Aalto University

OPEN ACCESS

This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Paasivaara, Maria; Lassenius, Casper

Communities of practice in a large distributed agile software development organization - Case Ericsson

Published in:
Information and Software Technology

DOI:
10.1016/j.infsof.2014.06.008

Published: 01/01/2014

Document Version
Publisher's PDF, also known as Version of record

Published under the following license:
CC BY-NC-ND

Please cite the original version:
Paasivaara, M., & Lassenius, C. (2014). Communities of practice in a large distributed agile software development organization - Case Ericsson. *Information and Software Technology*, *56*(12), 1556-1577. https://doi.org/10.1016/j.infsof.2014.06.008

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Information and Software Technology 56 (2014) 1556–1577

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Communities of practice in a large distributed agile software development organization – Case Ericsson

Maria Paasivaara *, Casper Lassenius

Aalto University, Department of Computer Science and Engineering, POB 15400, 00076 Aalto, Finland

ARTICLE INFO

Article history:
Received 22 April 2013
Received in revised form 28 May 2014
Accepted 9 June 2014
Available online 26 June 2014

Keywords:
Communities of practice
Large-scale agile software development
Scaling agile

ABSTRACT

Context: Communities of practice—groups of experts who share a common interest or topic and collectively want to deepen their knowledge—can be an important part of a successful lean and agile adoption in particular in large organizations.
Objective: In this paper, we present a study on how a large organization within Ericsson with 400 persons in 40 Scrum teams at three sites adopted the use of Communities of Practice (CoP) as part of their transformation from a traditional plan-driven organization to lean and agile.
Methods: We collected data by 52 semi-structured interviews on two sites, and longitudinal non-participant observation of the transformation during over 20 site visits over a period of two years.
Results: The organization had over 20 CoPs, gathering weekly, bi-weekly or on a need basis. CoPs had several purposes including knowledge sharing and learning, coordination, technical work, and organizational development. Examples of CoPs include Feature Coordination CoPs to coordinate between teams working on the same feature, a Coaching CoP to discuss agile implementation challenges and successes and to help lead the organizational continuous improvement, an end-to-end CoP to remove bottlenecks from the flow, and Developers CoPs to share good development practices. Success factors of well-functioning CoPs include having a good topic, passionate leader, proper agenda, decision making authority, open community, supporting tools, suitable rhythm, and cross-site participation when needed. Organizational support include creating a supportive atmosphere and providing a suitable infrastructure for CoPs.
Conclusions: In the case organization, CoPs were initially used to support the agile transformation, and as part of the distributed Scrum implementation. As the transformation progressed, the CoPs also took on the role of supporting continuous organizational improvements. CoPs became a central mechanism behind the success of the large-scale agile implementation in the case organization that helped mitigate some of the most pressing problems of the agile transformation.
© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).

1. Introduction

Agile methods, and in particular Scrum [1], have become mainstream in contemporary software development in both small and large organizations [2]. Based upon iterative and incremental software development [3], agile methods were originally created to support small teams of highly experienced developers working in a single team room [4]. Consequently, they rely heavily on face-to-face communication, limiting the maximum practical size of the development team [4].

The "home ground" for agile software development practices is small teams of highly experienced people building small, non critical systems with highly volatile requirements in an organization with a high degree of acceptance of uncertainty [5].

However, agile development is increasingly adopted in large organizations running big software development projects employing multiple teams distributed to several geographical locations [2,6]. Adopting agile methods in this context introduces new challenges related to scaling, such as inter-team coordination, effective knowledge sharing between the teams, design without a defined architecture or properly defined requirements, as well as all the challenges of distributed projects [7].

Despite these additional hurdles, companies report having successfully applied agile practices in large projects [8,9]. While a few case studies and experience reports on adopting agile methods in projects involving several teams and several geographical locations exist, most reports are from small projects involving only a few teams, often involving altogether less than thirty developers.

* Corresponding author.
E-mail addresses: maria.paasivaara@aalto.fi (M. Paasivaara), casper.lassenius@aalto.fi (C. Lassenius).

http://dx.doi.org/10.1016/j.infsof.2014.06.008
0950-5849/© 2014 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).

While practitioner literature, mostly authored by consultants, contains advice on scaling agile development to larger contexts, see, e.g. [10,7], academic studies providing evidence on scaling agile is still scarce [11]. In particular, there is little evidence on which scaling practices actually provide value, under what circumstances, and how to successfully introduce them.

One of the practices recommended by the consultants, is introducing Communities of Practice (CoP) to help with knowledge sharing, organizational and process development, and coordination [10]. While Communities of Practice have been described and used widely in other contexts, see e.g. [12,13], their use in the context of professional software development, and in particular in scaling agile development, has received little attention in the research literature.

Another significant issue that large software development organizations have to deal with when adopting agile methods is how to handle the organizational transformation to agile [14,15]. Large organizations often have institutionalized processes and organizational structures that provide a poor fit with agile development. Thus, in addition to understanding what a good end state should look like, managing the transformation from the starting state to a successfully working agile implementation can provide significant challenges.

In this paper we help mitigate the research gap on the usage of Communities of Practice in scaling agile development by presenting the motivation, development, and use of Communities of Practice (CoPs) in a large-scale agile software development organization at the telecommunications infrastructure provider Ericsson.

The paper is structured as follows: Section 2 provides an overview of related work, Section 3 describes the case background, research goals and methods, Section 4 presents our results, and finally Section 5 discusses the results and concludes the paper.

2. Related work

In this section, we discuss related work on communities of practice. While the importance of communities is widely acknowledged, e.g. in open-source development, in this paper we do not discuss communities in general, but focus on *organizational communities of practice*. More specifically, we are interested in the use of communities of practice in professional software development organizations, in particular those adopting large-scale agile development. This focus is also reflected in our review of the literature.

In this literature review, we first discuss the definition, cultivation and success factors for communities of practice in general. Second, we discuss the role of communities of practice in software engineering.

2.1. Communities of practice

A community of practice, is a *group of people who share a concern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in this area by interacting on an ongoing basis* [13]. Communities of practice have been applied in a wide variety of industries and contexts [12,13].

Communities of practice have three important characteristics that sets them apart from other communities: a *domain, community*, and *practice*. The *domain* defines the area of interest in which the members collaborate to share and create knowledge. The *community* aspect means that members actively engage in joint activities, form relationships with each other, and share information. The *practice* aspect means that they develop a shared set of resources for addressing problems in their domain of interest. Table 1 compares communities of practice to other types of groups [13].

Communities of Practice can take on different roles in an organization, and the roles can evolve over time. For example, CoPs have been found to be useful when an organization changes from a functional structure to one based on product lines or projects. In such situations, CoPs can help mitigate problems, for example by providing fora for functional experts who used to work together, but in the new organization are scattered around in different product lines, to meet and continue to deepen their functional expertise.

In general, communities of practice can provide a wide range of both long-term and short term benefits to both the organization (s) in which they function, and to the members of the community [13,17]. Benefits to the organization include helping to drive strategy, starting new lines of business, providing an arena for problem solving, transferring best practices, developing professional skills, and increasing the retention of talent [16]. Benefits to the members include help with challenges, being better able to contribute to one's team, fun, enhanced professional reputation, and a strong sense of professional identity [13].

For an organization interested in gaining benefits from Communities of Practice, a central question becomes how to build and support them. Wenger suggests "shepherding" rather than creating them, and present seven principles for cultivating CoPs, listed Table 2.

The relationship between the formal organization and the communities of practice can vary. Wenger [13] lists five relationship types: the community of practice can be unrecognized, bootlegged, legitimized, supported or institutionalized. These relationship levels differ in the amount of organizational involvement in, expectations of, and support for the communities of practice.

While communities grow, evolve and die according to their individual needs, literature identified five stages of development that they go through: potential, coalescing, maturing, stewardship, and transformation.

In the potential stage there is not yet any community, rather a set of interested people that start networking around a topic of joint interest. Key issues that the network needs to deal with at this stage to evolve into a real community of practice are finding enough common ground between members to help them see the value of connecting, sharing knowledge and solving problems together. At this stage, having an active and passionate *community coordinator* is important.

When entering the coalescing stage, the community knows what exists in the organization with respect to its domain, and the community is officially launched and community events are arranged. At this stage, coordinators are still crucial to the success of the CoP. The main challenge is to incubate and deliver immediate value to the members and the organization.

At the maturing stage, the community has delivered immediate value, proving its worth, and the focus shifts to clarifying the focus, role and boundaries of the CoP.

The fourth stage, stewardship, is mainly concerned with maintaining momentum and keep the CoP going. At the final stage, transformation, the community ceases to exist, either by fading away, or by turning into some other structure, such as a social club; or become institutionalized, e.g. as a department.

2.2. Communities of practice in software engineering

In software engineering, and in particular when scaling agile software development, CoPs have been proposed as a possible solution for functional learning, and knowledge sharing between organizationally separate individuals with similar roles. Examples

Table 1
Communities of practice vs. other organizational groups [16].

	Purpose	Who belongs?	What holds it together?	How long does it last?
Community of practice	Develop member capabilities, build and exchange knowledge	Members who select themselves	Passion, commitment, identification with group expertise	As long as there is interest
Formal work group	To deliver a product or service	Everyone who reports to the group's manager	Job requirements, common goals	Until the next reorganization
Project team	To accomplish a specific task	Employees assigned by senior management	Project milestones and goals	Until project completed
Informal network	Collect and pass on business information	Friends and business acquaintances	Mutual needs	As long as people have a reason to connect

Table 2
Principles for cultivating communities of practice [13].

	CoP cultivation principle	Explanation
1.	Design for evolution	Expect the community to grow and evolve on its own
2.	Open a dialog between inside and outside perspectives	Help the community members understand what the community could achieve
3.	Invite different levels of participation	Allow for different activity levels and motivations of participant, e.g. coordinators, core, active and peripheral members, as well as external interest groups
4.	Develop both public and private community spaces	Both "formal" meetings, as well as one-on-one networking is needed
5.	Focus on value	Encourage members to be explicit about the value of the community; let value emerge, don't try to design it by force
6.	Combine familiarity and excitement	Have a "routine" program, but also include novel experiences, e.g. invited speakers
7.	Create a rhythm for the community	Find a suitable rhythm for the regular meetings

of these include a testing CoP for software testing, and a Scrum master CoP [10].

Empirical studies of CoPs in the software engineering literature are scarce. One study [18] describes how a small Norwegian software company used CoPs to facilitate learning. When adopting agile development, Nokia reportedly used a CoP-like approach to solve inter-team issues, and suggests that CoPs could be valuable to organizations when adopting agile in the large [19]. IBM Global Services report on their experiences with over 60 CoPs, and present a development model with seven stages: potential, building, engaged, active and adaptive [20].

Perhaps not surprisingly, software engineering researchers have developed tools for supporting CoPs. For example, Chau and Maurer [21] presents a set of tools for supporting inter-team coordination and knowledge sharing. However, they do not present any validation in a real organization, and wisely recognize the cultural change needed in a potential user organization as a major hurdle to be overcome for the tool to be successful.

We found it interesting that despite the fact that the practitioner literature, e.g. [10], suggests the use of CoPs, we were not able to find any in-depth study of the use of communities of practice in large-scale agile software engineering. In particular, the software engineering literature lacked insights into how to implement CoPs, how to use them, and what their practical value is. Such knowledge would help both managers and practitioners in organizations to better understand when and how to implement CoPs, and avoid making the same mistakes and run into the same challenges as other organizations already have done before them.

As this kind of basic actionable knowledge is missing in the software engineering literature, we think that even a single case study as presented in this paper can provide value both to organizations, as well as to the scientific community.

3. Research design

3.1. Case background

This paper is based on a single descriptive longitudinal case study [22] of a product development unit at Ericsson, developing a large and complex systems product—a node that handles a specific type of traffic in telecommunications networks. The development involved three global sites: Finland, Hungary and the US, and ca 400 persons in approximately 40 Scrum teams. The product had originally been developed in Finland, where also product management and product development responsibility was located. Hungary had collaborated with Finland on the product since 2006, whereas the US site was a recent addition to the program after an acquisition of a US company in 2011.

The development of the product started over ten years ago, and at the time of the study it was used by over 300 operators all over the world. The product is still in active development. The organization had used a traditional waterfall-type software process until the end of 2009, when the whole organization started what they called a "Lean and Agile transformation", during which Lean principles [23] and Scrum [24,25] were taken into use.

Ericsson Finland has been involved in R&D since it was founded in the early 1970s. During the 80s and 90s, Ericsson developed a strong plan-based project culture, and became excellent in running waterfall-type projects. The projects delivered products close to the planned schedules, and product quality was high. However, in the late 2000s, Ericsson felt that despite the success of their waterfall-type development and high product quality, they would need to be even better in the future to compete in the modern telecom world, in which competition is cut-throat. Especially, shortening the lead times and being able to respond fast to the customer needs, while maintaining the high quality, would be crucial.

According to our background interview of unit management, the product development unit experienced four problems that motivated the change from a waterfall process to agile and lean development. First, the plan-driven process required up-front scoping and investment decisions for the whole product release, leading to large scopes and very long lead times, typically up to two years from requirement specification to delivery. Second, change management was very bureaucratic and strict, further exacerbating the problem of poor responsiveness to customer and market needs. Third, the cost of quality was high and growing, as there was a big-bang integration followed by a long testing and

fixing period at the end of the project. Fourth, management considered personnel motivation to be lower than desired, as people worked mainly in small silos with very specialized software engineering and product knowledge, and very little insight into the larger whole of the product and project.

