

DISTRIBUTED SCRUM WHEN TURNING INTO MAINTENANCE: A SINGLE CASE STUDY

Minna Kilpala¹ and Tommi Kärkkäinen²

¹Tieto Finland Oy, Mattilanniemi 6, 40100 Jyväskylä, Finland
minna.kilpala@tieto.com

²University of Jyväskylä, Department of Mathematical Information Technology,
P.O. Box 35, 40014 University of Jyväskylä, Finland
tommi.karkkainen@jyu.fi

ABSTRACT

Global software development using agile methods is commonplace in software industry nowadays. Scrum, as the agile development management framework, can be distributed in many ways, especially concerning how the key roles are presented in different sites. We describe here a single case study of a distributed Scrum, mainly for maintenance of the already constructed web portal. Using a qualitative method, both working well and challenging parts of the software work, as experienced by the project stakeholders, are revealed and discussed.

KEYWORDS

Scrum, software maintenance, case study, agile project management

1. INTRODUCTION

Software industry always tries to increase productivity, i.e., to produce larger amount of better software more efficiently and with reduced costs. This has not been or will not be easy; many reports and assessments continue to illustrate large portions of failed or cancelled projects. During the last decades, globalization, agility, versatility of development tools and computing platforms, domain specificity (in models, languages, and application areas), more systematic reuse in the form of product lines etc. have been affecting the ways to realize the software development efforts. But, old wisdom [1] still applies, because complexity of software entities, their conformity requirements, pressures of change, and invisibility within the distributed and hyperlinked technological environment are constantly increasing, reinforcing the unavoidable essential complexity of software development.

Globally distributed software development (i.e., Global Software Development, GSD) as a cost-effective way for software companies to minimize distances and jump the time zones for 24/7 production is commonplace. Reduced costs but higher flexibility and adaptability in this context are searched for by incorporating agile methods. However, the tension between the formal contracts and their plan-driven fulfillment, and adaptively and independently in a single-site, with direct user-interaction, working agile team, is evident.

The agile development model studied in this work is based on Scrum [2], which is more an agile project management framework than a detailed depiction of practices a la XP. Scrum's main roles are Scrum Master (SM) linking the customers and the development team and also protecting the team from external interruptions during the development cycles. Product Owner (PO) manages

Natarajan Meghanathan et al. (Eds) : COSIT, SEC, SIGL, AIAPP - 2015
pp. 55–67, 2015. © CS & IT-CSCP 2015

DOI : 10.5121/csit.2015.50606

the Product Backlog (PB) describing the features to be included in the software product. The software is constructed incrementally in small iterations called sprints, where the Sprint Backlog (SB) describes the subset of PBs to be realized. User stories are mainly (but not necessarily) used to describe the features in the backlogs in a semi-structured manner.

We located one literature review which summarized the used practices of Scrum in GSD projects [3]. From the review one notices that most studies in the research area are industrial reports and a knowledge gap to have sufficient amount of actual scientific information seems to exist. The actual phase of software lifecycle being studied here is related to software maintenance, whose needs are classically depicted by the famous Lehman laws [4]. More precisely, in our case the distributed team is correcting and modifying a web-based information system related to a long-lasting customer relationship with a stable PO, being involved in corrective, adaptive, and perfective maintenance [5].

Agile methods and software maintenance appears to be a scarcely studied combination. There are apparent needs for tailoring and modifications in aligning agility and maintenance: on the one side, the spirit of agility is, in principle, related to developing from the scratch without a need to comprehend an existing software system; on the other hand, small sprints and continuous integration are clearly activities suitable and valuable in maintenance as well. Moreover, both adaptive (e.g., due to changed qualitative requirements) and perfective (new user requirement from PB) forms of maintenance are incremental in nature. Actually [6] provide a pertinent characterization of the interplay between agile project and maintenance activities: “Maintenance can be seen as an endless agile project whose product backlog is changing constantly”. The same authors, importantly, also point out the existence of Change request – change management type of activity resulting from a customer as PO mastering the Product Backlog. In [6] the overall analysis along the change management direction, provided recommendations on balancing the customer disturbance and agility, and being more involved in the customer’s planning process on aligning business and software changes.

