Towards Principles of Large-Scale Agile Development:
A Summary of the workshop at XP2014 and a revised research agenda[1]

Torgeir Dingsøyr,[1,2] Nils Brede Moe,[1,3]

[1]SINTEF,
NO-7465 Trondheim, Norway
torgeird@sintef.no
[2]Department of Computer and Information Science,
Norwegian University of Science and Technology
[3]Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden
nils.b.moe@sintef.no

Abstract. Large projects are increasingly adopting agile development practices, and this raises new challenges for research. The workshop on principles of large-scale agile development focused on central topics in large-scale: the role of architecture, inter-team coordination, portfolio management and scaling agile practices. We propose eight principles for large-scale agile development, and present a revised research agenda.

Keywords: Large-scale agile software development, architecture, portfolio management; project management, scaling, inter-team coordination, software engineering

1 Introduction

Since the formulation of the agile manifesto in 2001, agile methods have transformed software development practice by strongly emphasizing change tolerance, evolutionary delivery and active end-user involvement [1]. Agile development has received widespread interest, resulting in a shift of patterns of thought. Scrum is now a de facto standard for development in many countries, and other methods like extreme programming (XP) and elements of lean software development such as Kanban are in widespread use. Rajlich describes agile development as a paradigm shift in software engineering that “brings a host of new topics into the forefront of software engineering research” [2].

In the first special issue on agile development, in *IEEE Computer*, Williams and Cockburn [3] stated that agile methods “best suit *collocated teams* of about *50 people or fewer* who have easy access to user and business experts and are developing projects that are not life-critical”. The success of agile methods for small, co-located teams has inspired use in new domains: Companies increasingly apply agile practices to large-scale projects.

However, there are challenges with achieving the same productivity gains in these areas, as in the “home ground” of agile methods. Agile methods are based on the idea that high-quality software can be developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change [4]. As agile development techniques are used on large-scale projects, new challenges arise. “Agile in the large” was voted “top burning research question” by practitioners at the XP2010 conference [5].

Fundamental assumptions in agile development are severely challenged when using these practices in large-scale projects. Self-management is a central principle in agile methods, but studies from other fields than software development indicate that self-management can reduce the ability to effectively coordinate across teams [6]. Also while the teams need to self-manage, team members need to have an effective knowledge network and collaborate closely with experts outside the team in large-scale agile [7]. 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 [8]. An international survey on agile adoption[2] shows that agile practice has primarily been successful in small teams.

To address these challenges with agile methods in large projects, we organized a workshop at XP2013 which resulted in a suggested research agenda [9]. At XP2014, we wanted to strengthen this line of

[1] Postprint, published as: Dingsøyr, T., & Moe, N. B. (2014). Towards Principles of Large-Scale Agile Development: A Summary of the workshop at XP2014 and a revised research agenda. In Torgeir Dingsøyr, Nils Brede Moe, Steve Counsell, Roberto Tonelli, Cigdem Gencel, & Kai Petersen (Eds.), *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation* (Vol. 199, pp. 1-8). Berlin: Springer. https://link.springer.com/chapter/10.1007/978-3-319-14358-3_1

[2] Scott Ambler: Agile Adoption Rate Survey 2008, http://www.ambysoft.com/surveys/agileFebruary2008.html

research and organized a workshop on “Principles of Large-Scale Agile Development.” The aims of this workshop were to create 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 included a keynote on "Characteristics and principles for large-scale agile development" by Maarit Laanti [10] and a second keynote on “The leader role in large-scale agile development” by Lars-Ola Damm. All workshop members were asked to define what they meant by ‘large-scale agile.’ Further, we had paper presentations and group discussions on four topics: Architecture, inter-team coordination, portfolio management and scaling. The assignment for groups during discussion was to identify main principles within each core area. After these discussions we revisited the research agenda and did a voting on what should be high priority topics.

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 and add results from the workshop discussions to propose principles in the four areas architecture, inter-team coordination, portfolio management and scaling. Finally, we present the revised research agenda defined at 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?

The term ‘large-scale agile development’ has been used to describe agile development in everything from large teams to large multi-team projects to making use of principles of agile development in a whole organization.

There is an established discussion on what constitutes agile software development, with Conboy [11] providing the most thorough discussion. He defines agility as the continuous readiness “to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment.”

