The Role of Responsiveness to Change in Large Onboarding Campaigns

Darja Smite[1,2](✉) and Nils Brede Moe[1,2]

[1] Blekinge Institute of Technology, Karlskrona, Sweden
darja.smite@bth.se, nils.b.moe@sintef.no
[2] SINTEF Digital, Trondheim, Norway

Abstract. Onboarding is a process of organizational socialization of the new hires, that includes recruitment, orientation, training, coaching and support. While onboarding individuals into an organization is a rather straightforward task, little is known about 1) onboarding hundreds of developers and 2) doing it on a distance in outsourcing situations. Furthermore, the subject of sustainable growth with respect to organizational capabilities and culture is often overlooked. This paper reports findings from an exploratory multi-case study of two large onboarding campaigns. We collected empirical data from interviews, retrospectives, onboarding documentation and onsite visits. Based on the empirical study, onboarding hundreds of software engineers in a complex agile product development environment which lacks documentation and puts high demands on engineers' knowledge and skills is a challenging and costly endeavor. To save the costs and for practical reasons, large-scale onboarding is organized in batches with the first batch trained onsite, and the later batches trained internally. We report challenges faced in the two cases and discuss possible solutions. One core finding is that a good plan combined with the organizational agility, i.e., the responsiveness to change, together with organizational maturity determined the success of organizational scaling. The presented cases contribute to the scarce research on knowledge transfer and onboarding in a large-scale agile context.

Keywords: Onboarding · Scaling · Training · Teams · Large-Scale Agile · Software Engineering · Empirical · Case study · Sustainable Organizational Growth

1 Introduction

To satisfy the need for new features and services at a faster speed, agile software companies face the need to scale their development capacity. However, organizational scaling carries a number of challenges. The first challenge occurs when a company needs to scale fast. Since employing many competent engineers with technical and domain expertise might be difficult and costly, many opt for university graduates and junior engineers instead considerably lowering the competence level and diluting the organizational culture, which also negatively impacts overall performance and quality. The second challenge relates to the difficulty of employing local engineers [1]. As a result,

C. J. Stettina et al. (Eds.): XP 2023, LNBIP 475, pp. 132–148, 2023.
https://doi.org/10.1007/978-3-031-33976-9_9

many companies "scale out" by either teaming up with external suppliers nationally and internationally or globalizing via acquisitions and setting up own subsidiaries. However, barriers such as domain complexity, differences in processes and even in the use of agile processes (e.g., timeboxed vs. flow-based development), and employee turnover all create challenges when scaling out [2]. Further, cultural barriers have been found to be impediments to agile software development practices [3]. One particular significant factor leading to success when scaling is effective onboarding [4, 5].

Onboarding is the process of supporting new employees regarding their social and performance adjustment to their new job [4]. It is also known as organizational socialization. In software development companies, onboarding of new developers happens to replace retired developers, to replace developers who left or will leave, to replace existing developers who changes roles, to offload existing developers or increase capacity, or to incorporate new people with unique expertise who bring new ideas and thus help a company to innovate [5].

Traditionally recruitment has been studied as a process that happens within corporate boundaries. The meaning of the "onboarding" process is thus commonly related to facilitating the transition of the newcomers from being organizational outsiders to being insiders [6]. Onboarding can also happen in a project or a team by relocating already employed staff. Large emphasis in traditional onboarding is put on the social integration into a team and ensuring mentoring and support from teammates. When onboading an individual to replace a team member who retires, takes up a new role within the company or leaves the company, training is often performed by the person who is to be replaced, or by immediate teammates. Onboarding in general is a well-researched field with studies of diverse type of professions [4, 7–10], and some empirical studies about onboarding newcomers in software companies [5, 11–13] as well as recent interest in the context of agile software development [14].

Large-scale agile involves hundreds of engineers and tens of teams spread not only geographically but also across legal boundaries of a firm. Onboarding in such contexts may thus require recruiting and socializing engineers at a remote subsidiary or at a sub-contractor, which is obviously a challenging endeavor. The reasons for this are twofold. First, newcomers need to be integrated not only into the boundaries of the subsidiary or the outsourcing supplier, but also into the assignment or the project led by the organization. Second, all recruitment and onboarding activities typically happen on a distance, which is not a straightforward task with little available knowledge on the topic [5].

In this paper, we present two empirical cases of large onboarding campaigns in the context of global agile software engineering and discuss the practices that can facilitate the onboarding. Our research is driven by the following question: **What are the strategies for large onboarding campaigns?**

2 Background

Onboarding models suggest that organizational socialization usually starts with recruitment and continues with orientation, training, coaching and support [4]. While the recruitment and orientation can be viewed as an entrance to the company, training focuses on the readiness of the newcomers for the particular job within the boundaries of, e.g., an assignment, or a team. Similarly, training is given to existing company employees, when they are transferred to new assignments within the company. Coaching and support are common means to help employees climb up their learning curve and improve performance. Just like agile coaches, engineering coaches paired with new hires target performance strategies by coaching to improve work processes, and by teaching new techniques and providing technical guidance [15]. Training as well as coaching and support during the onboarding process are the two well-researched areas that in the area of software engineering are primarily researched in open-source projects [13, 16–18]. Further, research shows that providing support to new hires plays even a more prominent role in onboarding success, than training [13].