The empirical focus of this paper is on one organization’s one individual project. Hence, we address a single case study well characterized by the definition of a case study as provided in [7, p. 13]: “an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident.”

Goal of this paper is to find out if distributed Scrum can be used in software maintenance and if there are any benefits or disadvantages when doing so. Additional goal for IT Company where this study was done is to find out how this kind of work model can be improved. Research data was collected by semi-structured interviews from most of active project participants or stakeholders. Background information of case project is described by its project manager who is also another author in this study.

After this introduction we will present some studies close to our interesting area in this study as related work. From related work we will move to present our research methods and describe our study case. Study results are described after that. Eventually in discussion part our findings are compared to findings from related work with some ideas why we agree, disagree or can’t say anything.

2. RELATED WORK

GSD, agile software development, and their various combinations have been studied a lot during the last decade. Without any attempt of providing a thorough coverage, we summarize next some

key findings from a set of existing research. First, the distributed agile in general and Scrum in GSD are addressed. Then, the few works that were identified on agile maintenance are recapitulated.

2.1. Globally Distributed Software Development

An experience report from My Yahoo! on *adopting agile methods in globally distributed software development* was provided by [8]. Their distribution strategy was based on a hierarchy of Scrum Masters: Chief SM in the global product team, regional SMs in the regional teams, and local SMs in the local teams. Initially the following challenges were revealed: i) cultural differences in communication (reluctance to discuss negative issues), ii) disjointed and conflicting priorities within different products of the same company, and iii) team bonding and distributed collaboration with few or no face-to-face interaction. As a whole, using a refined form of distributed Scrum, roughly 1.5 increase in productivity was obtained (FPs/dev months) with the reduction of 64% of significant bugs.

Successful modifications of Scrum practices in GSD have been described in [9], [10], [11], [12] and [2]. In [2] Paasivaara et al. provide the following list: i) tool support for daily Scrums and Backlog management, ii) synchronized 4-week sprints (and 2-week sprints by the maintenance team to provide hot fixes), and iii) weekly Scrum-of-Scrums (SoS) joining together all POs (in one site) and SMs (in two sites). In addition, i) needs for multifaceted communication, ii) joint physical presence in the critical project phases, and iii) support of POs using domain expert network were nominated as the main external practices supporting global development actions. Challenges faced were related to bad network connections, cultural differences in daily scrums, and misunderstandings between POs and developers. However, the overall experiences turned positive, because improvements in communication, trust, motivation, and product quality were documented. In a follow-up, multiple case study, the same authors [13] concluded the following further lessons (cf. [14]): i) distribute for benefits not for fashion, ii) focus on Scrum training and open communication, iii) arrange frequent visits between the sites, and iv) provide and utilize multiple communication tools especially for informal communication.

In a longitude case study of agile distributed project in Denmark and India [9], reasons why Scrum works were concluded with nine answers (summarized here in four principles): i) it can build relations, networks, trust, common language and shared target for the team, has good meeting structure for team communication and gives energy and motivation for the team, ii) it is useful in work coordination and progress tracking, iii) it creates boundary objects and boundary spanner roles, and iv) it has a simple quality assurance mechanism.

In [10] and [11] Sutherland et al. concluded that it is possible for distributed Scrum to be as productive or achieve the same velocity and quality as with a collocated team. In [12] Sutherland et al. list advantages: i) it can reproduce performance of collocated Scrum, ii) clear communication via meetings that are facilitated by Scrum, iii) high quality by applying XP practices, and iv) accurate and transparent estimation. Challenges faced in their study were: i) cultural differences, ii) sharing context and priorities, iii) managing customers with no agile experience, and iv) special local requirements like documentation on a customer language.

