Trends in Large-Scale Agile Development: A Summary of the 4th Workshop at XP2016

Nils Brede Moe
SINTEF
NO-7465 Trondheim
Norway
+47 930 28 687
nils.b.moe@sintef.no

Helena Holmström Olsson
Malmö University
205 06 Malmö
Sweden
+40 66 57319
helena.holmstrom.olsson@mah.se

Torgeir Dingsøy
SINTEF
NO-7465 Trondheim
Norway
+47 930 08 714
torgeird@sintef.no

ABSTRACT

Large projects are increasingly adopting agile development practices, and this raises new challenges for research and practice. The fourth workshop on large-scale agile development focused on the following topics: Distributed large-scale development, inter-team coordination, knowledge sharing, large-scale agile transformations, multidisciplinary work, and new ways-of-organizing for advancing agile practices.

CCS Concepts

Software and its engineering → Software creation and management → Software development process management → Software development methods → Agile software development

Keywords

Agile software development; large-scale software development; virtual teams, social capital, inter-team coordination; agile transformation, sourcing models, continuous software development, self-management, shared decision making, sociocracy.

1. INTRODUCTION

Agile development methods are increasingly adopted and used in other domains than small, co-located teams. The workshop on large-scale agile development focuses on challenges and research opportunities regarding use of agile development methods in large-scale projects and programs [1, 2]. Scale, complexity and distribution (which is often a consequence of scale) introduce a number of challenges. Quality concerns forces companies to govern and control the potentially fragmented multi-team development. Teams in large-scale projects need to reach agreement on many decisions with experts, managers and other teams. Individuals or even teams in large-scale projects will not have necessary knowledge regarding the whole system [3]. Teams

in large-scale projects also need to be highly aligned. If one team breaks the quality or functionality it will affect other teams. However, the need for aligning the work and the process, and for coordinating externally is a threat to team autonomy, which is the key to agility. Further, to have an emerging architecture could hamper project progress when many teams are working in parallel, and some practices like the scrum of scrum has been found to be inefficient in large projects [4].

While “Agile in the large” was voted “top burning research question” by practitioners at the XP2010 conference [5]. Research on agile methods has long focused on single teams and the software development function only.

To address these challenges with large-scale agile we organized a workshop at XP2013 which resulted in a suggested research agenda [2]. At XP2014, we identify “Principles of Large-Scale Agile Development” [1]. The aims of following workshops in 2014 and 2015 were to continue creating a community of researchers interested in this topic, to deepen the knowledge through identifying potential principles of large-scale agile development and to revisit the research agenda defined in 2013.

The workshop in 2016 included two keynotes: “Communities of Practice in Large-Scale Agile” by Casper Lassenius and Maria Paasivaara from the Aalto University, and “Scaling in Practice: Some Myths and Realities” by Evelyn Tian, Ericsson.

Further, the workshop consisted of nine paper presentations by academics and practitioners and group discussions on six topics: Distributed large-scale, inter-team coordination, knowledge sharing, large-scale agile transformations, multidisciplinary work, and new ways of organizing for advancing agile practices. Each topic was discussed in a group and presented in plenary.

Figure 1. One of the group discussions in the workshop.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

XP '16 Workshops, May 24 2016, Edinburgh, Scotland UK
© 2016 ACM. ISBN 978-1-4503-4134-9/16/05...\$15.00
DOI: <http://dx.doi.org/10.1145/2962695.2962696>

In this summary of the workshop discussions and introduction to the papers from the conference, we start by defining what we mean by large-scale agile development, then introduce the papers in this post-conference proceedings with results from the workshop. We hope this summary will inspire further research and provide valuable pointers to relevant research areas.

2. WHAT IS LARGE-SCALE AGILE DEVELOPMENT?

As in the previous three workshops on large-scale agile, we asked participants to define what they mean by "large-scale agile development". We see that many focus on aspects of size such as number of people involved in the development and number of teams. Participants used definitions such as between 3 teams or 30 developers, up to 5 teams and 50 developers. Further, the participants argued the need to distinguish between large-scale and very large-scale. Very large-scale was suggested to be around 9 teams or 100 people. Arguments for a definition based in the number of teams is presented in [6] where large-scale agile is defined as "*agile development efforts with more than two teams*" and very large-scale is defined as "*agile development efforts with more than ten development teams*".

3. TRENDS IN LARGE-SCALE AGILE DEVELOPMENT

In addition to short talks the discussions were arranged according to the world café methodology [7], with the goal of providing feedback on the papers and emerging topics presented at the workshop. We will now briefly describe the six topics addressed at the workshop.