Success of onboarding is often associated with the extent to which an organization succeeds to ensure that new employees possess different knowledge and skills [4]:

- Basic legal and policy-related rules and regulations,
- Understanding of their new jobs,
- Grasping organizational norms and culture,
- Establishing the necessary interpersonal relationships and information networks.

Social integration has recently gained attention. Onboarding is said to be more successful when the assimilation and integration of the new hires at the new job is assisted through educating them about the cultural norms, clarifying the expectations, helping to recognize formal and informal channels and assisting the integration into established communities [4, 9, 19]. Strong social connections are also said to improve productivity long-term [20]. In addition, in an agile environment the new hires might need to also adjust their behavior to the agile values and undergo a mindset change [14].

The emphasis, formality and duration of the different onboarding functions, may differ in different recruitment situations. For example, the job specificity and the entry competence of a newcomer may require less or more training, mentoring and support. Similarly, previous experience and expertise, whether specific for the job, related and or even unrelated may determine the amount of orientation and training [21]. Further, the amount of the simultaneous new hires plays an important role in determining how onboarding is organized. Large recruitment companies in agile companies tend to have more formalized onboarding functions, ensure extensive training and dedicated mentoring and support [5]. However, little is known about the practical aspects of large onboarding campaigns, especially in agile and global software engineering contexts.

3 Methodology

In this paper, we report our findings from an exploratory multi-case study of GlobCo and its collaboration partners. GlobCo is a large international company headquartered in Northern Europe that develops a wide range of software-intensive products and solutions, including generic software products offered to an open market and complex compound systems with customized versions. Agile ways of working including Scrum have been practiced since 2008, and as part of adopting agile methods, communities of practice (CoPs) were introduced early. The company has been globally distributed for a long time with many own sites across the world, and a network of external vendors. Here, we study two cases of large onboarding campaigns in outsourcing assignments for their two different vendors, namely Case A with ConsultCo from Poland, and Case B with SupplyCo from Croatia. All company names are pseudonyms to preserve confidentiality.

Our case study is a part of a larger study, the main focus of which is on scaling organizational efforts in agile companies. In the large study, we have researched individual onboarding cases of different size and in different contexts, including vendor switching [22] and knowledge transfer. The cases selected for the present study are all cases of onboarding hundreds of software engineers.

Our study follows the guidelines of the case study design proposed by Yin [23]. Our unit of analysis, the "case", is the onboarding of new software engineers at the outsourcing supplier side in the context of a client-supplier assignment. The goal of our exploratory study is to find out what is happening, seeking new insights, and generating recommendations based on the state-of-the-practice [24].

Table 1. Overview of the cases and data collection activities.

Case	Company	Interviews	Onsite visits	Documentation
Case A Onboarding of 150 engineers in 3 batches	GlobCo, Sweden and Canada	5 individual interviews	No	Assignment specification, follow up assessments and progress reports
	ConsultCo, Poland 1	2 individual interviews, retrospective (9 engineers and managers; 1,5h), group interview (7 engineers; 1h)	Yes	
	ConsultCo, Poland 2	2 individual interviews, retrospective (6 engineers; 1,5h), group interview (4 engineers; 1,5h)	Yes	
Case B Onboarding of 200 engineers in 3 batches	GlobCo, Poland	6 individual interviews	No	Assignment specification
	SupplyCo, Croatia	3 individual interviews, retrospective (9 engineers; 1,5h), group interview (7 engineers and managers; 3h)	Yes	

3.1 Data Collection

Our exploratory research is qualitative in nature. The data collected to answer our research question was collected through qualitative interviews involving individuals and groups and onsite observations (see Table 1 for details). We performed twenty-two semi-structured interviews with twenty-seven interviewees in total across all cases. The interviewees were selected by the companies and were set to represent the key roles in the onboarding campaigns. In both cases, we have interviewed the assignment owner from GlobCo, and two to seven representatives from the vendor companies. During the interviews, we asked the interviewees to recall the events of the transfer, and reflect on what went well, what did not go that well, and what recommendations could they give for future transition projects. All individual interviews were 45–60 min long, conducted via Skype for Business teleconferencing tool, while the group interviews and retrospectives varied in duration (see Table 1), organized at the outsourcing suppliers onsite in Poland and Croatia. Individual interviews were recorded with the consent of the interviewees using AudioNote (special software tool for recording and note-taking) while group sessions were led by one researcher and closed to live-transcribed by the second researcher. Some group sessions were organized as a retrospective session (see Table 1). To elicit the relevant problems, we chose to use an exercise where participants discuss issues that need to be dropped, added, kept or improved (DAKI). This exercise is suitable when there are many participants and issues. We then facilitated a discussion of the most urgent items by using the Lean Coffee technique.

In addition to the qualitative data, we received transition project documentation, containing detailed assignment specifications and knowledge transfer documentation. Assignment specifications were used to understand the assignment details, parties involved, and the amount of the work being transferred. Detailed transfer plans and personnel profiles were used to assess the scope of each onboarding project and associated activities. Travel plans were used to check which onboarding activities were carried out in co-location and how many people were given this opportunity. Progress reports were used to elicit the challenges and obstacles, as well as the carried mitigation actions reported during the onboarding campaigns.