Concerning the positive and measurable improvements in the projects when applying and tuning distributed agile as mentioned above, very different conclusions were provided in [14] by analyzing 66 European software development projects. Their results concluded “no statistically significant correlations between variables measuring outcomes of distributed processes and the type of development project followed—agile or structured”. However, identification of effective practices (from both worlds) remained as further study, because many process clusters with mixed practices were established as part of the quantitative analysis.

2.2. Agile Maintenance

In [15] the tailoring of Scrum in a single site – single organization *maintenance* situation was studied using the action research approach. The authors point out many differences between (forward) development and maintenance, noting, for example, that maintenance sprints can be subject to interruptions by urgent customer demands. Nine heuristics were suggested for agile maintenance (summarized here in five principles): i) balance sprints for multiple customer portfolio, ii) allow unexpected sprint changes, iii) emphasize knowledge sharing in the team, iv) balance face-to-face communication and documentation, and v) find other team motivators to compensate the lack of “now the software product is ready” in maintenance.

A Case study on a medium-sized Norwegian software company having adopted Evo (‘Evolutionary Delivery’ by Tom Gilb, close to Scrum) in maintenance was provided by [16]. Their findings concerning the experienced challenges were related to i) comprehension of the existing system, ii) effects of small change to many components (i.e., low cohesion), iii) unstable testing and lack of test coverage, and iv) need of a guru in the team, being both technical and domain knowledge champion. In conclusion, based on [16] and [15] on agility and maintenance, one notices that many original values of Scrum were abandoned when doing maintenance, e.g., related to self-organizing teams and frozen sprint Backlog without external interruptions.

3. RESEARCH METHOD

We use case study [7] as the research method. Suitable case was decided on two principles: We wanted to have globally distributed project which was already using Scrum practices and we wanted to collect experiences which could be later used to improve performance of the corresponding projects. The selected project under study is a maintenance project including also new development (i.e., perfective maintenance). Project maintains and develops company’s external web portal. The web portal is composed of main sites with several, different level country specific pages, and it is integrated to several different services which bring content to the pages. Some of integrations have legal requirements behind them.

The data was collected by semi-structured interviews of the project participants and stakeholders. The interviews were held via Lync and each interview was recorded. During each interview, notes were created on the fly and attendants were able to comment and correct them immediately. There were totally 12 interviews including all project participants and most participating stakeholders. The studied case project had also project manager but she was not interviewed because she was the one doing the study.

Interviews contained nine questions of which six were background questions, two larger, main questions, and one extra question. In the main questions, issues working well and needs of improvement were gathered and, in both questions, the interviewer encouraged participants to think all possible aspects related to the theme.

Interview data was then qualitatively analyzed. First each answer in the notes was coded and different answers were collected below the same main topic. For background and extra questions, only summarizing averages were calculated. When the open coding of cites from the interview notes was transformed into different levels of abstraction in axial coding, the categorizations of the two main themes were iterated around five times between the two authors commenting and proposing changes to one another’s category labels. Finally, the axial coding was stabilized to three different levels.

In the qualitative analysis of works well and improvement issues, it was counted how many times each issue was uniquely mentioned by an interviewee. When the preliminary results were obtained, a meeting was arranged with the team where the participants were able to comment the raised issues. In the second part of the meeting, three improvement needs were voted for further study and actions how to improve the activities in them were planned. Here, however, we only report the main findings from the two main issues leaving analysis of the planned changes for further studies.

3.1. Research question

Research question for this study was: Is it possible to use Scrum in globally distributed maintenance project and, if yes, how it is implemented and what benefits and disadvantages it brings? Research question will be answered with interviews. Regarding benefits and disadvantages we will also compare our results to other studies.

Interview questions were:

1. Your role and location in project?
2. When you joined team?
3. Your previous experience of Scrum?
4. Your previous experience of globally distributed projects?
5. Your previous experience of Drupal?
6. Your previous experience of other web development?
7. What is working well in project generally / personally?
8. What you'd like to improve in project generally / personally?
9. If you have questions or problems related to your responsibilities, who do you ask for help?

