

Agile Process Improvement in Retrospectives

Christoph Matthies
Hasso Platter Institute
University of Potsdam
Potsdam, Germany
christoph.matthies@hpi.de

Abstract—Working in iterations and repeatedly improving team workflows based on collected feedback is fundamental to agile software development processes. Scrum, the most popular agile method, provides dedicated retrospective meetings to reflect on the last development iteration and to decide on process improvement actions. However, agile methods do not prescribe how these improvement actions should be identified, managed or tracked in detail. The approaches to detect and remove problems in software development processes are therefore often based on intuition and prior experiences and perceptions of team members. Previous research in this area has focused on approaches to elicit a team’s improvement opportunities as well as measurements regarding the work performed in an iteration, e.g. Scrum burn-down charts. Little research deals with the quality and nature of identified problems or how progress towards removing issues is measured. In this research, we investigate how agile development teams in the professional software industry organize their feedback and process improvement approaches. In particular, we focus on the structure and content of improvement and reflection meetings, i.e. retrospectives, and their outcomes. Researching how the vital mechanism of process improvement is implemented in practice in modern software development leads to a more complete picture of agile process improvement.

Index Terms—Agile, Scrum, software process improvement, retrospective

I. INTRODUCTION

A vital feature of modern, agile software development processes is the focus on iterative development [1] and adaptation to the given context of a project and team [2].

A. Agile Feedback Cycles

The cycle of regularly repeating the analysis, design, implementation, and evaluation steps in a software development process enables frequent deliveries and timely feedback [2]. During development, feedback is not only collected regarding the created software, i.e. on quality or usefulness, but also regarding the execution of the process. By critically reflecting on the executed development process of the last iteration, problems within a team can be identified and improvements can be proposed, making the process more effective and more enjoyable for team members [3]. This concept is assigned a designation in most software development methodologies, such as *Kaizen*, the idea of continuous improvement in Lean or the *Inspect & Adapt* approach of Empirical Process Control in Scrum [4].

B. Feedback in Scrum

Scrum, the agile process framework most popular in industry [5], explicitly provides meetings for the collection of feedback, both on iteration outcome as well as the quality of the employed process. While these are regularly scheduled meetings, they are not the only opportunities for feedback within the process. Figure 1 shows an overview of the feedback cycles within the Scrum process.

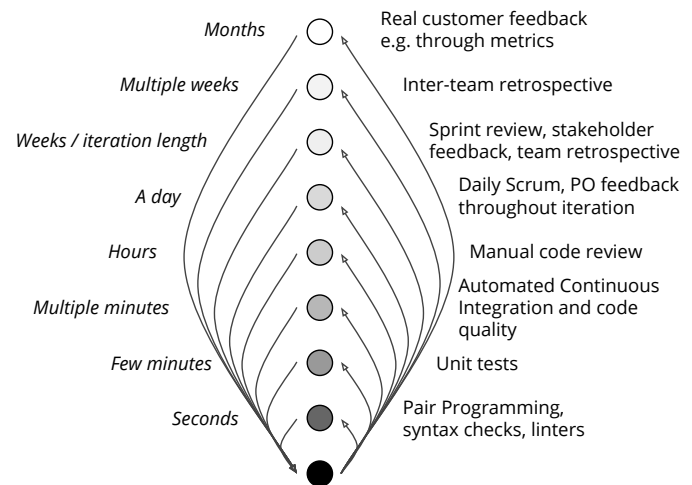


Fig. 1. Overview of feedback cycles and their time spans within the Scrum method. While Pair Programming provides feedback within seconds, feedback through retrospectives or software usage metrics can take weeks.

