Inter-team Coordination in Large-Scale Globally Distributed Scrum: Do Scrum-of-Scrums Really Work?

Maria Paasivaara
Aalto University
P.O. Box 15400
00076 Aalto, Finland
maria.paasivaara@aalto.fi

Casper Lassenius
Aalto University
P.O. Box 15400
00076 Aalto, Finland
casper.lassenius@aalto.fi

Ville T. Heikkilä
Aalto University
P.O. Box 15400
00076 Aalto, Finland
ville.heikkila@soberit.hut.fi

ABSTRACT

Scrum-of-Scrums meeting is mentioned in the literature as the mechanism for handling inter-team coordination in large-scale Scrum. However, how to implement it in projects with tens of teams is not explained. In this paper, we present a multiple case study on how Scrum-of-Scrum meetings were applied in two large-scale, globally distributed Scrum projects both employing at least twenty Scrum teams. We conducted 58 semi-structured interviews of project personnel, including managers, architects, product owners, developers and testers. Our results show that Scrum-of-Scrum meetings involving representatives from all teams were severely challenged: the audience was too wide to keep everybody interested and the participants did not know what to report that might be valuable to other teams, often ending up not reporting anything. As a solution, one of the case projects introduced feature-specific Scrum-of-Scrums meetings for 3-5 teams working on the same feature, which turned out to work well. However, challenges with coordination at the project level remained. The other case organization tried a site-based SoS structure that still did not work well.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management

Keywords

Agile Software Development; Distributed Scrum; Global Software Engineering; Inter-team Coordination

1. INTRODUCTION

Agile methods were originally designed for small collocated teams. Due to their popularity, they are nowadays also applied by large companies in large software development projects employing multiple teams that are distributed to several geographical locations. Scaling agile methods to this new context introduces new challenges, such as inter-team coordination, distribution of work without a defined architecture or properly defined requirements, as well as all the challenges of distributed projects [6]. Despite the challenges, companies have applied agile practices in large projects [5, 7]. While a few case studies and experience reports on taking agile methods into use in projects involving several teams and several geographical locations do exist, most are from small projects involving only a few teams and the number of developers in a project is often under thirty.

Practitioner literature contains some advice on scaling, [4, 6], but 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].

In this paper we focus on scaling practices for Scrum [11], which is one of the most widely adopted agile methods. For inter-team coordination and collaboration, the Scrum literature proposes and reports on the use of so-called Scrum-of-Scrum (SoS) meetings [11], but does not discuss how the SoS meetings are applied nor their benefits or challenges.

Research on experiences from larger projects consisting of more than ten teams is currently limited, and thus how to handle inter-team coordination in-the-large in globally distributed Scrum is not well understood. In this paper, we help shed some light on this by presenting a study on how Scrum-of-Scrums meetings were applied in two large distributed Scrum projects.

The paper is structured as follows: Section 2 presents the previous work, Section 3 describes the research goals and methods, Section 4 presents the results, and finally Section 5 discusses the results and presents our conclusions.

2. PREVIOUS WORK

The *Scrum-of-Scrums* meeting (SoS) is basically the only practice Scrum offers for inter-team coordination [10]. Recommendations for how to arrange SoS meetings can be found in the practitioner literature [2, 4, 11]. The basic format of the SoS resembles the Daily Scrum meeting, except that it deals with teams instead of team members. It is recommended that the SoS be arranged daily [10], or 2–3 times a week [2].

It has been suggested that the three questions answered in Daily Scrums [10] could be changed slightly to keep the SoS meeting interesting and effective [4]: 1) What did your team do since the previous meeting that is relevant to some other team? 2) What will your team do by the next meeting that is relevant to other teams? 3) What obstacles does your team have that affect other teams or require help from them? A fourth question to facilitate future coordination might be added: 4) Are you about to put something in another team's way? [2]

The SoS meeting is suggested to be time-boxed to last a maximum of 15 minutes, just like the Daily Scrums [4]. Cohn [2] suggests booking 30–60 minutes for SoS meetings in the calendar, so that if big problems show up at the meeting they could be discussed and solved right away, when all relevant people are present. An-

*ESEM'*12, September 19-20, 2012, Lund, Sweden.

other suggestion is to arrange self-organized follow-up meetings after the strictly timeboxed SoS [4]. Each team sends their representative to the SoS meeting. This member is normally rotating, either every time or after a couple of iterations. However, sending a Scrum Master (SM) is considered a bad practice, since it easily leads the Scrum Master role to slip towards that of a traditional project manager [4]. Another challenge of the SoS meeting is not make it into a status reporting meeting for management, but to keep it as a synchronization meeting between the teams [4].