3.2 Data Analysis

The data analysis was organized in iterations. The preliminary data analysis was performed upon the data collection. The interview protocol was used to capture the responses through simultaneous notetaking [25]. These initial "jottings" and summaries were later expanded based on the audio recordings whenever necessary. During the next stage, we used the preliminary coding to draft the case histories, which focused on the factors that hindered and factors that facilitated onboarding. We used an initial set of categories based on the risk factors identified in earlier research on knowledge transfer [26], adding new factor that emerged in the analysis. The case histories were then sent for validation to GlobCo's outsourcing partners, ConsultCo and SupplyCo, and later presented and discussed with the representatives from the client organization, GlobCo. Finally, we created the narratives (presented in Sects. 4.1 and 4.2) by revisiting the interview transcripts

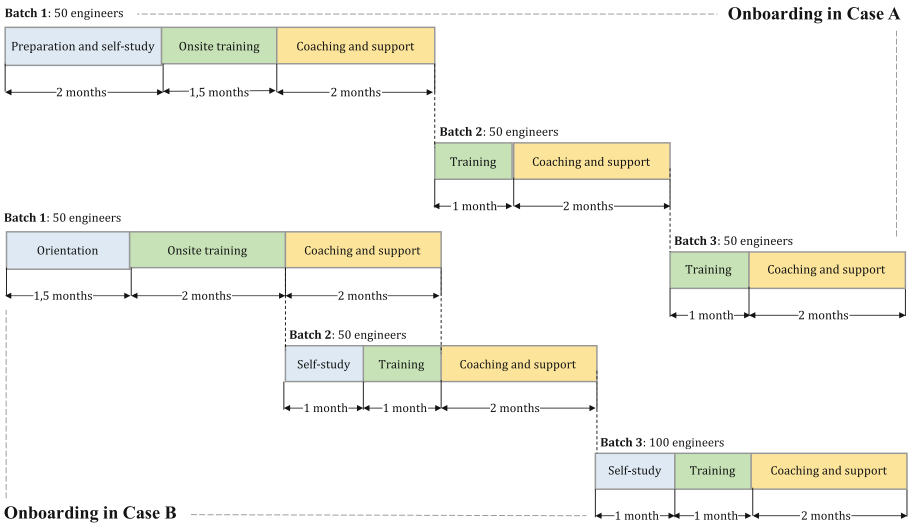


Fig. 1. Onboarding activities organized in three batches in Case A and Case B.

and notes from the group interviews and retrospectives, as well as the onboarding documentation. To explain what was going on during the onboarding, we also reconstructed the onboarding activities (Fig. 1). To provide the traces between our interpretation of the events and the data, we illustrate the core findings with transcribed quotations from the interviewees or the in-vivo notes taken during the group interviews and retrospectives.

3.3 Limitations and Threats to Validity

As any case study research, our study has several limitations [23]. Our findings are subject to interviewee bias, which we addressed by involving interviewees from all participating companies (data triangulation) and using progress reports (method triangulation). To increase the reliability and avoid false interpretations, we sent the draft of our report for review by key informants. Our study is also strongly bounded by the context of the studied companies (company sizes, size of the onboarding campaigns, outsourcing relationships involved) and the employed onboarding approach (three batches with relatively aggressive schedules), limiting the generalizability of the findings [23]. To support the readers in judging the applicability of our findings in their contexts, we detailed the case descriptions as much as possible. We believe that the findings can be of particular interest for managers involved in large-scale software development of complex legacy products facing the need for onboarding large numbers of engineers.

4 Insights into Large Onboarding Campaigns

4.1 Case A: Onboarding 150 Engineers

In ConsultCo, we studied a large recruitment and onboarding campaign for the new assignment by GlobCo that required 300 engineers to be onboarded as a replacement for the GlobCo internal engineers in Sweden, Canada and China. ConsultCo bid contained an offer of 170 engineers in China and 150 engineers in Sweden (the latter is the focus of our study). However, due to recruitment difficulties, ConsultCo could only hire 20 engineers in three different sites in Sweden while the vast majority (130 engineers) were hired in two subsidiaries in Poland.

Due to the scale of the assignment (150 engineers), the onboarding was organized in batches (see Fig. 1 – Case A), because organizing onsite on-the-job training for so many engineers simultaneously was regarded as practically impossible. The first onboarding cycle started with a 2-months long preparation and a period of self-study, followed by the onsite training phase, and coaching and support. The supplier company tried recruit experienced engineers from a GlobCo's competitor, who were later said to have much faster onboarding. Unfortunately, only half of the people could travel for the onsite training, so there were groups of engineers learning from the video recordings.