During coding and Pair Programming, feedback is generated in an immediate, conversational way. Unit tests and Continuous Integration services provide feedback on the scale of minutes, whereas manual code review and daily Scrum meetings happen only every few hours or once a day, respectively. During a sprint review meeting, the development team evaluates *what* was built in the last iteration. In contrast, in retrospectives the software development team discusses *how* it was built. It is discussed, what aspects of the performed work and collaboration worked well during the last iteration, and should consequently be continued, as well as what issues hindered work progressing and posed problems. The team then collectively decides which improvements could be made in the next development iteration.

II. RESEARCH OBJECTIVES

The main objective of this research is to gain a better understanding of process improvement approaches, particularly Scrum retrospectives, in agile software development teams in industry. As agile development methodologies are strongly influenced by industry practitioners, this is where the study should take place. By collecting the experiences of developers in a multitude of companies, working on different problems in different domains we can gain insights into how process improvement is executed in industry, what roles and meetings are involved and how teams can be further supported. In particular, the nature of identified action items is of interest and what type of ontology can be developed to describe, cluster and categorize them, e.g. technical issues vs. interpersonal issues. Furthermore, we aim to form further hypotheses on how exactly development teams could best be supported with additional tools for process improvement.

III. RESEARCH METHODOLOGY

An interview study with professional Scrum Masters, agile coaches or other roles, tasked with process conformance, process guidance or process improvement within multiple companies in different contexts is performed, focusing on the executed process improvement process in the company. Surveys of professional development teams have shown that agile methodologies, particularly Scrum, are widely understood and employed in a multitude of projects [6]. The vast majority of teams practicing Scrum hold retrospective meetings after each sprint [5]. In fact, in the 2017 *State of Agile* survey, more respondents used agile project tools, such as task boards, than relied on spreadsheets for project management tasks [6]. The majority of recruited companies rely on employees in Scrum Master or agile coach roles to mentor and advise development teams on agile methodologies and their implementation. These individuals have deep process knowledge, often know multiple development teams, and understand the challenges of the company's context. Interviews with these individuals are most suited to gain insights into how a company or individual teams implement development processes. Interviews are performed in a half-structured manner in order to allow exploring beliefs and attitudes as well as observing non-verbal indicators [7] and allowing questions to be appropriately adapted to different companies contexts. The main goal of the interviews is to establish how process improvement is handled in the company by the interviewee and to what degree development data is being used or considered useful. However, interviewees are asked beforehand to collect, if available, the minutes or protocols of the last three meetings that process improvement was discussed in. After the interviewing researcher has introduced themselves and has stated the goal of the study, i.e. learning about the implementation of agile process improvement in the teams the interviewee is knowledgeable in, the following topics will be discussed:

- What is the general idea of process improvement in the company? Do teams actively implement these ideas?

- What is the agenda for process improvement meetings? What is regularly discussed? Can you share or summarize the minutes of these meetings?
- If action items for improvements are decided upon, how are these tracked over time?
- Are tools or frameworks for process conformance measurement employed? If so, which ones?

As in related literature, interviews are transcribed [8] and coded with topics in an iterative fashion. We analyze the frequency of topics mentioned in the interviews and build topic clusters by company, team age, and other factors dependent on interviewees. We proceed similarly with the identified action items.

IV. RELATED WORK

The field of software process improvement (SPI) provides work, which is related to this research [9]. Santana et al. point out, that organizing software process improvement activities in agile development environments is different than in more traditional ones [10]. In their systematic literature review, which included 423 papers, they conclude that it is necessary to adapt existing SPI approaches or to create new methods for agile contexts. Similarly, Kuhrmann et al. note that there is a growing interest in agile methods as well as adopting agile principles in software process improvement approaches [11]. However, in accordance with our own research to date, they note that research mostly focuses on student labs. The authors conducted a comprehensive systematic mapping study of publications related to SPI over the past 25 years. They identified agile processes as becoming more relevant for software process improvement as companies adopt agile methods as a major research direction.

