

Deepening Our Understanding of Communities of Practice in Large-Scale Agile Development

Maria Paasivaara and Casper Lassenius

Department of Computer Science and Engineering, School of Science
Aalto University

POB 15400, 00076 Aalto, Finland

Email: {firstname.lastname}@aalto.fi

Abstract—Communities of Practice (CoPs) are groups of experts who share a common interest or topic and collectively want to deepen their knowledge. The use of CoPs is one of the practices that agile practitioner literature suggests for knowledge sharing and coordination in large-scale agile software development. Even though CoPs are largely used and studied in other industrial contexts, their usage in industrial large-scale software development has not received much research attention. In this paper we describe our findings from a case study of a large software development organization that successfully used CoPs to support their large-scale agile implementation, as well as their transformation from the waterfall model to Lean and Agile. In this case, the use of CoPs was one of the a key success factors. To advance both research and practice we suggest that future research should pay more attention on CoPs in large-scale agile, e.g. by studying how CoPs should be cultivated in the organization, how they work in practice, what they are used for, as well as the benefits and challenges of cultivating and implementing CoPs.

I. INTRODUCTION

Agile software development methods were originally designed for single teams of less than 10 people. However, during recent years, large organizations have increasingly adopted agile software development, leading to a need for scaling the methods. Scaling involves challenges such as coordination between several agile teams, lack of architectural planning, lack of requirement analysis, as well as all the challenges of distributed projects, as many large organizations are distributed [1]. Despite these challenges, a number of large companies have already chosen to adopt agile methods, even though there does not yet exist much written knowledge either on how to scale the agile methods to large-scale projects [2], nor on how to successfully conduct an agile transformation in a large organization [3]. The importance of better understanding this topic is exemplified, e.g., by the fact that practitioners at the XP 2010 conference listed the topic Agile and large projects as the number one top burning research question [4].

One of the practices that the agile practitioner literature, mainly written by consultants, suggests to help with knowledge sharing, organizational and process development, and coordination, is the use of Communities of Practice (CoP) [1], [5]. Even though CoPs have been described and widely used in other industrial contexts, see e.g. [6], [7], their use in professional software development, and in particular in scaling agile development to large organizations, has received little attention in the research literature.

This paper has two goals: First we present a case study on how one large and globally distributed software development organization used CoPs for two different purposes: to support their large-scale agile implementation, and to support their organizational transformation to lean and agile development. Our second goal is to motivate other researchers and practitioners to contribute knowledge to this important topic. We see a clear need for more research on the mechanisms on how to use and benefit from CoPs, as our research shows they can provide significant value to organizations, given that the organization taking CoPs into use knows how to cultivate and support them.

II. RELATED WORK

A. Communities of Practice

In his seminal work, Wenger defines a community of practice, as *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* [7]. Communities of practice have three characteristics that make them different from other forms of communities:

- 1) a specific domain, defining their area of interest,
- 2) an active community, meaning that members actively engage in joint activities, and
- 3) practice, meaning that the members develop a shared set of resources for addressing problems in their domain of interest [7].

Communities of practice have been widely studied in a variety of organizational contexts, however their use in professional software development has received remarkably little research attention.

B. Communities of Practice in Software Engineering

Practitioner literature on scaling agile software development [1], [5], suggest CoPs as a possible solution for learning and knowledge sharing between organizationally separate individuals with similar roles, e.g. between testers or Scrum Masters, in the form of testing CoPs or Scrum Master CoPs. However, case studies or experience reports on how CoPs actually have been implemented in practice are scarce. We found only a few studies reporting on the usage of CoPs in a software engineering context: a small Norwegian software company used CoPs to facilitate learning [8], Nokia build CoPs

to solve inter-team issues [9], and IBM report their experiences from over 60 CoPs [10].

Surprisingly, we did not encounter any in-depth study of the use of communities of practice in large-scale agile software engineering, even though CoPs are suggested in the practitioner literature. Does this mean that CoPs are not much used in real larger-scale agile implementations? Or are the experiences of their use and the benefits gained just not reported?

Our literature search showed that the current software engineering literature lacks insights into how to implement CoPs, how to use them, and what their practical value is. We believe that such knowledge would help 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. We would also like to believe that CoPs are widely used in large-scale agile implementations, but currently we don't know of any study showing this, nor its opposite.

III. METHODOLOGY

We conducted a single descriptive longitudinal case study [11], based on data collected from a large global telecommunications company. We studied the lean and agile transformation of an on-going product development program developing a large and complex systems product, a node that handles a specific type of traffic in telecommunications networks. The development organization was distributed to three global sites and involved around 400 persons in approximately 40 Scrum teams. We purposefully selected this information-rich case [11], as the case organization was a participant in a joint research program.

We collected the data during a three-year period by 52 semi-structured interviews and three observation sessions of CoP meetings. Altogether we performed five interview rounds during which we interviewed persons from different sites, roles and teams, and with differing amounts of experience. The roles interviewed included managers, Product Owners, Scrum Masters, Coaches and team members.