These problems created an uneasy feeling, in particular in the management team in Finland, and they started to search for solutions. There was a lot of interest in agile and lean software development worldwide in the industry, motivating the management team to study the agile practitioner literature, which seemed to provide solutions to their problems. As the main expected benefits of adopting agile software development, the organization saw breaking down functional silos and forming cross-functional teams, having closer relationship with the customers, and focusing on communication rather than documentation. However, given the strong process and project culture in the organization, and the large and complex product, they thought that basic agile practices would not be enough. Even though Agile software development, in particular the Scrum adoption, is the most visible part of the transformation, as it provides the day-to-day practices, lean thinking and principles were taken into the transformation already in the beginning. Actually, many founding ideas of this transformation came from lean, such as continuous improvement, removing waste, flow optimization in the form of end-to-end development and value stream thinking.

After careful planning, the transformation was started by forming the first pilot team at the end of 2009 and the second and the third team soon after that. The full-scale agile roll-out took place in 2010, during which the whole R&D organization—distributed between Finland and Hungary—was transformed to cross-functional Scrum teams of 7–8 persons. The Finnish teams were relocated to newly renovated team spaces in an old factory hall, while the Hungarian teams had team rooms. The first steps of this Lean and Agile transformation are described in more detail in [26], the new workspace solution in [27], and the continuous release planning process in [28].

3.2. Research goals and questions

The main research goal of the study presented in this paper was to investigate how a large globally distributed software development organization adopted the use of Communities of Practice as part of their transformation from a traditional plan-driven organization to Lean and Agile. We purposefully selected this information-rich case [29], as the case organization was a participant in a joint research program ensuring access. A longitudinal study was chosen to be able to understand how the transformation proceeded over time, and thus also study how the Communities of Practice developed and changed over time. As we were interested in studying a phenomenon in its in vivo context, without control over events or decisions, a case study approach is appropriate [22].

We approached this case in an exploratory manner aiming at understanding the basic reasoning behind and workings of the communities of practice. We stated the following basic research questions to guide our research:

- RQ1: What kinds of communities of practice were created in the case organization and how did they evolve over time?
- RQ2: What were the characteristics of successful communities of practice?
- RQ3: How did the role of communities of practice in the case organization evolve over time?
- RQ4: How did the case organization support the communities of practice?
- RQ5: How could the different purposes of communities of practice be classified?

3.3. Data collection

We collected the data mainly through semi-structured interviews, using five rounds of interviews over a two year period, complemented with observations of three CoP meetings. In total, we interviewed 52 persons. A timeline of the research is shown in Fig. 1.

3.3.1. Interview approach

We used the *general interview guide approach* [29], meaning that we had outlined a set of issues to be explored in the interviews, having the guide serve as a checklist to ensure that all relevant topics were covered. We performed the interviews in a highly conversational manner [29], to give space for deeper discussions, and to allow the interviewers to ask follow-up questions to gather more details on interesting topics that arose during interviews.

In particular during the two first interview rounds we concentrated on topics from the guide according to each interviewee's core knowledge, e.g., with the managers we had deeper discussions on the topics such as the lean and agile adoption and its reasons, whereas with the team members we could have deeper discussions on how they apply Scrum at the team level. Thus, the idea was not to discuss all the topics in the guide in detail level with all the interviewees, but to use the limited interview time so that we would get the best knowledge from each interviewee. We aimed for data saturation [30], i.e., when it seemed that an interviewee would not give additional information on a topic, we did not emphasize that topic in the interview. We did some minor modifications to the interview guides after the first few interviews, such as adding the Proxy Product Owner to the role list, when we discovered the organization had such a role. The interview guides most relevant for this paper—the one used during the first and second interview rounds, as well as the guide developed for the fifth interview round—are in Appendix A.

In most of the interviews, two researchers were present, one being the main interviewer and the other one taking detailed notes

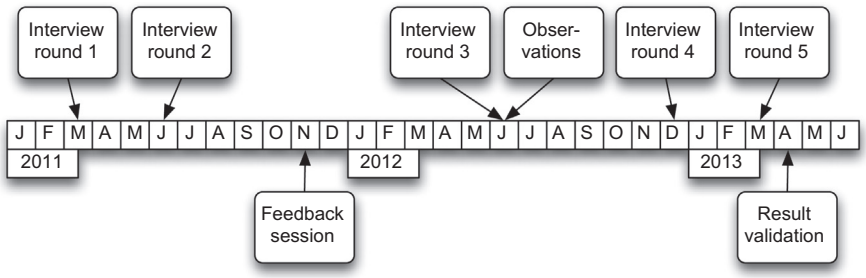

Fig. 1. Timeline of the research.

and asking additional questions. All interviews were recorded and transcribed.

3.3.2. Interview rounds

We performed five interview rounds, the details of which are shown in Table 3. When we started to study the organization, our main purpose was to study the Lean and Agile transformation. During the two first interview rounds our goal was to gather understanding of this whole large-scale transformation, e.g., why and how it is done, what the main challenges and successes were, and what challenges and lessons learned the organization had encountered for scaling agile development to a large distributed organization.

We selected the interviewees jointly with management, aiming for people with different lengths of experience and working in different roles. The first interview round of seven managers and a coach provided us with an overview of the organization's history, motivation for the transformation and the main transformation steps. To get a broader view of the transformation in the whole organization and to enable triangulation, during the second interview round we interviewed 31 persons—20 from Finland and 11 from Hungary—from different roles and teams, and with differing amount of experience in the organization.

The third and fourth interview rounds were follow-up interviews to study how the transformation was progressing. During the third round we discussed topics that we found interesting based upon the two first interview rounds, such as how the Product Owner structure and Scrum-of-Scrums practices were evolving. During the fourth round we focused on the integration of the US site in the transformation, as that was a recent development.

Based upon the two first interview rounds, the use of Communities of Practice arose as one of the most interesting topics for further study. The organization viewed CoPs as one of the main practices used to support scaling agile development. During the third and fourth interview rounds, we could see that the CoPs had evolved and had a high importance placed on them. Thus, it became evident to us that CoPs played an important dual role:

1. CoPs were one of the key elements supporting the on-going transformation from waterfall to agile development, or "the journey" as the organization called it, and
2. CoPs were a key practice to support the large-scale agile implementation.

For these reasons, we asked for a possibility to perform an interview round focusing only on CoPs. Fortunately, the organization agreed and we were able to do a fifth interview round. Moreover, we got an opportunity to observe three CoP meetings, of the Coaching CoP, Feature Coordination CoP and Feature Design CoP. This helped give us an understanding of what really happened in CoP meetings, and provided a possibility for methodological triangulation.

During the fifth interview round, we asked the interviewees about their past and current experiences in building and participating in CoPs. Two of the interviews were individual interviews of managers, one leading the lean and agile transformation and the other working as a head coach. With the manager leading the agile and lean transformation, we discussed CoPs in general and all types of CoPs. With the head coach, we focussed on the End-to-End CoP, but also talked about other types of CoPs. The rest of the interviews were pair interviews that focused on three types of Communities of Practice that we deemed interesting based on our earlier interviews and after discussions with the manager leading the agile and lean transformation: the Coaching CoP, Developers' CoP and Feature CoPs.

The Coaching CoP was chosen as it was one of the oldest CoPs and had largely evolved during its' existence. The Developers' CoP was deemed interesting, as it had already once died and then started again. The Feature CoPs were interesting, as they had turned out to be key elements of the large-scale agile implementation as they took care of a large part of the coordination and communication between cross-functional teams. The End-to-end CoP was interesting because it had an organizational wide focus, and management considered it important for optimizing the overall product development flow.

From the Coaching CoP, we interviewed two Coaches, and from the Developers' CoP, two developers, one of which also had the role of team coach. From the Feature CoPs, we interviewed representatives of two different CoPs; a Product Owner and a developer who doubled as a team coach. We discussed the End-to-End CoP in particular with the head coach, who led it. In all the interviews, we discussed the specific CoP or CoP type that the interviewees had most experience of, but briefly talked about other CoPs that the interviewees had participated in, as well as about general characteristics of successful and non-successful CoPs.

The interviewees for the pair interviews were chosen by the company representative, our main contact person. The interviewees represented experienced participants of the selected CoPs, and had experience of several different CoPs. In one of the pair interviews, a Hungarian interviewee participated through videoconference from Hungary, while all other interviewees were from Finland, where the interviews were physically conducted. In the manager interviews, we discussed the development of the organizational support for CoPs, or the "CoP culture" at a general level, as well as a few individual CoPs.

3.4. Data analysis

All interviews were transcribed by a professional transcription company. The main data analysis method used was qualitative coding of transcribed interviews [29,30]. While waiting for the

Table 3
Interviews.

Interview round	Focus	Interviewees	Roles
1. 3/2011	Transformation overview	Finland: 8	Managers: 7, agile coach: 1
2. 6/2011[a]	Deeper study of the transformation	Finland: 20, Hungary: 11	Scrum masters: 6, team members: 13, line managers: 3, product owners: 7, technical management/architecture: 5
3. 6/2012	Transformation follow-up: PO role, Scrum-of-Scrums	Finland: 3	Product owners: 2, team member: 1
4. 12/2012	Transformation follow-up: US site integration	Finland: 2	Managers: 2
5. 1–3/2013	Communities of practice	Finland: 7, Hungary: 1	Managers: 2, coaches: 2, team members: 3, product owner: 1

[a] The sum exceeds the total number of interviews, as some line managers had double roles, e.g. also worked as Scrum masters.

transcriptions, based on the interview notes we created preliminary categories for coding, a start list of codes [30]. This list included categories such as "Scrum", "CoP", and "Transformation". We coded the transcribed interviews in Atlas.ti, a qualitative data analysis software, using the preliminary categories, open coding, as well as axial coding [31]. This resulted in codes below the main categories, such as "Scrum: Daily", "Scrum: SoS" (Scrum-of-Scrums) etc.

We coded the two first interview rounds after the second interview round. Both authors of this paper did this coding together, dividing the interviews between each other, and using the same hermeneutic unit.[1] When coding the first few interviews, we cross-checked each other's coding to ensure agreement, as well as discussed the codes together agreeing on changes to the preliminary codes.

After the initial coding, we extracted specific codes from Atlas.ti, and did in-depth coding, both open and axial. Based upon this, we prepared a presentation for the case organization, describing our findings: successes, challenges and improvement needs seen by the interviewees.

The third and fourth interview rounds were coded in a similar manner, after each interview round, but instead of a feedback session the findings were briefly discussed with company representatives during research visits. Before the fifth interview round we extracted the passages coded by Communities or Practice and Scrum-of-Scrum meetings from the previous data. These were carefully analyzed before creating a semi-structured interview guide for the fifth interview round.

As the fifth interview round focused on CoPs, these interviews were coded from that view point. A start list of codes was created based on the interview notes. The codes included a few high-level codes such as the names of all CoPs, as well as codes describing positive or negative opinions: "challenge/problem" and "success/positive". The start list of codes was improved by adding codes that arose from the data while coding the first few interviews. Examples of these include "CoP leader", "decision making", "rhythm", "agenda", "CoP invitation", "mindset" and "community".

The final results, presented in this paper, were based on the analysis and comparison of both the CoP and SoS codes extracted from the first four interview rounds, as well as all the codes from the fifth interview round.

3.5. Validation

To ensure the validity of our findings we took several actions both when planning the research and during the data collection and analysis. First, we used three types of triangulation in data collection: data, investigator and methodological triangulation [29,32]. We interviewed a large number of persons from different roles and organizational levels to get as realistic picture as possible (data triangulation), in most of the interviews two researchers were present (investigator triangulation), and besides the interviews we observed three CoP meetings (methodological triangulation).

Second, the data was analyzed carefully together by both researchers, who communicated actively both during the coding process and analysis phase, which ensured that both researchers agreed on all the codes, analysis and the findings. Third, the correctness of our findings were validated in several ways by the case company representatives: conducting a company feedback session, discussions with the company representatives, and having company representatives read and comment on this paper.

The results based on the first and second interview rounds were presented to company representatives in a feedback session. To this session all interviewed persons were invited and around half of them joined the session. Both sites, Finland and Hungary, were represented. During the session, a lot of questions were asked and the audience was eager to discuss our findings and progress they had made after the interviews. The company found the feedback valuable, and while, e.g., management challenged some of our less flattering findings, personnel in other roles confirmed them, and no corrections or additions to our findings came out of the session.

Moreover, we presented and discussed our findings from the third and fourth interview rounds in research meetings with several case company representatives. After the fifth interview round the results, in the form of the first draft of this paper, were sent for commenting to all persons interviewed during that interview round, as well as to one other representative of the case organization, who was not interviewed, but was very experienced in CoPs due to having participated in and led several CoPs.

We received detailed comments from this last mentioned experienced person, as well as from one interviewed manager. The comments confirmed our findings, and the detailed comments included only minor clarifications to the text.

4. Results

As described in the background section, the case organization had started the lean and agile transformation in 2009 and the full-scale roll-out took place in 2010. The communities of practice were introduced at the same time as the full-scale roll-out. The main motivation for the case organization to institute the use of CoPs was that the CoPs were a proposed solution both by the consultants that the case company employed, as well as in the literature on scaling agile development [7,10].

In this section, we first give concrete examples on how and why the case organization built CoPs, and how they evolved over time. Second, we describe the characteristics of successful CoPs. Third, we discuss how the case organization supported the CoPs. Fourth, we describe the different phases the organization went through while building and using CoPs. We end the results section by presenting a typology of CoPs used in the organization.

4.1. CoPs in the case organization

In this section, we discuss our results related to RQ1: What kinds of communities of practice emerged or were created in the case organization? We present four examples of successful CoPs in the case organization: the Coaching CoPs, the Feature CoPs, the Developers' CoP, and the End-to-End CoP.

