Slack Me If You Can! Using Enterprise Social Networking Tools in Virtual Agile Teams

Viktoria Stray
University of Oslo, SINTEF
Oslo, Norway
stray@ifi.uio.no

Nils Brede Moe
SINTEF
Trondheim, Norway
nils.b.moe@sintef.no

Mehdi Noroozi
University of Oslo
Oslo, Norway
mehdin@ifi.uio.no

***Abstract*—Virtual teams rely on enterprise social networking tools such as Slack to collaborate efficiently. While such tools contribute to making the communication more synchronous and to support distributed agile development, there are several challenges such as how to interact with each other and how to balance the communication with other types of communication mechanisms such as meetings, e-mail, and phone. In this paper, we describe and discuss how a distributed global project used Slack. Some of the challenges we identified were related to language problems, using too much direct messaging when communicating, and unbalanced activity (33% of the users accounted for 86% of the messages). The positive aspects of using the tool were increased transparency, team awareness, and informal communication. Further, Slack facilitates problem-focused communication which is important for agile teams. Our study stresses the importance of reflecting on how virtual teams use communication tools, and we suggest that teams decide on guidelines on how to use the tools to improve their coordination.**

Keywords—communication, coordination, Slack, distributed teams global teams, agile software development, Enterprise Social Media

I. Introduction

Global software development (GSD) has become prevalent in recent years [1]. Some of the features that make GSD so popular is the possibility of working on the same software project simultaneously from wherever in the world and getting access to a larger workforce [2]. Further, global software development often relies on virtual teams in which team members are dispersed across different locations [3]. While the scenario of globally distributed teams is attractive on paper, in practice, there are many obstacles that such teams might face. For example, communication and coordination between team members and other stakeholders turn to become a significant challenge in distributed software development as well as trust, culture, and language [4] [5], [6].

Agile software development practices seem to address some of the inherent challenges following from distribution of work, like complexities in communication, coordination, and collaboration [7]. Agile philosophy favors flat organizations, open mindset, and informal communication instead of indirect formal communication patterns, and strict manager-led work division and allocation. For companies involved in GSD that want to work in an agile manner, this means that additional effort, investment, and consideration is needed to achieve a sustainable strategy that enables agile benefits [8]. Ramesh et al. [9] suggest four practices to improve communication in agile GSD; synchronize work hours, provide for informal communication through formal channels, balanced coordination, and constant communication. Further, enabling communication and coordination based on mutual adjustment across sites is one key to succeed with agile in GSD [10].

As a consequence, many organizations now start to use applications such as Slack or Yammer. These applications are known as Social Networking Platforms and can be categorized under communication methods that are called Enterprise Social Networking (ESN). ESN acts as a forum for its members where they can communicate with other co-workers in the same organization [11], facilitating knowledge sharing and easy access to expertise [12]. Further, ESN enables team leaders to take part in the daily communication to know what difficulties the team is facing at any time [13]. However, research on the use of such tools in virtual teams is scarce [14].

Motivated by the need of more studies on globally distributed software teams, and to understand approaches and solutions for coordination and communication in such teams, we studied a group of 30 developers, testers, team leaders, tech leaders and managers in one product center at a large international company. The group was working in agile virtual teams and was distributed between Norway and Poland. We interviewed and observed the team members in both countries, and we also collected and analyzed Slack logs, meeting minutes and other project material. In particular, we used the case study method [15] to investigate the following research questions:

RQ1: *What are the challenges of using enterprise social networking tools in agile virtual teams?*

RQ2: *How do enterprise social networking tools enable coordination by feedback in agile virtual teams?*

This paper is organized as follows. We present background in Section 2 and the research method used in Section 3. We report our findings in Section 4. In Section 5 we discuss the findings, implications for practice, as well as the limitations. We conclude and suggest future work in Section 6.

II. Background

In this section, we first present related work of communication in agile GSD projects, and then we describe Enterprise Social Networking and Slack which is a platform for team communication. Finally, we discuss Slack as a coordination mechanism in teams by drawing on established coordination theory.

A. Communication in agile GSD

Communication has a vital role in the success of GSD [16], and informal communication, in particular, plays a crucial role [16]. Further, face-to-face meetings are the most efficient and ideal type of communication [17], but in GSD

This is the author accepted version of an article published in 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)
https://doi.org/10.1109/ICGSE.2019.00031

where teams are be spread over continents with time zone differences, such meetings are limited. Therefore, team members communicate with each other using some form of internet-based communication tool [18]. However, in the presence of temporal distance, real-time and synchronous communication is challenging to achieve [19]. Therefore many distributed software development teams try to balance agile and distributed approaches, by requiring synchronous work and regular visits, to improve communication [2].

While new technology and new processes enable better communication in GSD, there are still many challenges. High pressure in mastering advanced communication technology and lack of nonverbal communication, as well as problems in forming trust between distributed teams and team members are some of them [20], [21]. Misunderstandings and conflicts might also happen in the absence of sentiment and meta-level information, such as body language [22]. Also, if the communication technology and tools are not adapted to the specific distributed context, they might lead to limited information sharing between distributed team members [14]. While regular and scheduled communication is vital in distributed teams [5], research also shows that developers are in need of ad hoc and informal communication [23], [24]. Therefore any shortcomings in communication between the team members will profoundly impact the success of GSD projects [25]. Receiving delayed feedback is another challenge often faced by distributed teams [26], and the time to receive a response increases dramatically with the use of only asynchronous communication tools [19]. Casey and Richardson [26] studied two distributed teams located in Ireland and Malaysia that used e-mail and found that use of only this type of asynchronous communication also increased the misunderstandings and ambiguity of information.

B. *Enterprise Social Networking*

The amount of communication tools and platforms that support distributed software development has increased substantially in the last years. Examples are Slack, Yammer, Microsoft Teams and Workplace by Facebook [27]. These applications are known as Social Networking Platforms (SNP) or Team Communication Platforms (TCP) and can be categorized under communication methods that are called Enterprise Social Networking (ESN).

Cardon and Marshall [28] surveyed business professionals to investigate the perceptions of using enterprise communication tools for team communication and coordination and concluded that ESN tools probably will become the primary communication tool for teams. Kane [29] looked into ESN and suggested a framework for considering the design decisions and possibilities when implementing such tools, for example, to consider how the design of the tool affect how people in the organization share and access content and how they interact with each other. Riemer et al., [30] studied the usage of Yammer, an enterprise communication tool, in the consultancy firm Capgemini to explain how social network emergence may originate from a grass-roots initiative.

