communication challenges are discussed in more detail in the following subsections. The challenges discussed in this paper are based on the challenges identified by Alzoubi and others [2016]. The focus of this paper is on student projects, and not all the challenges in the literature are necessarily relevant to them. For this reason, some of the challenges are only briefly mentioned.

3.5.1. Geographical and temporal distance

Geographical and temporal distance differences between teams and team members have been reported to be major factors for communication difficulties in distributed settings. According to the systematic literature review made by Alzoubi and others [2016], 76% of the papers analyzed in the study mentioned distance-related communication challenges, making it the main challenge of communication. Distance differences are directly or indirectly the reason for many communication challenges. Temporal and geographical

distance differences are mostly related to difficulties in organizing communication. However, indirectly, they can be one of the root causes for many types of challenges. For instance, geographical distance can indirectly lead to challenges with communication tools, because it can be the reason why such tools have to be used in the first place.

The geographical distance between team members and other stakeholders can greatly reduce opportunities to organize meetings and other synchronous communication. Geographical distance increases the costs of face-to-face meetings [Ågerfalk et al., 2005]. The higher the geographical distance, the more effort it requires to organize face-to-face meetings. For instance, collaborators who are working in the same building or city with good transport infrastructure could organize face-to-face meetings quite effortlessly compared to collaborators who are on different continents, thousands of kilometers away from each other. In some cases, it might not be possible to organize face-to-face meetings at all due to the high costs and effort required for organizing them.

Temporal distance on the other hand reduces opportunities for synchronous communication, which is the result of differences in time zones or work shifts [Ågerfalk et al., 2005]. Significant differences in time zones can be a barrier to communication [Robinson, 2019]. For instance, team members might not be available at the same time, which can make synchronous communication problematic. Even though, organizing communication can be more difficult in distant collaborations, lack of communication is often not the main problem. A recent study that investigated the effects of COVID-19 on team productivity found that communication problems were often related to the quality of communication rather than the lack of communication [Miller et al., 2021].

Distance differences can also decrease the possibilities for informal communication. In collocated settings, informal communication is a common way to exchange information as it happens naturally when team members interact with each other in a work environment. In distributed settings, informal or unintentional communication is less common, which might negatively affect team members' task awareness [Berry, 2011]. Lack of informal communication can also make knowledge sharing more difficult, especially in agile software development in which most knowledge is often tacit instead of explicit. Undocumented knowledge can only be shared with active communication. Without informal communication, there are fewer chances for sharing tacit knowledge [Razzak and Ahmad, 2014].

Distance differences also can reduce task awareness. Awareness can be defined as “an understanding of the activities of others, which provides a context for your own activity” [Dourish and Bellotti, 1992]. Awareness in software development is often about knowing what other team members are working on or knowing who has expertise in specific areas of development. According to Gutwin and others [1996], four types of group

awareness exist. *Informal awareness* provides knowledge about the presence and availability of other people in the work environment. *Social awareness* is the information individuals have about other people in social context or conversations (e.g., emotions, level of interest). *Group-structural awareness* provides knowledge about group processes and team members' roles, responsibilities, and status (e.g., how coding tasks have been divided among the team). *Workspace awareness* provides knowledge about a team's interactions with the work environment and its artifacts (e.g., knowing when someone makes contributions) [Gutwin et al., 1996].

Awareness is important for coordination and communication activities in collaborative environments, which lead to the development of common ground [Modi et al., 2013]. Awareness works differently in collocated settings compared to distributed settings. In collocated settings awareness forms naturally when individuals interact with each other in a work environment that supports both verbal and non-verbal awareness cues. However, in distributed settings, individuals can face challenges related to the task and team awareness because of the reduced awareness cues caused by distance differences between team members [Modi et al., 2013]. Informal communication is often important for task awareness, and the lack of it in distributed settings can cause team members to miss critical information [Bannerman et al., 2012]. Lack of awareness can also make normal communication more difficult. Without group awareness, team members might have to rely more on assumptions and perceptions, or in some cases, it can lead to isolation [Morrison-Smith and Ruiz, 2020].

Other communication difficulties caused by geographical and temporal distance include losing track of the work process, longer meetings, difficulties organizing meetings, communication delays, and lack of trust [Alzoubi et al., 2016].

Some practices can help teams overcome geographical and temporal distance problems. Alzoubi and others [2016] listed some practices based on the literature that can help to overcome the challenges. These include synchronizing work hours between team members, creating teams based on time zones or geographical location, minimizing dependencies, providing a more flexible working environment, dividing meetings into site-specific parts and parts common to every site, organizing face-to-face meetings or visits regularly, and using appropriate synchronous or asynchronous tools [Alzoubi et al., 2016].

Modern communication tools can make reliance on face-to-face communication less critical. Face-to-face communication might not be needed if other appropriate communication channels are available and the temporal distance between team members is small [Robinson, 2019]. Even though face-to-face communication can be replaced with communication tools, it is still recommended to use face-to-face communication when possible [Morrison-Smith and Ruiz, 2020]. Organizing project kickoff meetings face-to-face

can especially be important for the success of the project [Morrison-Smith and Ruiz, 2020].

3.5.2. Team configuration

Team size, the number of teams, and team coordination can cause communication difficulties, as identified by Alzoubi and others [2016]. Communication is easier for smaller teams and can cause difficulties for larger teams [Hummel et al., 2013]. Similarly, communication becomes more difficult the more teams are included in a project. Coordination between teams and team members can also become more difficult in distributed settings. For instance, a recent study showed that communication aspects such as social connection to the team, ability to brainstorm with team members, ease of communication with team members, and knowledge flow within the team have decreased due to the shift to remote working conditions caused by the pandemic [Miller et al., 2021].

Difficulties caused by these factors include early communication difficulties, difficulties in team formations, difficulties ensuring that team members communicate with each other, the unwillingness of certain team members to communicate, less teamwork, and slow team communication [Alzoubi et al., 2016].

Many practices can help to solve problems related to team communication. These include organizing face-to-face meetings at the beginning of the project, demonstrating the product after each iteration, organizing regular meetings, involving the customer in development, encouraging trust, and using synchronous communication tools [Alzoubi et al., 2016].

3.5.3. Customer communication

In agile software projects, customers are often involved in the development process throughout the life cycle of the project. They provide feedback to the developers, which makes customer communication important. In distributed settings, customer communication can be difficult due to distance factors. Sufficient customer involvement and communication can be difficult to achieve especially in distributed settings. Alzoubi and others [2016] found several consequences for insufficient customer involvement, including: less frequent communication with customers, weak relationships with customers, customer uninvolvedness, hiding information from customers, and miscommunication of requirements. To address these problems, regular meetings and active communication with the customers could be organized. Customer representatives could be used if the customers are unavailable for active communication [Alzoubi et al., 2016].

3.5.4. Project characteristics

Project characteristics, such as project domain, project architecture, and project type can cause communication difficulties. For instance, unclear software or system structure definitions can cause misunderstandings or unnecessary communication. However, communication challenges related to project characteristics have been rarely mentioned in the literature [Alzoubi et al., 2016].

3.5.5. Organizational factors

Organizational factors, such as organizational culture, project management processes, and communication tools and infrastructure can cause communication difficulties [Alzoubi et al., 2016]. Project management in distributed settings can be difficult because distance differences can reduce interactions that are required for effective leadership [Morrison-Smith and Ruiz, 2020]. Organizational culture can cause difficulties as well. For instance, organizational bureaucracy can decrease the efficiency and effectiveness of communication. Organizational culture should support rapid communication and trust between stakeholders [Alzoubi et al., 2016].

The use of communication tools can cause communication difficulties that might not be normally encountered in collocated settings, including, using unsuitable tools, technical incompatibilities between sites, bad quality of video conferencing communication and coordination, a lack of tool support, and extra costs of tool training [Alzoubi et al., 2016]. Furthermore, face-to-face interaction is different than interacting through communication tools. Communication, sharing, and interpreting information are done without gestural language in distributed settings [Abarca et al., 2020]. Compared to face-to-face interaction, computer-mediated interaction has less social information and cues (e.g., facial expression, gestures, contextual cues) that normally help individuals to better understand interactions with others. Reduced social information might lead to misunderstandings [Berry, 2011]. Communication activity might vary drastically between team members when using communication tools. Some team members might be very active communicators and dominate over the communication channels, while some team members might contribute very little. This can be especially problematic in self-managing agile teams that require balanced communication between different collaborators. Furthermore, team members who use communication tools for communication might prefer direct messages rather than public communication channels. In some situations, using too much direct messaging can hide relevant information from other team members [Stray et al., 2019].

Assessing and selecting appropriate tools and using different tools and communication models can help with the difficulties [Alzoubi et al., 2016]. Using a combination of

communication tools can increase the efficiency of communication and mitigate problems. However, too many communication tools and channels can cause difficulties to follow discussions that occur through multiple channels [Stray et al., 2019].

3.5.6. Human factors

Personal differences can cause communication difficulties, especially in distributed projects that often involve participants from different backgrounds. Alzoubi and others [2016] categorized challenges related to personal differences. The category includes differences in language, national culture (norms, values, language, style of communication), trust, and personal practices (attitudes and skills). The differences can cause a wide variety of difficulties, such as language misunderstandings, miscommunication, longer meetings, interpretation differences, and silence of participants [Alzoubi et al., 2016].

Alzoubi and others [2016] listed solutions for the problems based on the literature. These include organizing occasional face-to-face visits, addressing language problems early, speaking slowly and clearly, sending agenda to all participants before meetings, simplifying language, using wikis, recording meetings, and using asynchronous communication [Alzoubi et al., 2016].

4. Agile software development

The student projects that are studied in this paper used agile practices that are close to Scrum. One of the goals of this paper was to identify whether agile practices can help teams to solve communication challenges. For this reason, a brief look into agile software development is taken.

The rise of *agile software development* has been one of the biggest changes in the software development industry in the past two decades. Agile software development can be understood as software development that follows the values and principles defined in *the Agile Manifesto* [Beck et al., 2001] defines the main values of agile software development in the following way:

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

The manifesto places more value on the items on the left compared to the items on the right, while acknowledging that the items on the right should not be completely ignored. In addition to these values, the manifesto also defines twelve principles.

In the literature of agile software development, there seems to have been a lack of consensus of what the term “agile” means. The concept of agility has been used inconsistently by different authors and the variety of different agile methods and practices makes it even more difficult to give a common definition for them [Conboy, 2009]. Conboy [2009] addressed this problem and defined agility. According to Conboy [2009], agility can be defined as “the continual readiness of an information systems development to create change rapidly or inherently, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment”.

Agile practices and principles are quite common in software development today. The most recent State of Agile report shows that at least 94% of the respondents’ organizations are practicing agile methods, and 86% of the respondents had software teams in their organization that had adopted agile practices or principles. Agile practices and principles are not only limited to software development, and they can be used in other areas as well [Digital.ai, 2021].

4.1. Scrum

Scrum is the most popular approach to agile. According to a recent survey by Digital.ai, 66% of the respondents reported following Scrum, while an additional 15% of the respondents reported following derivations of Scrum [Digital.ai, 2021]. Scrum is a framework that is often used for software development projects. It helps teams, organizations, and individuals to solve complex problems and create value. It is a lightweight, iterative,

and incremental framework that provides the core practices of the Scrum theory, while also serving as a container for other practices that are compatible with Scrum practices [Schwaber and Sutherland, 2020].

Scrum is based on lean thinking, which is about reducing waste and focusing on the essentials, and empiricism, which asserts that knowledge comes from experience and decision-making is based on observations [Schwaber and Sutherland, 2020].

Scrum can be implemented only partly, but the result is not Scrum [Schwaber and Sutherland, 2020]. Derivations of Scrum are, however, quite popular. For instance, ScrumBan combines Scrum and Kanban is the second most popular agile approach according to Digital.ai [2021]. Scrum practices have also been combined with Extreme Programming (XP), and this combination has become more widely used than traditional XP [Digital.ai 2021].

4.1.1. Scrum teams

Scrum teams are usually small and nimble, but large enough to be able to make a significant effort in each Sprint. This is to enable better communication and productivity. Multiple Scrum teams that share the same product can be used if the team size grows too large. Scrum teams are also self-managing and cross-functional. Meaning that the team manages and performs their work and has all the necessary skills for it.

Scrum has three roles: *Developers* are responsible for creating the increments. *Product Owner* is a person who manages the Product Backlog and is responsible for maximizing the value of the product. *Scrum Master* serves as the leader of the team. They are responsible for ensuring that the agreed practices and processes are applied correctly. They provide guidance and communication between the team, the organization, and the product owner [Schwaber and Sutherland, 2020].

4.1.2. Scrum events

Scrum has four events: *Sprint Planning*, *Daily Scrum*, *Sprint Review*, and *Sprint Retrospective*. The events are contained in the main event, the *Sprint*. Sprints are fixed periods, usually 2-4 weeks, during which the other events take place. Sprints occur in a sequence; a new Sprint starts after the previous one has concluded. Scrum events are based on inspection, adaption, and transparency. In Scrum, decision-making is based on its artifacts. Artifacts need to be visible to the team doing the work and people receiving the work. Transparency is required for inspection, which is required for finding problems related to the progress or artifacts. If problems are found, adjustments must be made [Schwaber and Sutherland, 2020].

