Studying Onboarding in Distributed Software Teams: A Case Study and Guidelines

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

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

Marcus R. Goplen
University of Oslo
Oslo, Norway
goplen@live.no

ABSTRACT

Many companies have turned towards globally distributed software development in their quest for access to more development capacity. This paper investigates how a company onboarded distributed teams in a global project, and report experience on how to study such distributed projects. Onboarding is the process of helping new team members adapt to the existing team and ways of working. The goal of the studied onboarding program was to integrate Portuguese developers into two existing Norwegian teams. Further, due to the growing trend in utilizing globally distributed projects, and the challenge of conducting studies in distributed organizations, it is crucial to find good practices for researching such projects. We collected qualitative data from interviews, observations, Slack conversations and documents, and quantitative data on Slack activity. We report experiences on different onboarding practices and techniques, and we suggest guidelines to help other researchers conduct qualitative studies in globally distributed projects.

CCS CONCEPTS

• **General and reference** → **Empirical studies**; • **Software and its engineering** → *Software creation and management*.

KEYWORDS

Onboarding, Offshoring, Outsourcing, Global Software Engineering, Case studies, Research techniques

ACM Reference Format:
Nils Brede Moe, Viktoria Stray, and Marcus R. Goplen. 2020. Studying Onboarding in Distributed Software Teams: A Case Study and Guidelines. In *Evaluation and Assessment in Software Engineering (EASE 2020), April 15–17, 2020, Trondheim, Norway*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3383219.3383235

1 INTRODUCTION

Global software development (GSD) has become a standard way of doing software development. From a survey conducted in the United States from 2006, Oshri et al. [19] found that 2/3 of the IT companies outsourced work. When Moe et al. [16] took a closer look at the experience of four international software organizations, they found that promised benefits of GSD to be neither clear-cut nor guaranteed; all the four companies terminated their offshore contracts because of the low quality of the software being developed. While GSD is challenging, one particular significant factor leading to GSD success is related to onboarding. Muethel et al. [17] argue that recruitment and onboarding of new employees are essential for the success of dispersed teams. Further, Britto et al. [3] found that onboarding is challenging in GSD projects, and they identified practices that can be used to increase the chances of being successful when onboarding team members in GSD projects.

While evidence on GSD challenges and success factors exist, there are two main reasons why there is a need for more research in this area. First, there is a lack of industrial studies. Many studies of distributed work are studies of student teams [8], and therefore not always relevant for industry. Second, many studies are reported in such a way that implications for research and practice are hard to access. In their study on evidence reported from GSD studies, Smite et al. [25] found that most studies had unclear results and reported on general challenges of cross-border collaboration.

Although there is a need for more studies on GSD, several challenges need to be overcome for the studies to be relevant. It is essential to describe the context in order to reach valid conclusions when aggregating evidence [4]. Therefore, for research on GSD to have implications for practice and research, the context in which the industrial GSD studies were conducted must be described in detail. Further, data collection is challenging in GSD studies. In their study of four distributed projects across four countries, Moe et al. [15] experienced that because of limited availability of remote team members due to limited resources in the projects, it was not possible to interview project members from all sites.

Additionally, the travel cost of researchers has also been a challenge. Therefore, researchers seek alternative methods to reduce costs and increase the reach of their data collection [2]. One approach is using video conference interviews, which is a growing trend since the technology has made communication over distance much easier and convenient [31]. However, people may respond differently when using technology [2], than they would if they were faced with the same interview question in person.

Motivated by the need for more studies on GSD, the challenge of conducting such studies, and the importance of onboarding in GSD teams, we ask the following research question:

RQ: How to onboard distributed team members?

In addition, we want to share and reflect on challenges and good practices when studying globally distributed teams.

The rest of the paper is organized as follows. In Section 2, we present related work on onboarding and GSD project research. In Section 3, we provide the background of the studied empirical case

EASE 2020, April 15–17, 2020, Trondheim, Norway

ACM ISBN 978-1-4503-7731-7/20/04…$15.00
https://doi.org/10.1145/3383219.3383235

and research methodology. Section 4 describes the results of the empirical study, followed by a discussion in Section 5, in which we answer our research question and offer recommendations and implications for research. Finally, key conclusions and directions for further research can be found in Section 6.

2 RELATED WORK

In this section, we give an overview of challenges in globally distributed projects, and a introduction to onboarding. Onboarding is the topic being studied in our case study presented in the result section. Finally, we discuss challenges of studying global software development teams.

2.1 Challenges in Globally Distributed Projects

A GSD project consists of distributed teams who have team members collaborating on a common software project while working across geographic, temporal, cultural, political, and organizational boundaries to accomplish inter-dependent tasks. Some of the challenges experienced by GSD projects are related to cultural differences, different time-zones, values, and norms, which often results in challenges regarding transferring knowledge across remote sites [18]. Hole and Moe [10] found that a high level of trust is essential to reduce the need for standardization and direct supervision when coordinating work in a GSD project, and therefore trust increases the speed of communication and feedback. Further, they found that electronic chatting supports fast feedback and fast communication.

Moe and Smite [15] found the key factors to cause a lack of trust in GSD projects to be poor socialization and socio-cultural fit, increased monitoring, inconsistency and disparities in work practices, reduction of and unpredictability in communication; and a lack of face-to-face meetings, language skills, conflict handling, and cognitive-based trust. The effect of lacking trust was a decrease in productivity, quality, information exchange and feedback, morale among the employees, and an increase in relationship conflicts. It has been found that teams involved in distributed development have a high turnover [23]. To create high-performing GSD teams, trust must be built when new people join through a good onboarding process.