One thing that differentiates general social media such as Facebook and Twitter from ESN is that social media is open to public and everyone can use them, but access to ESN is limited to employees and those with whom the company needs to get in touch with regularly [31]. There are four attributes which a platform should provide for its users to be considered as an ESN [12]:

1. Being able to send and receive messages to individual members as well as the whole or a group of members.
2. Being able to choose and show particular coworkers as their communication partners.
3. Having the possibility to share files with other users.
4. Being able to view all the conversations that are done publicly, consisting of text messages and file shared.

If the ESN is managed correctly, it can boost the productivity and motivation of employees, enhance collaboration between the different actors in the organization, and lead to more learning for team members [32]. Moe et al., [13] found in a study of a virtual team between Norway and Ukraine that the primary use of the social software was related to asking questions and discussing problems, which increases productivity. ESN also enables people in different locations to communicate and become more involved in innovation, revenue creation, and socio-economic growth (Qi and Chau, 2016). However, Moe et al., [13] reported that the activity (number of messages) varied significantly among the team members indicating that not everyone in a virtual team is involved in the communication and committed to using the communication platform. Therefore it is essential to understand that there is a learning curve associated with any new tool, and organizations need to invest in training to reach their goals of increasing collaboration between their employees [33]. Developing an open culture among the employees and using an ESN which have similarities to social media that employees use in their private life may help to improve the usage of ESN by employees [34].

One of the problems that can be associated with the long-term use of ESNs in distributed projects is that employees might feel isolated because of limited face-to-face encounters with other people [35], but at the same time, ESN is not designed and meant to replace personal and physical encounters completely [36].

C. *Slack*

Slack is an ESN tool that was launched in 2014 and now has ten million active users every day [37]. It has turned a contemporary form of communication, i.e., texting, into a workplace app. Although lately some big players in the market, like Microsoft, Google, and Facebook, have launched products to compete with Slack, still 43 percent of Fortune 100 companies are using Slack. Its popularity among startup companies is the same, if not more than the big enterprises [38]. Slack is a multi-platform software, meaning that it can be used on all kinds of operating systems including mobile phones. In addition to instant messaging it is also possible for users to make voice and video calls and share files.

Messages in Slack can be seen and searched for only by those who are involved in the messages. Using emoji is another way of communicating in Slack, it can be both used by the user who is writing the message to communicate emotions, as well as by the readers of the message who can "react" to the message and communicate their emotion with the sender of the message. It is also common to get the attention of someone or all the members of a channel by using the @-symbol. In this case, users will immediately get a notification if they are online.

According to media synchronicity theory [39], instant messaging as a communication technology earlier scored low on the attributes symbol variety and parallelism. However,

This is the author accepted version of an article published in 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)
https://doi.org/10.1109/ICGSE.2019.00031

now we see that modern ESN tools are quite rich in providing these two facets. The "editability" feature is unique in ESNs such as Slack and cannot be found in other ways of communication (e.g., e-mail). When communicating on Slack, the users can edit their posts without any time limitation, and a small "(edited)" sign will be shown after the message so that the others will know that the message is changed. As a consequence, a person may respond more quickly and without as much thought as when writing an e-mail, knowing they have the ability to edit the message at a later time. A person may also delete their message if they regret sending it. These features might lower the threshold to communicate with each other in a distributed project.

As Slack is a quite new tool, there is little research yet. However, Stoeckli [40] looked into Slack chatbots and integrations from an affordance perspective and explored their constraints within enterprises while Lin et al., [41], through an exploratory study, tried to find out how developers use Slack and what kind of benefits it could give the team.

D. Slack as a coordination mechanism

Effective coordination and coordination mechanisms are essential for having success in GSD projects. Van de Ven et al. [42] propose three coordinating modes: by programming or codification (impersonal mode), and coordination by feedback on an individual level (personal mode) or a group level (group mode). Once implemented, the impersonal coordination mechanisms are codified and require minimal verbal communication between people. Examples include pre-established plans, checklists, and process documentation. Coordination by mutual adjustment or feedback is based on informal communication. In the personal mode, people in the organization make mutual task adjustments through either vertical or horizontal channels of communication. In the group mode, the mechanism for mutual adjustment is vested in a group through scheduled or unscheduled meetings. In the group mode, the mechanism for mutual adjustment is vested in a group through scheduled or unscheduled meetings. This division is mainly in order to differentiate between the more routine encounters and the informal conversations between co-workers [42].

Slack supports coordination through feedback on individual and group mode and both vertical and horizontal channels. The coordination on Slack can be categorized as vertical when team members are communicating with their manager. When team members communicate in group channels they are exercising group mode coordination; a group of people can follow and join the conversation. Williams [43] describe how Slack can be seen as a chat room where the whole company and its different teams can be broken into smaller channels for group discussion. Channels are often created to discuss a particular topic and are like the old chat rooms. These channels can either be public or private. Public channels are visible to the entire team, and all the team members can join them while private channels require an invitation to join. Most often the communication on Slack will be unscheduled as communication is often triggered by someone asking a question, sharing information or participating in a discussion. However, team members will sometimes schedule a Slack conversation or Slack video call at a specific time. Additionally, Slack also supports impersonal mode of coordination, for example, automatic messages posted on Slack through bots and integration with other systems.

III. Data Collection and Analysis

A. About the case

Geosoft (a pseudonym) is a large software company that produces and sell specialized software for the engineering domain. Geosoft develops both mass-market software and customer-specific software on a contract basis. In addition to developing software in the main offices in Norway, Geosoft also develops software in its offices in Malaysia, Germany, China, USA, and the UK. Geosoft has over 15 years' experience with global software development.

We studied a group of 30 developers in four agile virtual teams in one Product Centre distributed across Norway and Poland. At the time of the interviews, the teams were named Front-end, Back-end, Operations, and UX. We observed and interviewed team members on both sites. The development teams were considered to be self-organized, and the team members could choose themselves which tasks to solve from the product backlog. The backlog was maintained by the team leads and the business leader. The business leader was located in Norway and was the one responsible for the daily decisions regarding the features in the backlog. The developers and testers at both sites performed code reviews.

B. Data collection

We collected chat logs, consisting of around 30,000 messages shared between the team members from March 2015 to August 2017. We observed four distributed stand-up meetings and conducted a group interview in Norway in August 2017. Additionally, we held informal interviews with two of the leaders. In September 2017, we facilitated four retrospective meetings in Poland where members from both the Polish and Norwegian site where present. We also had several informal conversations and unstructured interviews with the members outside of these retrospective meetings. Throughout the study, we communicated with our main contact at Geosoft who was available for off-site data collection via both e-mail and phone.