Sprint Planning event takes place at the start of each Sprint. Work to be done during the Sprint is decided in Sprint Planning. A Sprint Goal is defined for each Sprint. Based on the goal, work items are selected from the Product Backlog to be included in the Sprint.

Developers plan how the work will be done by turning the work items into Increments that meet the quality required for the project depending on how the Definition of Done has been defined. The Sprint Goal, the selected work items, and the plan for accomplishing them are called Sprint Backlog [Schwaber and Sutherland, 2020].

Sprint Review event is held at the end of each Sprint. During it, the outcome of the Sprint is presented to the key stakeholders and the progress is discussed. The purpose is to review the progress in the Sprint and decide how to proceed from it [Schwaber and Sutherland, 2020].

Sprint Retrospective is held at the end of each Sprint, and it concludes the Sprint. The purpose is to inspect the Sprint to identify ways to increase quality and effectiveness. During Sprint Retrospective, the team discusses what went well, what problems were encountered, and how they were solved. Then, improvements are identified and addressed in the following Sprint [Schwaber and Sutherland, 2020].

Daily Scrum is a regular team meeting, usually held every working day. The purpose of it is to inspect the current progress towards the Sprint Goal, adjust the Sprint Backlog, and produce a plan for the next workday. It improves communication among team members, helps identify problems, improves decision-making speed, and eliminates the need for other meetings [Schwaber and Sutherland, 2020].

4.2. Communication and Agile

Agile methodologies emphasize informal face-to-face communication. One of the reasons is the effectiveness and efficiency of face-to-face communication. As the sixth principle of Agile Manifesto describes: “The most efficient and effective method of conveying information to and within a development team is face-to-face conversation” [Beck et al., 2001]. Informal communication, on the other hand, is important because agile has less reliance on heavy documentation.

Agile software development requires adequate communication practices, but in distributed settings, sufficient communication tools are also required. According to Berczuk [2007], “agile is about people, but distributed agile requires good tools to help people communicate effectively over distances”. Lous and others [2018a] expand this idea by stressing the importance of tools supporting ad hoc practices and facilitating not only communication, but also collaboration, coordination, and awareness. Furthermore, customer involvement should be considered when selecting tools in agile projects. Communication tools and practices should be selected early in the project together with the customers to select the most suitable tools and increase the relationship with the customer [Ahmad et al., 2018].

In agile projects, face-to-face communication is the preferred way of communication as mentioned earlier. However, if face-to-face communication is not possible, other methods can be used. Face-to-face communication is not necessarily required for successful

communication between developers [Robinson, 2019]. Using multiple channels for communication is recommended. Video conferencing can be used for replacing face-to-face communication, but it should be supported by other rich media, such as screen sharing, for improved communication. Audio conferencing is not as effective as video conferencing, but it can be sufficient for unofficial meetings. Text chat might be effective for daily informal communication or asking questions. Email is often appropriate for more formal communication, such as customer communication [Ahmad et al., 2018].

4.3. Distributed Agile Software Development

Agile and lean practices have been used increasingly in distributed settings [Lous et al., 2017]. The recent COVID-19 pandemic has left its contribution to the trend, forcing many agile teams to distributed settings. According to the latest State of Agile survey, 89% of teams have been geographically distributed during the pandemic, and the trend will most likely continue post-COVID-19 [Digital.ai, 2021]. The combination of agile methods and DSD is often called *Distributed Agile Software Development (DASD)* or *Agile Global Software Development (AGSD)*, or different variations of them. However, this combination does not come without challenges. Popular agile methods such as Scrum or XP were not originally designed for distributed settings [Vallon et al., 2017]. Agile methods rely on informal processes while DSD relies on formal mechanisms [Ramesh et al., 2006]. Combining agile practices with distributed software development seems to lead to a contradiction. Traditional distributed software development requires heavy rules and formalism to enable coordination between long distances, while agile relies on face-to-face communication and collaboration, and favors small, collocated teams [Lous et al., 2017]. As a result, applying agile practices into distributed environments can introduce a variety of challenges [Ramesh et al., 2006]. There have been many studies focusing on identifying challenges in distributed agile. For instance, Lous and others [2017] conducted a systematic literature review, in which they identified 40 challenges related to Scrum practices in distributed environments, while also providing solutions and suggestions to them based on the literature.

When practitioners first started to apply agile practices to distributed projects, there was no common consensus on whether this “impossible marriage” would work [Kajko-Mattson et al., 2010]. The usage of agile practices in distributed environments has become quite common ever since, especially now with the pandemic [Digital.ai, 2021]. However, since many agile practices were not originally designed for distributed settings, they are often not suitable for distributed settings without modifications [Robinson, 2019]. As a result, agile practices must be modified before they can be used in distributed settings [Paasivaara et al., 2009; Lous et al., 2017]. For instance, practices that have been traditionally done with the help of face-to-face communication (e.g., Scrum daily stand-up meetings) can be modified to use videoconferencing instead. Modifying agile practices is

not a new phenomenon for practitioners. Inspection and adaption are some of the cornerstones of agile [Beck et al., 2001]. Agile methods such as Scrum provide flexibility, which allows them to be modified accordingly [Lous et al., 2017]. Agile methods such as Scrum are rarely used by the book, and often modified according to practitioners' needs [Diebold and Wagner, 2015]. There is no "Silver Bullet" for Scrum implementation, and multiple versions of Scrum exist [Lous et al., 2017]. Since agile practices are often flexible for modifications, the more important question is how to implement them successfully in distributed settings. Some approaches introduce heavier management practices to eliminate the challenges with the cost of reduced agility [Lous et al., 2017; Lous et al., 2018b]. However, there have been also studies focusing on the successful usage of agile practices in distributed settings. For instance, Lous and others [2018b] conducted a case study on a software development team that managed to successfully modify and use agile practices in distributed settings without reducing agility.

4.4. Agile practices and their benefits

Successful agile practices in distributed settings provide benefits similar to collocated settings. Agile methods such as Scrum provide new mechanisms for communication, social interaction, project management, and coordination that can be useful for distributed projects [Pries-Heje and Pries-Heje, 2011]. Agile forces teams and team members to communicate, which can help teams to overcome challenges related to communication [Hummel et al., 2013].

Agile can also provide important benefits for communication. The most popular agile practices include daily standup meetings, retrospectives, sprint/iteration planning, sprint/iteration reviews, and the use of Kanban or task board [Digital.ai, 2021]. These practices can provide direct benefits to communication. Daily standup meetings can increase awareness of the project status, enable quicker response to change, improve coordination, reduce cultural issues, and increase knowledge sharing [Ahmad et al., 2018]. They can also facilitate informal communication among team members [Hummel et al., 2013]. These benefits make daily meetings crucial for communication. Sprint planning meetings increase the team's awareness of the next iteration, help reduce misunderstandings and misinterpretations, increase mutual understanding and collaboration. Retrospectives can provide a way to assess teamwork, improve team practices, improve the transparency of the project, help stakeholders to understand the project standards, and help managers to supervise the project. The communication benefit of the use of task boards is that it conveys status information to stakeholders [Ahmad et al., 2018].

5. Case study

A case study was conducted to identify communication challenges and evaluate how they were managed by the teams. This chapter explains how the data was collected, shows background information about the student projects, and presents the results of the data collection for each project.

5.1. Context

In this case study, data is collected from student projects that were organized as part of a software project work course at Tampere University in Finland. During the course, teams of students are tasked to create new software applications or improve existing software applications for real-world clients. The course teaches students essential skills related to software projects. The project work course is normally organized twice a year during the fall and spring semesters. This case study focuses on the fall 2020 implementation of the course. The course was organized during the COVID-19 pandemic, which changed how students could interact during the project, and consequently provided an opportunity to study communication challenges in settings with limited access to face-to-face communication.

Teams were formed by the course personnel based on the students' preferences and skills. The course is required for BSc students that are majoring in computer sciences. Students could participate in the projects either as developers or project managers. Students that participate in the projects as developers are required to have completed at least the basic studies in computer sciences. They typically take the course during the 3rd year of their studies, and therefore, have at least basic skills and understanding of creating software applications. For some of these students, the course serves as the first experience working on a software project that has real-world clients. Students are expected to learn the essential skills related to software project work such as, participating as a group member in software projects, understanding the most common software development models, presenting project outcomes, creating documentation related to software projects, understanding ethical requirements related to project work, and learning to use main tools related to software projects. Students that participate in the course as project managers must have completed the course once as a developer. They are also required to have a basic theoretical understanding of project management practices through a course that must be taken beforehand. Project managers are normally MSc students with more experience than the students participating as developers. They are expected to take more responsibility for their team and project.

During the fall 2020 implementation of the course, there was a total of 10 teams, each with five or six team members. A total of 57 students committed to the course after the initial weeks. Seven of the teams had six team members and the three of the teams had

five team members. Two projects shared the same project manager, while the other project managers were only responsible for a single project. Each student was required to spend at least 115 hours (developers) or 161 hours (project manager) of work time on the project. The working hours are logged into a monitoring tool. The students were working only part-time on their project and might have had other courses and possibly a job. Most students were not familiar with each other before the course. In half of the teams, no one knew each other, while in the rest of the teams some team members were familiar with some of their team members.

Each team had its separate project distinct from others, and they did not collaborate. The projects lasted approximately 4-6 months. They began in September 2020, and most of them were complete by the end of January 2021, which was a loose deadline for the projects. Some of the projects required additional time and were completed in early 2021.

5.2. Data collection and analysis

At the end of the course, data was collected from the projects. There were three main methods of collecting data: final reports, personal reports, and project manager interviews. It could be also noted that since the author of this paper was one of the project managers, own observations might be considered as the fourth data collection method because the observations affected the results.

Final reports were the most crucial data source. Students in each project must write a final report at the end of their project. Most of the data used in this study were collected from these final reports. Normally in these reports, students write about the outcomes of their project, how it was implemented, and other important details. For this study, a new section was added to the report, in which students were asked to explain what communication tools and practices were used in the project, what communication challenges they encountered, and how they solved the communication challenges. These were open-ended questions. The reports represent the perspective of individual teams because usually, the entire team participated in the writing process. However, in some cases, it can be possible that the project manager wrote the whole report without asking for input from the team. Once the project reports were written, the course personnel then provided the section from each report to the author of this paper. Project-specific information was removed from the section by the course personnel and the author did not have access to the other sections of the reports. Also, it was made clear to the students that the section in the report was to be used for research purposes.

Personal reports were another method of gathering data from the projects. During the course, students must write two personal reports: One in the middle of the project and another after the project has ended. In these personal reports, students give answers to personal questions about the project and the course. For this study, an additional section

was added to the second personal report. The idea was to try to get answers to more personal questions and questions that students may not want to mention in the final reports, such as, whether cultural differences caused any communication problems, or whether there was enough communication during the project. Where final reports represent the team's perspective, the personal reports represent individual perspectives. A total of 56 students out of 57 students answered the personal report questions. To protect the identity of the participants, the course personnel provided the anonymized data from these sections to the author of this paper. The author of this paper only knew the team to which the answers belonged. Furthermore, the author did not have access to the answers in the other sections of the personal report. It was also mentioned that the answers would be used for research purposes.

Project manager interviews were the third method of gathering data from the projects. The author of this study contacted the project managers of each team and asked them about a possibility for an interview. Four out of nine project managers accepted the invitation, and an interview was scheduled for each project manager. The interviews took place in January and February of 2021. The interviews lasted between 45 minutes to 75 minutes. The interviews were semi-structured. During the interviews, the project managers were interviewed about their projects. The interviews started with a discussion about background information, which included a discussion about the team, project environment, and communication tools and practices. Then, the project managers were asked about communication challenges they encountered during the project and solutions to them. The main objective of the interviews was to discuss challenges that were not necessarily noticed by the project teams. During the interviews, challenges identified from the literature were discussed with the project managers to determine whether the challenge had been encountered by the team. Finally, the project managers were asked about the role of agile practices when dealing with communication problems.

The collected data from all the mentioned sources were then combined. The real names of the projects are not used in this paper. Instead, the projects were anonymized. The original projects had number identifiers from 1 to 10. For this paper, the projects were arranged in an arbitrary order, and they were given alphabetical identifiers (A, B, C, D, E, F, G, H, I, J). Team members are not mentioned by their names, instead, they are referred to as team members, developers, or project managers. Also, nothing critical regarding the projects is revealed in this paper.

5.3. Communication in the project teams

Different teams managed their communication differently. The project teams had to decide their communication practices and tools. As a result, some teams had better success with communication than others.

In the personal report, each student was asked whether there was enough communication in their team. The results are shown in Figure 1. Most students seemed to agree that there was enough communication in their teams. However, according to the results, lack of communication seemed to have been a more widely agreed problem in three projects, in which most of the team members agreed that there was a lack of communication. In two other projects, less than half of the team felt that there was a lack of communication, and in the rest of the projects, the whole team agreed that there was enough communication.