2.2 Onboarding

When a remote team is added to an ongoing project, this often leads to lower productivity [22], as the new team and team members need to climb up the learning curve. The learning process may take up to three years [24]. Therefore effective onboarding is important for GSD success. Onboarding is a process to help new employees adapt to social integration and learn the expectations of their new jobs quickly and smoothly [1]. Klein et al. [13] presents four distinct perspectives for onboarding research: *Stages* through which newcomers progress, *Actors* involved with the onboarding of newcomers, *Tactics and practices* employed by organizations for onboarding newcomers and *Content* to be learned by newcomers during the onboarding. Since this paper report on experiences on conducting a case study in a GSD setting using onboarding of remote developers and testers as a context, we elaborate further on the perspective presented by Klein et al. [13] as *tactics and practices*, by describing the main models of onboarding. First, we will describe Jones' model [11] before we present, in detail, Bauer's model [1] which was built upon Jones'.

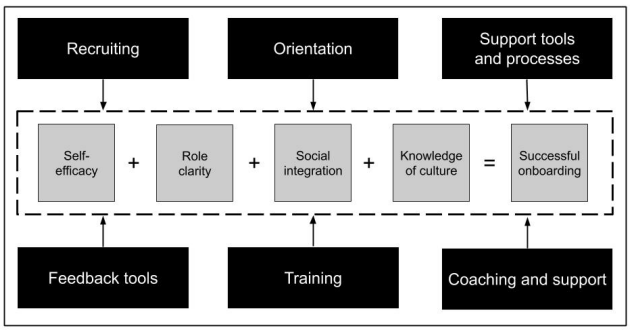

Figure 1: A model of onboarding, based on [1]

Jones [11] assumes that there are only two dimensions of onboarding tactics; *institutionalized* and *individualized*. The first occurs when tactics are implemented in structured programs, and newcomers receive formal group orientation and mentoring. The second takes place when newcomers start working from the beginning and must learn the norms, values, and expectations on-the-fly. Institutionalized onboarding is related to formal tactics, while individualized onboarding is related to informal tactics [3, 11].

Extending on Jones' work, Bauer developed a model, see Figure 1. Bauer [1] mentions Jones's [11] two dimensions, but presents them as formal (institutionalized) and informal (individualized) onboarding. She points to research that has found that companies that go for a formal onboarding have more efficient employees than those who go for an informal onboarding. Therefore, her model is based on institutionalized tactics [1]. There is a broad consensus in research that onboarding has four distinct levels, known as the Four Cs, which are the building blocks of successful onboarding [1, 11, 12]:

- *Compliance* is related to teaching employees basic legal and policy-related rules and regulations.
- *Clarification* is related to ensuring that newcomers understand their new jobs and what is expected of them.
- *Culture* is related to providing newcomers with a sense of organizational norms, including both formal and informal.
- *Connection* is related to the interpersonal relationships and information networks that newcomers must establish.

The success of an onboarding strategy is related to short-term and long-term outcomes. Short-term outcomes are associated with the adjustment of new employees to their new jobs. New employees go through a series of four adjustments [1]:

- *Self-efficacy*, which is about the extent to which the new employee feels comfortable to start work in the new job.
- *Role clarity* is about the extent to which the new employee understands their role and expectations in the company.
- *Social integration* is about the extent to which the new employee feels socially comfortable and including both of his colleagues and managers.

This is the accepted version of the publication. The final version is published in
EASE '20: Evaluation and Assessment in Software Engineering
https://doi.org/10.1145/3383219.3383235

- *Knowledge of culture* is about the extent to which the new employee understands the culture of the organization (including politics, values, and company language) and also to what extent the new employee fits into this culture.

Table 1: Six activities that affect the onboarding

Category	About
Recruiting	This is often a much more significant activity than many companies declare, and it is important to specify both soft skills and hard skills required of the developers and testers [6, 7]. Including recruitment in the onboarding process give the new employee both more precise and higher amount of information about the company and the job [10].
Orientation	Some form of formal introductory program is used to help new employees understand the most critical aspects of their new job and business, as well as company culture and values [13]. Bauer [1] also adds that it makes the new employees feel welcome by presenting them to others within the organization.
Support tools and processes	Of great value for the onboarding to be successful. Bauer [1] argue that a written onboarding plan: a formal document that should include timelines, goals, responsibilities, and support for each new employee. The plan is important to help them understand what to do and what kind of help to expect. The most effective onboarding plans are often in writing, communicated to the entire company, and continuously followed up.
Feedback tools	New employees need continuous feedback and guidance to understand their employees. During the actual onboarding, this activity is twofold; new employees apply for and receive feedback.
Training	A company must give a new employee the confidence, clarity, and skills needed to succeed in the company. Training can include training in both hard and soft skills, depending on the employee's ability to cope with the demands of the job.
Coaching and support	Mentors can teach newcomers about the company, provide advice and help with job instruction [3]. Bauer points out that mentoring programs and the opportunity to attend informal meetings with colleagues help new employees more easily adapt to the new job [1].

The extent to which these adjustments have been achieved should indicate how successful the onboarding has been. Furthermore, Bauer presents six activities that affect the adjustments, see Table 1.

2.3 Studying GSD Projects

Conducting studies in a GSD context has been found to be challenging. Access to data sources is one challenge. Britto et al. [3] argue that a long term relationship with the distributed company is one reason for getting access to remote team members and managers. While the study by Moe and Smite [15] reported on the importance of the same type of relationship with a GSD company, the researchers could not get access to some of the remote team members because the projects had ended when the data was collected and the managers did not let the developers participate in the interviews. One reason could be that the GSD setup was motivated by saving cost, and allocating time for interviews would increase the cost.

Further, Moe and Smite [15] found lack of trust to be a problem between sites, therefore, letting researchers from one site access another site's developers in a low-trust situation could be seen as a problem. A high level of trust is, therefore, essential in such studies. Stray et al. [30] studied 12 teams in four countries and found that when they traveled, the socializing with the participants during lunch, dinners and social events increased the trust and gave valuable insight in the distributed teams. One reason, as suggested by [2], was that the participants became aware of the similarities between the researchers and the participants. While it is valuable for researchers to travel in a GSD study, a challenge is related to the cost of conducting the research. Traveling to a remote site requires time and money.