3.2. Case Description

This case study is done on web maintenance and development project in global IT Company. Company's main area is North Europe, but it has offsite activities in several locations like in India. Both agile methods and global delivery model (GDM, term used for GSD in company) are in constant use in company and it is common to have both in same project. Same trend was also visible in the background questions; almost all participants had more Scrum or GSD experience than just this project. In table below average experience in years from background questions are presented.

Table 1. Average experiences in background questions

Question	Average (years)
Time in team	1.5
Scrum experience	2.6
Experience in globally distributed projects	3.7
Drupal experience	2.3
Web development experience	9.4

Web development and maintenance project had been on going about a year when this study was done and, before that, there was a development project which originally created the web portal. Between the development project and the project in a current form, there was a clear change of responsibilities from development to maintenance where also most of the project team members were changed. Average time in the project was 1.5 years while shortest time was 0.8 and longest time was 2.3 years.

The Web site was developed with Drupal which is open source Web Content Management System software. Half of participants had no earlier experience of Drupal but everyone had at least two years of experience at the web development.

The Scrum team was located in the three different offices: PO and one team member in Stockholm, Sweden, SM and two team members in Helsinki, Finland, and four team members in India, Pune. Other stakeholders were located in Helsinki and Stockholm and Project Manager in Jyväskylä, Finland.

When asking project members about their role in the project, there were actually ten different roles mentioned of which two were for the other stakeholders. Scrum team had an architect, test engineer, senior developer, Front End developer, and UX designer, in addition to usual Scrum roles of SM, PO, and a team member. Each team member had one role.

All scrum roles took care of their duties according to Scrum principles with exception that team members focused on own strength areas instead of everyone doing everything inside team. In addition to those project manager took care of additional management issues like handling project in company's ERP system, following project budget, communicating with project stakeholders and searching for new team members when needed.

The project applied sprints of two weeks which started with time limited sprint planning and ended in time limited sprint reviews. There were two grooming meetings each week which were used for presenting new issues in PB and providing story points for them. They weren't always needed, but they were reserved in beforehand anyway. Retrospectives were held regularly after each sprint. After the sprint reviews, there were bug hunts of which the whole team participated. On every second Monday, after last Wednesday review, all changes were deployed to production. All above team meetings including daily scrums were held via video meetings with Hangout and each team member had a web cam. All meetings were distributed and everyone was participating from own computer in the office, some Finnish and Swedish participants occasionally also from other places like home. Hangout was also used for smaller meetings and phone calls, whenever there was need to discuss or share screen to show something and threshold to start Hangout meeting was quite low.

All participants were encouraged to discuss and ask questions in the meetings and Flowdock as chat tool was in very active use in project. Flowdock was used for daily communication, general information sharing and for asking more details of user stories currently in development. It is remarkable how well communication succeed in project when there haven't been any visits to meet other participants face-to-face.

Project didn't have any special project room or corresponding even if most of Pune and Helsinki participants were located close to each other. Project status (PB and SB) was updated continuously in Jira task tool and its Scrum Board was used to visualize sprint status and tasks. Task statuses visible in Scrum Board where: to do, in progress and done. There were also special Jira dashboards which collected generally or role based most important issues of current sprint like one for PO which included showing all issues waiting for her approval and sprint burndown chart.

Project had also shared integration test environment where all changes were committed after peer code review. Continuous flow of actions in project code base was shown in Flowdock.

4. RESULTS

Regarding to our research question we were able to confirm that it is possible to use Scrum in globally distributed maintenance project. Project we interviewed was fully functional and no lack in service level was reported. We were able to catch more improvement needs than working well issues, but none of them were so severe that it would have prevented project work. We were able to identify some improvements that need to be handled still but we were also able to identify some clear benefits of Scrum usage.

Common conclusion based on working well issues and improvements is that this model is worth of use, but it requires some more development. Process and team issues were largely identified in both sides, which gives a contradictory result, but when looking into details it becomes clear that some actions and new skills are still needed while adjusting Scrum to this environment.

The axially coded information from the interviews for the two main categories is summarized in two mind maps as provided in Pictures 1 and 2. Results are described here separately for both main themes.