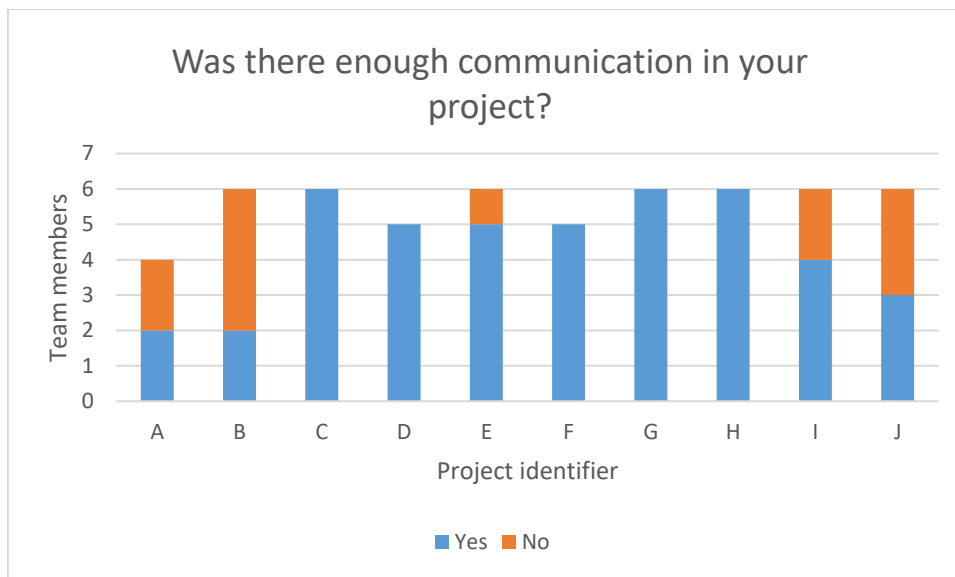


Figure 1: Students were asked about the sufficiency of communication during the project (Team A had five team members, but one answer was missing).

The students were also asked whether an increased amount of face-to-face communication would have helped the teams (Figure 2). For this question, 39 students answered “Yes”, and 17 students answered “No”. Only in three teams, most of the team did not find the need for increased face-to-face communication. It seems that even though most students and teams had enough communication, they still would have preferred more face-to-face communication.

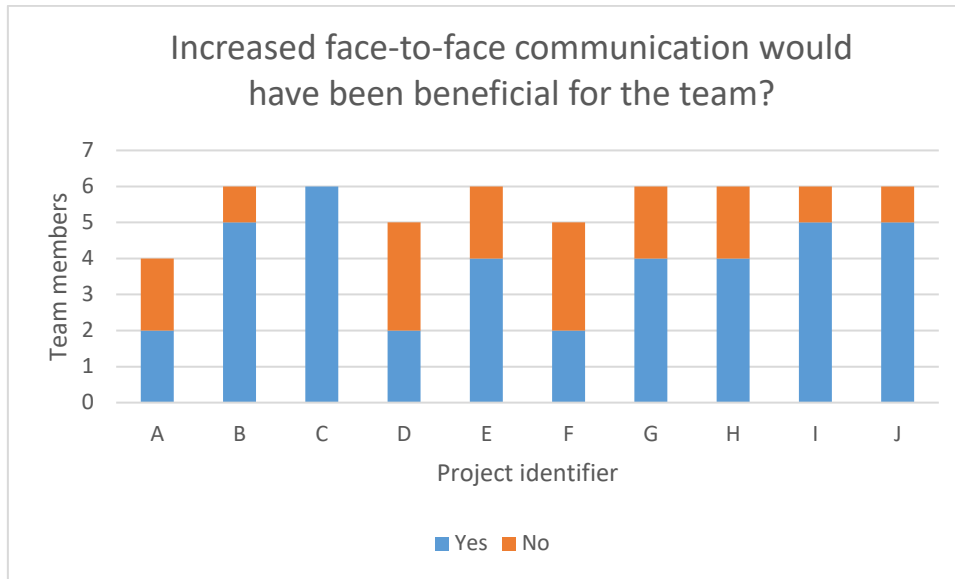


Figure 2: Students were asked whether an increased amount of face-to-face communication would have been beneficial for the team (Team A had five team members, but one answer was missing).

5.3.1. Communication tools

The project teams used different communication tools depending on their needs and prior experience using them. The main communication tools used by the project teams are shown in Table 1.

Each team seemed to have one primary communication tool that was used the most. The most common primary communication tool was Slack, which was used by four teams as a primary communication tool. Microsoft Teams was used by one team, and Discord was used by three teams as a primary communication tool. Discord offers similar functionality to Slack and Microsoft Teams, but it is mostly targeted at internet communities, while Slack and Microsoft Teams are more commonly used by organizations. Lastly, Telegram was the primary communication tool of the two teams. Telegram is an instant messaging software. It has fewer features compared to the other tools mentioned, but it can be a good choice for student projects if more advanced features are not needed. Prior experience or familiarity with the tools was most likely a major factor deciding the main communication tool, which would explain the usage of tools such as Discord or Telegram.

The student teams used various tools for online meetings. Zoom and Google Meet were tools used only for online meetings. Zoom was used by four teams (B, C, D, J) and Google Meet was used by three teams (H, I, and J). Also, Microsoft Teams (E and F) and Discord (C and I) were used for online meeting purposes. For team A, it is not entirely clear what tool they used for organizing meetings. Some teams used different tools for team meetings and client meetings. For instance, team C used Zoom for client meetings and Discord for team meetings, while team J used Zoom for team meetings and Google Meet for client meetings.

Some teams used instant messaging software, such as WhatsApp or Telegram as secondary communication tools. For instance, team E used WhatsApp at the beginning of the project as a backup communication tool because the team members were not familiar with their main communication tool, and WhatsApp was a more reliable tool for reaching the team when needed.

Task boards were used by all teams for project management and increasing task-related communication and task awareness. One team did not specifically mention the tool they used (Team I), one team used Jira (Team H), and the rest of the teams used Trello.

Traditional email communication was mentioned at least by seven teams. However, it was most likely used to some extent by all teams, but not all teams explicitly mentioned it. Email communication was mostly used by the teams for communication with their clients and project supervisors. Some teams also mentioned using it for distributing meeting notes or other documentation to the people involved with the project. Although, some teams also invited their clients and supervisors to their main communication platforms, such as Slack or Microsoft Teams, and might have handled the client communication through the communication platform, which might have reduced the need for email communication.

There are also other potential communication tools that the teams might have used but they are not explicitly mentioned. Only one team mentioned having used file-sharing tools, such as OneDrive. However, file sharing is often integrated into modern communication tools, which might explain why file-sharing tools were not mentioned. It could also be that the students did not particularly consider file-sharing tools as obvious communication tools. Also, the teams were required to use version control systems during the project. Modern version control tools such as GitHub often include communication features, but none of the teams specifically mentioned them.

	Slack	Telegram	Discord	Microsoft Teams	Zoom	Google Meet	WhatsApp
A		x					
B	x				x		
C		x	x		x		
D			x		x		
E	x			x			x
F		x		x			
G				x			
H			x			x	
I	x					x	x
J	x				x	x	

Table 1: Main communication tools used by the teams

5.3.2. Communication practices

Students communicated regularly with their team members throughout the project. The teams were able to decide their communication practices, although there were some required meetings that involved customers and project supervisors.

Regular online meetings were organized by all teams. Two teams reported having regular team meetings twice a week (teams B and C), while others reported having a single meeting per week. Many teams also organized occasional unofficial meetings when needed. Weekly team meetings served mostly as status meetings, in which the progress of the project was discussed by all team members. The weekly team meetings were also often the only time when all team members would be online at the same time, which made them crucial for communication about anything that needed to be addressed by all team members. Activities in these meetings that were mentioned include, discussing the status of tasks, discussing tasks that needed to be done next, discussing how the tasks should be divided among the team or assigned to certain team members, addressing problems that had been encountered, discussing technical details, planning upcoming meetings or other events, and updating task boards or other documentation. None of the teams mentioned following any strict agenda in their team meetings, but many teams had some sort of repeated pattern in their team meetings. Several teams seemed to loosely follow agenda similar to Daily Scrum in their weekly meetings. The duration of these meetings varied from team to team. Some teams mentioned them taking 30-60 minutes on average, while others reported that sometimes weekly meetings lasted for several hours.

In addition to regular weekly meetings, many teams (A, B, C, D, and E) mentioned having team coding sessions or workshops, in which the team would collaborate on tasks together or at least have a possibility to communicate with others while working on individual tasks. The teams that organized coding sessions reported them to be useful. However, many teams started organizing them rather late in their project or organized them very irregularly (e.g., when multiple team members had trouble with their tasks, or when the team was running out of time and had to make quick progress). Several project managers said that they could have organized more coding sessions to increase communication and productivity.

Outside of team meetings, the teams relied mostly on text-based asynchronous communication channels. Synchronous communication outside of regular team meetings was possible but often limited because students often had different schedules and were online at different times.

In addition to intra-team communication, the project teams also communicated regularly with their customers and supervisor. During the project, there was a total of five mandatory meetings with the customer and supervisor: a project plan meeting at the beginning of the project, three sprint review meetings during the project, and a final meeting

at the end of the project. These meetings had a predefined agenda in written form that was sent to the participants before the meeting. In addition to required meetings, additional meetings were also organized with customers by some teams. Team E had regular weekly meetings with their customers, while other teams had additional customer meetings when needed. However, most customer communication was managed through email and other text-based communication channels.

Meeting notes were required for the mandatory customer meetings. However, some teams reported using them for regular team meetings as well. It seems that meeting notes were useful for checking what had been discussed or decided in the previous meetings. For instance, in a situation when a team member was not able to attend a meeting.

Webcam usage was quite minimal among the teams. None of the teams used webcams extensively. Some teams used them occasionally (e.g., during certain customer meetings). One project manager also mentioned utilizing webcams for encouraging other team members to communicate by showing an example as an active communicator. Most teams, however, relied on audio communication and screen sharing during meetings.

Face-to-face meetings were rare due to COVID-19 restrictions. Only three teams mentioned having face-to-face meetings (teams B, E, and F). Two of the teams organized face-to-face meetings at the start of their projects to help team members to get to know each other better and to set up development environments for all team members. One team had a face-to-face meeting in the middle of the project, which was requested by their customer. Even though there were some face-to-face meetings, the teams had to rely on communication tools for communication and collaboration.

Communication practices improved over time in the project teams, as team members became more familiar with each other and found better ways to manage communication. This was often a result of challenges that the teams encountered. Although, in some cases, due to lack of information, it is not entirely clear whether certain practices were added later or whether they had been used from the very beginning. For instance, some teams attempted to keep their meetings as effective as possible, and they had certain practices for making them more effective (e.g., having troubleshooting sessions after team meetings so that time would not be wasted on it during the actual team meetings, or organizing separate meetings for frontend and backend developers). However, in some cases, it is not clear whether these practices were added later (e.g., once the team realized they were spending too much time on meetings), or whether they had used them from the beginning. The student teams were not exactly professional teams, and they had limited experience working on software projects. Consequently, they might have been lacking knowledge about successful or unsuccessful communication practices. Many of the practices used by the teams were likely reactive rather than planned.

5.4. Agile practices

The development process applied by the project teams can be generally considered agile or iterative. The teams did not closely follow any specific approaches or frameworks, such as Scrum. However, they did utilize some common practices that added agility to their projects. The students followed certain rules and guidelines defined by the course personnel. The guidelines defined some general practices, including some practices that can be considered agile. The guidelines and rules were mostly related to scheduling and managing the work, as well as organizing meetings with the customers and the supervisor. The teams had much freedom to decide their internal work practices.

Different teams managed their projects differently. There were two main roles: project managers and developers. Project managers in this case were more experienced students who already had at least some experience from software projects, as well as theoretical knowledge of managing projects. They led their teams and had certain extra responsibilities (e.g., writing a weekly report) compared to developers. Project managers also participated in the implementation process similar to the other developers. The teams could decide their management practices based on their needs. Some teams relied more on their project manager to handle managerial tasks, such as planning and assigning work. Some teams were more self-organizing and relied less on their project manager, which is the more agile way.

The work progress was divided into iterations. The initial iteration was reserved for gathering initial requirements, setting up the development environment, learning the basic tools required for the project, and creating a quick prototype. Apart from the initial iteration, the teams were required to divide their work progress at least into three main iterations, although, having more iterations was recommended. Given the schedule of the course, the length of each iteration was approximately 3-4 weeks for teams that chose to only have the required three main iterations. However, many teams chose to divide their work into shorter iterations (e.g., 2-week iterations), and consequently they had more iterations.

A review meeting took place at the end of each iteration. In the review meetings, the team would meet their customers and supervisor, and present the outcomes of the latest iteration and the plan for the next iteration.