3 RESEARCH METHODOLOGY

3.1 Research context

Case study is a suitable research methodology for GSD research since it studies contemporary phenomena in its context [32]. In order to draw valid conclusions when aggregating evidence from a case study, it is important to describe the context in which the study were conducted [20].

The case company, hereafter called "Norbank," is a bank based in Norway. In line with the IT industry in general, this company experienced an increased demand for more developers and testers. In spring 2018, the management of the IT department decided to expand with developers and testers from Portugal.

Norbank signed a contract with an international company, called "The agency", that would build up a separate department for the bank in Portugal that would be managed by The agency. Since the employees were not employed by the bank, it was not an insourcing setup, rather an outsourcing model. However, because of the active involvement from the bank, we classify the relationship as a partnership model.

The partnership model can be explained as a model where The agency hires the new team members, and are responsible for them, but the bank is involved in such a way that team members feel like they are working for the bank. The bank had one-to-one conversations with all the employees in Portugal, showing that they took

This is the accepted version of the publication. The final version is published in
EASE '20: Evaluation and Assessment in Software Engineering
https://doi.org/10.1145/3383219.3383235

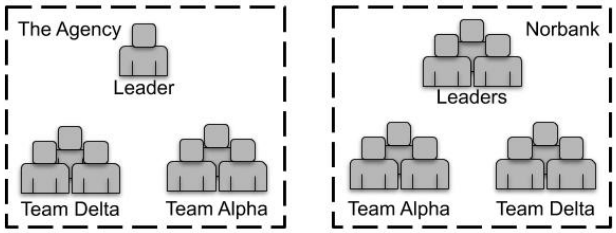

Figure 2: The agency and Norbank

Table 2: Case study team composition.

Team Alpha			**Team Delta**		
Country	**Role**	**#**	**Country**	**Role**	**#**
Norway	Team Lead	1	Norway	Team Lead	1
	Delivery Man.	1		Tech Lead	1
	Product Spec.	3		Developer	2
	Tech Lead	1			
	Developer	5			
	Tester	2			
Portugal	Tech Lead	1	Portugal	Tech Lead	1
	Developer	3		Developer	4
	Tester	1			
	Total	**18**		**Total**	**9**

full responsibility for making the remote team members feel that they were a part of the bank.

We have named the teams Team Alpha and Team Delta, see Figure 2. Team Alpha is a cross-functional team, which means that the team is comprised of members with different technical backgrounds. The team consists of one team leader, one delivery manager, three product specialists, three testers, and ten developers across remote sites who work on both front-end and back-end solutions for Norbank. Team Delta is a functional team consisting of one team leader and eight developers across remote sites who work with IT-architecture solutions for Norbank. See details in Table 2.

3.2 Data collection

There are several different sources of data that can be used in a GSD case study. Since a major weakness of case studies is that the data collection and analysis is more open to interpretation and researcher bias [5], it is important to use several data sources in order to limit the effects of one interpretation of one single source. Runeson and Höst [21] found interviews, observations, archival data, and metrics to be applicable to software engineering case studies. For this case study, we conducted interviews and observations and analysed Slack logs and documents. See details in Table 3.

Some of the challenges of studying distributed teams can be addressed if the researchers can travel to all sites where the team members work. Otherwise, they need to conduct high-quality video-interviews. In our study, some interviews were conducted face-to-face (8) and the rest through communication tools such as Skype or GoToMeeting (10). All the interviews were semi-structured, and all the face-to-face interviews had an interview guide with questions regarding topics from Bauer's model.

It is essential that the interviewee is ensured confidentiality and that the interviewee trusts the interviewer. Trust has been found to be improved by face-to-face meetings [15]. It is not recommended to start the interview with sensitive questions (e.g. concerning opinions of colleagues, why things went wrong) before a climate of trust has been obtained. Before we interviewed people using video, we met them in meetings and workshops to introduce them to the research and to build trust.

Observations can be conducted in order to investigate how a certain task is conducted by software developers. In today's current software development methods, there are many meetings (daily meetings, planning meeting, retrospective meetings), which makes observing meetings an important type of data. In meetings, participants interact with each other and thus generate information about their teamwork and how the collaboration can be improved [29].

Steinfield et al. [26] highlights the importance of using triangulation across disparate sources of data when studying distributed teams, and the need to verify observers' impressions across locations to avoid a partial view of a distributed group interactions. However, observing several sites involved in the collaboration, e.g., at the same time is troublesome because of the need for many researchers and the need for travel. We observed team members from both sites in meetings and workshops, but only in one location (Norway). These observation sessions lasted from 10 minutes to 1 hour. We sat behind the developers, taking notes on their dialogues, interactions and activities.

To achieve triangulation, adding more data sources is essential. Archival data is an important data source. Examples of archival data are meeting minutes, documents from different development phases, organizational charts, financial records, and previously collected measurements in an organization. Many software teams use social software [28] which document the communication and coordination. Since distributed teamwork requires that the team members communicate via electronic communication tools (e.g., Slack, Jira), much of the activity that the researchers are interested in can be retrieved from such tools. Other types of archival data refers to, meeting minutes, development process, organizational charts, financial records, and previously collected measurements in an organization. We relied on meeting minutes, onboarding and training programs and data from the use of Slack.

We used mainly data from interviews when exploring the different phases of the onboarding process, but we also relied on documents with schedules from the introduction program to prepare for the interview and to cross-check our data. Being able to use the archival data to prepare, helped the researchers to focus the interviews on certain aspects. Observational data was used to understand the teamwork and collaboration across sites.