Experiences reported in scientific articles of the usage of SoS meetings are very limited, and most of the articles discuss the usage of SoS meetings in projects with only a few Scrum teams [1, 14, 13]. We found only three research articles briefly reporting experiences from projects over ten Scrum teams [5, 8, 12].

For scaling up the SoS meetings when having a lot of teams, nested Scrum meetings, i.e., Scrum-of-Scrum-of-Scrum meetings (SoSoS) have been suggested [2]. For example, one could arrange normal SoS meetings with representatives from seven Scrum teams, after which each SoS meeting sends their representative to a higher level SoSoS meeting. However, whether this suggested structure has worked anywhere in practice is not reported in the paper describing it [2]. Also Schwaber [11] suggests that SoS meetings may be arranged in large organizations at multiple levels, with progressively higher levels of staff meeting less frequently. A case study presents a project with 14 teams, distributed to four time-zones that had scaled their daily SoS meetings to a two-level nested structure, where they had multiple daily SoS meetings led by senior SMs, who then reported in the project SoSoS [12]. The biggest challenge was the information flow up and down the nested structure. Another case study reports the usage of weekly two-level nested SoS meetings in a project consisting of 21 teams, and mention the weekly SoS as the best way to mitigate the problem of having two teams solving the same problem that a third team had already solved [5].

As an alternative to SoS meetings, consultants report on suggesting arranging a weekly Open Space to a client with 30 teams that was dissatisfied with the SoS. However, the authors do not report whether the client was more satisfied with the Open Space than the SoS [4].

As neither the scientific articles nor the practitioner literature report detailed experiences on using SoS meetings, there seems to be a clear gap in the literature that we hope to partly bridge with this multiple case study.

3. METHODOLOGY

3.1 Research Goals and Data Collection

For this multiple case study, which is part of a larger study on adopting Scrum in-the-large, the goal was to understand how the case organizations handled inter-team coordination in large-scale distributed Scrum. We purposefully selected two information-rich case projects [9] from two companies participating in a joint research program. Both projects were globally distributed and were developing similar large telecom infrastructure systems, consisting of both software and hardware. Our study was purely focussed on software development. We collected the data using 58 semi-structured interviews, see Table 1. We selected the interviewees jointly with management at the case organizations, aiming for people with different length of experience and working in different roles.

All interviews were recorded. In addition, one researcher took detailed notes. 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.

Table 1: Case Projects and Data Collection

	Case A	Case B
Product	Telecommunications, started from scratch	Telecommunications, 10-years old
Process	Scrum	Incremental change from waterfall to Scrum
Scrum experience	2,5 years	1,5 years
Sites and # of teams	Finland (10 dev. teams), India (6 dev. teams), Germany (2 test teams), Greece (2 test teams)	Finland (18 teams), Hungary (7 teams)
Interviews	19 (Finland 16, Greece 3)	39 (Finland 28, Hungary 11)
Roles[a]	Managers (4), Agile coach (1), Scrum Master (1), Developers (5), Testers (2), Line managers (2), Area Product Owners (4), Architects (1)	Managers (6), Agile coach (1), Scrum Masters (6), Team members (13), Line managers (3), Product/Proxy product owners (7), Technical management / architecture (5)
Interview length	Managers, coach: 1.5–3h, others 1–1.5h	Managers, coach: 2–3h, others 1–2h

[a]The sum exceeds the total number of interviews, as some line managers had double roles, e.g. also worked as Scrum Masters

In this paper, we focus on the experiences of inter-team coordination, for which both case organizations used Scrum-of-Scrum meetings. The research questions we aim to answer are:

1. How have Scrum-of-Scrums meetings been applied in large-scale distributed Scrum projects?
2. What are the benefits and challenges of Scrum-of-Scrums meetings?