4.1. Issues working well

Issues working well have two main categories where the most issues belonged to: Team and Process. Tools, mostly collaboration tools, were the next biggest top group, and Communication, especially team communication, close to it in size. Atmosphere when interviewing people of the issues working well supported these findings: participants sounded to be generally happy to use Scrum and they seemed to know each other at least on some level even if there haven't been any face-to-face meetings.

In picture below findings from well working issues are shown in mind map. Findings strongly suggest that it is possible to use Scrum in globally distributed maintenance project and that it will create some benefits. Scrum model is able to bring some of its benefits and here they are visible especially in form of advantages for team and in Scrum process generally. Each top level will be described in more details below picture.

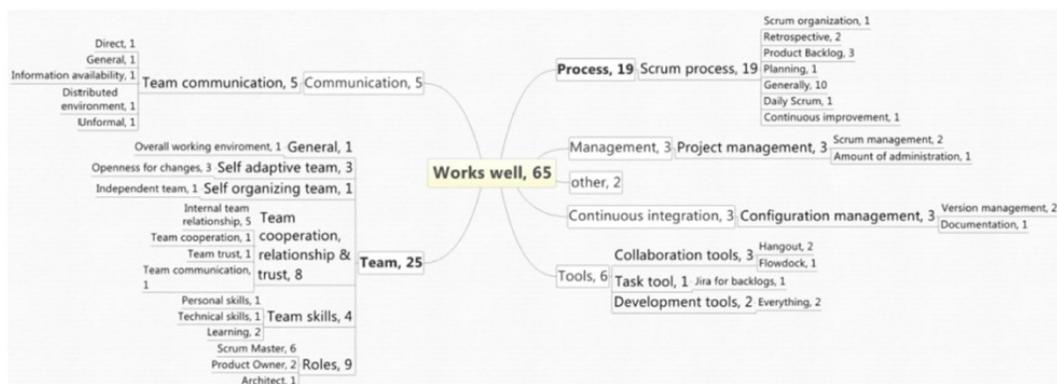


Figure 1. Works well issues mind map

4.1.1. Team

Within team, Roles got the highest amount of mentions and, especially, a good Scrum Master was found to have a central effect. The SM was good in facilitating meetings, and coordinating and supporting team. Another important issue was the combination of Team cooperation, relationship and trust, where especially internal team relationship was mentioned. Helping each other, positive attitude and asking questions are examples of things mentioned there. Also skills and Scrum related issues were mentioned: self-organized and self-adaptive team where self-adaptivity was presented as openness for changes. Therefore, we are able to say that Scrum in globally

distributed maintenance project was well-addressed by the team. It improved team spirit and trust and especially good Scrum Master was the key person in the team.

4.1.2. Process

All the good things in Scrum were linked to the top category of Process. Most comments were on the level that Scrum in general is good, but we were not able to identify exact practices behind these opinions. Interviewees generally liked to participate in a Scrum project, they saw that the use of Scrum brought many good things to the project, and it was also implemented well.

4.1.3. Other issues

Also issues linked to the categories of Tools, Communication, Management, Continuous Integration, and Other were revealed. In Tools, most mentions were for collaboration tools and, especially, Hangout as video meeting tool and Flowdock as instant messaging tool were mentioned. In Communication, there were several separate mentions but they were all related to team communication. In Management, all mentions were on Project Management and Continuous Integration was all about Configuration Management. There were two mentions in Other which we were not able to group: “service is working well” and “if no one notices your work, you are doing it well”.

4.2. Needs of improvements

Improvement needs received more comments than the working well issues and they had three main top groups: Process, Team, and Communication. It is very interesting that Process and Team were also biggest top groups in the working well theme. It seemed to be easier for participants to identify improvement needs. One possible reason for this is that participants were used to have frequent retrospectives where they continuously handled improvement needs.

In the picture below findings from improvement needs are shown in mind map with three levels. Each top level will be described in more details below picture.