3.3 Data analysis

To analyze the interviews and observations, we used Nvivo, a tool that allows the user to select text and link it to created nodes (categories), which makes it easier to categorize and compare the data. The third author coded the material, using open coding or "post-form" coding, looking for material related to the elements in Bauer's

This is the accepted version of the publication. The final version is published in
EASE '20: Evaluation and Assessment in Software Engineering
https://doi.org/10.1145/3383219.3383235

Table 3: Empirical data collection and analysis

Data Source	Location	Time	Participants	Data gathered
Interviews and informal conversations	Norway (communication tools)	Mar 2019 - jun 2019	Interview: Norbank, 6 participants (1 product owner, 1 head of finance, 2 team lead, 1 delivery man. and 1 product specialist)	Planning, execution and experiences from the onboarding process
	Portugal (communication tools)		Interview: The Agency, 3 participants (2 tech leads and two interviews with 1 site manager)	
	Norway (face-to-face)	Apr 2019	Interview: Norbank, 8 participants (1 prod specialist, 5 developers and 1 tester)	
	Norway and Portugal (communication tools)	Aug 2018 - Aug 2019	Informal conversations on preparation of the onboarding process and status on the onboarding process. 6 conversations with HR, and development lead	
Observations	Norway	Apr 2019	Norbank, both teams, 1 stand-up meeting each	How does the communication tools work, how is the collaboration, communication etc.
	Norway and Portugal	Sept 2018	Workshop on autonomous teams with developers from both sides during a visit in Norway	
Slack analysis	Norway	Sep 2018- Mar 2019	Team Alpha and Team Delta	Activity level in channels, conversations supporting data collected from other sources
Documentation	Norway	Sep 2018- feb 2019	System descriptions, specific role descriptions, list of team members, years of experience in the company, and process descriptions	Context of software development, onboarding process (costs, activities), activities and team information

model described in chapter 2.2. During the coding process, all authors had regularly meetings discussing the findings.

We received Slack logs from eight different channels where each half were used by one of the teams. The format of the files was of the JSON type, which made it not directly legible. For example, a message had a user ID (user), timestamp (ts) and a message (text) such as this:

"user": "X"
"ts": "00:14PM"
"text": "I was trying to call you on skype, <@Y>."

To create Figures 3 and 4 on Slack activity, we had to look up the user ID against a separate file (users.json) to find which nationality this person belonged to, and then find the number of instances of this unique ID. We created a table in Excel with all the unique IDs and the number of instances and generated the Figures using these charts.

4 RESULTS

For this case study, we wanted to find out how to onboard distributed team members, and whether Bauer's onboarding model [1] is applicable for globally distributed teams. Additionally, we were interested in uncovering good practices when studying globally distributed teams.

4.1 Onboarding new team members

In order to investigate whether Bauer's onboarding model [1] is suitable for globally distributed teams, we looked at the extent to which the case company has taken into account, or has achieved, the different parts of her model. This includes the six activities described in the background chapter that affect the onboarding: recruitment, orientation, training, coaching and support, support tools and processes, and feedback, as well as the four adjustments: self-efficacy, role clarity, social integration, and knowledge of the culture.

In Table 4, we have plotted the extent to which the different parts of the model have been taken into account by the two teams. We categorized the findings to show whether the onboarding practices had been covered into 1) yes, 2) some and 3) little/none. The table was created by comparing all data sources, including face-to-face and Skype interviews, analysis of Slack (both in the form of reading logs and by making usage graphs), from observations of the Norwegian developers' workplace and stand-up meetings, and received documents.

As can be seen from Table 4, there was a slight difference between the two teams. Both teams made use of a mentoring program (tech people in Norway mentoring team members in Portugal), but this was somewhat better used by Team Delta, which probably explains the difference. We will come back to the mentoring program later. In general we found that Norbank covered, or to some degree covered,

This is the accepted version of the publication. The final version is published in
EASE '20: Evaluation and Assessment in Software Engineering
https://doi.org/10.1145/3383219.3383235

Table 4: Result of onboarding: Team Alpha (α) & Delta (δ)

Activities and Adjustments	Covered by onboarding practices		
	Yes	Some	Little/ None
Recruiting	α,δ		
Orientation		α,δ	
Support tools and processes		α,δ	
Feedback tools		α,δ	
Training		α,δ	
Coaching and support	δ	α	
Self-efficacy	δ	α	
Role clarity	α,δ		
Social integration	α,δ		
Knowledge of culture	α,δ		

all of the activities and adjustments presented by Bauer [1]. This is an indication that the concepts in Bauer's onboarding model are applicable for globally distributed teams.

In the following, we report results from the onboarding process, starting with how the process was organized.

4.1.1 Recruiting. Norbank wanted to hire all the Portuguese developers at once so that they could do an institutionalized onboarding. They succeeded with the goal, and everyone joined at the same time, except for one person that was hired some weeks before the rest. Such an onboarding process, as mentioned in the background, is a structured program where newcomers receive formal group orientation and mentoring.

From both face-to-face and Slack interviews, we found that Norbank perceived the recruitment process as successful. The Agency had long experience regarding recruitment in remote locations. One of the Norwegian leaders said that The Agency had a comprehensive recruitment process and was capable of hiring skilled software developers. One essential criterion in the recruitment process was to get people that had values that were compatible with the existing values of the bank. One said: *"We have succeeded in getting people who are easy to relate to, so it worked well"* and *"what was good was that they did a good process in advance when hiring them"*.

The Agency was responsible for the first phase of the recruitment, finding the candidates and conducting the first interviews. In the second and third interview round the bank contributed with their own managers and developers when giving a brief introduction to the company and on the technical part of the interview. As one Norwegian developer said: *"I participated in many interviews and gave feedback on whom we should proceed with technically"*. Already in the interview phase the onboarding had started, as the Portuguese became aware of what was expected from their role in the company and was introduced to the the company's culture. At the same time, the final interview round also allowed the Norwegian developers to get to know their new colleagues. While the results of the hiring process were perceived as very satisfactory, the recruitment process ended up being a lot more time-consuming and expensive than expected.