We analyzed the data by categorizing it with help of the qualitative data analysis software Atlas.ti¹. We validated our findings by both arranging a feedback session for our interviewees and letting them read and check our case study report.

IV. RESULTS

A. Communities of Practice in the Case Organization

Our case organization started the lean and agile transformation as they saw that their current plan-driven way of working would not enable them to be competitive enough in the future. Communities of practice were introduced as one of the large-scale agile practices and tried out as part of their full-scale lean and agile roll-out. The main motivation for the case organization to introduce CoPs was that the CoPs were proposed both by the consultants that the organization had

TABLE I. CHARACTERISTICS OF SUCCESSFUL CoPs

Characteristic	Description
Interesting topic	A topic that excites participants and can provide concrete benefits or improvements in the daily work of the participants.
Passionate leader	A leader who builds, motivates and maintains the community, keeps the meetings on track, and communicates actively.
Proper agendas	Clear agendas help keep meetings focused and short, and avoids unnecessary babbling. Agendas available before the meetings helps participants to choose, which CoP meetings to participate in. Anyone is welcome to add topics to the agenda.
Decision making authority	The CoP can make decisions within its realm of expertise that do not need managerial approval to be implemented.
Open community	Anyone is invited to participate in CoP meetings; this avoids the us-vs-them syndrome.
Supporting tools	Tools, like wikis, are needed for agendas, knowledge sharing etc. For distributed meetings good quality videoconference equipment is necessary.
Suitable rhythm	Meetings are held often enough to handle upcoming issues efficiently, but not too often so as not to have "empty" meetings. Meetings can also be held on a need basis.
Cross-site participation	When experts reside at different sites, cross-site synchronous participation is encouraged.

hired, as well as by the practitioner literature on large-scale agile development.

During our case study the organization had over twenty CoPs. When we asked our interviewees to name the most successful CoPs, the same 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.

B. Characteristics of Successful CoPs

We asked our interviewees to list the characteristics of successful CoPs. Based on their answers we identified eight such characteristics: 1) an interesting topic and concrete benefits to participants, 2) a passionate leader, 3) proper agendas, 4) decision making authority, 5) an open community, 6) supporting tools to create transparency, 7) a suitable rhythm, and 8) cross-site participation when needed. These characteristics are explained in more detail in Table I.

C. The Role of CoPs in the Case Organization

The role of communities of practice in the case organization evolved over time. We could identify three main phases: 1) initially, the CoPs worked as a support mechanism for the agile transformation; 2) in the second phase, the main role of CoPs moved to support scaling, the large-scale lean and agile implementation in the case organization, and 3) finally, the CoPs established their place as the main forum for continuous organizational and process improvement.

During the first phase, the transformation phase, the level of knowledge on CoPs was still low in the case organization: people were still learning what CoPs are and how they work. During this phase, the CoPs were either role-based, such as in the Scrum Master CoP or Functional

¹www.atlasti.com

Verification CoPs, or they were used to replace some functions that did not exist anymore in the new organization structure. Managers and coaches initiated the CoPs and intensively supported the building of CoPs at that time.

In the next phase, the scaling phase, the goal of the organization moved from the transformation on how to get the scaling, in particular inter-team coordination to work in practice. During this phase, CoPs gradually replaced the not so successful Scrum-of-Scrums as an inter-team coordination mechanism. Scrum-of-Scrums meetings of representatives from around 25 teams were tried for a while, but that did not work, as these teams did not share enough common interests. Many times, they could not even understand each other. Instead, so called Feature Coordination CoPs, a kind of Scrum-of-Scrum meetings for a few teams working on a common feature, emerged as a well-working mechanism. Similarly, Feature Design CoPs, for designing the feature to be developed by a few teams together, were successful. Drivers for setting up these new CoPs were no longer managers, but coaches, Scrum Masters or Product Owners.

Finally, after the scaling challenges were solved, the organization gradually moved to the continuous improvement phase during which they concentrated on applying Lean thinking, e.g. optimizing the end-to-end flow by removing bottlenecks. CoPs turned out to be a natural fora for jointly designing continuous improvements. CoPs aiming to improve the way of working and optimize the end-to-end flow, such as End-to-End CoP and Coaching CoP established their place.

D. Organizational Support for CoPs

An important success factor for the CoPs seemed to be the supportive atmosphere for building, using and participating in CoPs. For building this atmosphere the openness of participation was a key: all the CoPs were open to anybody who wanted to participate in. People were free join a CoP meeting just to learn new things or to learn what is going on. Participants were not required to be an experts on the topic to join a CoP meeting. It was also acceptable to participate irregularly, e.g., when the agenda happened to contain a topic of interest.

Another topic that contributed to building the supportive atmosphere was that CoP participation was valued in the whole organization: a person was allowed to spend as much time in the CoPs and participate in as many CoPs he or she deemed necessary. CoP participation was valued both by management and colleagues. However, nobody forced anyone to participate in CoPs either—the participation was strictly on a voluntary basis.

