

Planned Research: Scaling Agile Practices in Software Development

Kathrine Vestues^(✉)

Department of Computer and Information Science,
Norwegian University of Science and Technology, 7465 Trondheim, Norway
kathrine.vestues@idi.ntnu.no

Abstract. Agile methods are increasingly being applied to large scale and distributed software development. While there is much evidence to support the efficiency of agile practices in small co-located team, less is known about the applicability of these practices to large scale projects. This paper gives an outline of planned research on the scaling of retrospectives. By using retrospectives as an empirical lens I will try to gain insight into the limitations and benefits of agile practices in large scale and distributed development.

Keywords: Agile · Large scale agile · Distributed agile · Retrospectives

1 Introduction

Agile methods were originally seen as best suited for small co-located teams with easy access to users and business experts, developing non-life-critical systems [1]. However, the good results achieved in small, co-located teams has led organizations to apply agile practices and principles to large scale and distributed software projects. Several papers and surveys show that both practitioners and researchers recognize a need for further knowledge about scaling of agile practices – a need that is further emphasized by the apparent gap in research revealed by a preliminary literature review.

This paper gives an outline of planned research on the scaling of retrospectives. By using retrospectives as an empirical lens, I hope to contribute to a better understanding of how agile practices scale in distributed and large projects.

The next section contains a short overview of some of the literature in the field, tying it to the research proposed in this paper. I will then go on to present the research problem, research methods and future agenda.

2 Background

Early research on agile methods focused mainly on implementation and adaption of agile practices in small, co-located teams [2]. As the use of agile methods became more popular, and the number and quality of studies increased [3], the research scope gradually expanded to include use of agile in new contexts. Two such contexts are agile in distributed teams, and agile in large scale projects.

2.1 Distributed Agile Development

In most multi-team project settings distribution will be an inevitable side effect. While team members often are co-located, the different teams tend to be distributed across multiple buildings or even countries. When looking at large scale agile, it will therefore be relevant to look at how practices scale across distances, as well as across teams.

Several papers have been written about distributed agile. Among the topics treated are pair programming in globally distributed projects [4], whether distributed development can be agile [5], and inter team coordination in large scale Scrum [6, 7]. Little seems to have been written about scaling of retrospectives across distributed teams.

2.2 Large Scale Development

At the XP2010 conference, Freudenberg and Sharp [8] compiled a list of “top ten burning questions” based on feedback from practitioners. Agile and large projects were on top of the list. Dingsøy and Moe [9] summarize the large scale agile research challenges that were discussed at the International Conference on Agile Software Development (XP2013). Among the 8 topics listed, two of them concern scaling of practices: “Which agile practices scale and which do not? Why and when do agile practices scale?” and “How can agile practices be adopted efficiently in large projects?”. Following the taxonomy given by Dingsøy et al. [10], project size is determined by the number of team. A project is considered large if it has 2–9 teams.

The literature review revealed several papers on large scale agile [7, 11, 12], but none discussing the scalability of retrospectives.

2.3 Retrospective Practices

The secret to more successful project management is learning from the past [13]. Dybå et al. also emphasize the importance of reflection and learning in the IEEE special issue on reflective practices [14]. Retrospectives are one way of achieving such reflection and learning.

The term “retrospective” was first used by Kerth [15], but was soon adopted by the agile community. Being a key practice within agile methods, the retrospective has been given much attention by both practitioners and researchers. Several books offer practical advice on planning and running retrospectives [16, 17]. There has also been done research on specific retrospective techniques [18] and how retrospectives contribute to the software improvement processes in agile development [19]. Little research has been dedicated to retrospectives in large scale and distributed projects.

The interest of practitioners and researchers in scaling of agile practices, and the lack of research on retrospectives in large scale projects makes the research questions presented in the next chapter highly relevant.

3 Research Problem

The (main) goal of the research is to contribute to a better understanding of how agile practices can give value in large scale software development.

To reach an answer, I will look into the following underlying research questions in detail:

Research question 1: How can retrospectives be adapted to suite large scale and distributed projects?

Research question 2: How does the retrospective practice contribute to process improvement and sharing of knowledge across teams?

4 Research Methodology

The research will be done using interpretive case studies, divided into 3 phases:

Qualitative survey: To indicate relevance of the above research questions, I will start by performing interviews with 5–6 key informants. These informants will be chosen among project managers and Scrum masters in large scale, distributed projects in public and/or private sector in Norway.