4.1.1. Existing communities of practice

During the fifth interview round, we asked our interviewees how many CoPs the organization currently had, but could not get a definitive answer. Some respondents reported around 20, and some that the number is somewhere between 20 and 50. Looking at the organization wiki page where you could find links to all CoPs, our interviewee noted that the list does not seem to be up-to-date. Some CoPs that he knew exist were not there, some that were there, had according to him already ceased to exist. However, when we asked our interviewees what are the most successful CoPs, the same CoP names started to come up: Feature CoPs, Coaching CoP, Developers' CoP, End-to-End CoP, Functional Verification CoPs, and a few other testing related CoPs. Next, we will have a look into these four first mentioned successful CoPs: what they are and how they have evolved. Table 4 summarizes the key findings on these CoPs.

[1] The analysis "container" in Atlas.ti.

Table 4
Examples of CoPs in the case organization.

	Coaching CoP	Feature CoP	Developer CoP	End-to-end CoP
Predecessor	Scrum master CoP	SoS and system CoPs	Previous developers' CoP	Program weekly
Predecessor's role	Sharing experiences in applying Scrum practices	SoS: inter-team coordination, Syst.CoP: feature design	Design rules etc.	High-level R&D progress monitoring and way of working
Participants	Scrum masters/coaches	Representatives of cross-functional teams	Developers	Persons wanting to make a difference: managers, product owners, coaches, team members
Current role	Improving the whole organization and its competences	Supporting coordination and design between teams developing a common feature	Software craftsmanship, unifying tools and technologies over product areas	Improve the way of working and optimize the end-to-end flow
Location and distribution	Cross-site (Finland/Hungary), and site-specific CoPs	According to the distribution of the product areas, mostly cross-site	Finland, Hungary	Finland, Hungary, US
Rhythm	Weekly	Feature coordination CoPs weekly, Feature design CoPs on a need basis	Bi-weekly	Weekly
Challenges faced	Lack of common activities and goals after day-to-day problems were solved	Organization-wide SoS did not work: several trials to find a solution to inter-team issues	The first trial ceased to exist due to lack of a CoP culture	Involving the US site due to the time-zone difference
Successes	Promising new start with goal to improve the whole organization and its competences	After many trials a functioning solution for inter-team coordination and design	Promising start-up after the death, with a passionate leader, interesting topics and goals	Broad representation of organizational levels and sites facilitates decision making and moving things forward

4.1.2. Coaching CoP

The Coaching Community of Practice (Coaching CoP) was one of the first CoPs that started. In the beginning, it was called the Scrum Master's CoP as at that time the case organization had mainly Scrum Masters. Later on, as the number of coaches started to grow, as many of the teams changed from Scrum towards Kanban or Scrumban [33], the name was changed to Coaching CoP.

4.1.2.1. Early stages. Our interviewees reported that in the beginning the Scrum Master CoP was working quite well, as people wanted to learn more, improve their teams and search solutions for their problems.

> *And then we have the Scrum Master CoP, where we discuss these things that have been challenging in the teams and try together to find solutions to those.*
>
> – Scrum Master, Finland 2011

> *Then there [in the Scrum Master CoP] was a lot more discussion on the basic routines, how we organize retros, what this agile thing is, and how we should behave and what is our role... And then it was really active, I have to praise that community, because that helped me to come into this role. I participated in an Agile course and Scrum Master training, but I had not come into this lean and agile thinking in the same way as I did when I saw the mentality here [in the Scrum Master CoP]. It helped me to come in.*
>
> – Coach, Finland 2013

At that time the Scrum Masters tried to "practice as they preach". Thus, the Scrum Master community had several common events during each week. They held a daily scrum meeting "the Daily" for Scrum Masters, and had a CoP "backlog", and held "Backlog Grooming" and "Impediment Handling" sessions. They also had weekly "Training" sessions, to which they either invited internal or external people to present on interesting topics, like conflict handling or they just watched a video together and discussed it. However, the CoP experienced several problems, including attracting participants to the meetings, and getting the idea of a coaching backlog to really work, as illustrated by the following quotes:

> *Coach 1: "We have had several things that, e.g., every morning Daily and there comes those who come. We have tried to work the way we want the teams to work."*
>
> *Coach 2: "We have failed miserably."*
>
> *Coach 1: "I don't know. We haven't had a continuos backlog at any time. We have had trials of a Coach backlog and Impediment backlog during this two and half years. ... But they have always died for some time, then somebody wakes them up again."*
>
> – Coaches, Finland 2013

The Scrum master/coach community has never had a real leader or even a facilitator. The meetings were organized around a loose agenda with discussion items collected by the participants to wiki pages. A brief memo was written during the meeting for those not present to be able to find out what was discussed.

> *We have rotated the responsibility for facilitation amongst the participants. Usually, it is the one who is there first and connects his laptop to a projector. Those meetings have been quite self-organizing, not much facilitation, we have had the agenda ready, so we have just discussed those topics one after another. (...) The number of participants has been quite low recently, I guess people have so much other things going on. (...) But we also write the wiki memo so it is possible to follow what has happened.*
>
> – Scrum Master, Finland 2011

4.1.2.2. Crisis and maturation. However, after this first phase of the agile transformation was over, the Scrum master/coaching CoP was in trouble. The participants did not form a tight community with common goals or even a common backlog for work tasks. People started to vote with their feet: some participated only occasionally, when they had the time, and some totally stopped showing up for the meetings, as they found that the meetings dealt with irrelevant topics, did not provide sufficient follow-up on issues, and participants had difficulties agreeing.

> *But to be honest, I was fed up with that [Scrum Master] CoP last year. (...) We discussed irrelevant issues there. That is my view. (...) For me it wasn't an issue to discuss who should take the laptop to a demo session, the Scrum Master or the team, so come on, that is not relevant at all. (...) And I started to be bored there. And I decided not to take part of that. (...) I thought at that time that was not important for me, because I didn't gain anything.*
>
> – Developer, Hungary, 2011

We don't practice what we preach. The way of working in this community is not what we are trying to coach for others, we don't have a backlog, we don't follow-up in the sense that we would visualize what we are doing. We don't catch things up. We don't agree on things together—that is missing from this community. It is probably one of the reasons why this CoP doesn't work and people get frustrated.

– Manager, Finland 2013

However, the CoP did not totally die, even though the number of participants in the meetings was low—around three to ten—for some time. Instead, the community started to search for better forms and goals.

During the fifth interview round, most of the teams used a combination of Scrum and Kanban, "Scrumban" [33] and had a team coach, instead of a Scrum Master. Most of the team coaches were dedicated to one team, but some were shared between two teams. Some of the coaches were more "technical", i.e., they concentrated mainly on technical tasks in the teams and took care of the coach role in addition to their developer role. A few of the team coaches were also line managers. Moreover, the organization had a few full-time organizational coaches, including the head coach, who concentrated on the development of the whole organization. Thus, the participants of the coaching community were at that time quite diverse, both regarding their roles and the length of their experience in the coach role.

At the time of our fifth interview round, the Coaching CoPs had been further developed. As the Finnish site had a few teams working for a different product, and the Hungarian site had other products all the time, there was a need for site-specific Coaching CoP meetings, in addition to the product-specific Coaching CoP. Thus, the first 30 min of the weekly Coaching CoP meeting was site-specific—in Finland only— followed by the cross-site product specific CoP for the next hour as a videoconference. Thus, the participants could choose, whether they wanted to participate in one or both parts of the session. Hungary also had site-specific Coaching CoPs, but at a separate time.

The organization wide Coaching CoP, that we studied, had also broadened its goal and responsibilities. The participants were no longer concentrating on solving and discussing day-to-day problems of the Scrum Masters and Coaches. Instead, their goal was to improve the whole organization and its competences.

. . .[currently] we are planning a Coach day. We tackle problems and discuss how we could improve this organization, and our competences. Meaning the competences of the whole organization.

– A Coach, Finland 2013

To support the organizational development and to align the work, the active coaching CoP members had built a "Cool Wall" and a "Coaching backlog" that were placed in the common area for the whole development organization in the Finnish office.

On the "Cool Wall" they listed the main topics that the organization needed to develop further and divided them into sub-areas under columns "seriously uncool", "uncool", "cool" and "sub zero". Under the column "Seriously uncool" were topics that needed a lot of work, and the "sub zero" column contained items that were "pretty cool".

The coaching backlog, on the other hand, was a backlog of concrete actions, divided into tasks, and placed into Kanban board with the columns, "not started", "In progress" and "Done". This backlog was recently created, and during our fifth interview round we could see how the backlog was already in active use with a lot of tasks. For example, "Software Craftmanship Day" and "Coach Day" were under preparation, as well as more technical items, such as test automation improvements had started in the form of coding camps and courses. At this time there were a few full-time organizational coaches driving this improvement step. The idea was to have all coaching tasks that required support from the community on the wall.

The idea of this (the Cool Wall and Coaching Backlog) is that it is not only the coaches building this, but the whole organization (. . .) Meaning management, Product Owners, team members, coaches. (. . .) And then the coaches decide what they do to these things and then we have a Kanban Board where we have a backlog.

– A Coach, Finland 2013

When interviewing a technical team coach, we noticed that the ideas of the organizational coaches were noticed in teams, as the team coach mentioned that he aimed to prioritize the development team's work according to the priorities at the "Cool Wall" and "Coaching Backlog":

. . .I think it [the Cool Wall] is a good thing and I have taken something from there also to our own team. If we need to choose between two tasks and one helps more than another to move things on the cool wall forward, then we will do more those tasks and support that work.

– Team Coach, Finland 2013

In the fifth interview round, we discovered that another activity, started already in the beginning, was still active, namely a weekly meeting to learn and discuss new lean and agile topics together. That "learning community" meeting was called the *Coaching Guild* and had recently concentrated on watching and discussing video clips on agile development.

One of the recent challenges of the Coaching CoP has been that the Coaching Community now had persons at different levels in their coaching career, there were *experts*, the experienced organizational coaches, as well as new team coaches who have team coaching as a side job, in addition to being a developer. As these persons have a totally different level of knowledge on agile methods, as well as different problems in their daily work, it did not seem to be easy to find discussion topics that would be interesting and important to all.

Even though this Coaching CoP and the coaching community was active during the whole organizational transformation, and thus probably the most successful example of a learning and knowledge sharing community in this organization, it has also had serious challenges. The community clearly suffered from a lack of common activities and goals. Only learning together and sharing knowledge was not enough to keep the community alive. The community did not have any leader, even though the organization had a head coach during our fifth interview round. The recent activities of the organizational coaches in building the "Cool Wall" and Coaching backlog for advancing the organization wide issues seem as a successful start to involve and align the whole organization. This seems to be a good beginning to create a common goal and common activities for the coaching community and even for the whole organization.

4.1.3. Feature CoPs

The Feature CoPs evolved out of Scrum-of-Scrum meetings. In the beginning of the agile and lean journey, the case organization had around 25 cross-functional teams, all using Scrum. According to Scrum, the main coordination mechanism between the teams are Scrum-of-Scrums (SoS) meetings [25]. That is, after each team has had its' own Daily Scrum meeting, in which each team member answers the three Scrum questions,[2] the team sends a representative to a Scrum-of-Scrums meeting, where each team representative answers similar questions on behalf of his or her own team and that

[2] 1. What have you accomplished since the previous meeting? 2. What are you planning to do before the next meeting? 3. What obstacles are in your way? [24].

way shares information between the teams. In the case organization these organization-wide Scrum-of-Scrum meetings were not organized daily. Initially, they were held three times a week, later once a week, and finally they ceased altogether.

At that time, the development was distributed to two sites, in Finland and Hungary. Thus, the SoS meetings were organized using videoconferencing. As the number of teams was large, and the teams were working on separate modules of the product using different technologies, they did not have much in common, and sometimes did not even understand what the others were doing. Thus, the SoS participants were not very interested in what others were reporting nor did they know what they could report that would be interesting or useful for the other teams. This led to a situation in which most of the teams reported "Nothing to report" since there was so little common ground. Obviously, this was a sign of a dysfunctional coordination mechanism. The organization tried different ways to improve the SoS meetings by facilitating and educating the teams on what to report and discuss there, as there still was a clear need for coordination between the teams. Many of our interviewees noticed problems due to this dysfunctional coordination and communication mechanism.

4.1.4. Feature SoS meetings

As a solution to the problems of organization-wide Scrum-of-Scrum meetings, the organization additionally started arranging weekly Scrum-of-Scrum meetings inside specific features, so called *Feature SoS* meetings. At that time, the features were very large, with 3–8 teams working on a feature for several months or even in some cases, years. The feature SoS meetings turned out to work well, as there was much more common ground than in the organization-wide SoS meetings.

The Feature SoS meetings covered topics that were important and interesting to discuss together frequently. The participants of the Feature SoS meetings were one or more team members from each team working for a specific feature, and in most features also the Product Owner or Product Owners participated, as well. In some cases, a whole team participated. As the implementation of most of the features were divided between the two sites, the Feature SoS meetings were organized using videoconferencing. As the following quotes indicate, the Feature SoS meetings started to work pretty well, and participants found them useful:

> *Feature SoS meetings are pretty good, because people participating in them work on the same things, talk "the same language" and have a common goal.*
>
> – Product Owner, Hungary 2011

> *This [Feature SoS] is a good meeting, since this is the only place where we are all together at the same time (...) Here we can discuss everything. We have tried to keep it this way, that we don't have agenda, but discuss what is done at different teams, if there are any problems or other common topics.*
>
> – Product Owner, Finland 2011

4.1.5. Feature CoPs

For some time, the organization had the organization-wide SoS meetings once a week and Feature SoS meetings for each of the features once a week, as well. However, soon the organization-wide SoS meetings died. As the word "SoS" had bad connotations after the not so successful organization-wide SoS meetings, and as the CoP culture developed in the organization, some started to call the Feature SoS meetings "Feature CoPs" instead.