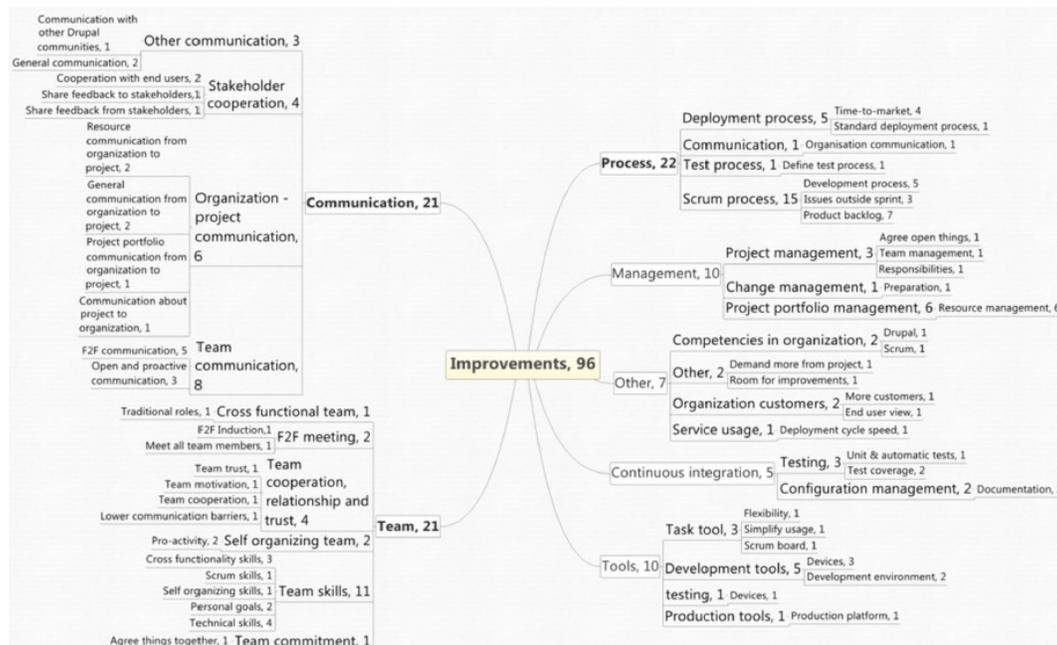


Figure 2. Improvement issues mind map

4.2.1. Process

Process was the biggest top category but only with one more mention than Team or Communication. Biggest part of Process improvements was Scrum Process, which, however, got almost as many mentions as in the Works well side. Concerning Improvements, interviewees were able to identify more specific issues.

PB was the most popular single item in Process Improvements and, there, six out of seven answers wanted more new development tasks in PB. The phase of development for the software was in a situation where the target was to maintain current service but also to create new functionalities. However, in some previous sprints the PO had prioritized mainly bug fixes in top of the PB. This would suggest that people are more interested in developing new issues than fixing bugs and Scrum is seen more as a development process model than as a maintenance process model.

In the development part related to Scrum, more flexibility and more time for real development work were the mostly raised issues. Issues outside project were mentioned too and there especially their interruptions to work. To reduce impact of these interruptions for each sprint one person was named to handle them. This is in line with suggestion above and also says that special attention must be paid on how to handle inevitable interruptions from production maintenance.

4.2.2. Team

Biggest part of team related improvements were related to skills in the team. Technical skills were the most popular issue and it is easy to see the connection to the background questions according to which the used technology was new for some team members. Another big part of team skills were the cross functionality skills. When it is compared to different roles in the team, it seems that the work inside the team was still done in a traditional (less self-adaptive) way where each participant had a certain role and corresponding tasks in the project. This suggests that more work is needed to develop team skills needed in Scrum model. This also creates need for further study: what will happen when we have fully cross-functional, self-adaptive team, will it change study results? Rest of the issues related to team improvements were mainly separated things, only mentioned once.