The teams organized regular team meetings. In normal agile projects, daily meetings (e.g., Daily Scrum) that eliminate the need for other meetings are common. However, organizing them daily is often not possible in part-time projects, in which participants have different schedules and work a lower number of hours. Instead of daily meetings, the teams organized regular meetings once or twice a week. On average, the students were able to spend around 10 hours a week on the project, which roughly corresponds to the number of hours a full-time employee would spend in 1-2 days. The ratio between the

number of meetings and time spent on implementation was similar to full-time projects. Even though the teams could decide their practices, many teams reported using similar practices to typical agile meetings (e.g., Daily Scrum). Many teams utilized the “Three Questions” from Daily Scrum in some form, giving the team a chance to discuss their progress since the previous meeting, their plans for the next few days, and problems they encountered. However, the student teams had often much longer team meetings compared to typical agile meetings (e.g., 15-minute Daily Scrum), as some teams reported that their meetings would sometimes take several hours.

Task boards were used by the project teams to track and organize tasks. Both Kanban and Scrum boards were used, depending on the needs. The teams were required to divide their work progress into tasks. For monitoring purposes, the teams had to report the number of tasks in the backlog, active tasks, completed tasks, and rejected tasks in their weekly report. The teams had different practices when using task boards. In some teams, the task board served purely as a project manager’s tool; it was rarely used by team members other than the project manager, and it was not necessarily even shown during team meetings. In some other teams, the project manager was responsible for managing and making changes to the task board, while other team members used it for gaining information (e.g., checking information about tasks). Only one team mentioned that their developers made changes to their task board independently. However, this number could be higher because not every team reported their practices in detail.

Other popular agile practices were less commonly used by the project teams. None of the teams mentioned sprint planning when asked about their agile practices. It is likely that the teams simply let their project managers plan their sprints, and no separate meetings were arranged for it. Sprint retrospectives were only mentioned by a single team, and even in that case, the practice was applied later in the project. However, many teams mentioned that they tried to improve their practices over time based on the feedback from normal interaction with the team. It is possible that the teams did not find retrospectives useful enough to be applied in such projects, or perhaps they lacked knowledge of utilizing them.

5.5. Most significant challenges

Students were asked in the personal report to list the most significant challenges their team faced during the project. The answers were given in text format, due to which some of the answers were more general, such as “communication”, while others were more specific answers with more detailed reasoning. Each student mentioned at least one challenge. The challenges were mapped into more general categories. The results are shown in Figure 3. The chart gives a general idea of how frequent the communication-related challenges were compared to some of the other challenges encountered by the project teams.

Based on the chart, the most common challenges the teams encountered are related to skill. Challenges in this category were reported 25 times by the students, and they were encountered in all projects except one. Most skill-related challenges were related to unfamiliarity with tools or technologies used by the project teams. Some students also reported skill differences between team members and a steep learning curve as significant challenges. Many teams seemed to lack the knowledge or skills required for the project especially at the beginning of their project, which seemed to have led to problems such as having to spend much time on learning.

The second most common category was time management, which was reported by 14 students in eight projects. This category includes challenges related to managing time and scheduling work. It includes challenges such as leaving too much work to the end of the project, not having enough time to finish all features, or not working enough early in the project. Many projects had a slow start, which might have caused an unbalanced workload later in the project.

Implementation-related challenges were reported by 13 students. These challenges were related to problems with technology, code, or the development environment. Only three teams had the problems. These projects had significant technical problems.

Communication was reported by twelve students in six different teams. However, it was mentioned by the majority of the team only in two teams. In the other four teams, it was only mentioned by one or two team members. Based on the answers, it seems that only a few teams had significant communication challenges.

Challenges related to clients were reported by seven students in four different projects. These were mostly related to problems with requirements, and how they were understood and communicated.

Team configuration, mentioned by six students, was a significant challenge in two projects. These challenges were related to early problems in team configuration. Team members quitting the project, teams being merged due to the lack of team members and having no project manager at the beginning of the project were mentioned by the students.

The part-time environment, in which the projects operated, caused some challenges according to five students in four different teams. These students reported having difficulties balancing other studies or full-time work with their projects.

Five students in three projects reported significant challenges related to project management or planning activities. These students mentioned difficulties with managing the scope, insufficient planning, and project management. Finally, motivation and time zones were both mentioned by two students.

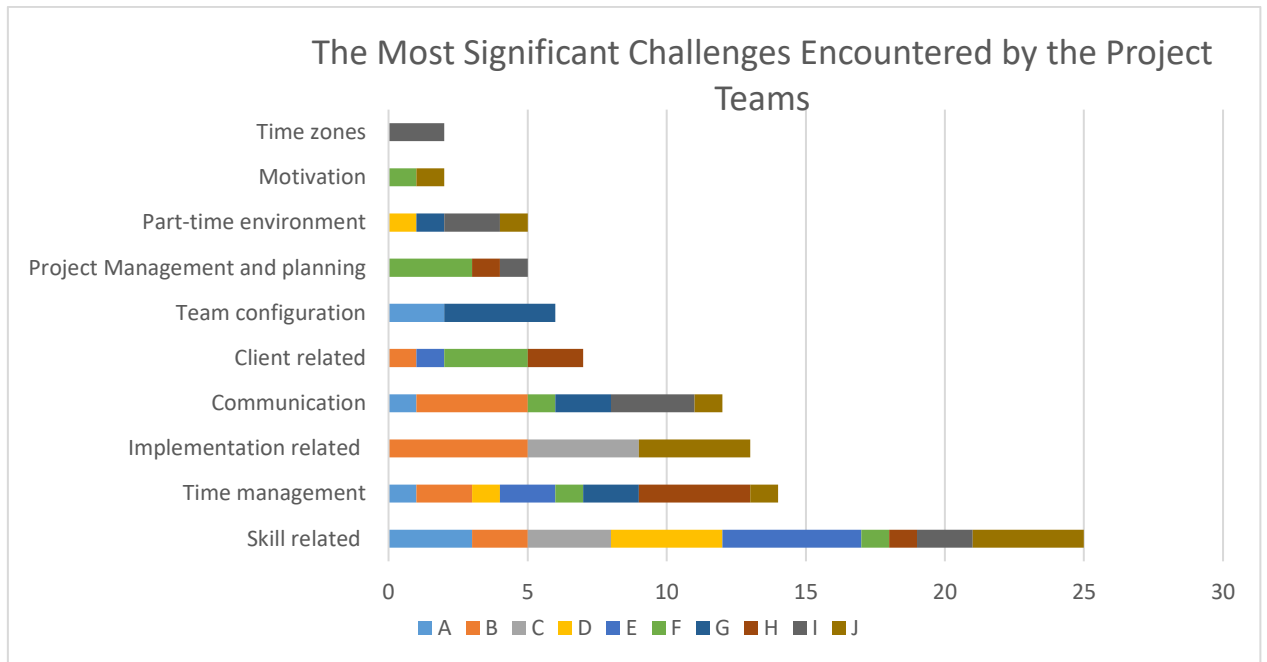


Figure 3: Challenge categories based on the challenges reported by the students. The chart shows the different challenge categories in the Y-axis and the number of answers related to the categories in the X-axis. The colors represent different teams.

5.6. Communication challenges

Communication challenges were identified based on the final reports and personal reports created by the students, as well as project manager interviews. This section addresses the challenges found in each project. Communication challenges for each team are separately explained in their subsections.

The challenges identified based on the material that was provided by the students are either treated as confirmed challenges or potential challenges. Due to the limited nature of the material, some of the challenges were only briefly mentioned in the material but they could not be fully confirmed as challenges. This might have been a case, for instance, if a single team member mentioned in their personal report a problem that was not confirmed by the team's final report or project manager, or any other team member. In some cases, the statements of different team members might have also contradicted each other. Furthermore, in some cases, it might have been even difficult to identify the main problem based on the statements of the team members due to limited provided information. In these types of situations, the challenges are treated as potential challenges because they cannot be fully confirmed. Most of the challenges identified based on the personal reports were treated as potential challenges. Confirmed challenges, on the other hand, were either in the final report, or they were mentioned by the project manager during an interview or multiple team members in their personal reports.

5.6.1. Project A

Communication challenges for Project A, are not entirely clear based on the material provided by the team. Three communication-related challenges were mentioned in the final report or reported by the team members, and several potential challenges were found based on personal reports. With half of the team answering yes, and the other half answering no, the team seemed to have mixed views on whether there was enough communication during the project. One of the team members mentioned that communication was a significant challenge for the team, while others mentioned the lack of technical skills, time management, and early problems with the project being the most significant challenges for the team. All team members agreed that the team's atmosphere was positive. Although one team member mentioned that it was better in the early stages of the project and that there was some frustration later in the project because the project did not go forward as planned.

The following communication challenges were reported:

Differences in schedules: In the final report, the team mentioned that availability of some team members was an issue. The team members had different schedules, which meant that they were often not available at the same time. According to the team, this caused communication difficulties such as delays in communication. The problem was also mentioned in the personal reports. **Solutions:** No solution was mentioned for this problem. However, one team member mentioned in their personal report that knowing the availability of other team members and being able to schedule more synchronous communication would have helped with the problem.

Passive team members: In the final report, it was mentioned that some team members did not always actively ask for help or share problems with the team. The team did not provide further details about this problem, and it was not mentioned in the personal reports either. However, it seems that the team was lacking communication at least outside of meetings. **Solutions:** In the final report it was mentioned that asking for help and sharing problems was encouraged, but it did not work very well. Meetings were most likely crucial for the team's communication and for solving any problems related to it.

Lack of task-related communication: The team had some problems with task-related communication. This problem was not mentioned in the team's final report, but it was mentioned by several team members. The team used Trello as their task board, and the team's project manager was responsible for managing it. One of the developers who worked closely with the project manager mentioned that Trello was a good way of keeping tasks updated. However, according to several other developers, tasks were not always detailed enough to know exactly what needed to be done, and assumptions had to be made. In addition to this, one developer mentioned that they did not always get clear

answers to their questions about tasks. It could be possible that the developers that interacted more often with the project manager had more task-related information available to them compared to those who had to rely more heavily on the task board, which did not always have all the necessary information. **Solutions:** None.

Potential challenges:

Early communication problems: The team mentioned that they did not have a project manager at the beginning, one of the team members quit the course early in the project and there was a lot of uncertainty because the team lacked experience. These team configuration problems in the beginning potentially affected early communication negatively, but the effects on communication are not clear based on the material.

No face-to-face meetings: The team did not have any face-to-face meetings. One team member mentioned that setting up the development environment caused problems for some team members, and it would have been easier to get started with the project if there were face-to-face meetings at the beginning of the project. However, this was only mentioned by a single team member and might not have been a significant challenge.

Language differences: The team had to use English as their main language for the project, but it was not the primary language for any of the team members. One team member mentioned in their personal report that this caused minor misunderstandings, while others did not report it as a problem. Most team members were native Finnish speakers and used Finnish when communicating with each other. One team member also mentioned that communication between the Finnish speakers was more relaxed, compared to situations where English had to be used.

Insufficient client communication: One team member mentioned that the communication with the client could have been more active, which might suggest insufficient client communication. However, the other team members did not report client communication as a problem.

5.6.2. Project B

Communication was a significant challenge for Project B. The team reported several problems related to communication in their final report, while two potential challenges were found based on personal reports. Most of the team members agreed that there could have been more communication during the project, and that access to face-to-face communication would have helped with the problem. The team's atmosphere was quite positive, although, one team member mentioned that it would have been more active if there was more face-to-face collaboration.

The following communication challenges were reported:

Differences in schedules: The team reported often having different schedules and working only part-time on the project while having other courses or a job. The team members were not always able to attend all meetings or check the team's main communication channel regularly, due to which they could not be able to answer important messages, communicate their progress to others when needed, or they might have missed crucial information related to their tasks. These problems lead to communication delays that were up to one week in the worst cases. **Solutions:** The team had a loose rule of informing others if they could not be able to attend a meeting. The team also tried to use a loose 24-hour rule, which meant that everyone was expected to check Slack at least once every day. These rules increased communication frequency, but they did not fully solve the problems, because not everyone followed them. The team also wrote short meeting notes if someone was not able to attend meetings. To avoid having to decide meeting times separately for each week, all internal team meetings were organized at the same time every week.

Unfamiliarity with communication tools: The team mentioned that some team members were not familiar with Slack before the project and that none of the team members used it for any other purposes than their project. Slack had to be opened specifically with the project in mind, which decreased the team's activity on Slack compared to using it for other purposes as well. The team also mentioned that their Slack was not always well organized. Some team members mentioned that it was sometimes difficult to find correct information because their Slack was divided into multiple chat channels, each with different purposes. It was also mentioned that at the beginning of the project, some of the team members were unaware of the different chat channels and might have missed crucial information. **Solutions:** The project manager mentioned that with time, the team became more familiar with Slack and learned how to better use it. The team mentioned that they could have switched to other platforms but, the team wanted to learn Slack. Adding other more familiar communication tools as secondary communication tools were also suggested, but according to the team, it would have made their communication more complex. However, they also acknowledged that relying on a single tool meant that it was often the only way of reaching the team.