4.1.2 Orientation. All the Portuguese developers went through a three-week-long visit in Norway, where they participated in various courses and social activities. The bank sector is a highly regulated industry. Therefore it is vital to teach employees legal and policy-related rules and regulations. The feedback was mostly positive regarding the three weeks visits; they learned a lot, and it was perceived as very positive to meet each other face-to-face for such a long period. The visit enabled the remote developers in building strong networks. While all the activities were seen as good, several commented that the stay was one week too long. Two reasons were mentioned. First, covering a lot of new technical and domain knowledge over three weeks was perceived as too much. There was too little time to start applying the new knowledge. Second, they miss their family and friends. As one Portuguese developer said when reflecting on his visit to Norway: *"We all agree that two weeks is more than enough for the onboarding."* This introduction program can be categorized as "Orientation" in the model by Bauer.

4.1.3 Support tools and processes. As mentioned earlier, both teams introduced a buddy-program, or better known as a mentor-program [1, 9]. Each Norwegian developer was initially set to be a mentor for one or two Portuguese developers. During the first stay in Norway, each mentor worked with the person they were responsible for on a task. After the stay, the mentor was the one responsible for delegating tasks, follow up on the tasks and answer questions if anything were unclear. The goal of the mentoring program was to offer better training and make sure all remote developers always had someone to ask for help. The program worked best for Team Delta. The reason was that in Team Delta, the developers had more time to do the mentoring. Team Alpha was given some urgent assignments, which resulted in a need to rearrange who was the actual mentors. Several developers had to stop mentoring, which resulted in one Norwegian developer becoming the responsibility of mentoring five Portuguese developers. Mentoring five new people was probably too much, especially since team Alpha was working in a more complex domain than Team Delta.

4.1.4 Feedback tools. The feedback activity is twofold; new employees both give and receive feedback. It emerged that Norbank has made sure that both types of feedback occurred during the onboarding. For example, Norbank held weekly feedback meetings on all aspects of the work and collaboration. Examples were retrospectives and "Improvement Friday" where members from both sites discussed what worked well and what could be improved. Further, leaders from Norway organized feedback meetings when they visited Portugal to identify what Norbank could do better to support the newly onboarded team members. One Norwegian manager commented: *"I've been watching how they work and I have attended standup via Webex meeting with the team down there. I have organized some meetings to give them the opportunity to tell me what they think we can do better*". Additionally, Norbank held one-on-one conversations every month with the team members in Portugal. A Norwegian leader said that if you want to include distributed team members, then it is important to treat them in the same way as the local ones. Norbank's system for providing and seeking feedback gave important input so Norbank could make the necessary changes to work processes, authority and decision

This is the accepted version of the publication. The final version is published in
EASE '20: Evaluation and Assessment in Software Engineering
https://doi.org/10.1145/3383219.3383235

making to improve the work situation. Offering many feedback tools made it easy to ask questions and give feedback.

4.1.5 Training. While the team frequently communicated, there was a need for regular visits. Both management and developers from Norway visited Portugal several times after the introduction program had finished. The management was there to follow up on the Portuguese developers. The Norwegian developers and architects were there primarily to help solving technical problems and to continue the training. When asked what visits were most important the Portuguese expressed the need for more technical personnel and less managers. Another important mechanism for training is the code review process. When a Norwegian developer reviews the code and gives feedback to an Portuguese developer, learning occurs. The Portuguese also came for visits in Norway, and we as researchers used such an opportunity to get to know them. Face-to-face meetings enable the creation of trust, which is important for good interviews. Ideally, we should have interviewed the developers while we met with them in Norway. However, the developers had too much on their agenda during their hectic visits, so there was no time for interviews.

4.1.6 Coaching and support. The mentor-program affected many of the elements in the Bauer model. The difference in how the two teams exploited the mentoring program, made Team Delta better on both coaching and support and self-efficacy (shown in Table 4). Most of the data used from evaluating the mentoring program was from interviews. However, we also relied on Slack to understand communication between the distributed teams. By studying Slack data we found that Team Alpha asked more questions than team Delta. This could also be explained by the fact that their domain was complicated, or that the developers had too little mentoring or guidance, resulting in the need for more questions.

4.2 Onboarding challenges

When investigating the onboarding challenges, we found that the four most mentioned challenges were missing domain knowledge (76%), communication tools (47%), unclear tasks (41%) and language barrier (35%). Because the challenges with the domain were anticipated (because of the Norwegian Bank domain), we decided to explore the challenge with communication tools as it also influences several of the categories in Bauer's model. Slack was one important communication tool. We found that that among the six most active users (number of messages) in all the public Slack channels (8), three of them were Portuguese (see Figure 3). It was people with responsibility or leader roles and people who needed help who were most active and therefore had a higher communication activity than others.

Figure 4 and 5 shows that the Portuguese developers were more active in Team Alpha Slack channels than in Team Delta's. The difference in activity level can probably be explained by the fact that we were told, in the interviews, that the tasks were clear, and the domain was easier for Team Delta than for Team Alpha. So it seemed that less complex tasks result in less need for interaction.