C. Data analysis

All interview and observational notes were imported into a shared database and discussed among the first and second author. Throughout the research we wrote memos which acted as a log and gave us the possibility to look at how our reflections evolved and why a particular decision was made during the research. For the analysis of the chat logs we used NVivo.

The exported chat logs from Slack were in the form of a compressed zip file which contained several folders which represented the channels in Slack. Inside each folder, there were lots of JSON files which each one containing the chat logs of one day. Inside the JSON file, every message was coded in a special format showing the ID of the sender, the message and a timestamp. We had to convert the IDs to usernames and the timestamps, which were in Epoch format, to a readable format. Finally, it was not possible to import JSON files directly into NVivo, so we converted the files to PDF before they were imported into NVivo.

This is the author accepted version of an article published in 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)
https://doi.org/10.1109/ICGSE.2019.00031

TABLE I. CODING SCHEME

Category	Explanation	Example from the logs
Coordination /General information	Giving general information	We will cancel todays internal demo meeting. There are too many people on holidays
General discussions	Questions and answers regarding general topics	Dates are ok, and no food allergy for me.
Problem-focused communication	Technical questions and discussion of possible solutions	Is it expected that user groups options endpoint returns nothing? It depends on how much we need to refactor, but there is better support for using a Java framework.
Technical Information	Giving technical information	Please ignore notifications from TFKR, I'm just cleaning up the view.
Socializing	Messages used for socializing	I would like to award this week's Commit Name Award to @Christian.
Emoji	Emojis used by users	:slightly_smiling_face:, :stuck_out_tongue:

In NVivo we coded parts of the logs manually, and we also conducted some automatic analysis offered by the tool (e.g., word frequency). For the manual coding in NVivo we applied descriptive coding. Descriptive coding uses a word or a short phrase to summarize the topic (as opposed to the content) and is useful for answering questions such as "What is going on here?"[44], see Table 1 for the final coding scheme.

As we have followed the team for many years, we have observed the team spirit and communication tone during visits [46], and have found that the tone in the teams overall is positive. We were, therefore, curious to see if automatic coding in NVivo would reflect this tone and we applied a sentiment analysis. Automatic sentiment analysis is to determine from the text whether the emotion or attitude of the statement is positive or negative [47]. NVivo looks at the sentiment of words in isolation and does not take context into account. As such, automatic coding is a good supplement to manual coding in NVivo, but cannot (yet) replace human coding.

IV. FINDINGS

First, we describe in general how the virtual teams collaborated to understand the role of Slack, followed by a description of the actual use of Slack. Next, we describe some challenges that the teams faced. Then we discuss how Slack may support agile virtual teams. Finally, we describe how they in a retrospective agreed on improving the use of the ESN tool Slack.

A. Team communication and coordination

The product center relied on a combination of Scrum and Kanban (ScrumBan) which requires frequent communication and coordination. Further, the teams employed agile practices such as daily stand-up meetings with the use of video, retrospective meetings, and demos and grooming meetings. Additionally, regular face-to-face meetings were essential for making the agile practices work. For example, they regularly had co-located retrospective meetings, see Fig. 1. The whole product center collocated every 6 or 12 months. Also, each time a new development effort was started, developers and testers involved would co-locate for one to two weeks. How co-location was done in practice is described in [46].

Fig. 1. A virtual team having a co-located retrospective meeting

Inspect and adapt was one core value for the product center, which led to continuously experimentation of agile practices for distributed teams. E.g., they tried out a virtual water cooler by having a video link up and running during working hours, and different variations on processes and tools for distributed work.

At the time of conducting the interviews at the Norwegian site, each day had six virtual meetings (stand-up meetings) scheduled for their virtual teams, see Table 2. During our observations of meetings, we found that team-members referred to discussions on Slack in a way that it was expected that everyone was updated on the latest information. Moreover, the team members did not wait for the daily meeting to address issues. As we observed in a stand-up meeting:

"[UX developer], do you have anything to add?"
"No, I got answers on Slack yesterday".

TABLE II. DAILY GROUP MODE COORDINATION

Start	End	Team meeting
08:50	09:00	Operations
09:00	09:15	UX
09:15	09:30	Front-end
09:30	09:45	Back-end
09:45	10:00	Support meeting
10:00	10:15	Bug triage

B. Use of Slack

In 2016, the virtual teams started using Slack as a communication tool. A project member, responsible for agile methods, told us:

"Many people were negative to introduce "another system" because we were already using Yammer, Wikis, Skype and others. But, we wanted to see if we could benefit from it in our distributed work."

Since a subscription to Slack channels was voluntary, they continued to use e-mail for information they needed to guarantee that everyone would get.

This is the author accepted version of an article published in 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)
https://doi.org/10.1109/ICGSE.2019.00031

TABLE III. CODING OF STATEMENTS

Type	Avg.	General	Back-end
General information/coordination	3 %	1 %	4 %
General discussions	15 %	16 %	13 %
Problem-focused communication	48 %	50 %	46 %
Technical information	20 %	6 %	33 %
Socializing	3 %	4 %	1 %
Emoji	13 %	23 %	3 %
SUM	100%	100%	100%

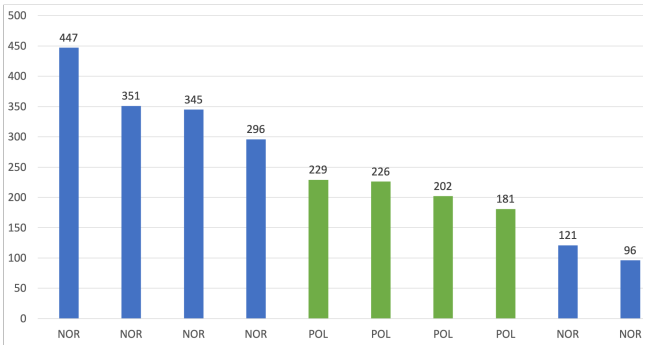

Fig. 2. User activity across three channels. NOR= Norwegian, POL = Polish

The Slack logs included 30 000 messages sent over 2.5 years in 70 channels. Most channels were no longer active and had been archived, indicating that Geosoft adapted Slack to their current needs. Because of privacy settings in Slack, we could not get access to the direct messages between two individuals.