3.1 Distributed Large-Scale

Large-scale agile projects and distribution often go hand in hand. Bass report on findings from nine companies running globally distributed projects. He found that the 21 teams on the nine companies relied on Scrum and Lean and further that they retained QA processes from plan-based development methods. An interesting finding in the work by Bass was that product ownership varies in most teams, and that few teams have the same setup. In some organizations the full product responsibility was given to the offshore site, while others relied on the workbench model - were everyone is onshore except for the developers. Sablies and Smite investigated the supporting roles for teams in two companies. They found that teams working on complex or unfamiliar tasks are required to coordinate within the team, with other teams, and with supporting roles in the rest of the organization cross company. Teams in large-scale agile got a huge external network, however the connections and networks between the sites are weak. From this we can argue that the workbench model described by Bass is very challenging when the offshore team is solving complex and unfamiliar tasks.

From the discussion in the world café it was highlighted the need for better understand how to support distributed virtual teams. Further there is a need for understanding how to organize the network of external experts, stakeholders and managers which the a team need to access to be able to solve complex tasks.

3.2 Inter-team Coordination

The workshop in 2014 focused on common norms and values as well as the importance of knowledge networks to facilitate inter-team coordination [1]. The workshop in 2016 extended this

discussion with findings from a multi-case study and in outlining a study seeking to identify factors that increase or decrease communication and coordination problems:

Bick, Scheerer, and Spohrer describe five approaches to Inter-team coordination, varying from top-down planning to bottom-up adjustment to a combination of both. This shows that a wide specter of coordination approaches are used in practice.

The common definition of coordination is "management of dependencies" [8]. A further finding is that in two of the five cases there were few signs of practices that focus on "active recognition, analysis and management of dependencies".

Crowston et al. outline a study on factors that increase or decrease communication and coordination problems, drawing on organizational discontinuity theory. Findings are believed to advance the conceptual understanding of these factors in large-scale agile development. The study can have implications also for other sectors in society that rely heavily on inter-team coordination such as the automotive and construction industry.

3.3 Knowledge Sharing

Large-scale development reaches the point when there is hardly any person who possesses all the knowledge regarding a system's development and evolution. Knowledge is therefore fragmented and teams often specialize in one or a number of areas (e.g. sub-systems, components, or type of features). Therefore the teams need effective knowledge networks that enable frequent knowledge exchange as reported by Sablies and Smite.

In the study presented by Rolland, the challenges involved when scaling agile to a large project is explored. With large-scale, the author refers to development projects involving multiple and multi-disciplinary teams, external experts and extensive complexity in relation to integration and migration of existing systems. To help address these challenges, the paper proposes a conceptual framework that focuses on the social and organizational aspects in sharing knowledge and coordinating in large-scale settings. Based on work by Carlile on knowledge boundaries [9], the framework recognizes that scaling agile comprises much more than just expanding the 'traditional' agile practices as scrum-of-scrums, sprint planning in teams, and a focus on collaboration within teams. The paper concludes that successful scaling of agile practices needs both structure and agility, and it questions the common assumption that structured methods and agility are opposing forces that need to be 'balanced'. Rather, structure is identified as necessary for agility – and vice versa. Similarly, Björnson and Vestues report on empirical research focusing on knowledge sharing and improvement in two larger case organizations. The cases are compared with a number of interesting insights on how agile practices were adopted, perceived and tailored. The authors emphasize that knowledge sharing and improvement practices are heavily case dependent and that what approach to take needs to be adjusted according to the type of project. Also, dynamic structures for learning as well as clear decision-making processes for continuous improvement are identified as critical.

While the world café discussed and agreed upon that that CoPs are essential for succeeding in knowledge sharing in large-scale, there is a need for better understating the concept of CoP and what it mean as an analytical lens. Further, there are challenges related to newcomers in large-scale projects and CoP that are not understood. How can the newcomers in a large-scale project easily become part of the knowledge sharing activities. Finally,

while the team should be self-managed, such teams might establish new knowledge boundaries.

3.4 Large-scale Agile Transformations

Introducing agile in a large organization is one frequently reported challenge in large-scale agile. Paasivaara and Lassenius argue that the difficulty is related to size of the organization, because large organizations change very slowly, and that adopting agile requires change to the organizational culture. In their systematic literature review Paasivaara and Lassenius identified 35 challenges of large-scale agile adoptions grouped into nine categories, and 29 success factors grouped into eleven categories. To understand which challenges and success factors are most important there is a need to perform a survey among the industry.

Further there is a need for case studies to understanding how to know how much effort you need to put into different parts of the agile transformation, how different scaling framework allow for different focus, and how the product architecture is a main driver for how to drive transformation.