How participants at the workshop defined large-scale agile development is shown in Table 1. We see that many focus on aspects of size such as number of people involved in the development, lines of code in the solution, number of development sites, number of teams, to definitions such as “agile in larger organizations.” Arguments for a definition based in the number of teams is presented in [12], where large-scale agile is defined as “*agile development efforts with more than two teams.”*

This definition excludes agile methods applied in large organizations from ‘large-scale agile’, and we propose that this is considered as a research direction on its own.

Table 1. Definitions by workshop participants at XP2014.

Definitions of large-scale agile development
Over 50 developers OR 1/2 million lines of code OR more than 3 sites / time zones.
Over 50 persons, over 5 teams, developing together the same product / project using agile method.
Agile being applied to more than one team, one project, one product.
Agile applied on the organisational level.
Truly agile development in a context of more teams than one person can manage, and larger products than few teams can handle.
When coordination of teams can be achieved in a new forum like a Scrum of Scrum forum.
Several arenas are needed for coordination, like multiple Scrum of Scrums.
Large teams -– how to get everyone on board with framework.
Big projects / Many people / Crucial to organisation / Customer focused / Flexible change / Many projects.
It is when you don´t know everyone else working in the same project/product.
Large-scale agile success depends on having the right structures in place “freedom to perform”.
Agile organisations are those that learn fast and are effective in creating value.
Multiple teams working together in order to deliver software artefacts.
When the values/principles or practices scale, extends to other functions, units of a company, i.e. beyond team and projects (+ how it is done).

Driven by many needs and challenges in organisations.
Emergent complex and adaptive approach, cultural based – a mind-set.

3 Towards Principles of Large-scale agile development

The workshop focused on four aspects of large-scale agile development: Architecture, inter-team coordination, portfolio management and scaling:

Architecture

Some of the critique of agile development and in particular of large-scale agile development has been how architecture is managed in such development efforts. Nord et al. [13] takes the position that agility is enabled by architecture, and architecture is enabled by agility. Architects work on three key concerns: Architecture of the system, the structure of the development organization and the production infrastructure. They identify a set of architectural tactics, which guide the alignment of the three concerns. Further advice to organising architectural work in large-scale agile development is provided by Eckstein [14], who argue that architectural work should be organized, depending on the number of changes and the level of uncertainty. Based on the above mentioned arguments and discussions at the workshop, we propose the following principles:

1. Architecture has a key role in defining how work is coordinated in large-scale development efforts.
2. The level of change and level of uncertainty will influence how the architecture work should be organized.

Inter-team coordination

A number of measures are important to coordinate teams in large-scale development efforts. Paasivaara et al. focus on the importance of defining common values through a study of an agile transformation project [15]. The article describes how value workshops were used to define common values, agree on interpretations and define behavioural implications. A similar approach is described by Nyfjord et al. [16], who focus on establishing common norms or conventions like "speaking the teams language". At the workshop, discussions in several groups focused on the importance of knowledge networks to achieve inter-team coordination.

In large-scale development, the needs for coordination of work appear on two levels – the team's level and between the team and the rest of the organization. Team members need to collaborate effectively within the team to accomplish their tasks, but also with experts outside the team, e.g. designers, architects, infrastructure personnel, and other stakeholders. Large-scale development needs an effective knowledge network.

Based on the articles and discussions at the workshop, we propose the following principles to achieve inter-team coordination in addition to standard mechanisms like the scrum-of-scrum meetings:

3. Common norms and values facilitate inter-team coordination.
4. Effective knowledge networks are essential in large-scale development due to the knowledge-intensive nature of software development.

Portfolio management

Agile project portfolio management is about handling several agile projects in the same portfolio. In traditional project portfolio management, the portfolio consists of a set of projects executed in isolation from each other and the changing environment [17]. However, in agile software development this is not valid as projects are flexible, feedback driven, and embracing change even at the end of the project. A new request from a client on a single project, may affect the portfolio as a whole.

Controls are essential when managing a project portfolio. Controls can be understood as an attempt by the organization to influence people to take actions and make decisions, which are consistent with the goals of the organisations. Controls can be formal and informal. The informal control is contained by the development group (clan) and the individual's ability to monitor and evaluate it selves according to the acceptable behaviour.

Agile project portfolio management can also be seen as handling a dynamic non-linear system. The theory of Complex-adaptive systems (CAS) can be used to study such systems [17]. In CAS self-organization emerges as agents interacts through simple rules that can change and adapt. Feedback is the driving force of change.