Managers and coaches offered their support for building CoPs, especially in the beginning a lot of support and encouragement was needed. It became clear to us that building and leading a new CoP was highly valued, and individuals who took responsibility of forming new CoPs were appreciated both by the managers and peers.

Besides the supportive atmosphere the case organization provided CoPs a good infrastructure including good quality videoconference facilities for cross-site CoPs and, e.g., wiki pages for sharing information such as meeting agendas and minutes, and other information to share.

E. Purposes of CoPs

Finally, we gathered information on the different purposes of CoPs in lean and agile software development. We identified the following four in our case organization: 1) knowledge sharing and learning (e.g. role-based CoPs), 2) coordination (e.g. Feature Coordination CoPs), 3) design (e.g. Feature Design CoPs), and 4) organizational development (e.g. End-to-End CoP). While the first one, knowledge sharing and learning can be seen as the basic purpose of CoPs, the other purposes are not well described even in the general CoP literature.

V. DISCUSSION AND CONCLUSIONS

Our case study showed that 1) CoPs can support a lean and agile transformation, 2) CoPs can support scaling agile to a large and distributed organization, and 3) building a CoP-friendly corporate culture is important for the successful cultivation and growth of CoPs.

For our case organization the CoPs became the most important inter-team coordination mechanism as the only practice that Scrum offers for that purpose, the Scrum-of-Scrums meetings, did not work. Moreover, the CoPs became a very important knowledge sharing and learning instrument in the organization. Based on this case study it is clear that the successful CoP implementation played a very important role for the whole success of both the lean and agile transformation and the large-scale agile implementation in this organization.

As we observed the organization during a three-year period we could see that in the beginning the CoPs were just one practice among others that the organization tried just because both the hired consultant firm and the large-scale agile literature had suggested that. Gradually, the role of CoPs increased and their position became stronger, as the organization noticed that CoPs really worked and provided clear benefits in knowledge sharing, inter-team coordination etc. It was clear that the CoP turned out to be an essential part of both their successful lean and agile transformation as well as their large-scale agile implementation.

This case study showed that when implemented well and supported, the CoPs can be a very powerful practice in large-scale agile.

However, as software engineering literature provides very little advice on how to use CoPs in practice and no in-depth case studies on experiences in using CoP, this is clearly an area that would need more research attention. Based on our case study we believe that also other large organizations planning to move to lean and agile could find CoPs highly useful. Therefore, more research and experiences on how to use CoPs in practice is needed. Our case study described how CoPs work, and are built and supported in one specific kind of context. We would need experiences from other contexts as well, to better understand the contextual factors, such as role of the existing organizational culture and structure, the role of national culture, products, etc.

We suggest that future research should study e.g.: how CoPs work in different contexts, how to provide organizational support for CoPs, what are the benefits and challenges of CoP, and how to best achieve the possible benefits of CoPs.

The implication of our results to practitioners are twofold: First, be brave and take CoPs into use: even single individuals can take initiative in creating CoPs. When 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. If your organization has CoPs already, you can utilize the 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. You may also influence the organization via CoPs, as CoPs are empowered to make decisions in their area of concern. By actively participating you can improve and influence even organization-wide issues.

Second, report your experiences of using CoPs, both successful practices and challenges, as well as benefits and useful advice. Your experiences are useful for other practitioners, as well as advance the general knowledge of CoPs in large-scale agile software development.

ACKNOWLEDGMENT

The authors would like to thank the case organization for its support of the 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),

REFERENCES

- [1] D. Leffingwell, *Scaling Software Agility: Best Practices for Large Enterprises*. Addison-Wesley Professional, 2007.
- [2] E. Hossain, M. A. Babar, and 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*, ser. ICGSE '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 175–184.
- [3] M. Paasivaara, C. Lassenius, V. T. Heikkilä, K. Dikert, and C. Engblom, "Integrating global sites into the lean and agile transformation at ericsson," in *Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on*, 2013, pp. 134–143.
- [4] S. Freudenberg and H. Sharp, "The top 10 burning research questions from practitioners," *IEEE Software*, vol. 27, no. 5, pp. 8–9, 2010.
- [5] C. Larman and B. Vodde, *Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum*. Boston, MA, USA: Addison-Wesley Professional, 2010.
- [6] E. Wenger, *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press, 1998.
- [7] E. Wenger, R. McDermott, and W. M. Snyder, *Cultivating Communities of Practice*. Cambridge, MA: Harvard Business Review Press, 2002.
- [8] A. Mestad, R. Myrdal, T. Dingsoyr, and T. Dyba, "Building a learning organization: Three phases of communities of practice in a software consulting company," in *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, 2007.
- [9] T. Kahkonen, "Agile methods for large organizations - building communities of practice," in *Agile Development Conference, 2004*, 2004, pp. 2–10.
- [10] P. Gongla and C. Rizzuto, "Evolving communities of practice: IBM global services experience," *IBM Systems Journal*, vol. 40, no. 4, pp. 842–862, 2001.
- [11] M. Q. Patton, *Qualitative evaluation and research methods*, 2nd ed. Newbury Park, Calif.: Sage Publications, 1990.