The training and knowledge transfer was kicked-off with a four-day long program in Sweden, and further activities were organized on-site in either Sweden or Canada. Admittedly, the GlobCo engineers were working under much stress and market pressure and could not dedicate due attention to the trainees. In retrospect, the time dedicated for onsite training was also regarded as insufficient. The reasons were threefold. First, the onboarding schedule generally was very aggressive (5,5 months for the first batch and 3 months for each of the following batches) due to large market pressure. Second, the product concerned was recognized as "extremely complex" with very large product areas and hardware and software architecture differing from competing products in the same area. Therefore, the threshold for learning the product was very high and the learning curve for understanding the system end-to-end was estimated to take 1–2 years. Even becoming at least somewhat productive according to the newly onboarded ConsultCo engineers was said to take 6–12 months. Finally, the product was insufficiently documented, especially concerning the legacy parts and new functionality. Besides, some material was outdated, resulting in the false feeling of obtaining knowledge. As described by an engineer:

> "[A colleague], who newly graduated from the university, was talking about this shock. He was reading the material and looking through presentations in the beginning, and he said [the company] has a lot of material and documentation, but there were two main challenges. First, the language [the company] has [includes] a lot of abbreviations. [...] And he had a hard time understanding the material. And then the second challenge was the outdated material. [...] He read it and he thought he has gained a lot of information. And then later when he saw the code he saw that the code differed."

Another engineer explained the challenge of understanding the product:

> "My feeling is that still the best source is the code itself. […] But there are still millions of lines of code. So it's not possible to read the whole code".

And another engineer estimates the time it takes for individual exploration:

> "It takes 1 hour for an experienced guy, but you need 1-2 weeks to figure it out on your own".

Nonetheless, GlobCo expected that by the end of the first onboarding cycle, engineers from the first batch will be able to further introduce and train the new teams locally with a minimum support from the original engineers in GlobCo.

In the following batches, several changes were made in the onboarding program. First, the self-study phase was skipped. Although the live trainings have been recorded, watching the trainings was not a popular learning activity. As an engineer explained: "People did not like reading material and watching videos for 2 months". Instead, the self-study parts of the training occurred in parallel with the integration in the teams and were based on the customized material adjusted to the local needs created by the engineers from the first batch. As an engineer explained:

> "In the next few months we prepared a checklist including mandatory training steps every new project member has to go through in order to get the knowledge because a lot of people that are joining our project don't have any [domain] background… There were some materials created during our internal trainings especially recorded trainings in Polish. It is always a little bit easier to work in your native language when talking about complex things".

Integration in the teams was organized by splitting the teams onboarded in the first batch (e.g. 10 teams of 5) and creating twice as many teams in the second batch (e.g., 20 teams of 5), and further increasing the number of teams in the third batch (e.g., 30 teams of 5). However, this meant that half of the engineers from the first batch had to change roles or product areas they worked with, resulting in the training efforts partially wasted. As an engineer explains:

> "In fact, some of the people who went to Canada or Sweden spent time on learning one area and then they were moved to another area. The area they received the knowledge about [during onsite training] didn't really match the features that they have received. […] People had to be spread somehow. […] I don't know if we really could have avoided that".

This was one of the reasons why competence build-up was slower than expected and the level of experience in some areas was notably low, as noted in the progress reports. Besides, the very lack of direct knowledge transfer from the original developers in this case had a prominent impact, as an engineer describes:

> "We can compare what type of onboarding [a colleague's name who was trained onsite] received and others who have watched the presentations. More should be sent for onsite training, but not necessarily all."

Onboarding in Case A had a mixed success. Despite good cooperation, initial enthusiasm and motivation to engage from the SupplyCo side, GlobCo did not have the time to organize efficient onsite training and upon the completion of the onboarding the speed of execution, delivery quality and ability to take full responsibility for assigned tasks was below expectations. There were other factors that contributed to this too. For example, ConsultCo organization in Poland was relatively new and thus did not have a safety net of colleagues that could support the newly recruited batches. Besides, the management acknowledged that the very complexity of the assignment from the start was underestimated and employed too many inexperienced people without the domain and process-related knowledge, or prior experience with working for GlobCo. Due to the large scale of the recruitment campaign, ConsultCo focused primarily on ramping up the number of engineers compromising on the competence front. Finally, the long periods of low productivity and other typical career changes led to a substantial amount of resignations causing significant damage on the capability front.

4.2 Case B: Onboarding 200 Engineers

In SupplyCo, we studied a large recruitment and onboarding campaign for the new assignment by GlobCo. Prior to the studied assignment, SupplyCo had extensive experiences in collaboration with GlobCo and thus had existing experience with their ways of working. The assignment overall was related to continuous development and maintenance of a very large and complex system, which had already involved over 1000 engineers from six different organizations and was said to take time to learn. The new assignment required onboarding of 200 engineers, which was similarly to Case A organized in three batches (see Fig. 1 - Case B).

The schedule chosen for the onboarding was rather aggressive due to market pressure (4–5 months for each batch and one year for the entire scaling). To ensure successful onboarding, SupplyCo's decided to organize onsite training at GlobCo in Poland for the first batch and put forward a group of highly-experienced inhouse engineers with related expertise and the right attitude – "not shy to push and motivated to get into work", who can put 110% effort to secure performance. Knowledge transfer started with the general presentations prepared by the client organization, but quickly shifted to real tasks and close integration into the teams of original engineers at the client site, whenever possible. A product owner who did not join a Polish team compares experiences with his colleagues:

> "A decision was made that the second team [of trainees] will split and join the existing Polish teams and they will work together with the Polish guys and try to learn as much as possible [...] And at the end, it turned out, I think, that those guys learned much more than the guys in my team that consisted of only the Croatian guys. That was the learning that guys that were part of the mixed teams learned much more than the guys that were left to learn on their own through support".