3.2 Data Analysis

All interviews were transcribed by a professional transcription company. While waiting for the transcriptions, we did a first round of coding based on the interview notes, creating preliminary categories for coding of the transcripts. We then coded the transcribed interviews in Atlas.ti using the preliminary categories, adding a few more while coding. Two researchers did this coding together, agreeing on and discussing codes while coding and using the same hermeneutic unit. For this paper, we extracted the codes and related quotations for “SoS”, “SoS Grande”, “SoS Feature”, as well as combined them with the codes “Problem” and “Positive” to find out what were seen as problematic or positive in Scrum-of-Scrum meetings.

3.3 Validation

We validated our findings by presenting them to both case companies in feedback sessions to which all interviewed persons were invited. During the sessions, questions were asked and the audience was eager to discuss. The companies found the feedback valuable, and while, e.g., management challenged some of our (less flattering) findings, personnel in other roles confirmed them, and no corrections to our findings came out of the sessions.

4. RESULTS

4.1 Scrum-of-Scrums in Case A

Case project A, developing a telecom product from scratch using Scrum from the outset, had grown from 2 teams to 20 teams distributed to four sites in 2,5 years. Teams were site-specific. Initially, while located at a single site in Finland, SoS meetings were held face-to-face daily, with one representative from each team present. The teams decided themselves who would represent them in the SoS meetings, using e.g. a round-robin approach. When teams from a second site (India) were added to the project, initially the project held daily project-wide SoS meetings using teleconferencing, using the same principle for participation.

However, these meetings were not considered to work properly, and were replaced by a structure consisting of two separate meetings: a *Finnish SoS* followed by a *Global SoS*, both led by the Finnish project manager. The project manager became a communications bridgehead, as he was the only one participating in both meetings. The project manager created a short memo in each meeting, and sent it to all project personnel by email. Later, when even more teams were added from two new sites, they also participated in the Global SoS, that now had team representatives from three sites, as well as the project manager from the fourth site.

Initially, the representative of each team answered the four SoS questions on the behalf of his or her team. However, when the number of teams grew larger, the meetings took longer and longer and the teams gave feedback in the retrospective that they were not that interested in what the other teams were doing. As a result, the first questions were left out, and teams only report if they have impediments or plan to put impediments in the way of other teams.

This has led to a situation in which the meetings are short — 5–15 minutes — and many teams report "No problems", which is not always entirely true, but the result of assuming that the other teams do not have to know or are not interested in their problems.

> *"...it would be good if people would really tell about the problems there. Sometimes it feels like everybody just says 'No problems', that everything is going ok, but later on comes up that this and this does not work. (...) And many are fighting with the same problem at the same time." — Tester*

Another problem and one of the reasons that the teams did not report problems seemed to be that they did not feel that they would receive help in the meeting:

> *"Maybe part of the reason is that in general you don't find solutions from there anyway. Now we are scattered around the world, so we don't have an absolute Scrum of Scrums." — Developer*

In Case A, managers admitted that inter-team communication and the SoS meetings did not work properly, and that they did not know how to improve it. The majority of the interviewed team members saw the SoS meetings as poor or even useless. Despite this fact, when we recently visited the company again, there was still no improvement to the situation.

4.2 Scrum-of-Scrums in Case B

Case project B consisted of 25 site-specific Scrum teams distributed to two sites, developing a 10-year old telecommunications product. The organization had started a transformation from waterfall to Scrum 1,5 years ago, and at the time of the interviews all software development teams were using Scrum.

In this project, SoS meetings were initially held using videoconferencing three times a week. Each team send a representative of their choice, most teams opting for a round-robin model. In this meeting, each team could report what they found important to share with the other teams. The problem was that the teams did not seem to know what to share, thus often ending up just reporting "Nothing to share". As a result, participants did not find these meetings useful, and some of our interviewees even claimed that it was a waste of time.