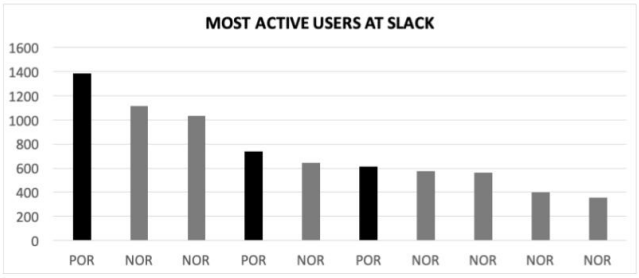

Figure 3: Slack activity: ten most active users

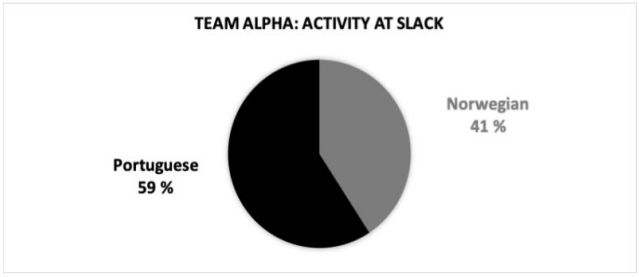

Figure 4: Slack activity: country/team (Team Alpha)

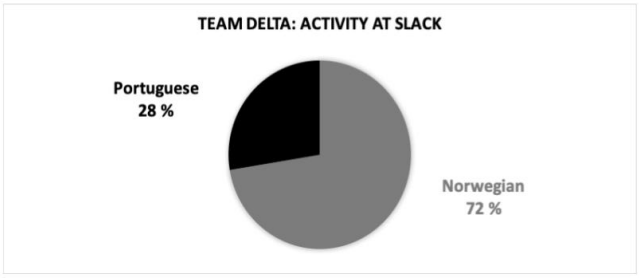

Figure 5: Slack activity: country/team (Team Delta)

The agile way of working was also a reason for the need for frequent communication and collaboration on the team level. Moreover, because of distribution, tools were needed. When the developers were distributed, they frequently participated in joint virtual meetings between the sites (e.g., daily meetings, planning meetings and retrospective) and one-to-one meetings by the use of video. While the company had access to good video systems, the teams experienced problems because of delays on the connection. The delay reduces the quality of the communication, because it became challenging to have good conversations cross sited. We as researchers experienced the same challenges when performing remote interviews with both Norway and Portugal. E.g., a delay with Portugal resulted in difficulties achieving a good flow in the interview and made it harder to explore more challenging and sensitive problems. Another problem was that the equipment sometimes did not work. We found the following conversation in one Team Alpha Slack channel:

This is the accepted version of the publication. The final version is published in
EASE '20: Evaluation and Assessment in Software Engineering
https://doi.org/10.1145/3383219.3383235

X (00:14PM): I was trying to call you on skype, Y.
X (00:14PM): Webex isn't working today
Y (00:15PM): We ran the standup without you guys. Sorry.

The interviews made us aware of the communication tool problem, and the Slack chat logs confirmed the finding. A third major technical problem was that some Norwegians did not have a good headset, which is important for being able to do a conversation without the need for booking a room, and for good sound quality. The problem with the sound quality was evident when comparing transcribing interviews with people not using and using a headset. It took longer time (because of bad quality) and some information was impossible to retrieve from the interviews.

5 DISCUSSION

Effective onboarding is a key issue in being able to succeed in GSD projects. There is a lack of research-based advice on how to onboard new developers to distributed projects. Further, few studies describe the experience with conducting GSD studies. Driven by our research question – *RQ1: How can developers be onboarded in distributed teams?* we have reported findings from a case study of onboarding remote developers from Portugal to a Norwegian bank. In the following, we discuss our findings with regard to the model suggested by Bauer [1] before discussing the use of different types of data collection in research on globally distributed teams.

5.1 Onboarding distributed team members

Norbank tried to hire all the Portuguese developers at once so that they could do a formal (institutionalized) onboarding, which is claimed to be a preferred method [1, 11]. The approach is a structured program where newcomers receive formal group orientation and mentoring [11].

There are four important components of formal onboarding [1, 11, 12]: Compliance, Clarification, Culture, and Connection. The financial domain is highly regulated, which was one reason why many resources were used on compliance - informing about legal and policy-related rules and regulations. Interviews in the hiring process, the use of mentors, and the tool Slack for mutual adjustment, all supported frequent clarifications and helped the newcomers understand their new job and expectations. During the recruitment process, one important criterion was to find people that matched the culture in the bank. Therefore, all interviews and visits focused on communicating the values and culture and on giving an insight into the organization's norms. The most important measure to build strong networks for the newly recruited people was the three-weeks visit. During this visit, the developers were introduced to important stakeholders, experts, and other teams. Further, they spent much time together with the Norwegian team members. When they got back, the frequent visits helped in strengthening the networks. Frequent visits for strengthening the communications, and conducting code reviews for learning and frequent feedback are in accordance with the study of Moe et al. [14].

As shown in Table 4, all the adjustments suggested by Bauer [1] were achieved, and all activities had been performed. Therefore, we argue that the case company performed an onboarding process of distributed teams in a partnership context, according to what is recommended. Our findings suggest that Bauer's general model of onboarding is applicable for globally distributed teams. One reason why the model seems to fit is that all the developers were recruited at the same time. Our findings are in accordance with the study of Britto et al. [3], who found that while the model is applicable, it needs to be adjusted, and it might be economically unfeasible to use it for onboarding a single developer.

While the onboarding process was seen as successful, we encountered some challenges when onboarding globally distributed teams. One of the biggest challenges was problems with communication tools. Challenges with the technical equipment that was used in virtual meetings resulted in lost working time and poor communication flow between the members. While the video communication was troublesome, Slack seemed to help a lot as a communication platform. Electronic chatting supports fast feedback and fast communication, which again strengthens the level of trust in a distributed project [10]. Bauer[1] did not mention the use of communication tools in her research, and this was probably because her research did not include distributed teams. Another challenge was the mentoring process. As Britto et al. [3] found, Norbank incorporate the new employees into already existing teams and relied on the ability of the experienced developers to help the newcomers in their learning process. However, since the experienced developers in one of the two team suddenly did not have the time to continue the mentoring job, one person was given the responsibility of mentoring 5 persons. Which probably was too much.

5.2 Studying globally distributed teams

We now turn to reflect on challenges and good practices when studying globally distributed teams. While it was not possible to visit all locations, which created some challenges, we found it helpful to have met the people we interviewed in advance during a company gathering. Further, we found that Slack was a valuable tool that we needed to investigate to understand the collaboration and communication going on in the organization. There exist a few studies of the use of Slack in GSD [27][28], however, few recommendations exist on how to analyze the data stores on the platform. Based on our experience in the results section, we have summarized benefits and challenges with the use of various data sources in Table 5.