Exploratory interpretative case studies: Initially, 2 cases will be chosen; one normal and one critical case. Data will be gathered through observation and interviews of projects members from different organizational levels.

Descriptive interpretative case studies: Not yet specified.

5 Dissertation and Publication

- Two workshop article synthesising the findings from literature review
- Two conference article on the preliminary findings from the case studies
- One journal article at the end of the study reporting on the full case studies

6 Future Agenda

Future work will be to select a theoretical foundation, conduct field studies and analyse and communicate the results.

Open Access. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, duplication, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, a link is provided to the Creative Commons license and any changes made are indicated.

The images or other third party material in this chapter are included in the work's Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work's Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt or reproduce the material.

References

1. Williams, L., Cockburn, A.: Guest Editors' introduction: agile software development: It's about feedback and change. *Computer* **6**, 39–43 (2003)
2. Dybå, T., Dingsøy, T.: Empirical studies of agile software development: a systematic review. *Inf. Softw. Technol.* **50**(9), 833–859 (2008)
3. Dingsøy, T., Nerur, S., Balijepally, V., Moe, N.B.: A decade of agile methodologies: Towards explaining agile software development. *J. Syst. Softw.* **85**(6), 1213–1221 (2012)
4. Flor, N.V.: Globally distributed software development and pair programming. *Commun. ACM* **49**(10), 57–58 (2006)
5. Ramesh, B., Cao, L., Mohan, K., Xu, P.: Can distributed software development be agile? *Commun. ACM* **49**(10), 41–46 (2006)
6. Lee, S., Yong, H.-S.: Distributed agile: project management in a global environment. *Empirical Softw. Eng.* **15**(2), 204–217 (2010)
7. Paasivaara, M., Lassenius, C., Heikkilä, V.T.: Inter-team coordination in large-scale globally distributed scrum: Do scrum-of-scrums really work? In: 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 235–238. IEEE (2012)
8. Freudenberg, S., Sharp, H.: The top 10 burning research questions from practitioners. *IEEE Softw.* **27**(5), 8–9 (2010)
9. Dingsøy, T., Moe, N.B.: Research challenges in large-scale agile software development. *ACM SIGSOFT Softw. Eng. Notes* **38**(5), 38–39 (2013)
10. Dingsøy, T., Fægri, T.E., Itkonen, J.: What is large in large-scale? *A Taxonomy of Scale for Agile Software Development*. In: Jedlitschka, A., Kuvaja, P., Kuhrmann, M., Männistö, T., Münch, J., Raatikainen, M. (eds.) PROFES 2014. LNCS, vol. 8892, pp. 273–276. Springer, Heidelberg (2014)
11. Dingsøy, T., Moe, N.B.: Towards principles of large-scale agile development. In: Dingsøy, T., Moe, N.B., Tonelli, R., Counsell, S., Gencel, C., Petersen, K. (eds.) XP 2014. LNBIP, vol. 199, pp. 1–8. Springer, Heidelberg (2014)
12. Moe, N.B., Šmite, D., Šablīs, A., Börjesson, A.-L., Andréasson, P.: Networking in a large-scale distributed agile project. In: Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2014, p. 12. ACM (2014)
13. Nelson, R.R.: Project retrospectives: evaluating project success, failure, and everything in between. *MIS Q. Executive* **4**(3), 361–372 (2005)
14. Dyba, T., Maiden, N., Glass, R.: The reflective software engineer: reflective practice. *IEEE Softw.* **31**(4), 32–36 (2014)
15. Kerth, N.L.: *Project Retrospectives: A Handbook for Team Reviews*. Dorset House Publishing, New York (2001)
16. Derby, E., Larsen, D., Schwaber, K.: *Agile Retrospectives: Making Good Teams Great*. Pragmatic Bookshelf, Raleigh (2006)
17. Kua, P.: *The retrospective handbook* (2013). E-book <https://leanpub.com/the-retrospective-handbook>

18. Lehtinen, T.O.: Development and evaluation of a lightweight root cause analysis method in software project retrospectives. Aalto University (2014)
19. Salo, O.: Enabling Software Process Improvement in Agile Software Development Teams and Organisations. VTT Publications, Espoo (2006)