Differences in motivation or goals: The team reported communication problems that suggest differences in motivation or goals. Some team members did not follow the team's rules. The team reported that occasionally some team members were unable to attend meetings, but they also failed to inform the team about it before the meeting or even after the meeting in some cases. The team had to often spend extra time at the beginning of the meetings figuring out who was going to attend the meeting. The project manager men-

tioned that they often had to contact the team members that were unable to attend a meeting to receive updates on their task progress because they also failed to communicate their progress outside of meetings. However, receiving a reply to the message would sometimes take several days, according to the project manager. The root causes for these problems are most likely motivation and having different goals, in addition to schedule differences as mentioned before. For some team members, the goal might have been to pass the course, which did not necessarily require attending all the meetings or following the rules the team had set. **Solutions:** None.

Minor technical problems: The team had some technical problems during meetings. According to the team, technical problems caused usually only minor disturbances. However, on one occasion, technical issues caused a meeting to take twice as long as it normally would have taken. **Solutions:** None.

Skill differences: The team reported some skill differences between team members. It caused difficulties understanding and explaining technical concepts, which lead to misunderstandings. However, according to the team, this was not a major problem. **Solutions:** None.

Insufficient client communication: According to the team, client communication was good enough for a student project, but it could have been better. Sometimes there was miscommunication and misunderstandings between the team and the customer. and the team had to make some assumptions occasionally. **Solutions:** The team mentioned that regular client meetings helped clarify misunderstandings and assumptions.

No webcams during meetings: The team decided not to use webcams during meetings. According to the team, this caused difficulties understanding how the team members reacted to certain things, knowing who was speaking, or knowing whether someone was paying attention or not. **Solutions:** None. It was the team's own decision not to use webcams.

Early communication problems: The team mentioned that communication was often very task-focused, achieving open discussion was difficult, and that the project manager had to often lead conversations. According to the project manager, this was a problem especially in the early stages of the project. The team mentioned that the problem might have been that the team members were not very familiar with each other before the project and that there was not much informal communication or team building at the beginning of the project, which might have led to a lack of cohesion. The remote nature of the project might have also affected it. According to one team member, "the atmosphere could have been livelier without COVID-19". **Solutions:** Communication improved over time.

Potential challenges

Inefficient use of task board: The team mentioned having a task board. Only the project manager used it, even though all team members had access to it. The task board was not shown during meetings either. Most task-related communication had to go through the project manager, instead of team members directly making changes to the task board and getting the information they needed from it. The team reported having two meetings every week, which allowed the team to discuss the recent progress. This most likely reduced criticality of the task board when it comes to awareness. However, some team members reported that assumptions about tasks had to be made occasionally. Not using the task board optimally might have not been a problem in itself. However, increasing the team's involvement with the task board could have potentially increased the team's communication effectiveness.

5.6.3. Project C

Communication was not a major problem in Project C. According to the project manager, communication problems were usually quite insignificant compared to other challenges. However, some communication challenges were reported in the final report and project manager interview. Personal reports did not reveal any new challenges. Everyone in the team agreed that their communication was sufficient during the project. They also agreed that more face-to-face communication would have helped the team, especially in the beginning. The team's atmosphere was positive, although it was also mentioned that it was "a bit stiff" at the beginning of the project when the team started working together. There were no significant conflicts between the team members.

The following communication challenges were reported:

Limited access to development tools: The most significant communication-related challenge for the team was the limited access to the equipment required for the project. The team used certain physical equipment that was required for building their software, but the problem was that not everyone in the team had access to them. The team mentioned that discussing the tools was sometimes very difficult (e.g., trying to explain how the tools worked), especially with those who did not have access to them. **Solution:** The team was able to demonstrate the software and the tools by using screen sharing and videos. According to the team, the situation would have been much easier if the team was able to have face-to-face sessions.

Lack of task-related communication: The team had some problems related to lack of task awareness. The project manager mentioned that the team members often worked on separate components, but they had only a little knowledge about what others were doing or how the components were related to each other. The project manager also mentioned that sometimes team members worked on the same components, but they had a

different understanding of its state because there was not enough communication between the team members. **Solution:** The team addressed awareness problems during meetings. The project manager mentioned drawing diagrams to get everyone on the same page if something was not clear. The team also used a task board, which was shown during meetings. However, it served mostly as the project manager's tool, and the team members did not use it. Encouraging team members to use the task board more actively would have most likely helped increase awareness and possibly mitigate some of the challenges related to awareness.

Skill differences: The team had some skill differences, which caused mostly minor misunderstandings. The project manager mentioned that it was sometimes difficult to explain technical concepts to less experienced developers. These problems were also confirmed by some of the developers. One of the developers said that it was sometimes difficult to understand task instructions without deeper knowledge in the subject area. It was also mentioned that reliance on screen sharing and audio communication during meetings made it even more difficult to explain things compared to normal face-to-face scenarios. The project manager also said that skill differences caused some team members to dominate communication. **Solutions:** The team did not mention any solutions to this problem.

Differences in schedules: The team experienced occasional difficulties finding common meeting times due to differences in schedules between the team members. **Solution:** Scheduling meetings well in advance.

Inefficient meetings: The team mentioned that communication during meetings was difficult at the beginning of the project because the team members did not know each other very well. This problem might also be related to a lack of agenda. One team member mentioned that in the beginning a lot of time was wasted on meetings because there was no clear agenda. **Solutions:** Agenda was added to all meetings. According to the project manager, adding agenda to all meetings made them clearer and more efficient.

Unfamiliarity with communication tools: The team mentioned that they encountered some usability problems related to screen sharing with the communication tools that they used. These problems were mostly caused by the lack of experience using the tools. **Solutions:** Improved over time.

Informal communication difficulties: The team tried to have informal communication during meetings, but they noticed that it was difficult to have one-on-one conversations during meetings because talking with one person meant that everyone in the meeting would hear the conversations, which made it more challenging to build connections. **Solutions:** None mentioned.

5.6.4. Project D

Project team D did not report any major challenges related to communication. Only one challenge was mentioned in the final report, and one potential challenge was discovered based on the personal reports. All team members agreed that there was enough communication during the project. They were also satisfied with the remote working arrangements and did not see a need for face-to-face communication. The team's atmosphere was positive, although a little reserved according to one team member. There were no major conflicts either.

The following communication challenges were reported:

Differences in schedules: The team reported a lack of options for scheduling meetings as a communication challenge. It was the only communication-related challenge they reported in the final report. This challenge was most likely a result of differences in schedules because many of the team members reported the time management and part-time nature of the project as significant challenges. **Solution:** The team solved this by scheduling meetings well in advance. They also mentioned setting up successful communication channels that made communication easy even without face-to-face communication.

Potential challenges:

Early communication problems: Two team members reported in their personal report communication problems that suggest early communication difficulties. One team member said that "it took some time before all learned to communicate things with proper depth". Another team member mentioned that "Nothing was discussed early because no one knew what they were doing", which most likely refers to lack of experience, which was a significant challenge as reported by the team. These problems were not mentioned in the final report. Due to lack of details, it is difficult to determine whether early communication was a real problem for the team, or what potentially caused it.

5.6.5. Project E

Team E was able to manage their communication without major problems. According to the project manager, the team did not encounter any major communication challenges, only small issues. A total of seven challenges were discovered based on the material provided by the team. All team members agreed that there was enough communication, although they also agreed that face-to-face communication would have helped. The atmosphere of the team was very positive, especially towards the end of the project, and there were no conflicts between team members.

The following communication challenges were reported:

Early communication problems: According to the project manager, the most significant communication-related challenge for the team was “breaking the ice” early in the project, because the team members were not familiar with each other. The project manager mentioned that communication was slow at the beginning of the project. It was mentioned that the team was often passive during conversations and that it was difficult to get reactions from the team. The project manager had to often take an active role in communication. **Solutions:** After a month or two, the team members started to become more familiar with each other, which made communication easier. According to the project manager, there was a night and day difference in how the communication functioned in the early stages compared to the later stages of the project. The project manager tried to take an active role in communication and encourage others to communicate by showing an example as an active communicator. The project manager mentioned that the team was not interested in using web cameras, but the project manager used it sometimes to show an example as an active communicator.

Unfamiliarity with communication tools: The team did not have prior experience using the main communication tools they chose for the project. It took the team some time to get familiar with them. **Solutions:** The team used a secondary communication tool at the beginning of the project. It was a more reliable tool for reaching the team because the team was already familiar with it and many team members used it for other purposes as well. The team stopped using the secondary tool later in the project when they became more familiar with their main communication tool.

Inefficient use of task board: The team had a task board, but only the project manager used it early in the project. One of the reasons for this was that the team members were not familiar with task boards. However, the team noticed it being a bad practice because it negatively affected the team’s communication and task awareness. **Solutions:** The project manager started showing the task board during meetings. By the end of the project, the team members were already familiar with the tool and were able to make changes to it independently. The project manager mentioned that the change greatly improved the team’s task awareness.

Technical problems: The team had occasional technical problems when using communication tools, such as having a bad connection, computer malfunctioning, or microphone not working. **Solutions:** None.

Differences in schedules: The team mentioned that it was sometimes difficult to decide meeting times because everyone had different schedules. One team member, in particular, did not have enough time for the project and could not always communicate properly, which caused long delays in answering messages. **Solutions:** The team used the

same meeting times every week, which eliminated the need of having to negotiate meetings times separately for each week. The team was aware of everyone's availability and could plan their activities in such a way that the inability to participate in the project did not cause any problems.

Skill differences: Lack of skills at the beginning of the project was a significant problem for the team. According to the project manager, some team members were more experienced than others, which sometimes caused difficulties understanding technical language and explaining technical concepts to less experienced team members. **Solutions:** The project manager mentioned that it did not cause real problems because each team member had their own tasks and responsibilities that fit their skills, which meant that they did not need to know all the technical details about others' tasks.

No face-to-face communication: One team member reported that it was sometimes difficult to explain things to the team by only using audio and online tools, as opposed to face-to-face scenarios in which it would be possible to physically demonstrate or show something. In an interview, the project manager agreed that it would have been easier to demonstrate things on-site and that there could have potentially been more spontaneous learning when team members would physically show each other how to do something in face-to-face scenarios. **Solutions:** However, the project manager did not see the lack of face-to-face encounters as a problem, because the team actively utilized screen sharing and other online tools that achieved the same results. According to the project manager, it was only a matter of getting used to the tools.

5.6.6. Project F

Team 6 reported four communication challenges, and one potential challenge was discovered based on the personal reports. Overall, the team managed their communication quite well, and there were no major challenges related to communication. It was mentioned that communication worked well since the beginning, and the team did not have to consider any specific strategies or rules for communication. However, the team also mentioned that some of the challenges were only noticed at the end of the project when the team had to report about them and there was more discussion about them. The entire team agreed that there was enough communication during the project, and most team members did not see the need for more face-to-face communication.

The following communication challenges were reported:

Insufficient client communication: The team encountered some challenges related to client communication. The team reported that they had to often make assumptions because the client could not always reply to messages in a reasonable time. The project manager further mentioned that initially the client provided too broad details about the

project and its goal and gave much freedom to the team to decide the implemented features. All this required much communication, which was not always available to the team. The team also had multiple client representatives, which made decision-making more difficult according to the team. **Solutions:** The team did not have any specific solutions to the client-related communication problem. The project manager mentioned that the problem was discussed at the final meeting, in which both the client representatives and the team agreed that there should have been more communication between the two parties. However, the team also mentioned that the iterative nature of the project allowed the team to have regular meetings with the client throughout the project, which eventually helped the team to clarify the misunderstandings. The team felt that these meetings were crucial for the project because the client was otherwise a passive communicator.

Lack of task-related communication: Task-related communication was another challenge for the team. According to the project manager, some team members did not communicate enough about their progress. Instead of reporting to the team when their tasks were completed, some team members waited until the next progress meeting to inform others about their progress. Furthermore, the team used Trello, but it was only used by the project manager, which meant that team members could not have selected new tasks for themselves without the project manager. Having only a single meeting every week meant that some team members might have finished their tasks early and then had to wait several days until getting new tasks. This problem was also mentioned by one team member in their personal report. **Solutions:** To solve this problem, the team wanted to add a second weekly meeting, but it was already too late in the project for such changes. The project manager also tried to encourage the team to use Trello, but it did not solve the problem. The project manager felt that the team failed to properly use Trello and that the team should have spent more time on it at the beginning of the project.

Keeping track of multiple communication channels: The team also mentioned in their final report that it was sometimes difficult to keep track of all communication channels. The team used both Microsoft Teams and Telegram, which meant that both had to be checked regularly to stay in the communication loop. **Solutions:** None mentioned.

Differences in schedules: The team had different schedules, which made organizing meetings sometimes difficult. Also, some team members were not able to always check Telegram, the team's main communication channel, regularly. However, no significant delays in communication were reported due to these problems. **Solutions:** The team organized weekly meetings at the same time every week to not have to negotiate meeting times separately for every week.

Potential challenges:

Skill differences: The project manager mentioned that sometimes it was difficult to explain technical concepts to less experienced team members. However, none of the team members mentioned this problem in their personal reports.