V. SUMMARY

While the Scrum methodology is prescriptive regarding the roles, meetings, and artifacts involved with agile software development, it offers little actionable details regarding one of its main strengths, the focus on iterative process improvement. Much research has therefore focused on how to best manage and structure retrospectives, however, little research has been conducted targeting the outcomes of retrospectives and how identified action items can be tracked. As a first step in this direction, we have begun an interview study with those roles in companies, tasked with process improvement, conformance, and coaching. We aim at gaining an understanding of how process improvement is conducted in these teams, as a starting point for supporting teams in these activities. A core component of agile methodologies is the idea of self-organizing teams, i.e. teams that do not need management to give them structure and processes, but who create structures that work best for them [12]. Our vision is that this idea could also apply to software improvement processes, where teams can autonomously decide how their improvement approach should work but can take guidance and inspiration from research. In order to take first steps in this direction, the problem domain needs to first be better understood using empirical methods.

REFERENCES

- [1] K. Schwaber, "Scrum development process," in *Business Object Design and Implementation*. Springer, 1997, pp. 117–134.
- [2] H. Kniberg, *Scrum and XP from the Trenches*. C4Media, 2007.
- [3] K. Schwaber, *Agile Project Management with Scrum*. Microsoft Press, 2004, vol. 7.
- [4] K. Schwaber and J. Sutherland, "The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game," Tech. Rep., 2017. [Online]. Available: <http://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>
- [5] Scrum Alliance, "The State of Scrum Report 2017 Edition," Scrum Alliance, Tech. Rep., 2017. [Online]. Available: [https://www.scrumalliance.org/scrum/media/ScrumAllianceMedia/FilesandPDFs/StateofScrum/StateofScrum_{_}2016{_\]FINAL.pdf](https://www.scrumalliance.org/scrum/media/ScrumAllianceMedia/FilesandPDFs/StateofScrum/StateofScrum_{_}2016{_]FINAL.pdf)
- [6] VersionOne Inc., "The 11th Annual State of Agile Report," VersionOne Inc., Tech. Rep., 2017. [Online]. Available: <https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2>
- [7] K. Louise Barriball and A. While, "Collecting data using a semi-structured interview: a discussion paper," *Journal of Advanced Nursing*, vol. 19, no. 2, pp. 328–335, feb 1994. [Online]. Available: <https://doi.org/10.1111/j.1365-2648.1994.tb01088.x>
- [8] I. Richter, F. Raith, and M. Weber, "Problems in Agile Global Software Engineering Projects especially within Traditionally Organised Corporations," in *Proceedings of the Ninth International C* Conference on Computer Science & Software Engineering - C3S2E '16*. New York, New York, USA: ACM Press, 2016, pp. 33–43. [Online]. Available: <https://doi.org/10.1145/2948992.2949019>
- [9] S. Zahran, *Software Process Improvement: Practical Guidelines for Business Success*. Essex, UK, UK: Addison-Wesley Longman Ltd., 1998.
- [10] C. Santana, F. Queiroz, A. Vasconcelos, and C. Gusmao, "Software Process Improvement in Agile Software Development A Systematic Literature Review," in *2015 41st Euromicro Conference on Software Engineering and Advanced Applications*. IEEE, aug 2015, pp. 325–332. [Online]. Available: <https://doi.org/10.1109/SEAA.2015.82>
- [11] M. Kuhrmann, C. Konopka, P. Nellesmann, P. Diebold, and J. Münch, "Software process improvement: where is the evidence?: initial findings from a systematic mapping study," in *Proceedings of the 2015 International Conference on Software and System Process - ICSSP 2015*. New York, New York, USA: ACM Press, 2015, pp. 107–116. [Online]. Available: <https://doi.org/10.1145/2785592.2785600>
- [12] R. Hoda, J. Noble, and S. Marshall, "Organizing self-organizing teams," in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - ICSE '10*, vol. 1. New York, New York, USA: ACM Press, 2010, p. 285. [Online]. Available: <https://doi.org/10.1145/1806799.1806843>