4.2.3. Communication

Communication issues were more evenly distributed within the different communication areas. Biggest group of communication was in the team communication where need for face-to-face communication received most mentions. The team has never actually met and it was obvious that they wanted to see each other. Another part in team communication was open and proactive communication and it is understandable that it is not so easy with persons you have never seen. Also cultural aspects need to be considered here because communication in the Nordic countries is different from India even if they weren't separately mentioned.

The next biggest issue in communication was the communication between the organization and the project. It was also the area where frustrations were visible in the interviews. It seems that fitting Scrum project in the organization and in response of maintenance was not well enough supported from the organization's side. There were some clear communication gaps between organization and project, like communication of other projects, where the same persons participated. The analyzed case is based on a project organization (meaning here that almost all work in this unit of company was made in projects) but on the maintenance side in company, ITIL (Information Technology Infrastructure Library for IT service management, see e.g. <http://www.itil.org.uk/>) practices were followed too. This suggests that more work is needed in how to implement maintenance Scrum project in company's processes to ensure communication between project and organization.

Another issue in communication was related to communicating with stakeholders. In stakeholder communication, there were two aspects with the same need of improvement present: stakeholders or team members didn't know how to get content in project's service in case they wanted to publish something in web portal maintained here. This refers to the phase before PB and to usage of the service.

4.2.4. Management

In the Management category, Resource Management in the Project Portfolio Management was clearly the biggest issue. There was concern that team wasn't stable, because constantly other projects were ongoing with the same members and new people joined the team. Project's role as internal development project, in a company whose main business area is to sell IT services to other companies, explains this partly. It is likely that this will also create a threat for achieving same performance as in co-located Scrum teams, but this question is left for further studies. Project and Change management got a few mentions here too.

4.2.5. Tools

In the Tools category, Development Tools created half of the mentions. Virtual servers, shared configuration management, connections and other technical issues have higher requirements in a globally distributed environment. It seemed that there were some differences in devices between locations. The project used Jira's Scrum Board as task tool and this also got some separate mentions.

4.2.6. Continuous Integration and Other

Testing and Configuration management were mentioned in Continuous integration. Especially testing coverage and documentation were found to need improvement. The separate, individual mentions without clear main context were gathered to the Other category.

4.3. Limitations

There are some identified limitations in this study. First, same person was participating in project as Project Manager and in study as Researcher. It has some limitations to results: we weren't able to interview project manager and we had to ignore all her opinions from project in order to be objective. Second, when interviewer already knew everyone to be interviewed, it might have had some effects in results. This was handled by keeping open atmosphere during interviews, encouraging participants to speak freely and letting them check notes on the fly. End result, this is now a personal opinion based on amount of results by participants, is that we were able to collect much more information than if interviewer would have been some unknown person. Third, we must consider that there were participants from three different countries (Finland, India and Sweden) with different cultures.

5. DISCUSSION

We were able to confirm that Scrum can be used in globally distributed maintenance project. Anyway, there are some aspects we found that must be addressed: i) interruptions from production, ii) ensure that team has required skills, and iii) communication between project and organization. These are partly in line with related work and in following chapters we will compare our results to them. Generally we were able to identify some same issues, some issues not, and we agree that it is possible and it might bring some benefits to use Scrum GSD.

In [3] were identified seven challenging factors. Synchronous Communication was not an issue here because biggest time difference was 4.5 hours between Sweden and India. The next biggest challenging factor in their literature review was *collaboration difficulties* and those were not visible in our case. Cooperation got only one mention and when we combined all closely related issues we still have a very small group. This is especially interesting result due to fact that there haven't been any visits or face-to-face happenings in the maintenance project and it is commonly

seen as important factor in global implementations of Scrum [13] and [17]. Communication bandwidth challenge wasn't either visible in our case but it was likely due to heavily used video conferences and chat, and the same situation was with the tools, because tools supporting distributed environments were ready and offered by the company. The same tools are also in use in other projects of the company.