5.6.7. Project G

Team G was able to complete their project without major communication problems. Two communication challenges were reported in the team's final report, and one potential challenge was discovered based on personal reports. Everyone in the team agreed that their communication was sufficient during the project. Most team members also agreed that more face-to-face communication would have been beneficial for the team.

The following communication challenges were reported:

Differences in schedules: The team reported occasionally having long delays in communication, which made it difficult to get answers or help in real-time from team members. It was mentioned to be especially problematic in situations when there were dependencies between tasks of multiple team members, which often led to having to wait for a reply before being able to continue work. This challenge was mostly a result of having differences in schedules and working times. **Solutions:** None mentioned.

Early communication difficulties: The team reported difficulties with communication early in the project. In the final report, the team mentioned that early in the project meeting absences were often not reported beforehand and that it was not clear who was going to attend meetings or continue with the project. Furthermore, one team member mentioned in their personal report that everyone in the team focused only on their tasks and that everyone was passive on Microsoft Teams, which was their main communication channel. These problems are most likely a result of early problems with team configuration. The team mentioned that two original team members quit the project early and the team had to be merged with another team. **Solutions:** The team noticed these problems quite early and addressed the problems during a meeting, after which communication started improving. However, the project manager mentioned that it took a long time until the team learned to communicate properly.

Potential challenges:

Skill differences: The team had also significant skill differences, which may have caused communication difficulties. One team member reported that skill differences caused misunderstandings and confusion. However, several other team members acknowledged the skill differences, but they did not see them as a problem.

5.6.8. Project H

Communication was not a significant challenge for Team H, and the team reported only minor communication challenges. Overall, three communication challenges were reported in the final report, and three potential challenges were discovered based on the material. All team members agreed that there was enough communication. Most team members also agreed that more face-to-face communication would have been beneficial for the team.

The following communication challenges were reported:

No webcams during meetings: The team did not use webcams during meetings, which made communication more difficult (e.g., difficult to know when someone wanted to speak or difficult to understand the reactions of team members. In addition to this, they also mentioned that during meetings, one team member was sometimes limited to only communicating by using text chat. **Solutions:** None mentioned. The team mentioned that using web cameras during meetings could have helped with communication and increased team cohesion, but it was their decision not to use them.

Language differences and personal differences: According to the team, English was used as the primary communication language, but not all the team members were proficient at it. Language differences caused some misunderstandings. This was also confirmed in the personal reports by several team members. One team member reported that some team members were quite passive in communication, which was most likely due to language differences. **Solutions:** None mentioned.

Minor technical problems: The team reported that their communication tools worked well in general, but some minor technical problems that affected communication (e.g., microphone not working) were encountered. **Solutions:** None.

Potential challenges:

Miscommunication with the client: In the personal reports, some team members described communication problems related to client communication that was not mentioned in the final report. It was mentioned that assumptions had to be often made because the project requirements were not always clear to the team. However, it was also mentioned that the team usually noticed when assumptions were made and managed to communicate about them. One team member pointed out that the team would have probably achieved a better understanding of the client's wishes if they could have had face-to-face meetings with the client. The team also had regular meetings with the client, which most likely helped resolve assumptions. One team member also mentioned that the client overpromised what they could provide for the team, which caused some issues. These problems suggest miscommunication with the client.

Skill differences: Two team members reported in their personal reports that there were differences in skill levels between team members. Those differences may have caused misunderstandings.

Task-related communication problems: The team reported that they had a task board, but it was not used until the later stages of the project. The team mentioned that it could have been used earlier to increase awareness and avoid miscommunication, which suggests that the team might have had some task-related communication problems. However, this problem was only briefly mentioned.

5.6.9. Project I

Team I reported one communication challenge in their final report, while the personal reports revealed at least two more. The team seemed to have mixed opinions on how communication was handled in the team. Three team members considered communication as a significant problem, while others did not consider it as a problem at all. In addition to this, two of the team members mentioned the team's atmosphere being negative towards the end due to problems that the team had. Interestingly, the rest of the team reported that the team's atmosphere was very positive. The team seemed to have significant problems, but they are not entirely clear based on the material provided by the team.

The following communication challenges were reported:

Time zone differences: The team reported communication problems caused by time zone differences, which was the only challenge reported in the final report. The team had members from multiple different countries in different time zones. The team reported having at least a four-hour difference in time zones between the team members. This temporal distance caused challenges at least in scheduling meetings. As one of the members said: "We used to have meetings at 6 pm Finnish time which is 10 pm for me. Generally, the meetings were 1-2 hours. So sometimes it created problems for me." It seems that the team had problems finding a time slot for meetings that was suitable for everyone. Some members also mention that the communication was more difficult due to the time zone difference, but they did not describe the problem further. **Solutions:** The team mentioned that they were using their communication channels actively but did not specifically mention how they handled the problem.

Potential challenges:

Language differences: In addition to time zone differences, there is also evidence for communication challenges caused by language differences. The team had members with varying skill sets from four different nationalities. They mention trying to capitalize on diversity. However, the diversity seemed to have also caused some issues. Several

team members mention having difficulties understanding the accent of other team members, which caused misunderstandings. However, for some, this was only a problem in the beginning before being familiar with the team. Overall, it seems that the personal differences did not cause major problems, and one of the team members even mentioned that the diversity made the project more interesting.

Unsuccessful communication practices: Two team members reported communication as a significant problem for the team and considered the team's atmosphere negative towards the end, which considerably differs from other team members' answers. They also mentioned that tasks and project goals were not always communicated clearly enough, especially later in the project. None of the other team members mentioned any task-related problems. The team also did not mention anything about how they managed tasks in general. Furthermore, the two team members also mentioned problems related to meeting practices. According to them, weekly meeting times were sometimes announced only a couple of hours before the meeting, and not everyone was able to attend them on such short notice. Sometimes even the project manager forgot the meeting, and the team had to wait an hour before canceling the meeting. This suggests that the team did not have regular meeting times, and instead meetings had to be separately decided for each week. None of the other team members mentioned these problems either. Since none of the other team members mentioned these problems, it could be that these two team members were out of the communication loop. They might not have had access to all the communication channels that the other team members used. Another explanation is that the other team members were aware of the problems but forgot to or did not want to report them. In any case, the team seemed to have some unsuccessful communication practices that negatively affected the team's performance.

5.6.10. Project J

Team J encountered a total of seven communication challenges. These were discovered based on the team's final report and project manager interview. According to the team, their communication could have been better, but it was sufficient for completing the project. The project manager said that in the real world, their communication practices would have potentially caused major problems. The team had mixed opinions about whether there was enough communication. However, the team seemed to mostly agree that more face-to-face communication would have been beneficial. The team's atmosphere was positive, and there were no significant conflicts between the team members.

The following communication challenges were reported:

Skill differences: The team experienced minor communication problems related to skill differences between team members. According to the project manager, the project was quite difficult for the team. Most team members had no prior experience using the tools and technologies required for the project. The project manager explained that misunderstandings were common and that it was sometimes difficult to explain technical concepts to less experienced developers. Furthermore, the skill differences caused the more experienced developers to dominate communication. The project manager explained that the three most experienced developers handled 85-90% of the team's communication, while others were more passive in communication. **Solutions:** According to the project manager, it was expected that the more experienced developers dominated communication, and it did not cause any larger problems. The team made sure to involve others in communication as much as possible by asking their opinions frequently.

Differences in schedules: One of the major problems related to communication was the usage of Slack. The team reported that sometimes the discussion on Slack was very passive. Some team members would not use it very often, which caused long delays in communication. The main reason for this was differences in schedules. Some team members mentioned in their personal reports that they were sometimes simply too busy to open Slack regularly. It was also mentioned by one team member that personal differences may have caused some passivity in Slack communication. **Solutions:** The project manager encouraged the team to install Slack on phones and enable notifications, but it did not solve the problem. The team also tried using a 24-hour rule, which meant that every team member was expected to check Slack at least once every day on weekdays. According to the project manager, the rule helped a little bit, but it did not solve the problem.

Unfamiliarity with Slack: The team used Slack as their main communication tool, but it was new to most team members. The project manager further explained in an interview that the team did not have prior experience using Slack and that they did not use it for other purposes than the project itself, which meant that it had to be opened specifically for the project and thus it was easy to forget to open it regularly. **Solutions:** The project manager suggested using other communication tools instead of Slack, but the team did not support the idea, and the team kept using Slack.

Motivation or differences in goals: Motivation or differences in goals was another factor that caused long delays in communication. As the project manager said, "the goal was to complete the course rather than getting to know each other". Some team members did not always follow the team's rules or communicate properly. For instance, the project manager mentioned that some team members simply stopped working and communicating when they finished their part. Apart from potentially receiving a lower grade, there

were no real consequences for doing not following the team's rules or communicating properly. **Solutions:** None mentioned.

Unfamiliarity with visualization tools: The project manager mentioned that it was often difficult to visualize things without using physical tools the team was familiar with, such as whiteboards or paper. The team tried using software tools that provided similar functionality, but they did not have experience using them, which made using them difficult. **Solutions:** With time, the team learned to use the software tools adequately for their purposes.

Limited communication methods: The team decided not to use webcams during meetings, which caused some communication problems. They used webcams only for the first meeting, after which they decided not to continue using them due to a personal preference. However, the project manager noticed that it was often difficult to read the situation during meetings (e.g., knowing whether someone was paying attention, or whether the team agreed or not). Also, one team member sometimes preferred communicating only by using text chat during meetings instead of audio communication, which occasionally made communication more difficult. **Solutions:** No solutions were mentioned for these problems.

Minor technical problems: The project manager mentioned that the team had minor technical problems occasionally that may have disrupted communication during meetings. **Solutions:** None.

6. Results

This chapter summarizes the results and addresses the research questions. The identified communication challenges are presented in Tables 2–6. Each table shows challenges, their consequences, potential solutions, and affected teams (teams that were potentially affected by the challenge are shown inside parenthesis) for one challenge category. The identified challenges were categorized based on adaptations of the categories identified by Alzoubi and others [2016]. Distance-based challenges can be found in Table 2, and they are further discussed in section 6.1. Challenges related to organizational factors are in Table 3, and they are discussed in section 6.2. Only communication tool-related challenges were identified for this category. Project management processes were likely a factor for some of the difficulties the teams encountered, but they could not be easily identified as such. Also, the student projects were not affected by organizational culture. For these reasons, Table 3 is only for communication tools. Challenges related to customer communication can be found in Table 4, and they are further discussed in section 6.3. Challenges related to team communication can be found in Table 5, and they are further discussed in section 6.6. The team communication category was adapted from the team configuration category of the original paper, and it contains communication challenges related to communication and coordination between team members. This change was made because the teams were not affected by challenges related to team size and the number of teams. After all, each project had only one team and the teams were small. Finally, challenges related to human factors can be found in Table 6, and they are further discussed in section 6.7.

6.1. Communication challenges related to distance factors

Distance factors caused some challenges for the teams (Table 2). The geographical distance was not a significant challenge for the teams, even though none of the teams were able to organize regular face-to-face meetings due to COVID-19 restrictions. Two teams reported that informal communication was more difficult without face-to-face encounters, which caused difficulties for the team members to become familiar with each other. One team mentioned difficulties demonstrating things without face-to-face communication. Furthermore, one team lacked certain physical development tools, which caused some communication difficulties, but the team was able to solve the problem with online communication tools.

Temporal distance on the other hand was one of the most common challenges for the teams, and it was identified almost in every project. The part-time nature of the project is the most likely cause for it. As one team mentioned in their final report, “the main problem was: agile projects need frequent communication. Meanwhile, students do not like to be bothered frequently”. Differences in schedules were a common challenge that almost

all teams encountered, as it caused problems such as delays in communication or difficulties finding common meeting times. Differences in time zones caused similar problems, but it was only reported by a single team, as most teams had their members located in Finland. Even though the temporal distance was a common challenge, most of the problems were quite easy to solve, and the teams managed to handle them in most cases.

Category	Cause	Consequence	Solutions	Teams Affected
Geographical distance	No face-to-face communication	Eliminates some spontaneous learning instances, difficulties to demonstrate things	-	E
Geographical distance	Lack of informal communication	Lack of cohesion	Improved over time	B, C
Geographical distance	Limited access to development tools	Communication about physical development tools was difficult because not everyone had access to them.	Using videos or screen sharing to demonstrate the tools	C
Geographical distance	No face-to-face meetings	Difficulties getting started with the project	-	(A)
Temporal distance	Differences in schedules	Delays in communication	24-hour rule, being aware of team member's availability, enabling notifications that notify about new messages	A, B, E, G, J
Temporal distance	Differences in schedules	Difficulties finding common meeting times	Fixed meeting times, scheduled meetings well in advance	B, C, D, E, F
Temporal distance	Differences in time zones	Difficulties finding common meeting times	-	I
Temporal distance	Differences in schedules	Team members not able to attend meetings	Meeting notes, being aware of team member's availability	B

Table 2: The identified communication challenges related to distance differences are listed in the table.