Therefore, after a while the project added another set of meetings called *Feature SoS* meetings. These were held by 3–5 teams working on the same feature[1]. The Feature SoS meetings took place once a week. In addition, the project still arranged the project wide SoS meeting once a week, which now was called the *Grande SoS*.

All interviewees found the Feature SoS meetings useful, since there a small group of people with common interests and goals could share, discuss and even solve problems together:

> *"This [Feature SoS] is a good meeting, since this is the only place where we are all together at the same time (...) Here we can discuss everything. We have tried to keep it this way, that we don't have agenda, but discuss what is done at different teams, if there are any problems or other common topics." — Proxy Product Owner*

> *"Feature SoS meetings are pretty good, because people there do the same things, talk "the same language" and have a common goal." — Proxy Product Owner*

However, the Grande SoS meetings were seen as problematic according to most interviewed persons, as evidenced by the following quotations:

> *But the [Grande] SoS meetings between features, it's very difficult to see the added value, because people do not talk about the same things, it just doesn't work, it's too big." — Proxy Product Owner*

> *"It [Grande SoS] has not worked well. (...) We are still looking for the role of this meeting and things that should be discussed there." — Developer*

Participants of the Grande SoS meetings did not clearly know what to share with each other, they were not interested in what others were doing and sometimes could not even understand each other's problems, since technologies used in different parts of the product differed. On the other hand, another frustration was that the weekly rhythm for the meeting was too slow for dealing with real problem situations requiring daily communication between the teams.

> *"I think that teams have not recognized the things that would be really interesting and useful to share with others. (...) And the features are quite separate from each other, and teams have done things that are internal to one feature and thus felt that it might not be interesting to other features. There just have not been enough connecting points." — Scrum Master*

5. DISCUSSION AND CONCLUSIONS

Our results can be summarized as two main findings: SoS meetings seem to work poorly when they have too many participants with disjoint interests and concerns; and smaller, focussed inter-team meetings with participants having joint goals and interests, seem to have a better chance of being perceived as successful.

Both case projects started by applying the SoS meetings "by the book", as advised by the agile consultants they employed — "This is the Scrum way". Doing it this way, they answered all the 3-4

[1]In this context, a *feature* is a big development task, comprising several epics, often involving several teams and taking several months to develop.

SoS questions and involved representatives from all teams in this common meeting for the whole project.

However, when the number of teams grows, it is clear that all 20 team representatives cannot answer all the Scrum questions in the 15 minutes allotted for the meeting. Instead of lengthening the meeting, both case projects started using a model in which only one issue was discussed: impediments. However, this solution did not turn out well — people did not know what to report, were not interested in what teams working on other parts of the product were doing, or could not understand the other areas if they were discussed.

To help alleviate the problems with the project-wide SoS, both case projects adopted a kind of multiple-level model. Case A used a site-and-size based approach, having one daily face-to-face SoS in Finland, where half of the teams were located; and another daily SoS by teleconferencing involving the Finnish project manager and the teams from the other sites. Case B used a model in which the weekly project-wide "Grande SoS" was augmented with weekly feature-specific SoS meetings. Of these two models, quite logically, the content/architecture-based model rather than the geographically-based one seemed to work much better. However, interestingly, both case projects still recognized the need for project-wide inter-team synchronization, but did not have any good solutions to the problem.

People in both projects had a history of working in waterfall type development, in which global visibility was neither needed nor required from the developers. However, in Scrum everybody should care about the whole product, and according to Scrum, teams are responsible for inter-team coordination. Maybe the change of mindset was not yet successful.

One could also argue that our case projects just did not practice the SoS meetings long enough to get them working properly. However, both projects had already used and improved this practice for a quite a long time and still kept trying. Both projects were all the time looking for improvements.