Based on the article by Sweetman et al. and the discussion in the workshop, we propose the following principle of large-scale agile portfolio management:

5. Continuously feedback from the portfolio to project levels enables the teams and project members to take decisions that are consistent with the goals of the large-scale agile portfolio.
6. Continuously feedback from the project level to the portfolio level enables changing the portfolio to optimise the value of the large-scale agile portfolio.

Scaling

Applying principles of large-scale agile development, requires an understanding of the term "scaling". Without a proper understanding of the term, inappropriate methods may be applied. Power [18] argues that there are three contexts of agility and scale: 1) agile use in a large organisation 2) agile used in a large development effort in a large organizations, and 3) the large organization it selves is agile. Understanding the context is essential when choosing the right approach for improving the agility of the teams and organization.

Eklund et al. [19] argues that scaling agile software development in embedded systems is a question of scaling agile in two dimensions: First increasing the involved number of teams and utilize agile practices for mid- and long-range planning such as release planning and road mapping. Second, scaling the system engineering activities executed in each sprint, to a truly iterative practice instead of a stage-gated planned approach. A cross-functional team must have the ability or support to perform activities at several abstraction levels in a systems engineering V-model in each iteration or sprint.

Based on the article by Power [18] and Eklund et al. [19] and the discussion in the workshop, we propose the following principle of large-scale agile portfolio management:

7. Describing the context for agility and scale is essential for understanding how to improve agility in large-scale agile.
8. For large-scale embedded systems development, agility should scale both with respect to the number of involved teams, and the systems engineering activities in each iteration due to the co-dependency of software and hardware development.

4 Revised Research Agenda

In the workshop, the researchers and practitioners revisited the research agenda defined in 2013 [9]. The research agenda was first presented and then the topics where discussed both from an academic and a practitioner perspective. The research agenda was then modified before topics were given priority by the workshop participants. The topics included on the agenda were categorized as high or medium priority.

Table 2. Revised research agenda for large-scale agile software development.

Priority	*Topic*	*Description*
High	Organisation of large development efforts	Organizational models, portfolio management, governance, project management, agile product-line engineering.
	Variability factors in scaling	Identify what factors are important in large projects that influence the development process.
	Inter-team coordination	Coordination of work between teams in large-scale agile development.
	Key performance indicators in large development efforts	Identify appropriate metrics are to monitor progress and support transparency.
	Knowledge sharing and improvement	How to ensure feedback for learning, use of knowledge networks and learning practices.
	Release planning and architecture	Coordinating and prioritizing functional and non-functional requirements, continuous delivery, minimizing technical debt.
Medium	Customer collaboration	Practices and techniques for product owners and customers to collaborate with developers in large-scale projects.
	Scaling agile practices	Determine which agile practices scale and which do not. Understand why and when agile practices scale.
	Agile contracts	Understand if contracts can change the mind-set of customers from upfront planning to agile principles. Uncover what legal limitations exist in contracts

	that reduce agility in large projects.
Agile transformation	Efficient adoption of agile practices in large projects.
UX design	Integration of user experience design in large projects

With the suggested research agenda in Table 2, we hope this will foster a continuous discussion over research agenda, and be an inspiration for future research.

Acknowledgement: The work on this article was supported by the SINTEF internal project “Agile project management in large development projects”, by the Swedish Knowledge Foundation under the KK-Hög grant 2012/0200, and by the project Agile 2.0 which is supported by the Research council of Norway through grant 236759/O30, and by the companies Kantega, Kongsberg Defence & Aerospace and Steria. We are grateful to Tor Erlend Fægri at SINTEF for comments on an earlier version of this article, and to all workshop participants for engaging discussions.

References

[1] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, "A Decade of Agile Methodologies: Towards Explaining Agile Software Development," *Journal of Systems and Software,* vol. 85, pp. 1213-1221, 2012.

[2] V. Rajlich, "Changing the paradigm of Software Engineering," *Communications of the ACM,* vol. 49, pp. 67 - 70, 2006.

[3] L. Williams and A. Cockburn, "Agile Software Development: It’s about Feedback and Change," *IEEE Computer,* vol. 36, pp. 39-43, 2003.

[4] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies," *Communications of the ACM,* vol. 48, pp. 72-78, 2005.

[5] S. Freudenberg and H. Sharp, "The Top 10 Burning Research Questions from Practitioners," *IEEE Software,* pp. 8-9, 2010.