We coded two of the channels manually, see result of the coding in Table 3. Almost half of the statements were coded as problem-focused communication (21% questions and 25% answers, on average). There were little social talk and almost no usage of emojis in the back-end channel, these numbers were much higher in the General channel.

Figure 2 shows user activity in the three essential channels General, Back-end and Front-end. The ten (of 30) most active users had 86% of the messages. Of these ten, the two Norwegian tech leads where the most and third most active user and two senior developers were the second and third most active. One Norwegian tech lead commented on her role and the need to interact with others:

> ”*some developers and testers are lacking the technical skills, so you need to help and support them.”*

When analyzing each channel separately we found that the Back-end channel had a slightly different pattern, see Fig. 3. The contribution was more balanced between the countries in this channel. One reason was that the Norwegian tech leads were posting in several channels, and therefore in sum were the most active users across channels.

We conducted automatic coding of the four channels General, Help, Back-end and Operations. A word frequency query in these channels revealed that “works” was the most frequent word in the logs, followed by “smiling” (see Figure 4). Smiling is such a frequent word because smileys or

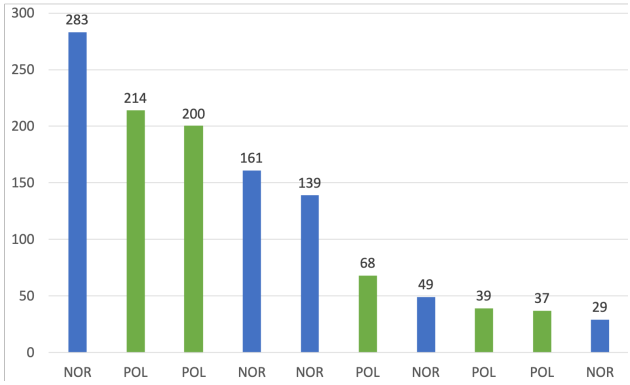

Fig. 3. User activity in the Back-end channel. NOR= Norwegian, POL = Polish

Fig. 4. Word cloud of the 100 most frequent words in four of the channels

positive emojis are shown in the logs as for example “:smile:”, “:smiling_face:” or “simple_smile”.

We also performed an automatic sentiment analysis in these four channels to get a view of the overall tone. Generally, the words were coded as neutral, see Table 4. The product owner explained:

> *“How personal we talk in open channels is kept within certain limits.”*

The General channel had a few more positive statements than negative statements, the same was the case in Back-end. However, Operations had 2% negative statements, and no statements were coded as positive. Surprisingly, the Help channel had more negative than positive statements.

TABLE IV. AUTOMATIC SENTIMENT ANALYSIS

Type	Avg.	General	Help	Back-end	Operations
Mixed	7 %	3 %	19 %	4 %	0 %
Negative	8 %	10 %	11 %	8 %	2 %
Neutral	78 %	73 %	64 %	77 %	98 %
Positive	8 %	14 %	6 %	11 %	0 %
Total	100%	100%	100%	100%	100%

This is the author accepted version of an article published in 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)
https://doi.org/10.1109/ICGSE.2019.00031

C. Challenges

1) Language

A challenge when using Slack in virtual teams was related to the different native languages people have. We found that in some cases, people wrote in Norwegian or Polish, probably because they found it was easier to write in their native language and the communication was directed to people in their own country. However, this prevented other (virtual) team members from understanding and engaging in the communication. This issue was also discussed in one retrospective meetings, where they discussed the importance of using English as the working language.

Another challenge was that some of the Polish developers were not that proficient in English, and this probably reduced the motivation and capability to write in Slack. As one Norwegian tech lead explained:

> *"We have struggled somewhat because of language issues. Domain knowledge in combination with low language skills has made it difficult. It was a real problem that especially one person was really bad in English writing."*

As two of the team members in Norway spoke English, the Norwegians were used to English as a working language, and therefore the language barrier was smaller for them.

2) Unbalanced activity

Another challenge was the unbalanced activity. The agile team is supposed to be self-managed, which require everyone to communicate with everyone and that the communication is balanced and not only one-way. Some users were very active, while some posted just a few messages and did not contribute much. We found that 33% of the users wrote 86% of the messages, and the four most active users were Norwegian tech lads and senior developers. This unbalance was perceived as a problem and discussed in a group interview. A teach lead stated: "*Some team members don't ask any questions [in the channels], maybe it's a cultural difference."*

The most active user was a Norwegian tech lead, whose main task became to support the developers, and in particular, she used a lot of time communicating with the Polish developers. An excerpt from the Slack logs give an example of a Polish developer asking her for clarification:

> **Developer:** to me this seems like test data that has been added to the develop database by mistake.. but maybe I'm wrong.. @techLead?

In the group interview, this tech lead elaborated on her role in the project which affected her activity on Slack:

> *"I ask a lot of questions to avoid misunderstandings. I want to make sure that we clarify terms and concepts so that we agree on the definitions, which are important."*.

3) Using Too Much Personal Mode of coordination

Another challenge was the use of private messages instead of communicating in public channels. One interviewee suggested that half of the messages on Slack were direct messages between team members. Several others confirmed this opinion regarding the use of direct messages. Generally, we found that the Norwegian developers and testers wanted more of the communication to be open while the Polish project members appreciated having private conversations. The Norwegian product manager stated: *"One should use the "public" channels and not direct messages, to increase learning."*

The managers and tech leads encouraged the use of channels and not direct messaging to be more transparent and to share more knowledge. However, it was hard to change as one tech lead explained: *"I try to encourage them to write messages in open channels, but they still continue to send personal messages*."

One explanation could be that the Norwegian developers and had used social software for a longer period and were therefore more motivated to communicate in Slack channels. While the developers in Poland were used to more one-to-one communication in their teams and a more hierarchical organization. Another explanation could be that remote developers in Poland noticed that the Norwegians answered more quickly when contacted directly. As one Norwegian developer explained:

> "*While I try to answer as quick as possible in the channels, I give faster feedback when a person sends a direct message."*

The Norwegians tech leads hoped that with more communication being open, others could learn from the discussions and that the team members would need less support as time went by.

4) ESN workspace structure (channels and members)

A final challenge was to not have too many channels so it would be difficult to follow relevant discussions but at the same time channels needed to have a narrow focus to know where to discuss what topic. The product manager explained: "*A major challenge has been to find a balance in the number of channels and make the channels as relevant as possible.* Another project member stated: *"We have created channels for different topics so that team members do not have to explain the context every time they send a new message. But, it does not work perfectly"*

Having many channels lead to another problem. During a retrospective meeting, one topic discussed was that people found it difficult to know who should join which channel.