Moreover, other Feature related CoP meetings were started between the team members working for a specific feature, e.g., System CoP, for systemization, i.e., high-level collaborative system design. The systemization work was previously, in the waterfall model, done by specific designers. When creating the new cross-functional teams, the idea was that the teams would take care of the systemization, as well. However, in the beginning most of the teams did not have enough knowledge to do systemization, thus instituting systemization CoPs that had participants from teams developing that feature helped:

> *It is a community formed of ad-hoc of persons, who want to or are interested in making decisions. (...) Let's say that in the next sprint we are going to have a user story with some complex task. If we have identified that, we aim to arrange some system planning beforehand, to draw a sequence diagram. To do a proposal. The System CoP invitation is send to all teams [working on that topic], that every team should send at least one representative. And then we make a decision; these technical decisions that are very difficult to make in the beginning. (...) I think this works fine and we have managed to make that kind of decisions that people don't complain about afterwards. And everybody has a possibility to affect this decision making. And when the decision is made, it holds.*
>
> – Developer, Finland 2011

4.1.6. Feature coordination CoPs

At the time of our fifth interview round the product had been divided into four product areas, two of them divided between the teams in Finland and Hungary, one between the teams in Finland and the US, and one had just local teams in Finland. Each of these areas arranged Feature CoP meetings between the teams working in this area. All the areas arranged weekly SoS style CoP meetings, where the team representatives reported what their team had been doing, what they were planning to do, and if they had some obstacles. Often there was some discussion in the end. In most cases, each team sends a representative chosen as round-robin style to these meetings. From some teams a few team members joined, and sometimes even the whole team participated. Some of our interviewees still called these weekly coordination meetings Feature SoS, some called them Feature CoPs, and when asked, they said that they did not think that the name matters. For clarity, we refer to all of these as *Feature Coordination CoPs*.

Besides the Feature Coordination CoP meetings, the product areas arranged Feature CoP meetings on technical topics, e.g., for systemization, architecture planning and making decisions related to technical issues. From now on we call these *Feature Design CoPs*. Feature Design CoPs were normally invited on a need basis and there the team representatives were often different persons than in the Feature Coordination CoP meetings, as different things interest different persons.

> *In practice we have noticed that it is better to have this [Feature Coordination CoP] as a general meeting and then have separate meetings for technical things, as those meetings have slightly different participants, those who are interested in that, so it has been easiest to have separate meetings.*
>
> – Product Owner, Finland 2013

Our interviewees emphasized that most often, the Feature CoP participants join the meetings on voluntary basis—those who are interested or need to be there attend. Each team just makes sure that they have at least one representative when needed. To some of the Feature Design CoP meetings, persons from other closely related areas might be invited, if they have technical connections that need to be discussed together, or to learn from another area that has dealt with similar issues earlier.

In the fifth interview round, we interviewed persons from two product areas. The first area had five teams: three in Finland, and two in Hungary. Consequently all Feature CoP meetings were organized using videoconferencing. In this product area, the weekly

Feature Coordination CoP meetings had a time slot of one hour. Normally, the first half an hour was used for a SoS style coordination discussion, and the rest was used for discussing the general "way of working" issues, e.g., testing, continuous integration and other improvements.

Previously, they had more technical discussions in the meeting, but noticed that it is better to have technical Feature Design CoPs separately, as the participants might be different. The Feature Coordination CoP meetings always had a facilitator, who typically was one of the team coaches. The facilitator, who had taken the role of a "community coach", also saw to that communication between the teams was working on a daily basis. In addition, the product area had a wiki page, where anybody could add topics on the agenda, even though in practice it was often the team coaches and the Product Owners who added most of the topics. Besides the Feature Coordination CoPs, this product area had started to arrange regular Feature Design CoP meetings once or twice a month.

The second product area had only three teams, all located in the same large open office in Finland. Their daily communication was easy, and thus their weekly Feature Coordination CoP meetings were arranged only when a particular need arose. Instead, they arranged Feature Design CoP meetings on specific topics, whenever someone felt that there was a need. For example, when making big architectural changes, refactoring, they arranged Feature Design CoP meetings to which they even invited people outside their product area to receive broader input. These were considered successful events.

> *This year we haven't felt SoS to be important, as we have only three teams sitting side by side. The teams or individuals from the teams discuss daily. We have had a few SoS meetings when somebody had done some bigger changes and wanted to explain that, and then informed beforehand that "In the SoS I will explain this to everybody, come to listen if you want". And those have been useful. The SoS is a natural meeting to share that kind of information.*
>
> – Developer, Finland 2013

However, previously, one year ago, this product area had been distributed: in addition to the three Finnish teams, there were three subcontractor teams, all working at different locations. Then, the weekly half-hour Feature Coordination CoP meetings had been very important, as the subcontractor teams could ask questions in the meetings and somebody would take responsibility for helping them.

> *It was working well at that time, we had many teams and we had useful things to discuss there. It was not only that "we are doing this and that" and nobody really is interested, but then it was useful, people asked for help, received help and told what they had done. (...) Then there was also a teaching aspect for those subcontractor teams, when others told what we had done and why.*
>
> – Developer, Finland 2013

It became clear that in the both product areas, the culture for inviting Feature Design CoP meetings was very open and well-functioning. Whenever there was a need for a meeting, somebody invited it, people with an interest in the topics participated, and decisions were made.

> *These meetings (Feature Design CoPs) are invited by individuals, it is not Product Owner driven. When somebody feels that we need to do something, he takes the responsibility and invites a meeting. And luckily, there has been interest, and the people that have been needed have participated, and also those that have been interested to learn more have participated. Thus, it is not only the best gurus that discuss together. Instead, it is an open meeting that everybody who wants may participate in. And the invitation is send to everybody. I think it is very important, because if there will be cliques, bad things will happen.*
>
> – Developer, Finland 2013

The regular Feature Coordination CoP meetings had found their place and were working well, when the area had several teams distributed at different sites. When only a few teams were working locally, the need for regular meetings was less important. One of our interviewees even commented that he was happy that in their product area they had accepted that regular Feature Coordination CoP meetings are not needed, so they do not waste their time on organizing meetings just because it is a routine, but have meetings only when they really are needed.

4.1.7. Developers' CoP

4.1.7.1. First trial. The Developers' CoP was one of the first CoPs that started during the early phases of the transformation. However, it did not find its' place back then, and ceased to exist. One of the reasons for its' early death, according to our interviewees, might be that back then, this CoP concentrated too much on design rules. Moreover, at that time, the CoP culture and mindset was not yet fully developed. In particular for developers, it might have been difficult to see the benefit from participating in that kind of CoPs. Third, this was a site-specific CoP, only involving Finland, even though collaboration between the sites would certainly have been beneficial.

> *I think the problem was that when this Developers' CoP in the old days was started it was here only and not really (...) Hungary wasn't really part of that one. (...) And it stayed on for roughly a year, but then it died because nobody participated.*
>
> – Developer, Finland 2013

4.1.7.2. Restart. In the beginning of 2013, the Developers' CoP was restarted. This time around, it concentrated on a popular topic in the organization, namely software craftmanship.

> *It was off more than a year. But then, as things came up in the chats, that people are really struggling with these day-to-day development issues, then we talked on a coffee break that now I think we need to do something about this, and restart the Developers' CoP and then it was agreed. Now we start it again. A whole lot of stuff then came up and I think it was a good decision to restart it.*
>
> – A Developer, Finland2013

The Developers' CoP is now organized as a videoconference between Finland and Hungary. However, it was decided that the new site, the US, will not be included due to the eight-hour time difference.

> *It's really like one or a maximum of two hours' time slot where everybody would be present. (...) So that is why we decided that the Developer CoP doesn't include [the US site]. It's sad. I think we should really talk with them as well, But due to the time difference, it's really hard.*
>
> – Developer, Finland 2013

At the time of our fifth interview round, the Developers' CoP had a very good restart and a passionate leader who had plans and goals for the CoP:

> *I think the main thing I want to build with the Developers' CoP is kind of ... really a community around it. So that people that currently are not participating, would come there. To discuss their problems. What I would like that to be, some kind of forum where we could share the good practices and try to avoid the bad practices. That's the direction I want that CoP to go. (...) We could actually drive things forward with this CoP.*
>
> – Developer, Finland 2013

The Developers' CoP had already established a mailing list for active members. The idea was to have a forum for asking and answering questions. In addition, the CoP had a "software craftmanship" chat forum, which was one of the most active chat forums in the organization. The CoP wiki page, contained both meeting agendas and minutes from previous meetings, as well as useful material, such as coding style guidelines, instructions for how to set up environments, etc.

As the product was divided into four product areas, the Developers' CoP aimed to break down the walls between the different product areas. The product areas had used different technologies and tools, but now they were increasingly unifying these, adding to the amount of joint topics to discuss and share. Thus, the Developers' CoP might have an important role in sharing knowledge between developers, and thus better possibilities to succeed than it had during the first attempt.

I have high expectation with this Dev. CoP. At least that through that different areas would talk more with each other.

– Developer, Finland 2013

4.1.8. End-to-end CoP

The idea of the End-to-End Flow Optimization CoP, or simply the End-to-End CoP, as most of our interviewees called it, is to improve the product development flow through the whole organization and to remove bottlenecks that inhibit that flow.

4.1.8.1. Predecessor. The End-to-End CoP was born out of a meeting called the "Program Weekly" led by the Program Manager. The Program Weekly was a high-level monitoring meeting, monitoring product development progress at a high abstraction level, but participants also discussed the way of working and improvement ideas. This meeting slowly died, and after half a year the idea to start a CoP around the same topic came up. At the time of the interviews, the End-to-End Cop had been active for a bit over a year.

4.1.8.2. Current stage. The end-to-end CoP was arranged once a week as a videoconference. However, if the agenda did not look good, the meeting was canceled. The CoP had 20–40 participants, both from Finland and Hungary, and sometimes even a few participants from the US site, if the meeting was scheduled so that they could participate. A few local meetings had been organized as well. Participants came from all over the organization: team members, managers, coaches, and Product Owners. According to our interviewees, the participants were active persons who wanted to make a difference in the organization and improve things. The meeting was facilitated by the Head Agile Coach.

There, we concentrate on what are the challenges, what we should do to be more efficient as an organization. If we don't have enough test environments, or certain knowledge, or the way-of-working is bad, or collaboration is not working. From a very broad scale you can bring in topics and there is the head coach and often other managers, people who can easily take things forward.

– Product Owner, Finland 2013

At the time of the first interviews of the fifth interview round, the leader of the End-to-End CoP wanted to test whether this meeting really is needed:

I have a test going on now (. . .) I haven't booked the meetings yet. If there will be more than five questions on when it will continue, it will continue. (. . .) Because in general, when you have a meeting, people will come just because it exists, but if there is a need for it, then people will ask for it.

– Manager, Finland 2013

During our later interviews it became clear that people had asked for these CoP meetings, thus they were organized again.

According to all the interviewees who mentioned the End-to-End CoP, this CoP was seen as one of the successful ones: it was well-organized with a clear leader, a good agenda and interesting and important topics with the potential to advance the organization and affect the daily lives of the participants. In addition, the broad scale of representatives from different organizational roles, from team members to managers ensured that different points of view from the whole organization were represented. Having people from all organizational levels made it easier to make big decisions and move things forward. The fact that all sites, Finland, Hungary and sometimes also the US were represented was seen as beneficial. However, having the US site as a participant clearly caused challenges, as it would mean that the meeting should be arranged in the evening time at the other sites, and early in the morning in the US. For a large meeting like this, that might mean a reduction in the number of participants.

Even though the topic of this CoP was very broad and it could be difficult to attract attention from a large group of people from different parts of the organization, the feedback so far had been very positive according to all interviewees mentioning this CoP. Having a good and interesting agenda, and a leader that keeps the discussion active seemed to be the key success factors of this particular CoP.

This end-to-end flow optimization CoP . . . it requires a pre-prepared agenda and topics beforehand, that people can see "whether it touches me or not". If that would be missing, I believe that participation could be low. . .

– Manager, Finland 2013

I like that CoP [End-to-End CoP] and I like the problemsldots or I don't like the problems that we are talking about there, but I like that we are talking about the problems and, we are trying solve them.

– Developer, Hungary 2013

4.2. Characteristics of successful CoPs

In this section we discuss our results related to *RQ2: What were the characteristics of successful communities of practice?* We used a subjective definition of success, i.e. we considered a CoP successful if the interviewees reported it to be so. Furthermore, we explicitly asked our respondents to list characteristics of successful CoPs. Based upon our data, we were able to identify eight characteristics of successful CoPs in the case organization. These are summarized in Fig. 2, and discussed in more detail below.

4.2.1. Interesting topic and concrete benefits

The main characteristic of the successful CoPs seemed to be that the participants felt that they really gained something from every single CoP meeting. The benefit should preferably be something concrete with an impact on their daily work.

In the end it really depends on people. . . . And that people feel like that they receive something. They have a benefit from participating. . . . So every meeting should provide something. Something tangible that you can take with you.

– Developer, Finland 2013

I participate in the Developers' CoP just because it is an interesting topic, and there we discuss topics that interest me. Topics that interest me are the reasons why I participate in those CoPs that I participate. Nobody has told me that "why don't you participate this, or that you should go there".