3.5 Multidisciplinary Work

Laanti argues that the problem with traditional project thinking agile, are the same whether you develop software or hardware. She has identified practices that are critical to an organisation moving from traditional to agile and lean ways of working when developing systems with software and hardware:

1. Two-level rolling planning
2. Synchronisation
3. Cadence
4. Key decision points

All changes of practices require a change in the culture.

Faster feedback brings faster learning, and there is a need to better understand how to achieve faster feedback when developing solutions containing both hardware and software. At the same time this is difficult because hardware has always been developed using plan-based methods, and it is challenging to share unfinished hardware. Implementing agile in hardware development requires support from the whole organisation, and a change in the culture.

3.6 New Ways-of-Organizing

Large-scale agile development requires supporting structures that do not hinder, but foster agility. During the workshop, a number of papers that recognized the need for new ways-of-organizing for advancing agile practices were presented. Common topics in these papers are the challenges associated with aligning agile ways-of-working with longer development cycles as experienced in hardware-intensive companies, the scaling of agile practices in highly distributed contexts and how to implement self-organizing principles in large-scale agile development teams in order to increase team responsibility and independency.

In a study conducted by Laanti, the concept of Lean-Agile hardware development is introduced, and it is concluded that this is not only possible but also brings a number of benefits to companies. The study shows how a company that was already experienced in using agile practices for software development expanded these to involve also hardware development and was able to replace the traditional milestones with a Lean-Agile framework. The key in this new framework was the separation of

content review and decision-making that made the milestones less heavy and hence, the company was able to act more nimble. As an empirical account based of large-scale development, the paper is an interesting example on how new ways-of-organizing can accelerate the adoption of agile practices at the same time as these drive new organizational forms.

In focusing on self-organizing as one of the core values of agile development, Eckstein explores how sociocracy, and the principles of ‘double-linking’ and ‘shared decision making’, can enable teams operating in large-scale contexts to adopt more independent ways-of-working. In the paper, the author provides a rewarding explanation of how sociocracy suggests alternative ways for decision-making and principles on how to organize to increase involvement and engagement are outlined. The author concludes that as an emerging way of organizing, sociocracy supports the agile values in general and the large-scale in particular. As recognized by the author, sociocracy provides a step-by-step approach that allows organizations to change gradual, and with its detailed discussion of the four core principles of sociocracy, this paper provides a comprehensive overview of the impact of such a change.

4. CONCLUSION

This year’s workshop continued the focus on the topics inter-team coordination and knowledge sharing from previous workshops, and also raised discussions on the new topics of large-scale development in relation to distributed development, multidisciplinary work and new ways-of-organizing.

5. ACKNOWLEDGMENTS

The work on this article was supported by the Agile 2.0 project (Research council of Norway grant 236759) and the Smiglo project (Research Council of Norway grant 235359). Thanks to all workshop participants for engaging discussions and to program committee members for reviews. We are also grateful to co-chair Morten Elvang for organizing this year’s workshop.

6. REFERENCES

- [1] Dingsøy, T. and N.B. Moe, Towards Principles of Large-Scale Agile Development: A Summary of the workshop at XP2014 and a revised research agenda, in *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation*, T. Dingsøy, et al., Editors. 2014, Springer: Berlin. p. 1-8.
- [2] Dingsøy, T. and N.B. Moe, Research Challenges in Large-Scale Agile Software Development. *ACM Software Engineering Notes* 2013. **38**(5): p. 38-39.
- [3] Moe, N.B., et al., Networking in a large-scale distributed agile project, in *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 2014, ACM: Torino, Italy. p. 1-8.
- [4] Paasivaara, M., C. Lassenius, and V.T. Heikkilä. Inter-team coordination in large-scale globally distributed scrum: Do Scrum-of-Scrums really work? in *Empirical Software Engineering and Measurement (ESEM)*, 2012 ACM-IEEE International Symposium on. 2012.
- [5] Freudenberg, S. and H. Sharp, The Top 10 Burning Research Questions from Practitioners. *IEEE Software*, 2010: p. 8-9.
- [6] Dingsøy, T., T. Fægri, and J. Itkonen, What Is Large in Large-Scale? A Taxonomy of Scale for Agile Software

- Development, in *Product-Focused Software Process Improvement*, A. Jedlitschka, et al., Editors. 2014, Springer International Publishing. p. 273-276.
- [7] Brown, J. and Isaacs, D., *The World Cafe: Shaping Our Futures Through Conversations That Matter*: Berrett-Koehler Publishers, 2005.
- [8] Malone, T.W. and K. Crowston, The interdisciplinary study of coordination. *ACM Computing Surveys (CSUR)*, 1994. 26(1): p. 87-119.
- [9] Carlile, P.R., A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organization science*, 2002. 13(4): p. 442-455.