This is echoed by an experienced engineer:

> "Some developers were integrated in different Polish teams, but my team was assigned a new feature. We could ask for help, but we did not have a person

> in Poland and zero experience. We had someone to talk to, but that person was not working on the same feature. There is a steeper learning curve when you are integrated in teams".

To maximize the learning, some engineers rotated from team to team to be exposed to different parts of the product and different points in the lifecycle of a task. At the end of each day, the SupplyCo engineers organized daily stand up meetings to share their experiences (what have one learned) and follow up on any emerging issues (what challenges one faced). The first batch was deliberately assembled by fast learners who were expected to pass their knowledge to further batches. Besides, their competence has helped to overcome the challenges faced during the knowledge transfer, as engineers at the client organization had themselves limited experiences, as they have been onboarded into the assignment 1–2 years prior to the scaling endeavor. The challenge in Case B was similarly to Case A the product itself. As an engineer describes:

> "There's quite a lot to learn. That's the biggest challenge, to learn the product. And it is still a big problem. After 7-8 months, we are still learning the product".

Due to onsite training being "a big investment", as recognized by the GlobCo stakeholders, the second and third batches have been onboarded by the representatives from the first batch at the supplier site. However, onsite training was also available for engineers in the other two batches, on request. At the end of the first onboarding cycle, the management ran a retrospective to improve the onboarding for the following two batches. For example, there was a need to have a more personal onboarding, and so the joint orientation phase was replaced with the tailored self-study period adjusted for different roles, product areas and personal experiences. As a manager said:

> "Not all are going through the same path and pace".

Further, engineers from the first batch created learning guides for different learning groups (with a detailed agenda), video-recordings and detailed mind maps with the necessary competences and skills reused for the following batches. An engineer explains:

> "Later we actually created some presentations and made maps and also created learning tasks so that you can show people how to go through a task and then discuss after they tried to learn all what is needed in order to work. And one month later [the following batch] actually start to work with the team".

SupplyCo's training philosophy is described by a manager as follows:

> "With documentation it's not possible to address everything, so you need to figure out what are the crucial things. Then learning methods are also very important. Learning methods [...] must build confidence. [...] So, depending on the setup [...] it's going to be on-the-job training, different tasks within the team or individually".

Onboarding in Case B has been regarded as successful. All interviewed GlobCo representatives mentioned good preparation, very good communication between the parties, and experienced people nominated by the supplier as the key success factors. SupplyCo further describes that they were able to find employees with prior experience of working on GlobCo assignments, and thus ensure process and domain knowledge. The company acted as a safety net, which was not the case for ConsultCo, a relatively new company with many inexperienced new hires. Besides, SupplyCo have shown exceptional motivation to receive the assignment and demonstrate their capabilities.

4.3 Cross-Case Analysis: Challenges of Onboarding Hundreds of Engineers

In this sub-section, we report the top five challenges particular for the large onboarding campaigns and the helpful strategies emerging from the two cases.

Challenge 1: A Large Onboarding Campaign is a Large Investment. Recruitment and onboarding of hundreds of software engineers in both cases required enormous practical and financial support. Despite the awareness that the best knowledge transfer is organized through on-the-job onsite training, GlobCo could not arrange hundreds of desks and workstations to accommodate all engineers in their premises. Even recruitment of so many engineers can be challenging, as experienced by ConsultCo in Case A. Besides, since the suppliers in our cases were situated on a distance, flying everybody for onsite training required substantial budget resources.

Helpful Strategies: To avoid high costs and impracticalities of onsite training for hundreds of developers, companies can organize onboarding in batches. In this approach, the first batch is formed by highly competent and fast learning engineers and sent for onsite knowledge transfer. Upon completion of their onboarding, the first batch shall be able to support onboarding of the following batches.

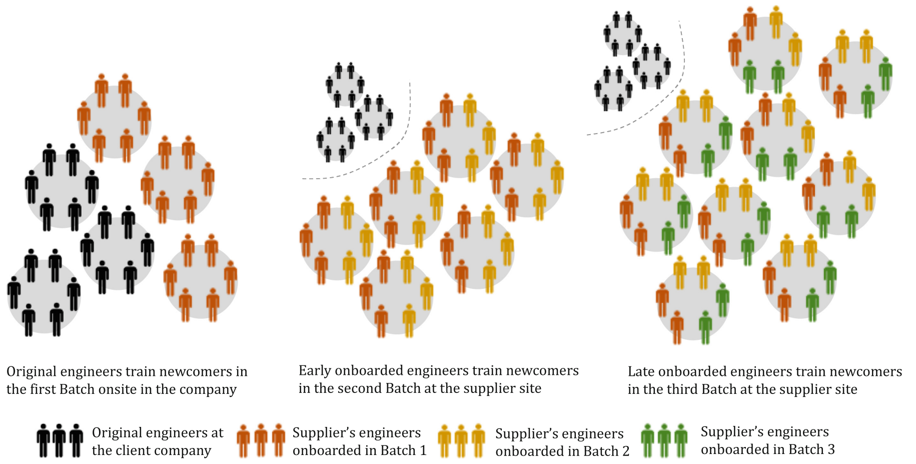
Challenge 2: Multiple-Batch Onboarding by Splitting Dilutes the Accumulated Competence. The batch-based scaling in our cases was organized by splitting. In Fig. 2, we illustrate the onboarding of 54 engineers in three batches, where the first batch comprises three teams trained by the original engineers, which are then split into halves and then scaled with the second batch engineers. The splitting process repeats for the third batch. Every new batch evidently lowers the overall level of experience in the team. If repeating the process each team will have just one engineer trained onsite by the original engineers in the fourth cycle, and after the fifth cycle some teams will not have any engineers trained onsite. In Case B, the third batch consisted of as many developers as in the previous two batches, which resulted in having teams without any first batch engineers. A young engineer from the third batch in Case B explained:

> “None of our team members had [domain] experience, none of the team members worked here. We were a team with only new persons. It is not a good practice, although we had a good solution to this – everybody had contacts [with experts] and we had a buddy team somehow. But better to have a mix [of competent and new people].

Helpful Strategies: Splitting teams to mix the competent and new people is a useful practice, but with some limitations. When splitting results in not having enough competent people on the team, companies may try pairing the new teams with a “Buddy team” or a “Sister team”, who then provide the necessary mentoring and support. Buddy teams were practiced in Case A during the onsite training, as described by a manager:

> “It’s when [the client company] teams work very close with new teams in [SupplyCo], instead of having 1:1 developer to developer relationship.”

Our results also clearly show that there is no one recipe for team formation and that the decisions shall consider 1) the new hires’ competence levels, 2) the product area selected for onboarding, and 3) the competence levels of the existing teams. Case B representatives mentioned that their strategies varied from extending some teams (which required having smaller teams in the beginning), splitting some teams (especially in the areas that require many engineers), and forming some new teams (especially in the areas that were not covered by the training for the first batch).



Original engineers train newcomers in the first Batch onsite in the company

Early onboarded engineers train newcomers in the second Batch at the supplier site

Late onboarded engineers train newcomers in the third Batch at the supplier site

Original engineers at the client company
Supplier’s engineers onboarded in Batch 1
Supplier’s engineers onboarded in Batch 2
Supplier’s engineers onboarded in Batch 3

Fig. 2. An example of the onboarding of ~ 50 engineers organized in three batches by splitting.

Challenge 3: Multiple-Batch Onboarding by Splitting can Result in Ill-suited Onboarding Efforts. The success of scaling by splitting depends on the tasks that the onboarded teams receive. The onboarding program of the first batch could not possibly cover the whole system, and thus onboarding was done on selected areas. Naturally, when splitting the initially formed teams, some engineers had to work on something they have not been trained for. As a result, these engineers perceived their training at least partially as a waste. However, as a manager from Case B explained, even when having a good plan, the problem is not always possible to avoid due to changes:

> “We had to onboard 200+ people. Can we take half of the product first? That was our plan. But reality was completely different. Once you start [the training], [the scope] starts changing, because the [operational] need is there (points in one

> direction), then the need is there (points in another direction), then the need is there (points in the third direction). So, you need to adjust."

Helpful Strategies: One way to address this challenge is to follow Case B example and offer on demand onsite training, for all in the first batch and some in the subsequent batches. Another strategy was to ensure that the self-study parts of the training occurred in parallel with the integration in the teams to make the training relevant (Case A). Alternatively, companies may also consider not forming the teams too early and instead assigning engineers from the first batch to cover all product areas, taking into consideration the additional resources planned in the subsequent batches (in the example on Fig. 2, this would require forming nine pairs of two in the first batch or having more teams in the first batch and fewer later).

Challenge 4: Poor Onboarding Experiences may Result in New Hires Leaving. When onboarding employees into a complex software product with poor or no documentation, and especially when employing inexperienced engineers, it may result in long periods of low productivity and confusion, in some cases leading to resignations. In Case A, the resignations among already onboarded employees caused a loss of accumulated experience and expertise. Attrition especially among those trained onsite has devastating impacts since every individual plays an important role in their team, when scaling by splitting the teams. As an engineer from Case A describes his experience:

> "A funny situation: someone has made the configuration [of the testing environment]. But he has left, and now nobody knows how to configure it."

Helpful Strategies: Countermeasures for the high levels of complexity and gaps in documentation include recruiting competent engineers with domain and potentially even process experience. This allows to form teams, in which inexperienced engineers are supported by more experienced engineers, ideally with prior engagement with the company. As an engineer from Case B explains:

> "We divide the team mixing more experienced and less experienced people. That's how a lot of companies do this onboarding."

Other helpful strategies identified in our two cases include on-the-job training, integration in or pairing with experienced teams for mentoring and support, on-demand onsite training, and fostering shared learning within a team being onboarded through regular daily synch sessions and retrospectives to improve onboarding experiences.

Challenge 5: Disproportional Organizational Growth During Large Onboarding Campaigns can be Devastating. Turnover in Case A was caused not only by the poor onboarding experience, but also for more typical reasons, including promotions, relocation to other projects, personal reasons. Onboarding in Case B, in its turn, was threatened by the turnover among the trainers on the client side, as a manager describes:

> "Attrition can be on both sides. If on the client's side, you are left without the trainers. In [Case B] we had over 20% attrition. That was a problem."