6.2. Communication challenges related to communication tools

Category	Challenge	Consequence	Solutions	Teams Affected
Tools	Technical problems with tools	Disruptions in online meetings	-	B, E, H, J
Tools	Unfamiliarity with communication tools	Passive use of tools	Using alternate tools, using multiple different tools	B, E, J
Tools	Unfamiliarity with communication tools	Making mistakes or not being efficient with the tools	Learning the tools over time, using other available tools	B, C, J
Tools	No webcams during meetings	Less social cues, which made communication more difficult	-	B, H, J
Tools	Team members restricted to using text chat during meetings	More difficult to communicate	-	H, J
Tools	Inefficient use of task board	Reduced communication and awareness	Increasing team's involvement with task board	E, F, (B, H)
Tools	Using multiple communication tools	Difficult to keep track of all communication channels	-	F

Table 3: The identified communication challenges related to communication tools are listed in the table.

Communication tools were a significant source of communication challenges for the project teams (Table 3). Technical problems with communication tools were encountered by several teams. It was mostly reported as a minor challenge because it mostly caused minor disruptions in communication. The unfamiliarity of communication tools was also a challenge encountered by several teams, and it led to problems such as passive or inefficient use of communication tools. To solve the problem, the teams had to learn to use the tools or use alternate tools. Furthermore, the number of communication channels and keeping track of all of them caused difficulties at least for one team.

Limited communication methods were a problem for some teams. Not using webcams was a common preference in many teams, but it caused communication difficulties at least for three teams. The difficulties were mostly related to reduced social cues which made communication more difficult. In addition to this, two teams reported that some of

their team members were occasionally restricted to text chat communication during meetings, which made communication more difficult. No solutions were suggested for these problems because they were caused by personal preferences or limited working environments.

The usage of task boards caused also challenges for some teams. Inefficient task board practices, namely the usage of task board solely as project manager's tool, were reported by several teams. The causes for the problem are not entirely clear, but one likely cause might be the lack of experience or knowledge. Some teams might have lacked knowledge in how to manage and utilize their task board efficiently. Only one team reported having solved the problem by increasing the involvement of the team with their task board.

6.3. Communication challenges related to client communication

Category	Challenge	Consequence	Solutions	Teams Affected
Client communication	Insufficient communication or miscommunication with the client	Assumptions had to be made about requirements or goals	Regular meetings and communication with clients	B, F, (A, H)

Table 4: The identified communication challenges related to customer communication are listed in the table.

Client communication caused challenges at least for two teams (Table 4). Miscommunication or insufficient communication with customers were identified. They often led to assumptions about the requirements or goals of the project. For some teams, this was a problem mostly in the beginning, but in some cases, the problems persisted throughout the project. Regular meetings and communication with the clients helped solve and mitigate these problems.

6.4. Communication challenges related to team communication

Team communication was challenging for several teams (Table 5). Early communication problems were common for many teams. In most cases, they were caused by the unfamiliarity of team members, which makes sense considering that most students in the teams were unfamiliar with each other before the course. In some cases, early communication problems were caused by team configuration problems, such as team members quitting the project in early phases or teams having to be merged. However, early communication problems were usually solved over time when team members became more familiar with each other. Some teams were also able to solve the problems by addressing them during meetings.

Lack of task-related communication was a challenge at least for three teams. In these teams, team members did not communicate enough about their tasks, which led to problems such as assumptions having to be made or lack of awareness. For solving the problems, the teams mentioned the usage of task boards as a potential solution. However, the efficient usage of task boards proved to be another challenge, as mentioned earlier. Similar to task boards, one team used visualization tools to visualize task dependencies between the team members and achieve increased awareness. One team also suggested that adding more progress meetings could have improved task-related communication.

Lack of agenda during meetings was a problem at least for one team. It caused inefficient meetings. However, the team was able to solve the problem by adding a clear agenda to all meetings.

Category	Challenge	Consequence	Solutions	Teams Affected
Team communication	Team members were unfamiliar with each other	Early communication problems	Improved over time, project manager encouraged others to communicate by showing an example as an active communicator	B, C, E
Team communication	Early team configuration	Early communication problems	Addressing problems during a meeting	G, (A)
Team communication	Lack of task-related communication	Lack of task awareness	Using task boards, visualizing dependencies between tasks, adding more progress meetings	C, F
Team communication	Lack of task-related communication	Assumptions had to be made	-	A
Team communication	Lack of agenda during meetings	Inefficient meetings	Adding agenda to all meetings	C

Table 5: The identified communication challenges related to team communication are listed on the table.

6.5. Communication challenges related to human factors

Category	Challenge	Consequence	Solutions	Teams Affected
Human factors	Differences in motivation or goals	Not following rules or participating enough in communication	-	B, J
Human factors	Skill differences	Difficulties understanding or explaining technical concepts	Giving team members tasks and responsibilities that match their skills	B, C, E, (F)
Human factors	Skill differences	Some team members dominated communication	Involving less active team members by asking their opinions	C, J
Human factors	Skill differences	Misunderstandings	-	J, (G, H)
Human factors	Passive team members	Team members not sharing information or asking help	Encouraging communication	A
Human factors	Language differences	Misunderstandings, passiveness in communication	-	H, (A, I)

Table 6: The identified communication challenges related to human factors are listed in the table.

Communication challenges related to human factors were identified almost in all projects (Table 6). The identified challenges include language differences, the passiveness of team members, skill differences, and differences in motivations or goals. Even though quite many teams encountered these challenges, no solutions were often mentioned, which suggests that these challenges may have been difficult to solve.

Skill differences were the most common communication challenge related to personal differences. Skill differences caused many types of problems, such as difficulties explaining or understanding technical concepts, experienced team members dominating communication, and misunderstandings. However, they were usually quite minor problems. Some solutions to the problems were mentioned. Two teams mentioned that they tried to ensure that the less experienced team members were involved in communication by asking their opinions regularly during meetings. It was also mentioned that assigning suitable tasks for team members eliminated some of the trouble with having to communicate about unfamiliar concepts. However, most teams did not mention any specific solutions for the problems related to skill differences.

Communication problems related to language differences were encountered in teams that had to use English as their main language. Most teams had only native Finnish speakers, and thus they did not have any language problems. Language differences caused misunderstandings and passiveness in communication. The teams that had language difference problems did not mention any specific solutions for them. Passiveness in communication and similar problems were also encountered in other teams for other reasons than language differences. In one team, passiveness in communication was a result of personal differences, and the team mentioned encouraging communication as a solution. Differences in motivations and goals were a challenge at least for two teams. The teams reported problems such as team members not following rules or participating enough in communication. No solutions were mentioned for them either.

6.6. Answers to research questions

6.6.1. Answer to RQ1

RQ1: What communication-related challenges can be found in student software projects?

The teams encountered many communication challenges during their projects (Tables 2 – 6). However, most of the challenges were quite minor and easily solvable. The more significant communication challenges were often team-specific. Overall, the most common challenges for communication were differences in schedules (8 teams) and differences in skills (7 teams). They were the only challenges encountered by most of the teams. Other more common communication challenges included technical problems with tools (4 teams), unfamiliarity with communication tools (4 teams), insufficient / miscommunication with clients (4 teams), inefficient use of task board (4 teams), not using webcams during meetings (3 teams), the unfamiliarity of team members (3 teams), lack of task-related communication (3 teams), and language differences (3 teams).

The distributed project setting and limited face-to-face communication possibilities did not cause significant communication challenges for the teams. Availability of face-to-face interaction would have most likely eliminated some of the challenges such as time zone differences. Team members would have most likely had an easier time having more natural conversations and becoming familiar with each other, demonstrating code and physical development tools could have been easier, and with less reliance on communication tools some technical problems and unfamiliarity with tools could have been avoided. However, most of the communication challenges would have likely stayed the same even if face-to-face interaction was possible. Some teams even seemed to prefer the remote setting due to its benefits, such as, not having to spend time commuting to face-to-face meetings.

Not all challenges could be identified due to a lack of details of the material provided by the teams. Some teams were able to report their challenges in great detail, while some teams only shortly mentioned the most evident challenges. Several teams also mentioned that some challenges were only noticed or discussed later in the project when the teams had to address them for the final report and during the final meeting with the customers. It is possible that the teams were not aware of all the communication difficulties that affected them, and communication might have been a more significant problem than what was reported.

6.6.2. Answer to RQ2

RQ2: How were the challenges solved?

The teams were able to find solutions to most of the encountered challenges. Challenges were usually solved in a reactionary way when they caused problems rather than trying to prevent or mitigate them beforehand. Typically, when there were communication difficulties, they were addressed during a meeting and the team would decide how to deal with them. Some teams managed their communication difficulties better than others. As mentioned earlier, some teams noticed their communication difficulties later in the project when it was already too late to solve them. The teams were able to solve the most crucial challenges to complete their project. However, the less significant communication problems were often ignored, even if they were known. Certain challenges were also unavoidable/unsolvable (e.g., technical problems) or a consequence of conscious decisions (e.g., not using webcams), and no actions were taken to solve or mitigate them even though they might have caused some problems. Some problems might have also been too difficult to solve, and with limited experience or resources, the students might have been unable to solve them. The most difficult challenges to solve seemed to have been the ones related to human factors. For instance, language differences and differences in motivation or goals were quite significant challenges, but no solutions were mentioned for them.

6.6.3. Answer to RQ3

RQ3: How did agile practices help solve or mitigate communication-related challenges?

The student teams found agile practices useful for mitigating and solving communication-related challenges. However, the practices had to be applied to a remote working environment, in which the students had limited possibilities for face-to-face communication. Furthermore, the students were only working part-time on the project. Also, the students themselves were not professionals and might have lacked knowledge even about the basics. For these reasons, the teams often did not use agile practices to their full extent, and the usage of agile practices was not always efficient.

Task boards were used by all teams to organize and keep track of tasks. Several teams found them useful for increasing task-related communication and awareness, which consequently mitigated and solved problems related to them. However, many teams also encountered challenges related to task boards, most notably the inefficient use of them.

The project teams also used internal team communication practices that were influenced by certain agile practices. Progress meetings were organized normally at least once a week. In these meetings, the teams often discussed what everyone had done since the previous meetings, what problems they had encountered, and how they would proceed. A similar practice is commonly used in daily Scrum meetings. Some teams mentioned that it allowed everyone to speak about their tasks and problems, which increased task awareness and ensured that everyone would at least communicate their tasks at a minimum level. It was also mentioned that it gave meetings a clear structure or agenda, which made meetings more efficient.

Several teams reported that regular customer involvement helped improve communication with customers and solve misunderstandings and assumptions. All teams had review meetings after each sprint, which ensured that the teams had at least minimum communication with their customers. Some teams had more regular meetings and communication with their customers (e.g., weekly meetings with the customers) which allowed any misunderstandings to be solved quickly compared to teams that mostly relied on sprint review meetings. Some teams that had less communication with their customers acknowledged that the communication could have been more active, but at the same time noted that it was enough for a student project. In real-world projects would passive customer communication could have led to larger problems, as some of them mentioned.

Given the limited amount of material provided by the teams and the lack of details in the material, the effects of agile practices used by the teams could not be fully identified.

7. Conclusion

Software development projects are often complex, and they require sufficient communication which is not always easy to achieve. Communication becomes even more difficult in distributed settings in which face-to-face communication is limited. Communication breakdowns can cause significant problems and affect the project's success. For this reason, practitioners need to be able to identify communication challenges and address them accordingly.

This study identified communication challenges and their solutions based on software projects worked by university students. For these teams, communication was mostly a minor challenge. Implementation, time management, and lack of experience caused more significant problems for the teams. Many communication challenges were identified, but most of them were quite minor challenges that were either easy to solve or could be ignored without significant consequences. The more troublesome communication challenges that were more difficult to solve and caused more significant problems (e.g., language differences or motivation differences) were more team-specific and only affected certain teams.

The teams were able to find solutions to most of the communication challenges. Although, some teams managed their communication better than others. The student teams were also required to use some common agile practices, which generally had a positive effect on communication. However, agile practices were not always used efficiently due to a lack of experience, which might have been a challenge in itself.

Perhaps the one most notable observation about how communication and the challenges were managed was that the student teams were not necessarily interested in dealing with problems that they did not see as a threat to their project success or ability to pass the course. For instance, none of the teams used webcams actively, even though it made communication more difficult during meetings. Perhaps the students acknowledged that their project was only a part of a university course that can be passed whether the project succeeds or fails, rather than a serious real-world project with more significant consequences. As a result, the students may have had different goals and lower motivation to handle communication and the challenges it causes, compared to industry professionals working in real-life projects. The way communication was handled by the teams might have been sufficient for a student project, but in real-world settings, their practices would have potentially caused more significant problems.

This study identified many communication challenges that are relevant to university student projects. However, not all the possible challenges could be identified. Students with limited experience may not be aware of all the problems that affect their projects. Also, students may lack the motivation to provide detailed answers. For these reasons, it is important to carefully design how the data is collected and what type of questions are

asked from the students. In this study, students were asked more open-ended questions for which many students and teams failed to provide detailed answers, which limited the possibilities for identifying communication challenges. Perhaps multiple-choice questions could have been used as the primary type of questions, while being supported by open-ended questions, instead of fully relying on open-ended questions.