– A developer, Finland 2013

Thus, the CoP topic should be important and interesting for the participants. The CoPs that were build around something very concrete, that was closely related to the daily work of the participants, seemed to work well, e.g., testing related CoPs. CoPs dealing with topics that are more general and outside the "Daily Flow" of activities are more difficult to keep active and useful.

> *The more it (the CoP topic) has direct links to daily work of the persons, the easier it is to create this link, that "if I participate in that CoP, then it gives to my daily work this and this". The larger the distance from the daily work the more work you need to do to create that CoP, motivate people and keep the CoP active.*
>
> – Manager, Finland 2013

As people have limited time, they participate only in the CoPs they feel that will benefit them the most.

4.2.2. Passionate leader

The second factor we identified was to have an engaged, passionate leader or facilitator:

> *I guess they [the successful CoPs] have a driver who is passionate about it. ... I guess that's one of the most important stuff... I'm not saying that you have to have only one leader or driver, but at least one.*
>
> – Developer, Hungary 2013

Our interviewees emphasized that the CoP leader should preferably be an expert who people appreciate, within the area of the CoP. When people know that this expert is participating or even leading a new CoP, they can trust that important issues are discussed and decided there. Thus, it came clear that if a person who is "nobody" in that community decides to invite a CoP, even though the topic might be important, that CoP will most probably not succeed.

> *The better known and appreciated person he (the person setting up a CoP) is in that context, where he is building the CoP, the bigger chances he has to succeed.*
>
> – Manager, Finland 2013

Quite often a typical CoP leader in this organization was a coach, e.g., a team coach, who is interested in driving things forward in a wider organizational perspective. For example, the Developers' CoP was lead by a team coach, the End-to-End CoP by the Chief Agile Coach, and the Feature Coordination CoPs were facilitated by one of the team coaches in that area.

4.2.3. Proper agenda

Having a proper agenda for every CoP meeting was highly appreciated. Most often the agenda was sent in the CoP invitation, which was a calendar invitation send to everybody in the organization, and could also be found from the CoP's wiki page.

The agenda sent in the invitation was an important source of information for participants for deciding whether to participate in that specific CoP meeting at that time or not. If the agenda was good, i.e., included interesting and important discussion topics or decisions to be done, it was more probable that an individual decided to participate in it. In Feature Coordination CoPs, the agenda was basically always the same, but also those meetings informed participants if there was something special on the agenda, e.g., discussion on some special topic or decision making.

The agenda of the most CoPs could be found on the CoP's wiki page, and anybody could add a topic to the agenda in wiki.

A good agenda was felt to ensure that something really happens in the CoP and the participants feel that they gain something from participating in that CoP meeting. Even though everybody could add their items on the agenda, it was clear that, in many of the CoPs, it was the responsibility of the CoP leader to ensure that the agenda was good. If there was no proper agenda, some of our interviewees felt that it might be better to cancel the meeting.

> *I really like if there is a proper agenda, for a meeting... Otherwise, if there is no such kind or leader or driver, CoPs can end up without no agenda. People will start to feel that it's just wasting time because you are just talking about basically nothing.*
>
> – Developer, Hungary 2013

Besides having an agenda, it seemed to be important that the CoP leader or facilitator lead and even time-boxed the discussions to keep discussions going nowhere from taking up valuable time. Moreover, our interviewees emphasized that decisions or actions taken in a CoP meeting should be followed up in the next meetings, and thus added into the agenda of the following meetings, as well, as follow-up items.

4.2.4. Decision making authority

Many of our interviewees mentioned that decisions are and should be made in the CoPs. They felt that it was part of the organizational culture that the CoPs had an authority to make decisions. They pointed out that this decision making culture existed already during the plan-driven era, but the lean and agile transformation had emphasized it even more.

> *Currently I guess it's [Dev. CoP] more of a this kind of discussion forum, but I think we have authority... yes, I know, we have the authority to decide even.*
>
> – Developer, Finland 2013

The different sites have had slightly different cultures regarding decision making. At the Finnish site, people have been more active decision makers, whereas at the Hungarian site, people have been encouraged to be more active in decision making. This difference is most probably due to the different national cultures, as well as, to the fact that previously the Hungarian site was in the position of subcontractor.

Our interviewees felt that it is important to let the participants know already in the CoP invitation whether decisions will be made in that meeting and on which topic. Then, people who want to participate in making that decision know to participate.

> *This kind of CoPs or meetings where we want to decide something. Usually the invitation will clearly state that "we would like to have a decision about this, so if you are passionate about this ... make your voice heard".*
>
> – Developer 1, Hungary 2013

> *True. And that's really the main thing that the invitation directly says that "we now decide, choose to come or not to come." If you choose not to come, then that's your opinion then. That you relay on the others to decide for you.*
>
> - Developer 2, Finland 2013

The interviewees did not feel that it was important that all persons affected by the decisions participated in making them. As everybody is informed about the meeting and the agenda, those who are interested and want to participate attend. For example, a Developers' CoP meeting had around ten participants from four development teams, when the whole product had almost 40 teams, our interviewees felt that this CoP meeting was fully eligible to make decisions:

> *...if we can decide with those ten people that this is the direction we want to go. We can actually be quite sure that the rest will agree, as well. But if we can't agree with ten people that we want to do this, then we need some more discussion. ... They [the participants] were*

passionate. And that's good for me at least. If you show your interest, then you have the authority decide. If you don't care... well, that's, you can choose to do so. ... It's really up to... yourself to do what you want.

– Developer, Finland 2013

However, not all felt that the CoPs had totally solved the decision making challenges, instead this could be one area for further development in the organization's CoP culture.

...we still have had problems in decision making. People feel that the same things are talked over a week after a week. ... If you have only a democratic approach, it doesn't work, it is inefficient. You start to have challenges as people feel that it is not my problem. It's the CoP's problem. But the CoP is only a tool for decision making, and decision making is often the biggest challenge there.

– Manager, Finland 2013

4.2.5. Open community

Building a successful CoP requires that a community around the topic already exists, or can be created. When we asked what people had learned on building CoPs, one of the interviewees replied:

Maybe the biggest learning has been that just sending invitations and creating a Wiki page or somethingldots that is not yet a CoP. It requires something else to fly. And that else requires preparation, and some clear, tightly defined topic that enough persons can identify themselves. Then it works. But you need to create the communityldots

– Manager, Finland 2013

First, there needs to be enough people interested in that topic. Building a community requires talking to a large number a people and identifying topics and problems that are common to them, and something that they feel they would benefit from discussing and solving together.

The company culture regarding the CoPs and the attitude on how much, e.g., individual developers care about the bigger picture and participating in common activities counts.

I think that everything depends on what people want by themselves, whether they want to participate in, make a change. It would be difficult to get anything forward by just booking meetings. When people have the power to vote with their feet: "An invitation came. I couldn't care less. I'm working now".

– A developer, Finland 2013

As an important part of the company culture our interviewees mentioned the openness of the CoPs. They considered it very important that the CoPs do not form into cliques, e.g., groups of experts that meet and decide something behind closed doors. Instead, they emphasized that successful CoP culture is open, the invitation is send to everybody, the agenda and decisions can be found in the Wiki and persons not in expert positions regarding the CoP topic are welcomed to join in CoPs to participate or just to listen and learn.

...they all have the same principle that all are invited, the whole organization, and then anybody can participate. They try to be as open as possible so that there wouldn't form any cliques, because it would mean hidden information.

– Developer, Finland 2013

4.2.6. Supporting tools to create transparency

Most of the CoPs we studied had wiki pages. The CoPs were listed in the organization's internal wiki, with links to each individual CoP's pages. On the CoP's wiki page you could find at least the meeting agendas and minutes from the previous meetings, sometimes also other information. Up-to-date information on the wiki page was seen important to CoP success. It was seen as the most important informing channel that contributes to the transparency.

There in the Wiki all can access that information, so even though you might not be able to participate in some CoP during some week, you can see from the Wiki what decisions were made, how things agreed in the previous meeting have progressed. I think it has succeeded well, that the information is very transparent, what happens in the CoPs, that they are not closed Communities cliques. Even though I woudn't visit the Developers CoP at all, so if I follow the wiki page I'm pretty much on the map on what they have talked about and decided there.

– Developer, Finland 2013

Besides the wiki, some of the active CoPs had also a mailing list and a chat channel to discuss topics between the meetings. For example, the Developers' CoP had both an active chat forum on "Software Craftmanship", where developers could discuss their daily issues, as well as a mailing list to discuss broader topics. Having these active supporting forums were seen as signs of a functioning community.

There in the Wiki all can access that information, so even though you might not be able to participate in some CoP during some week, you can see from the Wiki what decisions were made, how things agreed in the previous meeting have progressed. I think it has succeeded well, that the information is very transparent, what happens in the CoPs, that they are not closed communities, cliques. Even though I wouldn't visit the Developers CoP at all, so if I follow the wiki page I'm pretty much on the map on what they have talked about and decided there.

– Developer, Finland 2013

4.2.7. Suitable rhythm

Finding a suitable rhythm for the CoP meetings was considered important. The suitable rhythm is related to the number of topics on the agenda. The interviewed CoP leaders saw as their responsibility to take care of that there are a suitable number of interesting topics on the agenda for every CoP meeting. If that was not the case, they would rather cancel a meeting than frustrate the participants with a poor agenda. They also found it important to adjust the meeting rhythm accordingly.

For example, the Developers' CoP was organized bi-weekly. The leader of this CoP explained that his idea was that this CoP should not meet too often, as there are many CoPs and other activities that take time from the participants. He even commented that if the current bi-weekly rhythm turns out to be too fast, he is ready to adjust the rhythm to every three weeks or even to every four weeks, or cancel some meetings if there is no proper agenda. He felt that it is important to have a good agenda and a proper number of interested people, instead of organizing meetings where people could get bored. Therefore he also emphasized that even though anybody can suggest an item to the agenda, it is important to time-box the items, to prevent discussions to continue for too long.

Most of the CoPs in this organization were organized either bi-weekly or weekly. Only a few of the CoPs took place on a need basis.

4.2.8. Cross-site participation when needed

Some of the CoPs were organized between the sites and some only inside a single site. The Functional Verification CoPs and the Coaching CoPs had both site-specific and cross-site instances. Based on the interviews, it seemed that both cross-site and site-specific CoPs were needed, even though most of the interviewees

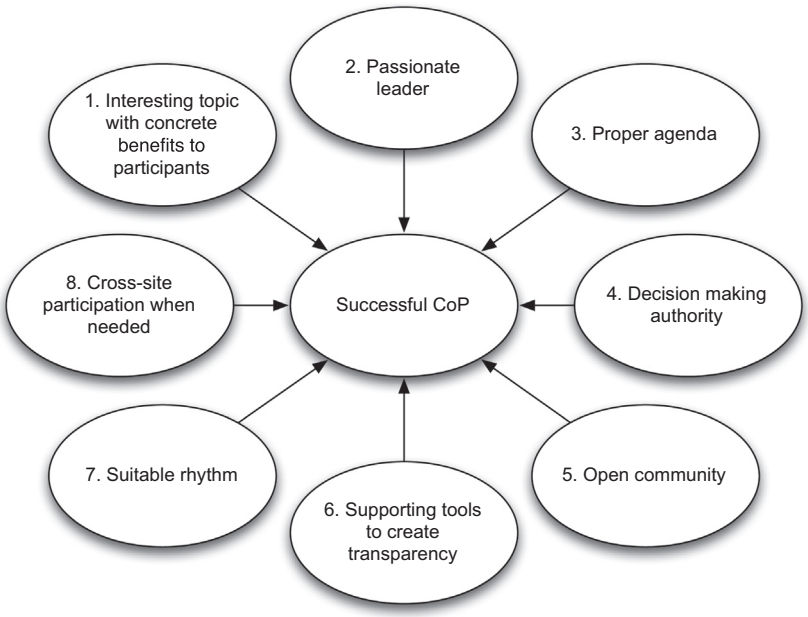

Fig. 2. Characteristics of successful CoPs.

supported the idea of having mainly cross-site CoPs, if possible. Furthermore, they would have liked to have the other sites more involved in the CoPs than currently. The reason for this was that the product was common and the issues discussed in CoPs around that same product should be effectively shared between the sites. The cross-site CoP meetings were supported by good quality video-conferencing spaces.

The biggest challenge the organization had for organizing cross-site CoP meetings was the eight hour time-difference between the newly joined US site and the two other sites. This made involving that site difficult, as it would require adjustment to the schedules of participants at all sites, which was not seen as a possible solution in the long run. Thus, the US site was involved in only a few CoPs.

The second reason for not organizing only cross-site CoPs were site-specific issues. In particular, all sites had other products in addition to the common one. Thus, some of the interviewees felt that sharing information between the products inside one site and discussing site-specific issues might be beneficial, as well. For example the Coaching CoP had started to organize site-specific meetings at both in Finland and in Hungary, in addition to the cross-site meetings. A few organizational coaches were shared between the products in Finland, and saw it as an opportunity to share knowledge between different products.

The participation in the CoP meetings both from the Finnish and the Hungarian sites were considered similar, i.e., the number of participants from each site to the cross-site CoP meetings was relative to the number of teams in both countries. A few interviewees saw a slight difference in the eagerness to participate in the discussions, as most Hungarians had less experience with the product and software development in general than the Finns. Moreover, some interviewees felt that the Hungarian national culture might somewhat limit the discussion, e.g., on the problems if managers were present in a CoP meeting, like in the End-to-end CoP.

> *I have a feeling that they (Hunagarians) are not active as they do not have a lot to give. They have only a few years of experience whereas here people are much more experienced in general. But they are listening, so that way they will certainly learn.*
>
> – Product Owner, Finland 2013

4.3. The evolution of the role of CoPs

In this section, we discuss our findings related to *RQ3: How did the role of communities of practice in the case organization evolve over time?* We describe how the role of communities of practice evolved by first growing into a support mechanism for the agile transformation, then supporting scaling, and finally establishing its place as the main forum for continuos improvement. Our findings on the role of CoPs in different transformation phases are summarized in Table 5.