D. Support for agile teamwork in virtual teams

Agile methods highlight principles such as communication, collaboration and transparency [49] as well as being able to "sense and respond" [50] or "inspect and adapt"[51].

We found that Slack facilitated these agile principles in the distributed project. As the product manager commented: *"We use Slack to share knowledge, communicate frequently and, to enable continuously learning."*

Slack enables some of the same transparency in distributed projects as co-located projects with an open office landscape. One example is that team members used Slack to notify each other about their presence. In 2016, a developer suggested: *"Should we have an #out-of-office channel here on slack? Easier than sending email imo."* This channel was created two days later and actively used for informing each other of presence across sites, see an excerpt of a log in Table V. Team members also notified each other of why they could not attend team meetings (e.g., daily stand-up meetings).

This is the author accepted version of an article published in 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)
https://doi.org/10.1109/ICGSE.2019.00031

TABLE V. EXCERPT FROM CHANNEL #OUT-OF-OFFICE

User	Time	Statement
1	06:17	I'll be bit late today. There is something called as 'Easter Bake Sale' at my daughter's school for her class today at 9:15 which I'm going to attend to.
2	06:17	Good luck with that:-)
3	06:18	Thanks!!
4	07:35	I'm running late today. I'll be in as soon as possible.
5	08:08	I'm down with fever. I'll try to work some from home when the drugs kick in. Available on slack and phone all day.

They also discussed the importance that if someone knew an answer to a question without being involved in the discussion, they should respond immediately and not wait for others (for example the person mentioned with @) to answer. They encouraged fast feedback as one team member posted on Slack:

"(…)Please give me feedback ASAP. And as always, we aim to "fail fast"; if we see problems, we'll deal with them as they appear."

To quickly sense and respond, information has to be easily accessible. Slack was built around the principle of having easy access to information; the name came from the acronym Searchable Log of All Conversation and Knowledge [52]. Team members often posted information to have it accessible for future reference. The team members also used Slack to share files, especially screenshots. A Norwegian team member offering help to a Polish team member: *"Yes. I can help. Do you have a screenshot of how this should look like when it is done?"*

E. Guidelines.

As the virtual teams wanted to improve how they interacted with each other using ESN, they often discussed in retrospectives how to communicate in the project and on Slack. In one co-located retrospective it became evident that it was important to formalize slack guidelines so everyone had the same understanding and to make it easier for newly hired people to get up to speed. The project members created the following guidelines:

More open communication. Open communication in channels was favored over direct messages between two people.

Each team should have a main channel. E.g., front-end, back-end and UX.

More separate channels. Rather than posting too many messages in team channels or the general channel**,** they wanted more narrow and specific channels. All channels should have a descriptive purpose so that it would be easy to know what issues to discuss in which channel.

Less communication of features and bugs in other tools. While they still used other tools (such as Microsoft VSTS) to discuss features and bugs, they wanted to move this type of communication only to Slack, e.g., by creating new channels for discussions of specific features or bugs. They believed it then would be easier to browse the history of the discussions of that issue, if the discussion was only in one place.

More short-lived channels. A developer who first started to work on a feature should make a new channel for that feature (similarly for bugs). All new feature- or bug specific channels had to be mentioned in the general-channel. They decided that such short-lived channels should have a specific prefix, showing that it was a feature-discussion, for example. The channel should be archived when the feature had been implemented, or the bug had been solved, to reduce the number of channels.

V. DISCUSSION

For this study, we aimed to understand and explain how a virtual agile team being distributed across sites used the Enterprise Social Networking tool Slack to coordinate and communicate. Agile software development relies on frequent interactions, and since distance makes people communicate less [6] virtual teams need tools that can mitigate the distance and lower the communication barriers. We found that by using Slack, the agile teams were able to rely on mutual adjustment. While we found that Slack supports the virtual teamwork, there are some barriers. We now turn to our first research question: *What are the challenges of using enterprise social networking tools in agile virtual teams?*

We identified four challenges: language, unbalanced activity, using too much personal mode of coordination and finding the right ESN workspace structure. We will now discuss these challenges.

Language is an often-reported challenge in distributed projects [6] and was also present in our study. When team members in distributed projects have to communicate in their second language, the quality of the communication is reduced [5], which explain why some messages were in Norwegian or Polish. However, this behavior excluded other team members. Further, we found that the Norwegians were used to speaking English, which can explain why they dominated more in the Slack channels. Our finding is in accordance with a study of a distributed agile project [53], that found that people who were confident in their second language dominate more in meetings. One of the reasons that the Norwegians were more confident was that English had been their working language for several years. From this, we argue that language and unbalanced activity were related. We now continue to discuss the unbalanced activity.

Another reason for the imbalance between the two sites was that the tech leads and senior developers were in Norway while more junior developers were located in Poland. Our findings are in agreement with Smite et al., [54]; newly hires and less experienced people communicate less frequently than more experienced team members. Apart from this, the Norwegians had worked longer together and then probably also had stronger ties. Strong ties and a higher knowledge level, influence the frequency of communication [55], which then probably results in more frequent communication among the Norwegians.

While the analysis of the Slack logs showed an imbalance, we believe that the communication in reality was more balanced. Since the Polish developers and testers favored direct messaging (personal mode) over messages in channels (group mode) this affected the total analysis as the direct messages could not be included, there were fewer messages from the Polish team members in the public channels. Being less experienced and therefore having more need for help from an experienced person could also be one explanation for why

This is the author accepted version of an article published in 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)
https://doi.org/10.1109/ICGSE.2019.00031

the direct messaging were appreciated – meaning that they would get faster feedback. The unbalanced activity in Slack between the countries can also be explained by other relevant streams of research to address the adoption of methods and technology, e.g. the diffusion of innovation literature [56]. The Norwegians had used social software for a longer time. The earlier-mentioned language barrier may also be a reason why Polish members were less active on Slack.

We now turn to our second research question: *How do enterprise social networking tools enable coordination by feedback in agile virtual teams?*

Coordination mechanisms of type feedback is divided into personal and group mode. The personal mode is further divided into vertical (communication with a manager) and horizontal (communication with other team members). Group mode is divided into scheduled and unscheduled meetings. When solving complex tasks with a high degree of uncertainty, like in a distributed software project, the team need to rely on a high level of mutual adjustment in both personal and group mode [42]. We found a high use of both modes but that it was unbalanced. Further, the Polish developers and testers appreciated vertical personal mode over group mode as explained when discussing research question one. However, the Norwegians encouraged the move from personal to group mode.