5.3 Limitations

The main limitations of our study are the single-case design and the possibility of bias in data collection and analysis. The fact that we used a single-case design makes us more vulnerable to bias and eliminates the possibility of direct replication or the analysis of contrasting situations. Therefore, the general criticisms about single-case studies, such as uniqueness and special access to key informants, may also apply to our study. However, our rationale for choosing Norbank as our case was that it represents a critical case for investigating onboarding in globally distributed teams. Further, we use Norbank to investigate whether Bauer's onboarding model [1] is suitable for globally distributed teams. Our mode of generalization is analytical, i.e., we used a previously developed model as a template with which we compared the empirical results

This is the accepted version of the publication. The final version is published in
EASE '20: Evaluation and Assessment in Software Engineering
https://doi.org/10.1145/3383219.3383235

Table 5: Benefits and challenges of the use of various data-sources in GSD

Data source	Benefits	Challenges
Interviews (online)	Can conduct the interview when needed	Hard to build trust even when using video
	Can be conducted with less time and money wasted due to travel	Delays in the conversation if there are network challenges.
Interviews (face to face)	Easier to gain the needed trust	Need to be planned in advanced.
		Travel budget
Observations	Being able to observe coordination and communication in practice	Need to be planned in advance
		Travel budget
		Observing a meeting from all sites at the same time
Archival data (Slack logs)	Being able to collect from all sites on communication and collaboration. We know that users have different usage level	Analyzing Slack logs over time is challenging because of the amount of data and the type of data that is written in the channels.
	Can cross-check data	Graphs can describe the activity level but need to be combined with other data sources.
	All data is recorded, so you can use more time on analysing than documenting (versus observations)	Data in private channels is not accessible
Archival data (Documents)	Often quantitative data that supports triangulation when studying social phenomena	Hard to reference to
	Can cross-check data	Hard to get access to because of confidentiality
	Can reveal data that does not appear otherwise	

of the case study, which is similar to Yin's [32] concept of Level Two inference.

Another possible limitation is that we based much of our data collection and analysis on semi-structured interviews. First, we did not interview all the managers and developers. Second, there is a particularly uneven distribution of the number of interviews of informants from different countries. Fourteen interviews were conducted of Norwegian managers and developers, but only four in total from management and developers in Portugal. However, the use of multiple data sources made it possible to find evidence for practices and phenomena from more than one data source; we also observed, talked to, and interviewed the team members and managers over a period of several months, which made it possible to study the phenomena from different viewpoints as they emerged and changed.

6 CONCLUSION AND FUTURE WORK

In this paper, we investigated the strategies employed by a bank when onboarding remote software developers. Even though the bank relied on an outsourcing and partnership model (they did not hire the developers themselves), they relied on an onboarding model [1] intended for companies that are fully responsible for their employees. We evaluated the onboarding functions by using this model (ibid).

In response to our research questions, we learned that even if one organization apply the same practices and strategies for onboarding of all new employees, the adaptation of an onboarding model is affected by several factors. For example, the domain and complexity the teams operate in, the type of team, and how busy the team people are onboarded into are. One important implication of this finding is that the onboarding outcomes cannot always be predicted by management.

Facilitated by Bauer's model [1] (Figure 1), we analyzed the coverage and prevalence of the different onboarding functions. The mentoring program was one crucial activity. However, it had limited success in one of the teams because the team members suddenly did not have the time to do the job. Further, it seems that mentoring is a challenging job, and the mentors themselves need help in performing that job and balancing mentoring vs. solving their own task. We found it positive that new employees are helped into existing teams and rely on the ability of the experienced team members for speeding up the learning process. However, a potential drawback of this approach is that mentoring hinders the productivity of the mentors, who experience frustration when they are not able to do their work.

In this article, we also have described the importance of using multiple approaches to data collection when studying globally distributed teams. Triangulation with multiple sources of evidence (interviews, observations, documents, chat logs) from all sites increases the validity and allows the researchers to "see" the teams in different ways. Further, we have described the benefits and challenges of the use of various data sources in research on global software development projects and organizations. Even if the researcher is not able to travel to the remote site, we argue that the researcher should try to meet the interviewees before conducting the interviews. In most cases, the remote workers travel to the main site. If the decision is to do interviewing on Skype or similar video conference tools, it is of utmost importance that there is no time lag when talking.

Future work should explore how single individuals are onboarded when they are the only ones being recruited at a given time. There

This is the accepted version of the publication. The final version is published in
EASE '20: Evaluation and Assessment in Software Engineering
https://doi.org/10.1145/3383219.3383235

is also a need to study the mentoring process seen from the mentor. One interesting question is how mentors (who often are the most experienced developers) balance mentoring and solving their own tasks.

ACKNOWLEDGMENTS

We would like to thank the case company for providing us access to their personnel and data record, and to their sincere interest in our study. This work was supported by the A-team project, partly funded by the Research Council of Norway under grant 267704.