4.3.1. Transformation support

CoPs were initiated in the case organization as part of the full-scale agile roll-out during 2010, when the whole development organization at both sites was restructured into cross-functional teams. As CoPs were suggested both by the employed consultants and the literature on scaling agile [7,10], building them was seen as a natural part of the transformation.

However, the first steps of this effort were not easy. The organization tried to institute CoPs by managerial decision, an approach that did not turn out particularly well, as illustrated in the following quote:

> *We tried to establish the CoPs by coaches and managers, and we thought that this CoP is needed and then we tried to get people to join in. But it is a voluntary meeting. People need consider the discussions to be interesting or important—then they would show up. We found this empty spot for a few months, before we understood that CoPs emerge when they are needed. And I think we are still experimenting with that.*
>
> – Manager, Finland 2013

In the beginning, finding people to build or even participate in the CoPs was difficult, as the understanding of what the CoPs really are, and the CoP culture were not yet established in the organization.

> *. . .it was the mindset, the thinking that "I" could participate in any community and contribute. That was not familiar, it was not seen as an opportunity. It was not seen what "I" could do there. It was not concrete enough.*
>
> – Manager, Finland 2013

Table 5
Role of CoPs in different transformation phases.

Phase	Transformation	Scaling	Continuous improvement
Goal of the phase	Implementing basic Scrum, broadening the knowledge in the cross-functional teams	Getting scaling, especially inter-team coordination, to work	Applying lean thinking: optimizing the whole, end-to-end flow, continuous improvement
Why CoPs?	Literature suggests CoPs	CoPs gradually replaced the not so successful Scrum-of-Scrums as an inter-team coordination mechanism	Lean suggests continuos improvements and CoPs seem natural fora
Level of CoP knowledge	Learning and practicing what CoPs are and how do they work	Experimenting to use CoPs for new purposes	CoPs established their place as a central fora for a wide range of purposes
What kind of CoPs?	Role-based CoPs and CoPs to replace functions that did not exist anymore in the new organization structure	CoPs to support coordination and design between teams developing a common feature	CoPs aiming to improve the way of working and optimize the end-to-end flow
Who were the drivers setting up the CoPs?	Managers and coaches	Product owners, Scrum masters, coaches	Coaches at different levels
Examples of CoPs	Scrum master CoP, framework CoP, functional verification CoP	Feature design CoPs, feature coordination CoPs	End-to-end CoP, coaching CoP, developer's CoP

As instructed by the consultant literature that the case organization was using [10,7], the first CoPs were build around specific roles distributed in the cross-functional teams, such as Scrum Masters, testers and developers. The cross-functional Scrum teams of initially 7–8 persons were formed with the idea that all teams would have "all needed knowledge" to build end-to-end user stories that would provide value to end customers.

However, in practice this was not possible. The ten-year old product was so large, complex and built using a large number of components employing different technologies that having single teams being able to develop any feature would be practically impossible. Moreover, having experts on all the components or even all the previous phases of the development (e.g. system design and different levels of testing) in every team was not possible to accomplish in practice, as some areas had only a couple of experts.

During our first two interview rounds it became clear that one of the biggest challenges was that the new cross-functional teams needed to broaden their knowledge to be fully functional, while at the same time keep up with the new development. The communities of practice provided some solutions to these problems the organization experienced during the transformation.

First, the communities of practice offered a forum, where people could discuss how to apply agile development in practice and together solve challenges related to the way of working. The Scrum Master CoP was a good example of this kind of community, discussing and solving day-to-day challenges on applying agile, e.g. sharing good practices on how to arrange retrospective meetings in teams.

Second, communities of practice were forums in which cross-functional team members could share and broaden their knowledge on specific areas together, and solve problems and make decisions related to that area. Most of the experts were now divided into different cross-functional teams, thus the communities of practices were natural forums for them to continue working together and make decisions. For example, the Functional Verification CoP was this kind of forum.

> *The functional testers, when they were split into teams, they were left with this CoP, and it in a way works very naturally, because before they belonged to the same group and worked together, so now it is very natural and well-functioning. They share testing issues and make decisions related to testing.*
>
> – Scrum Master, Finland, 2011

Moreover, as all the teams could not have all kinds of experts, teams were lacking knowledge. For example, the cross-functional teams were supposed to do system-level design by themselves, despite the fact that many teams did not have the required knowledge, as there were not enough system designers to assign to all teams. The System CoPs were instituted to mitigate these problems, and became forums where the few experts and the team members who wanted and needed to learn more could do design work together, make design decisions and learn from each other.

> *System knowledge is hard to find, ... so there was a community of practice established for this system work and now we are trying to empower people in that group so there are representatives from each team...*
>
> – Developer, Hungary 2011

> *...I found it a good idea to have separate ... to group those people who are interested in the same topic. Because we have cross-functional team ... It's really good to have these CoPs, because those guys who, for example, who are good at systemizing, they can have discussion together.*
>
> – Manager, Hungary 2011

Third, during the transformation it was noticed that some function or organization that had taken care of specific responsibilities in the old organization did not exist anymore. Thus, some of the CoPs were instituted to handle the work of old functional teams in the old plan-driven world. One example was the Framework CoP, which was started in the beginning to take care of the work of the old "Process, Methods and Tools" functional organization. The participants of this CoP were the people who had been responsible for process development in the plan-driven world. However, as the participants were felt to be too disconnected from the actual work, team members did not join the CoP. Instead, it became a meeting for the old process developers only, and did not gain the momentum needed to survive.

> *Some of the CoPs were born as we have previously had some function or organization that has taken care of some things. When that didn't exist anymore, so we created a CoP that takes care of those things. ... The Framework CoP is a good example of that kind of CoP. However, it turned out that when it is not clearly part of the flow of doing things, so ... it is difficult to accomplish real results. ... It dried up, it didn't fly. And it was a good experience to see that too. That this doesn't work.*
>
> – Manager, Finland 2013

The first CoPs that were created during the transformation were role-based, like Scrum Master CoP, Functional Verification CoP, Developers' CoP and System CoPs. The main purpose of these first CoPs was learning and knowledge sharing, both regarding the agile practices and broadening the knowledge of the cross-functional teams. This kind of learning and knowledge sharing CoPs are

exactly what the agile literature suggests. At this phase, it was mainly the managers and later on the coaches that were the main drivers for setting up the CoPs. As a summary we can see that these first CoPs initiated during the early phases of transformation were important support mechanisms for the transformation.

4.3.2. Scaling support

After the first CoPs supported the transformation by concentrating on making the new cross-functional teams fully functional, the next challenge was to get the scaling of agile, especially the inter-team coordination to work.

As discussed previously, the main inter-team coordination mechanism offered by Scrum, the Scrum-of-Scrum meetings, were tried, but the organization wide Scrum-of-Scrum meetings never delivered the expected value, and thus slowly died. This organization of over 25 teams was just too large for teams to have enough common ground for one common coordination meeting to be useful.

The next attempt, Feature Scrum-of-Scrum meetings, weekly coordination meetings between the teams contributing to a common feature, were more successful. The scope of these meetings was limited and the issues discussed were interesting and important to all participants. As the term "SoS" had bad connotations after the failure of the organization-wide SoS, the Feature SoS meetings were renamed to Feature CoPs. And that is what they were: communities of practice concentrating on coordination between the teams working for a common feature, thus we also refer to them as Feature Coordination CoPs.

Feature Design CoPs were partially born out of the early System CoPs that aimed to make design decisions together, and partly out of Feature CoP meetings where participants talked also design and architectural issues as part of the inter-team coordination. However, as it was noticed that persons interested in design and architectural issues were at least partially different than those participating in Feature Coordination CoP meetings it was natural to separate these meetings. In most areas Feature Coordination CoP meetings took place weekly, while Feature Design CoPs were arranged on a need basis.

In this organization, the main challenge of scaling agile from one team to several teams, the inter-team coordination, was alleviated by using CoPs. The Feature Coordination and Feature Design CoPs took successfully care of most of the inter-team coordination tasks. This, finally well-functioning support structure, was not suggested by the literature or consultants, but was gradually born out of the need to solve a problem. As could be seen, it took several attempts and trials before the organization ended up to the current structure. Thus, the drivers setting up these CoPs were no-longer managers, but those who saw the need: Product Owners, Scrum Masters, Coaches and team members.

At this stage of the transformation, the idea of CoPs was not new to the organization, but a well-functioning mechanism, thus it was a natural to try it for new purposes.

4.3.3. Continuous improvement support

When the main challenges of the transformation and scaling agile were solved, the organization continued their lean and agile journey by focusing on the lean side of the transformation. They started to look at the end-to-end flow, from requirements until they were implemented as part of the product, with the goal of optimizing the whole.

Lean thinking emphasizes continuos improvements, and CoPs seemed to be natural fora for that. Most CoPs had actually already from the beginning contributed to continuous improvements to the way-of-working. However, we noticed a clear change: when the most burning transformation related problems that affected the day-to-day activities were solved, many CoPs started changing their goals towards organizational development, looking at the organization as a whole.

The End-to-end CoP was the most self-evident example as it concentrated on removing the bottlenecks from the flow. Another clear example was the Coaching CoP that had moved from sharing knowledge of the implementation of Scrum practices in teams to advancing organization wide issues and trying to involve the whole organization into this development by creating the "Cool Wall" and the coaching backlog. Also other role-based CoPs like the Developer's CoP aimed at organization-wide improvements. These improvement initiatives were more or less led by the coaches at different level: the head coach was leading the End-to-end CoP, the organization level coaches were driving the Coaching CoP activities, and a team coach had initiated the reborn Developers' CoP.

At this phase of the transformation journey the CoPs had clearly established their place as a central fora, that could be used for a wide range of purposes: for learning and knowledge sharing, coordination and design activities, as well as organizational improvement.

4.4. Organizational support for communities of practice

In this section, we discuss our findings related to *RQ4: How did the case organization support communities of practice?* We present the two crucial elements of organizational support that we think enabled the success of the CoPs at Ericsson: a supportive atmosphere for CoPs, and infrastructure support.

4.4.1. Supportive atmosphere for CoPs

Based on our results the main success factor for the CoPs has been the supportive atmosphere for building, using and participating in the CoPs. Next, we will describe the three elements contributing towards building supportive atmosphere for CoPs: *openness of participation, participation valued*, and *support from managers and coaches for building CoPs.*

4.4.1.1. Openness of participation. As a general rule, the CoPs are open to anyone who wants to participate in them. This mindset can be seen in, e.g., that CoP calendar invitations are send to everybody. Of course, anyone can decline an invitation and not receive invitations to that same CoP anymore. Our interviewees emphasized this openness of the participation in the CoPs. They said that it is important that CoPs do not become meetings for only a select group of experts, or "gurus", because then cliques might start to form. Instead, the current CoP culture emphasizes that anybody can participate in any CoP, and that people can join in CoP meetings just to learn and to know what is going on, i.e., you do not have to be an expert on the topic to join in. It is also acceptable to participate in CoP meetings irregularly, e.g., when the agenda contains a topic you are interested in, or you just happen to have time. Several interviewees mentioned that they participate irregularly in some CoPs, depending on the topic and their other work engagements.

4.4.1.2. Participation valued. It is fully acceptable for a person to spend as much time in CoPs and participate in as many CoPs he or she deems necessary. Instead of being viewed by fellow team members as waste, CoP participation is valued by the colleagues. Nobody forces or asks anybody to participate in any CoPs either, the participation is on voluntary bases. The Feature CoP meetings are the only ones to which each team working on that feature sends their representative or representatives. Each team chooses their representative to these meetings and our interviewees mentioned that it is not a problem to find a volunteer or volunteers for that either.

The idea there is that they (CoPs) are on voluntary bases, that people who have passion to influence on things, to discuss with each other, and to make decisions they go to the communities of practice.

– Coach, Finland 2013

In principle, we have so many CoPs that you can just spend your time only in the CoPs, so everybody participates in what he feels as the most interesting and important for himself.

– Developer, Finland 2013

When we asked our spring 2013 interviewees how many CoPs they normally participated in, most reported that they participated in 2–3 CoPs per week. However, as our interviewees were active members of the community and even the developers we interviewed were part-time coaches, they most probably participated in more CoPs than regular team members. One of our interviewees mentioned that approximately one third of the people in the organization are active in that sense, and want to make difference, change things, and thus participate actively in CoPs as well.

I would say that it is around 30 percent of the whole organization who thinks this (the lean and agile transformation) is "the" important thing for them. And they participate in this kind of meetings (CoPs) a lot.

– Manager, Finland 2013

4.4.1.3. Support from managers and coaches for building CoPs. Even though we could see from the initial steps that managers should not force CoPs to be build, but CoPs should just emerge when there is a need, our interviewees emphasized that especially in the beginning a lot of support and encouragement is needed. When people do not clearly understand what a CoP is and what is it for, the CoPs do not just emerge. During our interviews we could hear that building and leading a new CoP was clearly highly valued, those individuals who took responsibility of building a new CoP were appreciated both by the managers and peers. Moreover, we noticed that many of the CoPs were led and supported by coaches, e.g. by a team coach, head coach, or community coach. Thus, the coaching community had clearly taken CoPs as part of their responsibility.