A large portion of the discussions on Slack were problem-focused communication. This type of communication involves problem-solving discussions where team members share knowledge and discuss solutions and is linked to positive team outcomes, and [48].

While activity in Slack channels support strengthening the awareness of what is going on in the team which is important for coordination by feedback, we found one specific channel that was dedicated for the purpose of sharing information on non-job activities. Distributed team members who have met and know each other personally have stronger ties and communicate better [55], [57]. Slack supported this e.g. when team members talked about their family members in the "out of office" channel (e.g., being sick, having a cake sale), the personal ties of the virtual members became stronger. Further, when the remote site knows why the person is absent, the level of trust is maintained. When the remote site do not know why there is no progress at the remote site, the level of trust is reduced [5].

An essential tenet of agile is transparency. Slack facilitates group mode of coordination because all the virtual team members can engage in the discussion. Whereas team members not included in an e-mail thread, for example, might miss out on relevant information. Alternatively, they might hold valuable information that they do not know that other project members need because they are not aware of the discussion going on.

A culture of experimentation and rapid feedback in agile teams is vital for succeeding in distributed projects [51]. In this project, we saw that managers encouraged team members on Slack to "fail fast". The virtual teams also experimented with the use of Slack and the structure of the workspace to continuously improve their communication.

The team members received fast feedback to their questions on Slack. Frequent communication builds trust and awareness of tasks and how they affect each other [58]. Further, they informed each other on what they were doing, which increased the transparency between sites. Increased transparency is a key for building build trust, and trust is vital for success in distributed teams [5].

Slack also supported building stronger and more autonomous teams, which is a prerequisite for the success of agile teams. Slack supported the teams and teamwork by:

- Increasing team awareness by supporting constant information sharing
- Facilitating network building (both team internal and external networks)
- Increasing the awareness of who knows what, which is essential for high-performing teams [59]
- Reducing the need for e-mail and other channels

Team awareness is the understanding of the activities of others [60] and is the result of recurrent processes of information sharing within a team [61]. The ongoing information sharing in Slack helped build team awareness. Our findings suggest that Slack facilitates constant informal communication through formal channels, which improves communication in agile distributed projects [9].

The team members experienced the introduction of Slack as a positive change; however, at the same time it was also an opportunity for senior members and tech leads to have more rigid control of each team member. One team lead reported asking many questions, to control that everything was understood and agreed upon. Our findings are in agreement with the findings of Moe [62] and Barker [63], who pointed out that self-managing teams may end up controlling group members more rigidly than with traditional management styles.

A. Implications for practice

First, the team members should agree on clear guidelines on how to use and not use the ESN tool, and regularly discuss these guidelines. Relying on the right norms emerging by themselves on how to communicate since the team members might not be aware of the norms or have not the same understanding of the norms based on their cultural background [64]. For example, the team may decide to have a rule that all team members communicate in English to reduce frustration and misunderstandings.

Second, organizations should continuously observe their teams and see how they use the tools. Some team members might not be familiar with the tools or might continue to use the former tools that they are accustomed to (e.g., email). For the teamwork in distributed teams to be successful, all team members should be involved in the discussions on the dedicated tool. As many companies now start to implement BizDevOps teams [50] it is especially important that also team members from the business side use the ESN tool.

Finally, managers may check the usage trends by analyzing the data provided by the communication and coordination tool. For example, examining the number of messages sent and received by the teams during a month and see the trends in the number of messages at different times during working hours. An analysis also makes it possible to identify people who do not send that many messages to help them master the tool efficiently.

This is the author accepted version of an article published in 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)
https://doi.org/10.1109/ICGSE.2019.00031

B. Limitations

As with all empirical studies, there are some limitations that we need to consider. First, we used a single-case design and thus have the possibility of bias in data collection and analysis. Therefore, the general criticisms of single-case studies, such as uniqueness and special access to key informants, may also apply to our study. However, our rationale for choosing Geosoft as our case was that it represents a valuable case for investigating the use of ESN in virtual teams as we had been observing their use of such tools for several years.

Another possible limitation is that some of our data collection is based on meetings that we facilitated as well as interviews. Also, while we gathered a large number of Slack messages, we only manually coded a small portion of them. However, the use of multiple data sources made it possible to find evidence for episodes and phenomena from more than one data source. We also observed, talked to, and interviewed the team members over a period of several years as they had been involved in a research project on global work, which made it possible to study the phenomena from different viewpoints as they emerged and changed.

VI. Conclusion and Future Work

In this paper, we shed light on communication and coordination in virtual teams. By analyzing conversations traded in Slack channels between two teams located in Norway and Poland we tried to understand how distributed teams communicate on Enterprise Social Networking tools.

Our research shows that there are signs that some users are very active while others post very few messages. We found that language skills and knowledge level, in particular, influenced the user activity. Further, experienced team members time favored messages in open channels, while less experienced people favored more direct messages (one-to-one communication). Openness and transparency are building blocks of collaboration and trust in agile projects.

In our case study, the team members were constantly communicating and coordinating with other virtual team members. This highlights the importance of using tools that can help teams to do such activities in the best and most efficient way possible. Our study suggests that it is very important for the team members to be comfortable with the tools they are using and that there should be clear guidelines for use of such tools so that everyone can benefit from them. According to our results, Slack supports frequent communication and collaboration, which can support globally distributed software development organizations and their teams.

Future work should compare the conversations between team members in more balanced teams. Further, it would also be interesting to better understand the role of private messages of the team members in an ESN. Additionally, the nature and degree of group mode and personal mode of communication and its influence on team performance would be highly relevant to investigate. Future research should also analyze how quickly people respond in open channels compared to direct messaging.

Acknowledgment

We are grateful to all the participants in the case study. This work was supported by the A-team project, partly funded by the Research Council of Norway under grant 267704.

References

[1] M. Paasivaara and C. Lassenius, "Could Global Software Development Benefit from Agile Methods?," in 2006 IEEE International Conference on Global Software Engineering (ICGSE'06), Florianopolis, Brazil, 2006, pp. 109–113.

[2] D. Šmite, C. Wohlin, T. Gorschek, and R. Feldt, "Empirical evidence in global software engineering: a systematic review," Empirical Software Engineering, vol. 15, no. 1, pp. 91–118, Feb. 2010.