[6] J. A. Ingvaldsen and M. Rolfsen, "Autonomous work groups and the challenge of inter-group coordination," *Human Relations,* vol. 65, pp. 861-881, Jul 2012.

[7] N. B. Moe, D. Smite, A. Sablis, A.-L. Börjesson, and P. Andréasson, "Networking in a large-scale distributed agile project," presented at the Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Torino, Italy, 2014.

[8] M. Paasivaara, C. Lassenius, and V. T. Heikkila, "Inter-team Coordination in Large-Scale Globally Distributed Scrum: Do Scrum-of-Scrums Really Work?," *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM'12),* pp. 235-238, 2012.

[9] T. Dingsøyr and N. B. Moe, "Research Challenges in Large-Scale Agile Software Development," *ACM Software Engineering Notes* vol. 38, pp. 38-39, 2013.

[10] M. Laanti, "Characteristics and Principles of Scaled Agile," in *Agile Methods: Large-Scale Development, Refactoring, Testing, and Estimation.* vol. 199, T. Dingsøyr, N. B. Moe, R. Tonelli, S. Counsell, C. Gencel, and K. Petersen, Eds., ed: Springer Verlag, 2014.

[11] K. Conboy, "Agility From First Principles: Reconstructing the Concept of Agility in Information Systems Development," *Information Systems Research,* vol. 20, pp. 329-354, 2009.

[12] T. Dingsøyr, T. E. Fægri, and J. Itkonen, "What is Large in Large-Scale? A Taxonomy of Scale for Agile Software Development," *accepted for publication at Profes 2014*, Helsinki, 2014.

[13] R. L. Nord, I. Ozkaya, and P. Kruchten, "Agile in Distress: Architecture to the Rescue," in *Agile Methods: Large-Scale Development, Refactoring, Testing, and Estimation.* vol. 199, T. Dingsøyr, N. B. Moe, R. Tonelli, S. Counsell, C. Gencel, and K. Petersen, Eds., ed: Springer Verlag, 2014.

[14] J. Eckstein, "Architecture in Large Scale Agile Development," in *Agile Methods: Large-Scale Development, Refactoring, Testing, and Estimation.* vol. 199, T. Dingsøyr, N. B. Moe, R. Tonelli, S. Counsell, C. Gencel, and K. Petersen, Eds., ed: Springer Verlag, 2014.

[15] M. Paasivaara, O. Vaattanen, M. Hallikainen, and C. Lassenius, "Supporting a Large-Scale Lean and Agile Transformation by Defining Common Values," in *Agile Methods: Large-Scale Development, Refactoring, Testing, and Estimation.* vol. 199, T. Dingsøyr, N. B. Moe, R. Tonelli, S. Counsell, C. Gencel, and K. Petersen, Eds., ed: Springer Verlag, 2014.

[16] J. Nyfjord, S. Bathallath, and H. Kjellin, "Conventions for Coordinating Large Agile Projects," in *Agile Methods: Large-Scale Development, Refactoring, Testing, and Estimation.* vol. 199, T. Dingsøyr, N. B. Moe, R. Tonelli, S. Counsell, C. Gencel, and K. Petersen, Eds., ed: Springer Verlag, 2014.

[17] R. Sweetman, O. O’Dwyer, and K. Conboy, "Control in Software Project Portfolios: A Complex Adaptive Systems Approach," in *Agile Methods: Large-Scale Development, Refactoring, Testing, and Estimation.* vol. 199, T. Dingsøyr, N. B. Moe, R. Tonelli, S. Counsell, C. Gencel, and K. Petersen, Eds., ed: Springer Verlag, 2014.

[18] K. Power, "A Model for Understanding When Scaling Agile is Appropriate in Large Organizations," in *Agile Methods: Large-Scale Development, Refactoring, Testing, and Estimation*. vol. 199, T. Dingsøyr, N. B. Moe, R. Tonelli, S. Counsell, C. Gencel, and K. Petersen, Eds., ed: Springer Verlag, 2014.

[19] U. Eklund, H. H. Olsson, and N. J. Strøm, "Industrial challenges of scaling agile in mass-produced embedded systems," in *Agile Methods: Large-Scale Development, Refactoring, Testing, and Estimation*. vol. 199, T. Dingsøyr, N. B. Moe, R. Tonelli, S. Counsell, C. Gencel, and K. Petersen, Eds., ed: Springer Verlag, 2014.