Further research could be able to identify more communication challenges from similar projects. As different projects have varying settings, people involved, and different needs for communication, different communication challenges could be discovered. For instance, time zones were not a major source for problems in the projects addressed in this paper because they involved participants mostly from the same time zone and area, but in some other projects varying time zones might be a more crucial factor for communication. Communication challenges may also change over time, and new trends can introduce additional communication challenges, but also solutions to the existing ones. The recent pandemic, for instance, might have already changed communication both in face-to-face and distanced environments. Thus, the topic of communication and its challenges in software development remains open for additional research.

References

- Victor G. Abarca, Pedro Palos-Sanchez, and Enrique Rus-Arias. 2020. *Working in virtual teams: a systematic literature review and a bibliometric analysis*. *IEEE Access* 8: 168923-168940.
- Muhammad O. Ahmad, Valentina Lenarduzzi, Markku Oivo, and Davide Taibi. 2018. Lessons learned on communication channels and practices in agile software development. In: *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, IEEE, 929-938.
- Yehia I. Alzoubi, Asif Q. Gill, and Ahmed Al-Ani. 2016. Empirical studies of geographically distributed agile development communication challenges: a systematic review. *Information & Management* 53 (1), 22-37.
- Paul L. Bannerman, Emam Hossain, and Ross Jeffery. 2012. Scrum practice mitigation of global software development coordination challenges: a distinctive advantage?. In: *Proc. of the 45th Hawaii International Conference on System Sciences*, IEEE, 5309-5318.
- Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, and Ron Jeffries. 2001. *The Agile Manifesto*.
- Steve Berczuk. 2007. Back to Basics: The role of agile principles in success with a distributed scrum team. In: *Agile 2007*, IEEE, 382-388.
- Gregory Berry. 2011. Enhancing effectiveness on virtual teams: understanding why traditional team skills are insufficient. *International Journal of Business Communication (Thousand Oaks, Calif.)* 48 (2), 186-206.
- Fabio Calefato and Christof Ebert. 2019. Agile collaboration for distributed teams. *IEEE Software* 36 (1), 72-78.
- Peter W. Cardon and Bryan Marshall. The hype and reality of social media use for work collaboration and team communication. *International Journal of Business Communication* 52 (3), 273-293.
- Erran Carmel, Alberto Espinosa, and Yael Dubinsky. 2010. "Follow the sun" workflow in global software development. *Journal of Management Information Systems* 27 (1), 17-38.
- Igor Čavrak, Marin Orlić, and Ivica Crnković. 2012. Collaboration patterns in distributed software development projects. In: *Proc. of the 34th International Conference on Software Engineering (ICSE)*, IEEE, 1235-1244.
- Kieran Conboy. 2009. Agility from first principles: reconstructing the concept of agility in information systems development. *Information Systems Research* 20 (3), 329-354.
- Eoin Conchúir, Pär Ågerfalk, Helena Olsson, and Brian Fitzgerald. 2009. Global software development: where are the benefits?. *Communications of the ACM* 52 (8), 127-131.

- Richard Daft and Robert Lengel. 1986. Organizational information requirements, media richness and structural design. *Management Science* 32 (5), 554-571.
- Alan Dennis and Joseph Valacich. 1999. *Rethinking media richness: Towards a theory of media synchronicity*. In: *Proc. of the 32nd Annual Hawaii International Conference on Systems Sciences*, IEEE, 10 pages.
- Alan Dennis, Robert Fuller, and Joseph Valacich. 2008. Media, tasks, and communication processes: A theory of media synchronicity. *MIS quarterly* (2008), 575-600.
- Philipp Diebold, Jan-Peter Ostberg, Stefan Wagner, and Ulrich Zendler. 2015. What do practitioners vary in using Scrum?. In: *Proc of the 16th International Conference on Agile Software Development*, Springer, 40-51.
- Digital.ai. 2021. 15th state of agile report. Available as <https://digital.ai/resources/state-of-agile>. Checked October 20, 2021.
- Paul Dourish and Victoria Bellotti. 1992. Awareness and coordination in shared workspaces. In: *Proc. of the 1992 ACM Conference on Computer-Supported Cooperative Work*. 107-114.
- Clarence Ellis, Simon Gibbs, and Gail Rein. 1991. Groupware: some issues and experiences. *Communications of the ACM* 34 (1), 38-58.
- Fabian Fagerholm, Alejandro S. Guinea, Jay Borenstein, and Jurgen Munch. 2014. Onboarding in open source projects. *IEEE Software* 31 (6), 54-61.
- Hugo Fuks, Alberto Raposo, Marco A. Gerosa, Mariano Pimentel, Denise Filippo, and Carlos Lucena. 2008. Inter-and intra-relationships between communication coordination and cooperation in the scope of the 3C Collaboration Model. In: *Proc. of the 12th International Conference on Computer Supported Cooperative Work in Design*, IEEE. 148-153.
- Imran Ghani, Angelica Lim, Muhammad Hasnain, Israr Ghani, and Imran B. Muhammad. 2019. Challenges in distributed agile software development environment: a systematic literature review. *KSII Transactions on Internet and Information Systems (TIIS)* 13 (9), 4555-4571.
- Carl Gutwin, Saul Greenberg, and Mark Roseman. 1996. Workspace awareness in real-time distributed groupware: Framework, widgets, and evaluation. In: *Proc of HCI'96 (People and Computers XI)*, Springer, 281-298.
- Kumi Ishii, Mary M. Lyons, and Sabrina A. Carr. Revisiting media richness theory for today and future. *Human Behavior and Emerging Technologies* 1 (2), 124-131.
- Samireh Jalali, and Claes Wohlin. 2010. Agile practices in global software engineering - A systematic map. In: *Proc. of the 5th IEEE International Conference on Global Software Engineering*, IEEE, 45-54.
- James Herbsleb and Deependra Moitra. 2001. Global software development. *IEEE Software* 18 (2), 16-20.
- Miguel Jiménez, Mario Piattini, and Aurora Vizcaino. 2009. Challenges and improvements in distributed software development: A systematic review. *Advances in Software Engineering 2009*, 14 pages.

- Alfredo Jimenez, Dirk Boehe, Vasyl Taras, and Dan Caprar. 2017. Working across boundaries: Current and future perspectives on global virtual teams. *Journal of International Management*, 23 (4), 341-349.
- Mira Kajko-Mattsson, Gayane Azizyan, and Miganoush Katrin. 2010. *Classes of distributed agile development problems*. In: *2010 Agile Conference, IEEE*, 51-58.
- Filippo Lanubile. 2007. Collaboration in distributed software development. In: Andrea De Lucia, Filomena Ferrucci (eds.), *Software Engineering. ISSSE 2006-2008*, Lecture Notes in Computer Science, vol 5413, 174-193.
- Pernille Lous, Marco Kuhrmann, and Paolo Tell. 2017. Is Scrum fit for global software engineering?. In: *Proc of the 12th International Conference on Global Software Engineering (ICGSE)*, IEEE, 1-10.
- Pernille Lous, Paolo Tell, Christian B. Michelsen, Yvonne Dittrich, Marco Kuhrmann, Allan Ebdrup. 2018a. Virtual by design: How a work environment can support agile distributed software development. In: *Proc. of the 13th International Conference on Global Software Engineering (ICGSE)*. IEEE, 97-106.
- Pernille Lous, Paolo Tell, Christian B. Michelsen, Yvonne Dittrich, Allan Ebdrup. 2018b. From Scrum to Agile: a journey to tackle the challenges of distributed development in an Agile team. In: *Proc. of the 2018 International Conference on Software and System Process*, 11-20.
- Courtney Miller, Paige Rodeghero, Margaret-Anne Storey, Denae Ford, and Thomas Zimmermann. 2021. "How was your weekend?" software development teams working from home during COVID-19. In: *Proc. of the 43rd International Conference on Software Engineering (ICSE)*. IEEE, 624-636.
- Audris Mockus and James Herbsleb. 2001. Challenges of global software development. In: *Proc. of the 7th International Software Metrics Symposium*, IEEE, 182-184.
- Sunila Modi, Pamela Abbott, and Steve Counsell. 2013. Negotiating common ground in distributed agile development: A case study perspective. In: *Proc of the 8th International Conference on Global Software Engineering*, IEEE, 80-89.
- Nils B. Moe and Darja Šmite. 2007. Understanding lacking trust in global software teams: A multi-case study. In: *Proc of the 8th International Conference on Product Focused Software Process Improvement (PROFES 2007)*, Springer, 20-34.
- Nils B. Moe, Daniela Cruzes, Tor Erlend Fægri, Jan Edvard Faugustad. 2016. Enabling knowledge sharing in agile virtual teams. In: *Proc. of the 11th International Conference on Global Software Engineering (ICGSE)*. IEEE, 29-33.
- OED Online. September 2021. Communication, n. Oxford University Press.
<https://www.oed.com/view/Entry/37309?redirectedFrom=communication> (accessed October 10, 2021).
- Mehmet Orhan. 2017. The evolution of the virtuality phenomenon in organizations: A critical literature review. *Entrepreneurial Business and Economics Review* 5 (4), 171-188.

- Maria Paasivaara, Sandra Durasiewicz, and Casper Lassenius. 2009. Using Scrum in distributed agile development: A multiple case study. In: *Proc. of the 4th International Conference on Global Software Engineering*, IEEE, 195-204.
- Lene Pries-Heje and Jan Pries-Heje. 2011. Why Scrum works: A case study from an agile distributed project in Denmark and India. In: *Proc. of the 2011 Agile Conference*. IEEE, 20-28.
- Balasubramaniam Ramesh, Lan Cao, Kannan Mohan, and Peng Xu. 2006. Can distributed software development be agile?. *Communications of the ACM* 49 (10), 41-46.
- Mohammad A. Razzak and Rajib Ahmed. 2014. Knowledge sharing in distributed agile projects: Techniques, strategies and challenges. In: *Proc of the 2014 Federated Conference on Computer Science and Information Systems*, IEEE, 1431-1440.
- Everett M. Rogers. 2003. *Diffusion of Innovations*. Free Press.
- Paul Robinson. 2019. Communication network in an agile distributed software development team. In: *Proc. of the 14th International Conference on Global Software Engineering (ICGSE)*, IEEE, 100-104.
- Ken Schwaber and Jeff Sutherland. 2020. The Scrum Guide – The Definitive Guide to Scrum: The Rules of the Game. Available as <https://scrumguides.org/docs/scrum-guide/v2020/2020-Scrum-Guide-US.pdf#zoom=100>. Checked November 05, 2021.
- James Simpson. 2002. Computer-mediated communication, *ELT Journal* 56 (4), 414–415.
- Slack. 2021. Slack vs. email: An easier, more organized way to work. <https://slack.com/why/slack-vs-email> (accessed August 23, 2021).
- Darja Šmite, Claes Wohlin, Zane Galviņa and Rafael Prikladnicki. 2014a. An empirically based terminology and taxonomy for global software engineering. *Empirical Software Engineering: An International Journal* 19 (1), 105-153
- Darja Šmite, Marco Kuhrmann, and Patrick Keil. 2014b. Virtual teams. *IEEE Software* 31 (6), 41-46.
- Sarah Morrison-Smith and Jaime Ruiz. 2020. Challenges and barriers in virtual teams: a literature review. *SN Applied Sciences* 2, 1-33.
- Igor Steinmacher, Ana P. Chaves, and Marco A. Gerosa. 2013. Awareness support in distributed software development: A systematic review and mapping of the literature. *Computer Supported Cooperative Work (CSCW)* 22 (2-3), 113-158.
- Viktoria Stray, Nils Brede Moe, and Mehdi Noroozi. 2019. Slack me if you can! using enterprise social networking tools in virtual agile teams. In: *Proc. of the 14th International Conference on Global Software Engineering (ICGSE)*, IEEE, 111-121.
- Leigh Thompson. 2000. *Making the Team: A Guide for Managers*. Upper Saddle River, NJ: Prentice Hall.
- Crispin Thurlow, Laura Lengel, and Alice Tomic. 2004. *Computer Mediated Communication*. London: SAGE Publications.

- University of Minnesota Libraries Publishing. 2017. *Business Communication for Success: GVSU Edition*. Grand Valley State University Libraries.
- Raoul Vallon, Bernardo J. da Silva Estacio, Rafael Prikladnicki, and Thomas Grechenig. 2018. Systematic literature review on agile practices in global software development. *Information and Software Technology* 96, 161-180.
- Richard West, Lynn Turner, and Gang Zhao. 2010. *Introducing Communication Theory: Analysis and Application*. Vol. 2 McGraw-Hill New York, NY.
- Jeanne Wilson, Michael Boyer O'Leary, Anca Metiu, and Quintus Jett. 2008. Perceived proximity in virtual work: Explaining the paradox of far-but-close. *Organization Studies* 29 (7), 979-1002.
- Pär J. Agerfalk, Brian Fitzgerald, Helena Holmström, Brian Lings, Björn Lundell and Eoin Conchúir. 2005. A framework for considering opportunities and threats in distributed software development. In: *Proc. of the International Workshop on Distributed Software Development 2005*, 16 pages.