[3] R. Jabangwe, D. Šmite, and E. Hessbo, "Distributed software development in an offshore outsourcing project: A case study of source code evolution and quality," Information and Software Technology, vol. 72, pp. 125–136, Apr. 2016.

[4] V. Casey and I. Richardson, "Virtual software teams: overcoming the obstacles," 3rd World Congress on Software Quality, 2005.

[5] Nils Brede Moe and Darja Smite, "Understanding a lack of trust in Global Software Teams: a multiple-case study," Softw. Process, vol. 13, no. 3, pp. 217–231, 2008.

[6] J. Noll, S. Beecham, and I. Richardson, "Global software development and collaboration: barriers and solutions," ACM inroads, vol. 1, no. 3, pp. 66–78, 2010.

[7] G. K. Hanssen, D. Smite, and N. B. Moe, "Signs of Agile Trends in Global Software Engineering Research: A Tertiary Study," presented at the Global Software Engineering Workshop (ICGSEW), 2011 Sixth IEEE International Conference on, 2011, pp. 17–23.

[8] Darja Šmite, Nils Brede Moe, and Viktoria Stray, Integrating Agile with an Offshore Strategy. Leanpub, 2018.

[9] B. Ramesh, L. Cao, K. Mohan, and P. Xu, "Can distributed software development be agile?," Communications of the Acm, vol. 49, no. 10, pp. 41–46, Oct. 2006.

[10] S. Hole and N. B. Moe, "A Case Study of Coordination in Distributed Agile Software Development," presented at the Software Process Improvement, 2008, pp. 189–200.

[11] P. M. Leonardi, "Social media, knowledge sharing, and innovation: Toward a theory of communication visibility," Information Systems Research, vol. 25, no. 4, pp. 796–816, Dec. 2014.

[12] P. M. Leonardi, M. Huysman, and C. Steinfield, "Enterprise social media: Definition, history, and prospects for the study of social technologies in organizations," Journal of Computer-Mediated Communication, vol. 19, no. 1, pp. 1–19, Oct. 2013.

[13] N. B. Moe, D. S. Cruzes, T. Dybå, and E. Engebretsen, "Coaching a Global Agile Virtual Team," in 2015 IEEE 10th International Conference on Global Software Engineering, 2015, pp. 33–37.

[14] L. L. Gilson, M. T. Maynard, N. C. Jones Young, M. Vartiainen, and M. Hakonen, "Virtual Teams Research: 10 Years, 10 Themes, and 10 Opportunities," Journal of Management, vol. 41, no. 5, pp. 1313–1337, Jul. 2015.

[15] R. K. Yin, Case study research: design and methods. Thousand Oaks, Calif.: Sage, 2002.

[16] E. Carmel and R. Agarwal, "Tactical approaches for alleviating distance in global software development," IEEE Software, vol. 18, no. 2, pp. 22–29, Apr. 2001.

[17] B. L. Kirkman, B. Rosen, P. E. Tesluk, and C. B. Gibson, "The Impact of Team Empowerment on Virtual Team Performance: The Moderating Role of Face-to-Face Interaction," The Academy of Management Journal, vol. 47, no. 2, pp. 175–192, 2004.

[18] B. L. Kirkman and J. E. Mathieu, "The dimensions and antecedents of team virtuality," Journal of Management, vol. 31, no. 5, pp. 700–718, Oct. 2005.

[19] H. Holmstrom, E. O. Conchuir, P. J. Agerfalk, and B. Fitzgerald, "Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance," in 2006 IEEE International Conference on Global Software Engineering (ICGSE'06), 2006, pp. 3–11.

[20] S. L. Jarvenpaa and D. E. Leidner, "Communication and Trust in Global Virtual Teams," Journal of Computer-Mediated Communication, vol. 10, no. 6, pp. 791–815, Jun. 1998.

This is the author accepted version of an article published in 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)
https://doi.org/10.1109/ICGSE.2019.00031

[21] T. R. Kayworth and D. E. Leidner, “Leadership Effectiveness in Global Virtual Teams,” Journal of Management Information Systems, vol. 18, no. 3, pp. 7–40, Jan. 2002.

[22] C. D. Cramton, “The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration,” Organization Science, vol. 12, no. 3, pp. 346–371, Jun. 2001.

[23] R. Kraut and L. Streeter, “Coordination in software development,” Communications of the ACM, vol. 38, no. 3, pp. 69–81, Mar. 1995.

[24] J. D. Herbsleb and R. E. Grinter, “Architectures, coordination, and distance: Conway’s law and beyond,” Software, IEEE, vol. 16, no. 5, pp. 63–70, 1999.

[25] L. Layman, L. Williams, D. Damian, and H. Bures, “Essential communication practices for Extreme Programming in a global software development team,” Information and Software Technology, vol. 48, no. 9, pp. 781–794, Sep. 2006.

[26] “The Impact of Fear on the Operation of Virtual Teams - IEEE Conference Publication.” [Online]. Available: https://ieeexplore-ieee-org.ezproxy.uio.no/abstract/document/4638663. [Accessed: 04-Feb-2019].

[27] P. Leroy, C. Defert, A. Hocquet, F. Goethals, and J. Maes, “Antecedents of willingness to share information on enterprise social networks,” in Lecture Notes in Information Systems and Organisation, vol. 2, Springer, Berlin, Heidelberg, 2013, pp. 109–117.

[28] P. W. Cardon and B. Marshall, “The hype and reality of social media use for work collaboration and team communication,” International Journal of Business Communication, vol. 52, no. 3, pp. 273–293, 2015.

[29] G. C. Kane, “Enterprise social media: Current capabilities and future possibilities,” MIS Quarterly Executive, vol. 14, no. 1, pp. 1–16, 2015.

[30] K. Riemer, P. Overfeld, P. Scifleet, and A. Richter, “Oh, SNEP! The dynamics of social network emergence-the case of Capgemini Yammer,” 2012.

[31] E. Turban, N. Bolloju, and T. P. Liang, “Enterprise social networking: Opportunities, adoption, and risk mitigation,” Journal of Organizational Computing and Electronic Commerce, vol. 21, no. 3, pp. 202–220, Jul. 2011.

[32] R. – D. Leon, R. Rodríguez-Rodríguez, P. Gómez-Gasquet, and J. Mula, “Social network analysis: A tool for evaluating and predicting future knowledge flows from an insurance organization,” Technological Forecasting and Social Change, vol. 114, pp. 103–118, Jan. 2017.

