Scaling Scrum in a Large Distributed Project

Maria Paasivaara, Casper Lassenius
Department of Computer Science and Engineering
Aalto University, School of Science
P.O. Box 19210, 00076 Aalto, Finland
Email: maria.paasivaara@aalto.fi, casper.lassenius@aalto.fi

Abstract—This paper presents a currently ongoing single case study on adopting and scaling Scrum in a large software development project distributed across four sites. The data was gathered by 19 semi-structured interviews of project personnel, including managers, architects, developers and testers. At the time of the interviews the project had grown in size during the past 2,5 years from two collocated Scrum teams to 20 teams located in four countries and employing over 170 persons. In this paper we first describe our research approach and goals. Then we briefly summarize the preliminary results of this ongoing study: we explain how Scrum practices were scaled, as well as discuss the successes and challenges experienced when adopting the agile practices and scaling them, while at the same time growing the project size at a fast pace. Even though this project has been very successful from the business point of view, it has experienced a lot of problems in applying Scrum, especially related to scaling the agile practices. Thus, it seems that adapting Scrum practices to work well in a large distributed setting is challenging.

Keywords—Scaling Scrum; agile software development; global software development;

I. INTRODUCTION

Agile methods, like Scrum and XP, were originally designed for use by single small teams, members of which are collocated, working face-to-face, preferably in team rooms. Nowadays, many companies developing large systems with multiple teams distributed to several geographical locations would like to reap the benefits of agile methods. Thus, it is necessary to scale the agile practices. Scaling involves several challenges, such as coordination between the teams, lack of architecture, lack of requirement analysis, as well as all the challenges of distributed projects [1]. Despite the challenges, several companies have already applied agile practices in large projects, and a few books (e.g. [1-2]) have been published describing how to scale agile methods. Even though very interesting, most of these books are written by consultants, who do not always provide much evidence to support their suggestions. Studies on how these scaling practices really work in practice, what kind of challenges there might be, and how to overcome the challenges are still rare [3]. A few case studies and experience reports on taking agile methods into use in projects involving several teams and even several geographical locations (e.g. [4-8]) do exist. However, most of these reports are from quite small projects involving only a few teams and the number of developers in a project is often under thirty. As a scaling practice for cross-team coordination and collaboration these studies often mention only Scrum-of-Scrum meetings, otherwise the agile practices used are normal team level agile practices. Research on experiences from larger projects consisting of ten or twenty or even more teams is currently very limited. Thus, more research is clearly needed on this topic.

This paper presents a currently ongoing study that aims to contribute to filling in that gap by describing a case study on adopting Scrum into use in a large globally distributed project. The studied project developing a new product decided to take Scrum into use already at project start up. Since then, the project has grown in size from two to twenty teams spread across four locations. In this paper we will describe what and how we are studying in this case project, as well as present some initial results.

The paper is structured as follows: Section 2 describes the research goals and methods, Section 3 presents some initial results, and Section 4 concludes the paper with a brief discussion.

II. METHODOLOGY

A. Case Study

This study is based on a qualitative multiple-case study approach [9]. Case study was chosen as a research method since we wanted to get an in-depth understanding on the adoption and scaling of Scrum practices in their real industrial environment. The studied case project was selected purposefully, i.e., by choosing an information rich case [10] as we encountered an extremely interesting project from a company participating in a joint research program and suggested a case study. As a result of our study the case company was hoping to gain understanding on how their Scrum adoption was progressing, what the successful practices were, and what kind of challenges and improvement opportunities there might be.

From the research point of view, in this study we focused on communication and coordination practices and challenges when adopting Scrum in a large globally distributed software development project.
B. Data Collection

We collected data using 19 semi-structured interviews. The researchers and case company representatives discussed the criteria for choosing interviewees. To get a good overview of the project, we first interviewed a few managers and an agile coach, who could give us a good picture of the project history, goals, growth, structure and the main practices used. The rest of the interviewees were selected from different roles, different Scrum teams and with differing length of experience in this project: Scrum Masters, developers, testers, line managers, area product owners and architects.

The interviews of the managers and an agile coach lasted 1.5–3 hours, while the rest of the interviews were shorter, taking 1–1.5 hours. The language of the interviews was either English or Finnish depending on the language skills of the interviewees. Both interviewers were native Finnish speakers.

The first round of interviews, altogether 16 interviews, was performed at the project main site, i.e., Finland. The second round of interviews, altogether 3 interviews, took place at one of the distributed sites, in Greece. One researcher performed all the interviews in Finland and the other researcher inter-viewed in Greece. The interviewer took detailed notes. In addition, all the interviews were recorded.

The interviews were relatively loosely structured and conversational in order to maintain adaptability to the roles and individual experiences of the employees in different roles. The interviewees were asked to describe their own experiences of the usage of Scrum, the Scrum practices used and the experienced successes and challenges in the project related to the usage of Scrum. In the interviews, the issues were dealt with both retrospectively and focusing on the current situation during the time of the interviews.