4.4.2. Infrastructure support

The organization provided several types of infrastructural support for both collocated and distributed CoPs, for example: videoconference, wiki, and chat. As many CoPs were distributed between sites, our interviewees saw the well-functioning videoconference system as the most important infrastructure need for the CoPs. This need had been well taken care of in the case organization. The organization had a large number of meeting rooms at all sites that had been equipped with high quality videoconferencing equipment. These rooms had microphones hanging from the roof, making it irrelevant where in the room the speaker sits, as the audio quality is excellent regardless of physical location in the room. The videoconferencing rooms have enabled effective distributed CoP meetings, and are in very frequent use.

These videoconferences are the best thing ever! ... Earlier we had normal phone conferences with poor quality cracking voice. ... Now when we have this surround environment and sharp voices, and you actually see the voices and can even follow the movements of the mouth that what a person is saying on the other end. This has been a really big improvement, this videoconference.

– Developer, site α 2013

Besides videoconferencing, the CoPs actively use the organization-wide wiki. Most of the CoPs have their own wiki pages with meeting agendas and minutes, as well as other information. Some of the CoPs even have their own chat forums and mailing lists that anybody could join.

4.5. CoP purpose classification

In this section we answer the *RQ5: How could the different purposes of communities of practice be classified?*

Based on CoP purpose we did a preliminary classification of CoPs in the case organization. The purpose of this classification is to clarify the understanding on the purposes of CoPs in lean and agile software development.

We found the following four different purposes for CoPs: (1) knowledge sharing and learning, (2) coordination, (3) technical work, and (4) organizational development, as summarized in Table 6. Next, we will briefly explain each of these purposes and give examples from the case organization. Even though these four purposes are clearly different, we can see that many of the CoPs had characteristics of a few of them.

4.5.1. Knowledge sharing and learning

For many of the CoPs, our interviewees state that knowledge sharing and learning was the basic purpose. This is also the most typical rationale for creating CoPs according to the literature [13].

In our case organization the best example of this type of a CoP was the Coaching CoP, as the main purpose of it was especially in the early phases of the transformation exactly that: to share experiences on how to implement agile from different teams and try to find solutions to problems together. The coaching community even arranged regularly another meeting called "Coaching Guild" to study new material together.

Other examples of knowledge sharing and learning are the other role-based CoPs that were build as forums for specific roles in different cross-functional teams to share knowledge, e.g., Developers' CoP and Functional Verification CoP.

Actually, we could say that all the CoPs have at least to some degree knowledge sharing and learning as one of their purposes.

4.5.2. Coordination

Our respondents deemed inter-team coordination to be the main purpose of certain CoPs, called Feature Coordination CoPs. In the Feature Coordination CoPs several cross-functional teams working inside one product area, or a feature meet and coordinate their work. The Feature Coordination CoPs were formed when the inter-team coordination mechanism offered by the Scrum method, the organization-wide Scrum-of-Scrum meetings, were tried, but did not bring the expected benefits.

The first step was to start Feature specific Scrum-of-Scrum meetings where each team developing that feature was represented, and this representative answered the three Scrum questions regarding their team. The Feature Coordination CoPs were based on the same idea, but over the time, the format offered by Scrum, started to change from a reporting meeting towards a discussion meeting, where coordination issues were discussed, not only raised.

4.5.3. Technical work

Another important purpose of CoPs according to our respondents is to work, design and make demanding design decisions together. The earliest CoPs of this type in our case organization were the System CoPs that aimed to gathering together a group

Table 6
CoP purposes in the case organization.

Purpose	Example
Knowledge sharing and learning	Role-based CoPs
Coordination	Feature coordination CoP
Technical work	System CoPs
Organizational development	End-to-end CoP

of individuals who could do the system design together, as all teams did not have enough system knowledge. Later on, the Feature Design CoPs had the same idea, but their main aim was to make high level design and architectural decisions together with all the teams involved in that specific product area. Sometimes even experts from a related area were invited.

4.5.4. Organizational development

Organizational development, i.e., developing the way of working in the organization is another purpose of CoPs.

The End-to-End CoP was a perfect example of this kind of a CoP in our case organization. It aimed at look at the product development flow through the whole organization, and to optimize the flow by together removing the bottlenecks. For this kind of a CoP it was important that all organizational levels and parts were represented in the CoP, so that sub-optimizations could be avoided and that the CoP could make or at least take care of that all the needed decisions are made.

Organization development was one of the main purposes of also the Coaching CoP and the Developers' CoP. Developers' CoP aimed to improve the level of Software Craftmanship and make the needed decisions to align ways of working at the software development level.

5. Discussion

5.1. Summary of the findings

In the case organization, CoPs evolved into a valuable mechanisms for knowledge sharing, inter-team coordination and communication, technical work and organizational development. The various CoPs became a central mechanism behind the success of the large-scale agile implementation in the case organization and helped to mitigate some of the most pressing problems of the agile transformation. Next, we summarize and discuss our findings related to the research questions.

5.1.1. RQ1: What kinds of communities of practice were created in the case organization and how did they evolve over time?

The case organization had tens of different CoPs, and had created a culture in which CoPs were formed as needed, and ceased to work when they either were dysfunctional, or had fulfilled their purpose. We chose four CoPs for deeper study, based upon the fact that the authors and interviewees jointly deemed them both interesting and successful: Coaching CoP, Feature CoPs, Developers' CoP, and End-to-End CoP. We described how each of these CoPs were created and evolved over time.

These CoPs, their evolution and purposes differ. In particular not all CoPs were only about learning and knowledge sharing. CoPs had three main roles: to support the agile transformation, to be part of the large-scale Scrum implementation, and to support continuous improvement.

The Coaching CoP was an example of a role-based CoP, the original main purpose of which was knowledge sharing and learning. This represents the perhaps most typical uses of CoPs. However, during the timespan of the study, we could see a slight change of the purpose of this CoP–at the time of our last interviews it was taking clear steps towards contributing to organizational development.

The Feature CoPs were created to provide a solution to the problem of dysfunctional organization-wide Scrum-of-Scrums. The main purpose of the Feature Coordination CoPs was coordination between the teams working in the same product area, whereas the Feature Design CoPs focussed on technical product design.

The Developer CoP was an organization-wide role-based CoP that initially run into problems due to the lack of a CoP culture. When later restarted, it had a passionate leader, focussed on software craftmanship and tools and was considered successful.

The End-to-End CoP gathered together people from a wide variety of roles to work on improving the overall product development flow across team and product area boundaries. The main purpose of this CoP was thus organizational development. The fact that the CoP had participation from both managers and practitioners and had decision making authority facilitated fast decision making and problem solving.

5.1.2. RQ2: What were the characteristics of successful communities of practice?

We identified eight characteristics of successful CoPs. First, a CoP needs an interesting topic, which is important to a group of people big enough, and the participants of the CoP need to get some concrete benefits for their daily work from every single CoP meeting. Second, every CoP needs a passionate leader or facilitator that takes care that CoP meetings have a proper agenda and contents and that the discussions are not endless. Third, a good agenda that is distributed before the meeting and that everybody may contribute to is an important factor for people in deciding whether to participate or not. Fourth, a CoP need to have decision making authority on the matters it handles. Fifth, each CoP needs a community around it, which is open and transparent. Sixth, CoPs need some tools to build organizational memory and transparency, e.g. wiki. Seventh, CoPs should have suitable meeting rhythm. It might be better to cancel a meeting rather than hold one without proper discussion topics. Eighth, a distributed organization should enable cross-site participation to CoPs to share learning and align the whole organization. For site specific topics, local CoPs might be considered.

These success criteria match well with the literature (cf. [13]), with the exception of decision making authority, which the literature did not mention. Decision making authority and the sense that the people participating in the CoPs were trusted to make binding decisions was considered very important for the success of the CoPs. Comparing the success factors to Table 2, Ericsson seemed to follow all of Wegner's principles for cultivating communities of practice.

5.1.3. RQ3: How did the role of communities of practice in the case organization evolve over time?

The needs with respect to CoPs might change over time. In the beginning of the agile transformation, CoPs had an important role in supporting the transformation, especially by broadening the knowledge in the cross-functional teams and sharing tips on how to implement Scrum.

After basic Scrum was working in the cross-functional teams, the next challenge was to support scaling, especially inter-team coordination. During this phase, the CoPs gradually replaced the not so successful Scrum-of-Scrums as an inter-team coordination mechanism.

Next, the organization concentrated on continuos improvement, i.e., developing and fine-tuning the way of working. The CoP turned out to be excellent forums for initiating and agreeing on improvements as persons eager "to make a difference" from different parts of the organization participated these CoPs.

While the literature on CoPs (e.g. [13]), acknowledges that both CoPs and their role evolve over time, we think that this particular instance is interesting in how it illustrates the roles of CoPs in large-scale agile transformation: from dealing with basic operational level issues to organizational development.

5.1.4. RQ4: How did the case organization support the communities of practice?

According to our results the main success factor for the CoPs was the supportive atmosphere for building, using and participating in the CoPs. We identified three elements contributing towards building supportive atmosphere for CoPs: openness of participation, participation valued by the organization, and managers and coaches support in building. In addition, infrastructure support, including e.g., wikis and videoconference facilities for distributed CoPs, is needed.

Developing an atmosphere that provides a fertile ground for CoPs does, however, require strong management support, as well as both time and patience. Initially, people in the organization did not understand why they should partake in the CoP meetings, and what the meetings were supposed to accomplish. The first CoPs were formed by management and the coaches, but slowly the organization as a whole started to understand the idea behind CoPs, and how to create and utilize them in the best way. Management learned that CoPs should not be created by management decision, but by creating a supportive atmosphere, as well as supporting infrastructure. The organization created a culture in which CoPs and participation in them was appreciated, by having open CoPs with invites sent to all, and by creating transparency into the CoPs.

The organization learned that CoPs cannot be kept alive artificially. To successfully form a CoP, a community needs to be built—just communicating to the whole organization that there is a new CoP does not work. On the other hand, the term CoP was also used for single or a series of ad-hoc meetings that required a group of people to come together to solve a common problem. After the problem was solved, the ad-hoc CoP was dissolved.

Our findings here match well with Wenger's [13] observations, as he emphasizes shepherding rather than creating CoPs.

5.1.5. RQ5: How could the different purposes of communities of practice be classified?

We presented a classification of CoPs based on their purpose. In our case organization, we found four main purposes for CoPs: (1) knowledge sharing and learning, (2) coordination, (3) technical work, and (4) organizational development. Even though these four purposes are clearly different, we can see that many of the CoPs had characteristics of a few of them.

Knowledge Sharing and Learning, the basic idea behind CoPs, was a goal, at least implicitly, in all CoPs we studied. Several CoPs were involved in organizational development, as well. The technical work, on the other hand, was a very specific purpose: designing together and making demanding product design decisions together across team borders. The coordinating work between the teams was also a quite a specific purpose, that some CoPs had as their main purpose, e.g. Feature Coordination CoP, but others as one of their minor purposes, e.g. Developer's CoP.

We think that this classification can help us understand the roles CoPs can have in large-scale agile software development, in various stages of the transformation. While Wenger [13] recognizes that CoPs can have a wide range of roles, the breadth of roles CoPs had at Ericsson – from coordination to creative technical work – was quite interesting.

5.2. Practical implications

In this section we briefly describe the main practical implications of our research, divided into two categories: organizational and management implications, and implications for practitioners.

At the organizational and management level, we have three main implications:

1. *CoPs can support a Lean and Agile transformation*. This case study shows that CoPs can be used effectively as a support mechanism during a large-scale Lean and Agile transformation, as they provide a forum to discuss the transformation, plan continuous improvements to the way-of-working, and share knowledge and tips regarding working practices between the roles and teams. Our study suggests that organizations wanting to do an organizational transformation can use CoPs as an effective mechanism to support the change.
2. *CoPs can support scaling agile to a large and distributed organization*. Basic Scrum [24] does not offer support for large-scal cross-team coordination. This case showed that CoPs can help provide what is missing from Scrum, i.e., efficient coordination and knowledge sharing mechanisms between the teams, as well as between the experts in the teams. Furthermore, our study shows that large and complex product might be supported by product area or feature based CoPs rather than basic Scrum-of-Scrums.
3. *Building a CoP-friendly corporate culture is important*. In order for CoPs to work in an organization, suitable organizational support is required. To support the creation and evolution of CoPs as part of the way of working, you should build a supportive atmosphere for CoPs, give them the necessary empowerment, as well as offer infrastructure support. Without this, CoPs might not emerge and grow well.

For practitioners working in the organization, we derive at least the following practical implications:

1. *Participate in CoPs and create a new one when needed*. If your organization supports CoPs and you see a problem or opportunity that needs to be addressed outside of your own team context, consider taking it up in an existing CoP or form a new one to deal with the issue, as well as similar ones.
2. *Use CoPs to learn and further your career*. Utilize CoPs to keep up-to-date about what happens in the organization at large, and to deepen and broaden your own product, technical and process knowledge.
3. *Influence the organization via CoPs*. CoPs are empowered to make decisions in their area of concern. By actively participating you can improve and influence even organization-wide issues.

5.3. Generalizability and threats to validity

This paper presented a single case study on how a large, globally distributed software development organization built communities of practice as part of their lean and agile transformation. The generalizability of the results to other contexts is likely limited.

We rely on the definitions of validity and reliability proposed by Yin [22]. Internal validity is not relevant, as this research is a exploratory case study [22].

The main threat to the construct validity of this research was the accuracy of the descriptions of the studied phenomena. During the spring 2013 interviews we concentrated on CoP, however, then the number of people interviewed was only eight. These interviews were extremely fruitful, but having a larger number of interviews, and having interviewees from a bigger number of CoPs could have given us a broader view, as the organization had over twenty CoPs. There is a possibility of positive selection bias with respect to the CoPs that might skew the results, since the CoPs to study was selected jointly with management at the case organization. Moreover, most of the spring 2013 interviews were conducted in Finland, only one in Hungary, and we did not have access to the US site. Having a better representation of these two sites would

have given a better understanding of the current situation regarding CoPs in the organization.