In both cases, the success of the onboarding depended on the ability of the companies to respond to changes and compensate the loss or lack of competences primarily through relying on internal resources. This was possible in Case B, because SupplyCo was a rather mature company, while this was a challenge in Case A, since ConsultCo was a relatively new company and had few resources to help in this situation.

Helpful Strategies: The problem of rapid organizational growth was discussed among the top managers in the group session at SupplyCo at length. The managers agreed that employing *too many* new engineers is not a good idea. According to a manager's opinion, recruitment efforts shall be proportional to the existing organization:

> "You should aim for 33% impact [on the existing base]. For example, when 6 teams are trained, you can then add 2 teams every 2 months. When teams [complete the onboarding] then the new teams enter the preparation phase."

And another manager explained:

> "If we do not have a good competence breakdown figure and a large enough base, we cannot speed up the scaling. Newcomers dilute the existing teams, but the rest of the organization cannot help us a lot. We need more time to have the ability to integrate the people in the organization".

There seem to be a threshold for reasonable organizational growth, unique for each organization and based on existing competence, resource availability and company size.

5 Concluding Discussion

In the previous sub-section, we described two stories of large onboarding campaigns. With respect to our research question: *What are the strategies for large onboarding campaigns?* We found that onboarding everyone at the same time is impractical and instead agile companies should organize large onboarding campaigns in batches (see Fig. 1). Further we know that effective onboarding of developers, especially in agile environment where detailed documentation is not emphasized, is organized through on-the-job onsite training [5, 14], however, in large-scale campaigns such a strategy is not possible. The often-used strategy further is to scale by splitting the recently onboarded teams (as shown in Fig. 2). However, the success of onboarding that puts engineers through these onboarding cycles is mixed. A deeper cross-case analysis suggests that the keys for successful large-scale onboarding are threefold.

1) **A good initial plan.** Companies shall have a good understanding of the complexity of the assignment and critically assess their own capabilities in terms of organizational maturity and size as well as existing in-house competences. A good onboarding plan for large campaigns includes a phase-based approach as suggested by related research [4, 5, 12] and a multiple-batch breakdown, a novel contribution of our study (see Fig. 2). Our findings shed light on the challenges and potential strategies for organizing multiple-batch onboarding, which resonate with the findings from an

unprecedent, transformational growth (although not as rapid as in our cases) reported by Kumar and Wallace [27]. Our results also suggest that the onboarding plan shall contain context-appropriate programs to avoid wasted training efforts. The helpful strategies included running self-study parts in parallel with on-the-job training (e.g., integration in the teams), which is consonant with existing research that emphasizes the importance of early productive work and exposure to work-related practices and ceremonies [5, 14]. Our findings contribute to existing research suggesting that mentoring and support can be organized through a "buddy team" or a "sister team" [6, 12, 14]. Prior research shows that such approach reduces the problems of mentors being unavailable for newcomers [14], and helps new teams to grow their social network, which is essential in large-scale agile [31]. And yet, our study clearly shows that following a plan will not take you far when changes arrive.

2) **Responsiveness to change.** Organizational agility, often seen as a key for success, is the ability of firms to sense environmental change and respond appropriately [30]. In our study, training targets shifted, recruitment got delayed, priorities of the ongoing operations changed, unforeseen gaps in documentation were uncovered and people started leaving. In both cases studied, supplier companies strived to learn from their experiences and respond to the emerging changes. Among the helpful strategies identified, many highlight the importance of constantly reevaluating the course of the onboarding through arranging daily synch meetings and retrospectives, as well as continuously improving the onboarding and training material. The importance of integrating the agile learning practices into onboarding are also presented in the model proposed by Gregory et al. [14]. Yet, our findings suggest that the ability to respond to changes requires certain organizational maturity.
3) **Organizational maturity**. Our study shows that large onboarding campaigns required certain organizational maturity and size to provide a safety net when the changes emerged. This is also highlighted by related studies from large and mature outsourcing suppliers who are able to address the challenges by relocating their key engineers to prioritized projects and customers [28, 29]. Our study thus suggests that large onboarding campaigns might not be for any company as the associated growth shall be proportional to the company size.

Evidently, the three keys to success generate further questions: How much planning should be built into the onboarding and scaling process? How much shall an organization invest into onsite training? How much scaling is appropriate for an organization of a certain size? Like the tale of the guitar player asking Buddha how best to tune his instrument, we find the famous answer, "Not too tight, and not too loose". A better understanding of the thresholds for planning, investments into onsite training and organizational growth is a valid direction for future research in this area.

Acknowledgements. We would like to thank the studied client company for their engagement in our research, and especially for opening the doors to their outsourcing partners, which is uncommon in outsourcing research studies. This research is partially funded by the ScaleWise project and the Swedish Knowledge Foundation under the KK-Hög grant 2019/0087, and the Research Council of Norway through the 10xTeams project (grant 309344).