REFERENCES

[1] Talya N. Bauer. 2010. Onboarding New Employees: Maximizing Success. (2010).
[2] Emily S Block and Laura Erskine. 2012. Interviewing by telephone: Specific considerations, opportunities, and challenges. *International journal of qualitative methods* 11, 4 (2012), 428–445.
[3] Ricardo Britto, Daniela S. Cruzes, Darja Ŝmite, and Aivars Sablis. 2018. Onboarding software developers and teams in three globally distributed legacy projects: A multi-case study. *Journal of Software: Evolution and Process* 30, 4 (2018), e1921. https://doi.org/10.1002/smr.1921
[4] Tore Dybå, Dag IK Sjøberg, and Daniela S Cruzes. 2012. What works for whom, where, when, and why?: on the role of context in empirical software engineering. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*. ACM, 19–28.
[5] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. 2008. Selecting empirical methods for software engineering research. In *Guide to advanced empirical software engineering*. Springer, 285–311.
[6] Raluca Florea and Viktoria Stray. 2018. Software tester, we want to hire you! an analysis of the demand for soft skills. In *International Conference on Agile Software Development*. Springer, 54–67.
[7] Raluca Florea and Viktoria Stray. 2019. A global view on the hard skills and testing tools in software testing. In *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*. IEEE, 143–151.
[8] Lucy L Gilson, M Travis Maynard, Nicole C Jones Young, Matti Vartiainen, and Marko Hakonen. 2015. Virtual teams research: 10 years, 10 themes, and 10 opportunities. *Journal of management* 41, 5 (2015), 1313–1337.
[9] R. Hoda, J. Noble, and S. Marshall. 2013. Self-Organizing Roles on Agile Software Development Teams. *IEEE Transactions on Software Engineering* 39, 3 (March 2013), 422–444. https://doi.org/10.1109/TSE.2012.30
[10] Steinar Hole and Nils Brede Moe. 2008. A case study of coordination in distributed agile software development. In *European Conference on Software process improvement*. Springer, 189–200.
[11] Gareth R. Jones. 1986. Socialization Tactics, Self-Efficacy, and Newcomers' Adjustments to Organizations. (1986).
[12] Howard J. Klein and Beth Polin. 2012. Are Organizations On Board with Best Practices Onboarding? *The Oxford Handbook of Organizational Socialization* (2012). https://doi.org/10.1093/oxfordhb/9780199763672.013.0014
[13] Howard J. Klein, Beth Polin, and Kyra Leigh Sutton. 2015. Specific Onboarding Practices for the Socialization of New Employees. *International Journal of Selection and Assessment* 23, 3 (2015), 263–283. https://doi.org/10.1111/ijsa.12113
[14] N. B. Moe, D. S. Cruzes, T. Dybå, and E. Engebretsen. 2015. Coaching a Global Agile Virtual Team. (July 2015), 33–37. https://doi.org/10.1109/ICGSE.2015.26
[15] Nils Brede Moe and Darja Šmite. 2008. Understanding a lack of trust in Global Software Teams: a multiple-case study. *Software Process: Improvement and Practice* 13, 3 (2008), 217–231.
[16] Nils Brede Moe, Darja Šmite, Geir Kjetil Hanssen, and Hamish Barney. 2014. From offshore outsourcing to insourcing and partnerships: four failed outsourcing attempts. *Empirical Software Engineering* 19, 5 (2014), 1225–1258.
[17] Miriam Muethel, Sarah Gehrlein, and Martin Hoegl. 2012. Socio-demographic factors and shared leadership behaviors in dispersed teams: Implications for human resource management. *Human Resource Management* 51, 4 (2012), 525–548.
[18] Ilan Oshri, Paul Van Fenema, and Julia Kotlarsky. 2008. Knowledge transfer in globally distributed teams: the role of transactive memory. 18, 6 (2008), 593–616. https://doi.org/10.1111/j.1365-2575.2007.00243.x
[19] Ilan Oshri, Julia Kotlarsky, and Leslie P. Willcocks. 2015. *The Handbook of Global Outsourcing and Offshoring 3rd edition*. Springer. Google-Books-ID: X7LIDAAAQBAJ.
[20] Kai Petersen and Claes Wohlin. 2009. Context in industrial software engineering research. In *2009 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE, 401–404.
[21] Per Runeson and Martin Höst. 2009. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering* 14, 2 (2009), 131.
[22] Darja Šmite, Nils Brede Moe, Thomas Krekling, and Viktoria Stray. 2019. Offshore outsourcing costs: known or still hidden?. In *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*. IEEE, 40–47.
[23] Darja Šmite and Rini van Solingen. 2015. What's the true hourly cost of offshoring? *IEEE Software* 33, 5 (2015), 60–70.
[24] D Šmite and R van Solingen. 2016. What's the real hourly rate of offshoring? *IEEE Software* 33, 5 (2016), 60–70.
[25] Darja Šmite and Claes Wohlin. 2011. A whisper of evidence in global software engineering. *IEEE software* 28, 4 (2011), 15–18.
[26] Charles Steinfield, Marleen Huysman, Kenneth David, Chyng Yang Jang, Jan Poot, M Huis in't Veld, Ingrid Mulder, Erik Goodman, John Lloyd, Timothy Hinds, et al. 2001. New methods for studying global virtual teams: towards a multi-faceted approach. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*. IEEE, 10–pp.
[27] Margaret-Anne Storey, Alexey Zagalsky, Fernando Figueira Filho, Leif Singer, and Daniel M. German. [n.d.]. How Social and Communication Channels Shape and Challenge a Participatory Culture in Software Development. 43, 2 ([n. d.]), 185–204. https://doi.org/10.1109/TSE.2016.2584053
[28] Viktoria Stray, Nils Brede Moe, and Mehdi Noroozi. 2018. Slack Me If You Can! Using Enterprise Social Networking Tools in Virtual Agile Teams. (2018).
[29] Viktoria Stray, Nils Brede Moe, and Dag IK Sjoberg. 2018. Daily stand-up meetings: start breaking the rules. *IEEE Software* (2018).
[30] Viktoria Stray, Dag IK Sjøberg, and Tore Dybå. 2016. The daily stand-up meeting: A grounded theory study. *Journal of Systems and Software* 114 (2016), 101–124.
[31] Jessica R Sullivan. 2012. Skype: an appropriate method of data collection for qualitative interviews? *The Hilltop Review* 6, 1 (2012), 10.
[32] Robert K Yin. 2017. *Case study research and applications: Design and methods*. Sage publications.

This is the accepted version of the publication. The final version is published in
EASE '20: Evaluation and Assessment in Software Engineering
https://doi.org/10.1145/3383219.3383235