C. Data Analysis

All interviews were transcribed by a professional transcription company. The first round of analysis was made based on the interview notes. Based on this analysis categories for coding were created. The transcribed interviews were then coded using the created categories. Finally, the data was extracted from the coded interviews.

D. Validation

After the first interview round the results were presented to the case company in a feedback session to which all inter-viewed persons, as well as other interested project participants were invited. A lot of questions were asked and the audience was eager to discuss. The feedback session showed that taking Scrum into use in this large project was much more controversial than the researchers had realized during the interviews: even though a larger part of the project personnel found this change as a step towards better, there were still a few persons who clearly resisted the change. Thus, the discussion mainly focused on the change and whether taking Scrum into use was for better or worse. However, our results on the practices used, the success achieved and challenges faced were regarded, e.g. by the PO and an internal agile coach as valuable and no corrections or comments against the findings were presented.

III. Results

A. The Case Project

The case organization is a large, global product development company. Most of the product development projects it performs are globally distributed and involve both software and hardware development. Earlier, the company used a traditional stage gate, waterfall model. During recent years, the company has started to adopt agile methods for software development. The studied case project was one of the first large projects using Scrum. In this case study we discuss only software development, even though it was part of a larger program also involving hardware development.

The case project started early 2008. The adoption of agile methods was deemed to be easy since the product line and product were new, and there were no legacy products to be supported at the same time. The project started as a collocated project in Finland. In the beginning it had only around 15 developers in 2 teams that were formed by combining and mix-ing two agile teams from two previous projects. During the first couple of months these teams started to build the product while practicing how to work. Subsequently, a couple of more teams were added, and the members from the two original teams coached the new teams.

The first distributed site was added five months into the project by hiring over 40 developers organized into six teams at a subcontractor in India. The first Indian feature team came to Finland for hands-on product training, as well as for basic Scrum training. The company had resources with testing expertise in Germany, as well as in Greece, thus these sites were added when more resources were needed. The Greek site had tested the previous product, so the domain was familiar.

At the time of the interviews the project had grown during 2.5 years from two teams to 20 teams (altogether over 170 persons) located in four countries: Finland (10 development teams), Germany (2 testing teams), Greece (2 testing teams) and India (6 development teams). None of the teams were distributed and the work between the sites had been divided with a goal in mind that a whole specific area would be done at one site. Development activities were concentrated in the Finnish and the Indian sites, whereas the German and the Greek sites took care of testing.
B. Coordination Practices for Scaling Agile Development

In this section we describe how coordination mechanisms between the teams were scaled: Area Product Owners, Common Sprint Planning, Scrum-of-Scrums, Common Sprint Demo, and Common Retrospective.

1) Area Product Owners: Larman and Vodde [2] suggest using Area Product Owners (APOs) for scaling the product owner role, a practice adopted by our case project. In addition to the main PO role, the project had a team of APOs. Each APO was responsible for features in one specific product area and worked with the teams developing those features. The role of APO was divided between two persons: a system architect and a product management representative. At the time of the interviews, the project had around 10 APOs, each fulfilled by two persons. The system architect worked closely with the teams, as well as communicated with the product management representative, who did not communicate with the teams. At the main site, the system architect was located on the same floor as the teams, but the product management representatives were located in another building.

The idea was that each product area would have a couple of development teams, and each feature would be implemented by a specific team. However, at the time of the interviews the situation looked somewhat chaotic. Because of the tight schedule, some features were split between several teams that worked on them concurrently. The teams received work from several APOs for the same iteration. One APO even mentioned that at most he had been working with eight teams simultaneously. The APOs could suggest which team would be preferably for implementing each new feature, but the preferences could not always be realized, since the best teams were requested by several APOs.

The collaboration and communication between the APOs and the teams had been challenging and was not working well. Some improvements were implemented, e.g. the APO and the team were allowed to arrange a requirements workshop and a design workshop on each user story before the sprint planning, as needed. These workshops were considered very useful. The daily communication between the APOs and the teams was considered good enough according to our interviewees.

2) Common Sprint Planning: Common sprint planning was done in a short, one-hour, start-up meeting for sprint planning. The PO and program manager led the meeting and each team sent a representative. First, some general information was shared, e.g. the PO told news about the market situation. Then, the user stories for the sprint were briefly presented and assigned to the teams. Afterwards, the teams moved to team spaces for detailed planning. In the evening each team sent an email of their commitments to the PO and program manager. The program manager updated the backlog, i.e. an Excel sheet based upon these.

3) Scrum-of-Scrums: One representative from each team participated in the daily Scrum-of-Scrums. The participant reported the impediments the team has experienced. The challenge of this meeting was that many teams just stated nothing to report, even though that might not be true, because it is the easiest for them. Thus, many of our interviewees felt that these meetings were not very useful in their current state.