[33] K. E. Anderson, “Getting acquainted with social networks and apps: picking up the Slack in communication and collaboration,” Library Hi Tech News, vol. 33, no. 9, pp. 6–9, Nov. 2016.

[34] P. Korzynski, “How does online social networking help leaders communicate? Evidence from the Fortune 500,” Asia Pacific Journal of Human Resources, vol. 52, no. 4, pp. 460–475, Oct. 2014.

[35] G. C. Kane, M. Alavi, G. (Joe) Labianca, and S. P. Borgatti, “What’s different about social media networks? A framework and research agenda,” MIS Quarterly, vol. 38, no. 1, pp. 275–304, Mar. 2014.

[36] X. Zhang and V. Venkatesh, “Explaining Employee Job Performance: The Role of Online and Offline Workplace Communication Networks,” MIS Quarterly, vol. 37, no. 3, pp. 695–722, 2013.

[37] K. Shawn, “Slack surpasses 10 million daily active users.” .

[38] S. Rodriguez, “Slack adds 1 million paying users amid increasing competition,” 2018.

[39] L. M. Maruping and R. Agarwal, “Managing team interpersonal processes through technology: A Task-technology fit perspective,” Journal of Applied Psychology, vol. 89, no. 6, pp. 975–990, 2004.

[40] E. Stoeckli, “Exploring Affordances of Slack Integrations and Their Actualization Within Enterprises – Towards an Understanding of How Chatbots Create Value,” Hawaii International Conference on System Sciences 2018 (HICSS-51), pp. 2016–2025, 2018.

[41] B. Lin, A. E. Zagalsky, M.-A. Storey, and A. Serebrenik, “Why Developers Are Slacking Off: Understanding How Software Teams Use Slack,” in Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion - CSCW ’16 Companion, 2016, pp. 333–336.

[42] A. Van De Ven, A. Delbecq, R. Koenig, and Delbecq, “Determinants of Coordination Modes within Organizations,” American Sociological Review, vol. 41, no. 2, pp. 322–338, 1976.

[43] O. Williams, “What is and how to use Slack: the ultimate guide to doing anything.,” 2015.

[44] J. Saldaña, The coding manual for qualitative researchers. Sage, 2015.

[45] D. R. Thomas, “A General Inductive Approach for Analyzing Qualitative Evaluation Data,” American Journal of Evaluation, vol. 27, no. 2, pp. 237–246, 2006.

[46] N. B. Moe, T. E. Fægri, D. S. Cruzes, and J. E. Faugstad, “Enabling Knowledge Sharing in Agile Virtual Teams,” in 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE), 2016, pp. 29–33.

[47] S. M. Mohammad, “Sentiment Analysis: Detecting Valence, Emotions, and Other Affectual States from Text,” in Emotion Measurement, H. L. Meiselman, Ed. Woodhead Publishing, 2016, pp. 201–237.

[48] S. Kauffeld and N. Lehmann-Willenbrock, “Meetings matter: Effects of team meetings on team and organizational success,” Small Group Research, vol. 43, no. 2, pp. 130–158, 2012.

[49] M. Laanti, O. Salo, and P. Abrahamsson, “Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation,” Information and Software Technology, vol. 53, no. 3, pp. 276–290, Mar. 2011.

[50] B. Fitzgerald and K.-J. Stol, “Continuous software engineering: A roadmap and agenda,” Journal of Systems and Software, 2015.

[51] P. Lous, P. Tell, C. B. Michelsen, Y. Dittrich, and A. Ebdrup, “From Scrum to Agile: A Journey to Tackle the Challenges of Distributed Development in an Agile Team,” in Proceedings of the 2018 International Conference on Software and System Process, New York, NY, USA, 2018, pp. 11–20.

[52] “Slack, the red hot $3.8 billion startup, has a hidden meaning behind its name,” 28-Sep-2016. [Online]. Available: https://nordic.businessinsider.com/where-did-slack-get-its-name-2016-9/. [Accessed: 01-Feb-2019].

[53] V. Stray, Y. Lindsjorn, and D. I. K. Sjoberg, “Obstacles to efficient daily meetings in agile development projects: A case study,” in IEEE International Symposium on Empirical Software Engineering and Measurement, 2013, pp. 95–102.

[54] D. Šmite, N. B. Moe, A. Šāblis, and C. Wohlin, “Software teams and their knowledge networks in large-scale software development,” Information and Software Technology, vol. 86, pp. 71–86, 2017.

[55] J. H. Gittell, “New directions for relational coordination theory,” in The Oxford handbook of positive organizational scholarship, 2011.

[56] E. M. Rogers, Diffusion of Innovations, vol. Fourth Edition. New York: The Free Press, 1995.

[57] S. Dorairaj, J. Noble, and P. Malik, “Understanding Team Dynamics in Distributed Agile Software Development,” in Agile Processes in Software Engineering and Extreme Programming, vol. 111, C. Wohlin, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 47–61.

[58] O. McHugh, K. Conboy, and M. Lang, Using Agile Practices to Build Trust in an Agile Team: A Case Study. 2010.

[59] K. Lewis and B. Herndon, “Transactive Memory Systems: Current Issues and Future Research Directions,” Organization Science, vol. 22, no. 5, pp. 1254–1265, 2011.

[60] P. Dourish and V. Bellotti, “Awareness and coordination in shared workspaces,” in Proceedings of the 1992 ACM conference on Computer-supported cooperative work - CSCW ’92, Toronto, Ontario, Canada, 1992, pp. 107–114.

[61] E. Salas, C. Prince, D. P. Baker, and L. Shrestha, “Situation Awareness in Team Performance: Implications for Measurement and Training,” Hum Factors, vol. 37, no. 1, pp. 123–136, Mar. 1995.

[62] N. B. Moe, “Key Challenges of Improving Agile Teamwork,” in Agile Processes in Software Engineering and Extreme Programming, vol. 149, H. Baumeister and B. Weber, Eds. Springer Berlin Heidelberg, 2013, pp. 76–90.

[63] J. R. Barker, “Tightening the Iron Cage - Concertive Control in Self-Managing Teams,” Administrative Science Quarterly, vol. 38, no. 3, pp. 408–437, Sep. 1993.

[64] V. Stray, T. E. Fægri, and N. B. Moe, “Exploring norms in agile software teams,” presented at the International Conference on Product-Focused Software Process Improvement, 2016, pp. 458–467.

This is the author accepted version of an article published in 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)
https://doi.org/10.1109/ICGSE.2019.00031