The literature suggests multi-level SoS meetings, but in reality there are almost no reported experiences on how they should be arranged nor whether they really work. In this kind of nested structure literature mentions information flow up and down the structure as a challenge. Our projects had challenges already with this very low hierarchy structure: What to report in a SoS meeting? What to report back to the team? The team representative often “forgot” to report back to his team or did not find anything important to report back.

Our overall findings—at least in retrospect—seem more or less obvious, and we were frankly a bit surprised at getting such results from our case companies. Both companies are "best in class" in their own field, and have a long history of successful software development. In addition, Case A was one of the organizations the consultants writing one well-known book on how to scale agile worked with. Despite this, the organization had serious issues (among other things) with inter-team coordination, and expressed frustration that neither consultants nor the literature really had provided any real help. This gives an indication that claims in the practitioner literature might be exaggerated and solutions over-simplified.

5.1 Limitations

The paper is based on only two case projects from two organizations, which limits the generalizability of the results. The results are based interviews during which we collected data on many topics, SoS practices being only one of the them. Thus, we could have received even more detailed information if we could have put more emphasis on SoS meetings. Unfortunately we have not yet had the chance of observing the actual SoS meetings in the organizations.

5.2 Future Work

More empirical research is needed on how to tackle inter-team coordination in large-scale agile projects. We plan to follow our case organizations to see how their SoS practices evolve, collecting more detailed data about their SoS meetings by observations, analyzing the meeting notes, as well as by additional interviews. We are also interested in performing additional case studies.

6. REFERENCES

[1] P. L. Bannerman, E. Hossain, and R. Jeffery. Scrum practice mitigation of global software development coordination challenges: A distinctive advantage? *Hawaii International Conference on System Sciences*, 0:5309–5318, 2012.

[2] M. Cohn. Advice on conducting the scrum-of-scrums meeting., May 2007.

[3] E. Hossain, M. A. Babar, and H.-y. Paik. Using scrum in global software development: A systematic literature review. In *Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering*, ICGSE ’09, pages 175–184, Washington, DC, USA, 2009. IEEE Computer Society.

[4] C. Larman and B. Vodde. *Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum*. Addison-Wesley Professional, Boston, MA, USA, 2010.

[5] E. C. Lee. Forming to performing: Transitioning large-scale project into agile. In *Proceedings of the Agile 2008*, AGILE ’08, pages 106–111, Washington, DC, USA, 2008. IEEE Computer Society.

[6] D. Leffingwell. *Scaling Software Agility: Best Practices for Large Enterprises*. Addison-Wesley Professional, 2007.

[7] M. Paasivaara, S. Durasiewicz, and C. Lassenius. Using scrum in distributed agile development: a multiple case study. In *Proceedings - 2009 4th IEEE International Conference on Global Software Engineering, ICGSE 2009*, pages 195–204, 2009.

[8] M. Paasivaara and C. Lassenius. Scaling scrum in a large distributed project. In *Proceedings - 2011 5th International Symposium on Empirical Software Engineering and Measurement, ESEM 2011*, pages 363–367, 2011.

[9] M. Q. Patton. *Qualitative evaluation and research methods*. Sage Publications, Newbury Park, Calif., 2nd edition, 1990.

[10] K. Schwaber. *Agile Project Management with Scrum*. Microsoft Press, Redmond, Washington, USA, 2004.

[11] K. Schwaber. *The Enterprise and Scrum*. Microsoft Press, Redmond, Washington, USA, 2007.

[12] H. Smits and G. Pshigoda. Implementing scrum in a distributed software development organization. In *Proceedings of the AGILE 2007*, AGILE ’07, pages 371–375, Washington, DC, USA, 2007. IEEE Computer Society.

[13] J. Sutherland, G. Schoonheim, and M. Rijk. Fully distributed scrum: Replicating local productivity and quality with offshore teams. *Hawaii International Conference on System Sciences*, 0:1–8, 2009.

[14] J. Sutherland, G. Schoonheim, E. Rustenburg, and M. Rijk. Fully distributed scrum: The secret sauce for hyperproductive offshored development teams. In *Proceedings of the Agile 2008*, AGILE ’08, pages 339–344, Washington, DC, USA, 2008. IEEE Computer Society.