4) Common Sprint Demo: The project had tried several ways to arrange sprint demos. In the beginning, when the project had only a few teams, common sprint demos were held in a big auditorium for all teams. When the project grew, they created a new practice: the program manager and PO visited each team, and the team gave a demo to them and other interested. At the time of the interviews the project had moved back to having a common sprint demo meeting. But, instead of giving a real demo, a representative of each team gave a short slide presentation of the team achievements. The way of demoing was hugely criticized by the interviewed team members: they felt that a presentation does not tell whether the software is good or not and you can even cheat, tell that something works, even though there might still be problems. Moreover, in a large common session there is little possibility for discussion. Thus the teams feel that they did not get enough feedback. Even though these common demos were open to all, only a few team members, except those presenting, participated.

At the time of the interviews there were plans to move back to real demos in each team space. The audience of these team demos would include the APO, who has worked with that user story or feature, as well as other interested stakeholders. This arrangement would enable the APO to give the team detailed feedback.

5) Common Retrospective: The project had tried a few ways to arrange common retrospectives. In the beginning, each team had their team specific retrospective first, after which everybody gathered for a common retrospective. Each team was advised to bring three issues that are too big for their own team to solve to the common demo. Three of all those issues brought by the teams were then chosen for discussion and solutions were drafted together.

The problems were often very big, such as communication between the APOs and the teams is not working, thus they could not be solved during one iteration, and the implementation of the solution was not followed up. Thus, our interviewees felt that no real improvements followed from the common retrospectives. Huge problems remained, and common retrospectives were considered useless. Since participation was voluntary, the team members voted with their feet, and finally the meeting had only very few participants.

The next trial was to replace the common retrospective with an "open space, where anyone could suggest discussion topics. Several discussions took place simultaneously. Our
C. Challenges

The implementation of the scaling practices: area product owners, common sprint planning, Scrum-of-Scrums, common sprint demo and common retrospective were clearly not working well enough at the time of the interviews.

Many of our interviewees, especially team members, complained that the common meetings did not give a good big picture or help enough in coordination between the teams. For some it was not even clear why these meetings were arranged and they saw it more or less as a waste of time to participate. Some teams sent their SMs to these meetings, instead of a rotating team member, since they felt that team members had other more important tasks to do. Some team members even thought that these meetings were mainly arranged for the higher level managers, like PO and program manager so that these persons could follow what was happening in the project and that teams were reporting to them in these meetings.

On the other hand, the managers had a totally different view. They saw that the main purpose of these common meetings was to share information between the teams, so that the teams can notice early enough when there is a need to coordinate and communicate between the teams. According to the managers the other main goal of these meetings was to give a big picture on what is happening in different parts of the project to the teams.

The idea of product areas, in each of which a few teams would be working with the help of one APO was not a reality in this project. Due to time pressure, user stories from a single feature were often distributed for implementation to several teams, even to different countries. This would have required a lot of coordination between the teams. Common meetings were the only formal coordination mechanism, but it clearly did not work. Of course APOs tried to coordinate, but there did not seem to be any clear mechanisms for that. APOs worked with several teams at the same time, the teams could be located at different sites and receive user stories from several APOs for the same sprint, thus creating conflicts regarding priorities.

Even though the project had tried several different ways of implementing some of the scaling practices, none of them worked properly at the time of the interviews, all the scaling practices received quite a lot of critique.

D. Successes

Even though this project still had a lot of challenges both in implementing the basic Scrum practices and ideas, as well as in scaling the Scrum practices, most of the interviewees had only good things to say about this process change, when comparing it to the waterfall type of process model used in previous projects. They thought that without Scrum they would not have had anything to show to the first customers. Thus, they saw that the usage of Scrum was the main reason why they got these customers on board.

The project had created a quite well functioning organization structure. Especially the role of APOs was seen as very useful. The project had managed the very fast growth of the organization while at the same time developed the product and created new releases for the customers to test.

IV. Discussion and Conclusions

This case study shows that the practices for scaling Scrum to a large distributed project are not as easy to take into use in reality as they seem to when you read about them from a book. In this case project many interviewed team members had a quite negative attitude towards the scaling practices. They clearly did not feel that they could influence the project out-side their own team, e.g. by participating in the common retrospectives. Nor did they feel that it would be their responsibility to coordinate between the teams working with a same feature. Could the negative attitude come from their background, from the waterfall model when they could concentrate only on their very limited part of the project? Or is this just too big a project for a single team member to make any difference?

A. Limitations

We studied one case project and performed 19 interviews of project personnel coming only from two of the four sites. Due to time and traveling restrictions most of the interviews took place at the main site, which was also the biggest site with all managers and product experts at place. Thus, this site was not that dependent on communication with the other sites. The results might have been at least somewhat different if some other sites and cultures would have been involved.

B. Future Research

In the future we plan to continue this study by concentrating on the challenged practices. Moreover, we plan to broaden the study to the other sites of the case project to collect their experiences as well.
ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support of the Finnish Funding Agency for Technology and Innovation (Cloud Software Program). Our thanks go to the Case Company for making this study possible and especially to all inter-viewed persons.

REFERENCES