References

1. Carmel, E., Tjia, P.: Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce. Cambridge University Press, Cambridge (2005)
2. Stray, V., Moe, N.B., Mikalsen, M., Hagen, E.: An empirical investigation of pull requests in partially distributed BizDevOps teams. In: IEEE/ACM Joint 15th International Conference on Software and System Processes (ICSSP) and 16th ACM/IEEE International Conference on Global Software Engineering (ICGSE), pp. 110–119 (2021)
3. Šmite, D., Moe, N.B., Gonzalez-Huerta, J.: Overcoming cultural barriers to being agile in distributed teams. Inf. Softw. Technol. **138**, 106612 (2021)
4. Bauer, T.N.: Onboarding new employees: maximizing success. SHRM Found. (2011)
5. Britto, R., Cruzes, D., Šmite, D., Šāblis, A.: Onboarding software developers and teams in globally distributed projects: a multi-case study. J. Softw. Evol. Process **30**(4), e1921 (2018)
6. Buchanan, B.: Building organizational commitment: the socialization of managers in work organizations. Adm. Sci. Q. **19**(4), 533–546 (1974)
7. Klein, H.J., Polin, B., Leigh Sutton, K.: Specific onboarding practices for the socialization of new employees. Int. J. Sel. Assess. **23**(3), 263–283 (2015)
8. Lynch, K., Buckner-Hayden, G.: Reducing the new employee learning curve to improve productivity. J. Healthc. Risk Manag. **29**(3), 22–28 (2010)
9. Derven, M.: Onboarding, Association for talent and development (2008)
10. Gruman, J.A., Saks, A.M., Zweig, D.I.: Organizational socialization tactics and newcomer proactive behaviors: an integrative study. J. Vocat. Behav. **69**(1), 90–104 (2006)
11. Britto, R., Smite, D., Damm, L.-O.: Software architects in large-scale distributed projects: an Ericsson case. IEEE Softw. **33**(6), 48–55 (2016)
12. Moe, N.B., Stray, V., Goplen, M.R.: Studying onboarding in distributed software teams: a case study and guidelines. In: Proceedings of the Evaluation and Assessment in Software Engineering, pp. 150–159 (2020)
13. Sharma, G.G., Stol, K.J.: Exploring onboarding success, organizational fit, and turnover intention of software professionals. J. Syst. Softw. **1**(159), 110442 (2020)
14. Gregory, P., Strode, D.E., Sharp, H., Barroca, L.: An onboarding model for integrating newcomers into agile project teams. Inf. Softw. Technol. **143**, 106792 (2022)
15. Stray, V., Tkalich, A., Moe, N.B.: The Agile Coach Role: Coaching for Agile Performance Impact (2021)
16. Fagerholm, F., Sanchez Guinea, A., Borenstein, J., Munch, J.: Onboarding in open source projects. IEEE Softw. **31**(6), 54–61 (2014)
17. Matturro, G., Fontán, C., Raschetti, F.: Soft skills in scrum teams a survey of the most valued to have by product owners and scrum masters. In: International Conference on Software Engineering and Knowledge Engineering, pp. 42–45 (2015)
18. Steinmacher, I., Conte, T.U., Treude, C., Gerosa, M.A.: Overcoming open source project entry barriers with a portal for newcomers. In: Proceedings of the 38th International Conference on Software Engineering, pp. 273–284 (2016)
19. Morrison, E.W.: Newcomers' relationships: the role of social network ties during socialization. Acad. Manag. J. **45**(6), 1149–1160 (2002)
20. Casalnuovo, C., Vasilescu, B., Devanbu, P., Filkov, V.: Developer onboarding in GitHub: the role of prior social links and language experience. In: ESEC/FSE, pp. 817–828 (2015)
21. Boh, W., Slaughter, S.A., Espinosa, J.A.: Learning from experience in software development: a multilevel analysis. Manage. Sci. **53**(8), 1315–1331 (2007)
22. Šmite, D., Moe, N.B.: Switching vendors: factors that matter when engineers onboard own replacement. J. Softw. Syst. **169** (2020)

23. Yin, R.K.: Case Study Research and Applications: Design and Methods. SAGE Publications Inc. (2017)
24. Robson, C., McCartan, K.: Real World Research, 4th edn. Wiley (2016)
25. Creswell, J.W.: Qualitative Inquiry and Research Design, 3rd edn. SAGE Publications Inc. (2014)
26. Smite, D., Wohlin, C.: Strategies facilitating software product transfers. IEEE Softw. **28**(5), 60–66 (2010)
27. Kumar, S., Wallace, C.: Patterns of identity and interaction in an agile community of practice. In: Proceedings of the International Workshop on Cooperative and Human Aspects of Software Engineering, pp. 71–78 (2019)
28. Moe, N.B., Šmite, D., Hanssen, G.K., Barney, H.: From offshore outsourcing to insourcing and partnerships: four failed outsourcing attempts. Empir. Softw. Eng. **19**(5), 1225–1258 (2013). https://doi.org/10.1007/s10664-013-9272-x
29. Sabherwal, R.: The evolution of coordination in outsourced software development projects: a comparison of client and vendor perspectives. Inf. Organ. **13**(3), 153–202 (2003)
30. Gren, L., Lenberg, P.: Agility is responsiveness to change: an essential definition. In: Proceedings of the Evaluation and Assessment in Software Engineering, pp. 348–353 (2020)
31. Moe, N.B., Olsson, H.H., Dingsøyr, T.: Trends in large-scale agile development: a summary of the 4th workshop at XP2016. In: Proceedings of the Scientific Workshop Proceedings of XP conference, pp. 1–4 (2016)