To increase the construct validity we took several actions: First, we used multiple sources of evidence: we interviewed multiple persons from different roles, as well as observed actual CoP meetings. Second, the interviews were analyzed together by both authors. Third, all the informants of the two first interview rounds were invited to a feedback session, where the results were presented, and the first draft of this paper was reviewed by the key informants.

The external validity of a case study concerns the domain to which the results can be generalized [22]. Based on our study, we can hypothesize about the significant characteristics of the domain. First, the system under development was large, multifaceted and technically demanding. Second, the case organization was transforming from a waterfall type process to agile development. Third, the number of development teams was relatively large. Fourth, the development was distributed to three geographical sites. The results are likely generalizable to single site development, but it is difficult to hypothesize how generalizable the results are when the development is distributed to a larger number of sites.

The main threat to the reliability of this research is the variability in the data collection. The data collection was conducted using the general interview guide approach [29], which introduced variability to the topics discussed in the interviews. However, the large number of interviewees and multiple interviewers allowed data source and investigator triangulation [29] which increased the reliability of the results.

5.4. Future research

In this case study, we got a preliminary understanding both of the role CoPs can play in agile adoption and implementation in a large organization, as well as related challenges and success factors. However, additional data is needed both to confirm these results, as well as to better understand the contextual factors, such as role of the existing organizational culture and structure, the role of national culture, products, etc.

In the immediate future we will continue to longitudinally study the CoPs at Ericsson, both related to this particular product, as well as in other products. In addition, we plan to study another large organization in the same industry sector, using interviews and observations as our main method. In order to get a better understanding of the contextual factors affecting the use and value of CoPs in large-scale agile development, organizations in other sectors and countries should also be studied, using e.g. survey instruments or case studies by other researchers.

Acknowledgements

We would like to thank Oy LM Ericsson Ab for making this study possible and all the anonymous interviewees and persons who gave us feedback for providing valuable contributions to this research. This work was supported by Tekes as part of the Cloud Software Finland program of DIGILE (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT and digital business).

Appendix A. Interview questions

A.1. Transformation questions

1. Background
   - What is your role, background and experience at Ericsson?
   - What kind of training have you had on Lean and Agile software development? Would you have needed more training?
2. Agile and lean transformation
   - Why was the transformation started?
   - Explain the timeline of the transformation.
   - Why did you choose Scrum? Why Lean?
   - What comes from Scrum in your way-of-working? What comes from Lean in your way-of-working?
   - What is still left from the waterfall model?
   - What kind of training have you arranged? By whom? To whom?
   - What are you still planning to change/improve?
   - What is your general feeling of the transformation? Good/bad/challenges? How have you solved the challenges?
   - What would you have done differently? How?
   - Has Lean and Agile solved the challenges you wanted to solve by this change?
   - How have you measured the change? What have you exactly measured? How does this change look according to your measurements?
3. Organization structure
   - How does your organization look like? (draw a picture!)
   - How has the agile transformation affected the roles and responsibilities?
   - What roles do you currently have in your organization? Are the following roles and their responsibilities clear: Project Manager vs. Scrum Master vs. Product Owner vs. Proxy Product Owner vs. Program Manager? Are the roles clear in your organization? Are there challenges/improvement needs regarding these roles or their application?
   - Comment on the organization structure: Good?/improvement needs? Do you still plan to change something regarding the organization structure?
   - How many persons do you have at the moment at each site? How many teams at each site? What are the team sizes at the different sites?
   - How are responsibilities divided between the different sites? Do you have different kind of tasks/roles at different sites? When did Hungary join in? Why?
   - How do you take care of the collaboration, communication and division of tasks between the sites? How does it work? Good?/improvement needs?
   - Do different teams have different roles?
   - Feature teams: What kind of knowledge/"roles" do you have in each team? Good?/challenges? What have you done to solve the challenges?
4. Coaching
   - External coaching: What kind of coaching/consulting have you had to support the transformation?
   - What kind of training have you arranged?
   - Have you had internal coaches? If yes, tell more about their role and tasks.
5. Team-level Scrum practices
   - Your own team: team size, what does your team do, connections to other teams?
   - Scrum practices of your team (explain all Scrum practices your team uses)
   - What is the length of your iterations? What is your opinion on the iteration length? Explain.
   - Cross-functional teams: How has the idea of the cross-functional teams worked out? Why?/why not?
   - Tell about the communication inside your team. Good/bad/improvement needs?

- How and when do you communicate with other teams?
- Do you know enough on what is happening in the other teams/elsewhere in the project? Is there something that you would need to know more? What? Why?
- How does the global distribution affect your daily work?
- What is working well in you team/regarding your team practices? What are the biggest problems? What should be improved? How?

6. Improvement suggestions
   - Is there something that should be improved in this project?
   - How?
7. Tools:
   - What are the most important tools that you use?
   - Tell a bit about each tool (for what is it used, who are using it, etc.) Good?/Bad?
8. Scaling agile/cross-team coordination practices
   - What are your current scaling/cross-team coordination practices?
   - Scrum-of-Scrums? CoPs? Feature Owners? Tell about each practice.
   - How have you taken into account the global distribution in cross-team coordination? How do you handle communication? Tools used to support the communication and coordination? What kind of challenges have you experienced?
9. Measuring
   - What do you measure?
   - How do you measure quality? Productivity? How has this changed compared to the way of working before the transformation?
   - What should be measured? Why?
10. Product management
    - How is product management taken care of?
    - Where do you receive the requirements? Describe the whole flow from the customer request to the requirement being part of the product.
    - How is customer communication taken care of?
    - What does the product owner do? What does the proxy product owner do?
    - Communication between the product owners/proxy product owners? Communication with the teams?
    - How is work divided between teams? Who does that?
    - What are the current challenges of product management?
11. Opinions
    - How do you think that the Lean and Agile transformation has succeeded?
    - How do you think that Lean and Agile software development fits in your organization? What are the benefits is brings? What are the challenges? Has your opinion towards the Lean and Agile changed somehow during the transformation?
    - What advice would you give others considering the application of Lean and Agile to a similar situation?
    - What are your expectations towards our research?

A.2. Communities of practice questions

1. Background
   - What is your role, background and experience at Ericsson?
   - Of which CoPs do you have experience?
2. Scrum-of-Scrums
   - Project wide SOS – Have you participated? How did it work? Good/challenges?
   - Feature SOS – Have you participated? If yes, to which feature SoS? How did to work? Good/challenges?
   - Do you still have SoS? If, yes, what kind?
3. (Only to managers) CoP culture
   - How did you start building CoPs?
   - How have you supported building CoP?
   - How have you built "the CoP culture"?
   - What has been challenging in building CoPs/CoP culture?
   - What are the most successful CoPs? Why?
4. CoPs:
   - Which CoPs do you currently have? Which CoPs have you had previously? Which CoPs have disappeared? Why?
   - When did you start building CoPs? Why did you start building CoPs?
   - What is the difference between SoS and CoPs?
   - How is a CoP born? How does a CoP die/is terminated?
   - What kind of a CoP is successful? Why? What makes a CoP well-functioning?
   - What kind of CoPs have not been not successful/well-functioning? Why?
5. Your own CoP participation
   - Which CoPs do you participate in? Or have participated?
   - Tell about each CoP you have participated in: the purpose of that CoP, what happens in that CoP, who are the participants, what is your general feeling of this CoP, positive/negative experiences
   - How does the global distribution affect on the CoPs? Experiences with Hungary/Finland/US? Your general feelings on the distributed CoPs? Success/challenges? How well different sites participate in distributed CoPs (Hungary/Finland/US)? Arranging CoPs over videoconference, how does that work?
   - Wiki: How is wiki used in the CoPs you have participated in? What kind of information can be found from the CoP wiki pages?
   - CoP invitations?
   - How would you improve the CoPs you participate in? How would you improve the CoP culture/support for building CoPs?

References

[1] K. Schwaber, M. Beedle, Agile Software Development with Scrum, Prentice-Hall, Upper Saddle River, NJ, 2002.
[2] VersionOne, Inc., 6th Annual State of Agile Development Survey, 2011. <http://versionone.com/state_of_agile_development_survey/11/>.
[3] C. Larman, V.R. Basili, Iterative and incremental development: a brief history, Computer (2003) 2–11.
[4] A. Cockburn, Agile Software Development, Addison-Wesley, Boston, MA, 2002.
[5] B.W. Boehm, R. Turner, Balancing Agility and Discipline: A Guide for the Perplexed, Addison-Wesley Professional, 2004.
[6] M. Lindvall, D. Muthig, A. Dagnino, C. Wallin, M. Stupperich, D. Kiefer, J. May, T. Kahkonen, Agile software development in large organizations, Computer 37 (12) (2004) 26–34.
[7] D. Leffingwell, Scaling Software Agility: Best Practices for Large Enterprises, Addison-Wesley Professional, 2007.
[8] E.C. Lee, Forming to performing: transitioning large-scale project into agile, in: Proceedings of the Agile 2008, AGILE '08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 106–111.
[9] M. Paasivaara, S. Durasiewicz, C. Lassenius, Using scrum in distributed agile development: a multiple case study, in: Proceedings – 2009 4th IEEE International Conference on Global Software Engineering, ICGSE 2009, 2009, pp. 195–204.
[10] C. Larman, B. Vodde, Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum, Addison-Wesley Professional, Boston, MA, USA, 2010.
[11] E. Hossain, M.A. Babar, H.-y. Paik, Using scrum in global software development: a systematic literature review, in: Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering, ICGSE '09, IEEE Computer Society, Washington, DC, USA, 2009, pp. 175–184.
[12] E. Wenger, Communities of Practice: Learning, Meaning, and Identity, Cambridge University Press, 1998.

[13] E. Wenger, R. McDermott, W.M. Snyder, Cultivating Communities of Practice, Harvard Business Review Press, Cambridge, MA, 2002.
[14] C. Fry, S. Greene, Large scale agile transformation in an on-demand world, in: Agile Conference (AGILE), 2007, pp. 136–142. http://dx.doi.org/10.1109/AGILE.2007.38.
[15] G. Benefield, Rolling out agile in a large enterprise, in: Proceedings of the 41st Annual Hawaii International Conference on System Sciences, 2008, pp. 461–461. http://dx.doi.org/10.1109/HICSS.2008.382.
[16] E. Wenger, R. McDermott, W.M. Snyder, Communities of practice: the organizational frontier, Harvard Business Rev. (January–February) (2000) 139–145.
[17] E. Lesser, J. Storck, Communities of practice and organizational performance, IBM Syst. J. 40 (4) (2001) 831–841.
[18] A. Mestad, R. Myrdal, T. Dingsoyr, T. Dyba, Building a learning organization: three phases of communities of practice in a software consulting company, in: 40th Annual Hawaii International Conference on System Sciences, 2007 (HICSS 2007), 2007.
[19] T. Kahkonen, Agile methods for large organizations – building communities of practice, in: Agile Development Conference, 2004, pp. 2–10.
[20] P. Gongla, C. Rizzuto, Evolving communities of practice: IBM global services experience, IBM Syst. J. 40 (4) (2001) 842–862.
[21] T. Chau, F. Maurer, Tool support for inter-team learning in agile software organizations, Lecture Notes Comput. Sci. 3096 (2004) 98–109.
[22] R.K. Yin, Case Study Research: Design and Methods, fourth ed., SAGE Publications, Thousand Oaks, CA, USA, 2009.
[23] M. Poppendieck, M. Cusumano, Lean software development: a tutorial, IEEE Softw. 29 (5) (2012) 26–32.
[24] K. Schwaber, Agile Project Management with Scrum, Microsoft Press, Redmond, Washington, USA, 2004.
[25] K. Schwaber, The Enterprise and Scrum, Microsoft Press, Redmond, Washington, USA, 2007.
[26] M. Paasivaara, C. Lassenius, V.T. Heikkila, K. Dikert, C. Engblom, Integrating global sites into the lean and agile transformation at Ericsson, in: 2013 IEEE 8th International Conference on Global Software Engineering (ICGSE), 2013, pp. 134–143.
[27] M. Hallikainen, Experiences on agile seating, facilities and solutions: multisite environment, in: 2011 6th IEEE International Conference on Global Software Engineering (ICGSE), 2011, pp. 119–123. http://dx.doi.org/10.1109/ICGSE.2011.20.
[28] V.T. Heikkilä, M. Paasivaara, C. Lassenius, C. Engblom, Continuous release planning in a large-scale scrum development organization at Ericsson, in: Agile Processes in Software Engineering and Extreme Programming, in: H. Baumeister, B. Weber (Eds.), Lecture Notes in Business Information Processing, vol. 149, Springer, Berlin Heidelberg, 2013, pp. 195–209.
[29] M.Q. Patton, Qualitative Evaluation and Research Methods, second ed., Sage Publications, Newbury Park, Calif., 1990.
[30] M. Miles, A. Huberman, Qualitative Data Analysis: An Expanded Sourcebook, second ed., Sage Publications, Thousand Oaks, CA, 1994.
[31] J. Corbin, A. Strauss, Basics of Qualitative Research, third ed., SAGE Publications, Thousand Oaks, CA, USA, 2008.
[32] T. Jick, Mixing qualitative and quantitative methods: triangulation in action, Admin. Sci. Quart. 24 (4) (1979) 602–611.
[33] C. Ladas, Scrumban: Essays on Kanban Systems for Lean Software Development, Modus Cooperandi Press, Seattle, WA, USA, 2008.