Our improvement results match only somehow for challenges at [8]. We couldn't identify any of communication improvements into *cultural differences*. This is interesting because we had two different cultures involved: Nordic (Finnish, Swedish) and Indian. Conflicting priorities were found in two places: organization – project communication and project portfolio management. If combined, they would create a middle size top group in our study. Team collaboration issues are not common in our results, and internal team relationship got several mentions in the Works well side. That would suggest that lack of face-to-face can be handled with correct use of communication tools, but people will still miss it.

We were not able to identify all corresponding improvement needs for the challenges proposed in [2]. Network connections were not mentioned and cultural differences were not much visible. *Team communication* was one major category in improvements, but there were not any special comments of misunderstandings. We are able to confirm some positive experiences anyway. Team in communication and relationship was well working issue and Team and Process were biggest top groups in here. Possible reason for this might be that we were already following some of the lessons and recommendations as described in Section 2: i) shared tool supporting Backlog management, ii) distribution to achieve certain benefits like competence availability, iii) availability of several experienced Scrum persons, and iv) presence of many open communication channels whose use was encouraged. Let us also note that we had only one Scrum team under analysis.

For [9] we can confirm that i) team relation related issues were on working side, ii) for coordination and progress tracking we can agree that we didn't get that kind of issues in improvement side, and iii) it creates clear boundary objects like PB and it has *clear boundary spanner roles* like SM. Especially SM was seen as well working and important role in the project. For [10], [11] and [12] we can't estimate if velocity is close to collocated team, and for [12] we can only say that we had small group of team communication in both sides and our improvement needs are not similar to their challenges.

Based on our study, we cannot conclude rigorously whether Scrum in a distributed maintenance project is either good or bad solution [14]. We can conclude that we found more improvement needs than working well issues but, on the other hand, it seemed to be easier for participants to find detailed improvement needs. Scrum process was among the biggest groups in both sides' even if different details and levels of detail were mentioned.

Some of the nine heuristics as concluded in [15] were visible here too. It is good to remind that our project was slightly different than theirs; we had only one (internal) customer. Scrum in generally was seen as a working solution and work coordination issues were seldom mentioned as improvements, so Scrum with its sprints was a working approach to organize the maintenance work. Issues outside sprint were mentioned in Improvements and, for that, Heeager et al. just proposes to accept them. This same has been found in other studies [16] and, clearly, we can't follow Scrum strictly. In our study, team skills had a strong part in the improvements side which at quick glance seems opposite to their study, but by noting that we had, for example, several different roles in the team, this likely means that in our projects there was still some knowledge sharing to do. In our case, it looks that balance was much more on documentation than in face-to-face communication and, for project collaboration that was a good solution. Anyway, people still missed personal communication possibilities.

Compared to [16] we can find some correspondences. Comprehension, change effects and need of a guru can be linked to our findings for the need to improve the team skills. Test coverage and

testing was mentioned few times, but in our study it was not among the biggest improvement needs.

ACKNOWLEDGEMENTS

The authors would like to thank Tieto and in particular the interviewees for participating in the study. The study was partly funded by TEKES/Digile as part of the Need for Speed (N4S) program.

REFERENCES

- [1] Brooks, F. P. (1987) “No Silver Bullet—Essence and Accidents of Software Engineering”, *Computer* 20, pp1-16.
- [2] Paasivaara, M., Durasiewicz, S., Lassenius, C. (2008) “Using Scrum in a Globally Distributed Project: A Case Study” *Softw. Process Improve. Pract.*, 13, Wiley InterScience, pp 527-544.
- [3] Hossain, E., Babar, M., Paik, H. (2009) “Using Scrum in Global Software Development: A Systematic Literature Review” *Fourth IEEE International Conference on Global Software Engineering*, pp175-184.
- [4] Lehman, M. M. (1980) “Programs, Life Cycles, and Laws of Software Evolution” *Proc. IEEE* 68, pp1060–1076.
- [5] Lientz, B. P., Swanson, E. B. (1980) *Software Maintenance Management* Addison-Wesley Longman Publishing Co., Inc., Boston, MA.
- [6] Omanovic, S., Buza, E. (2013) “Importance of Stable Velocity in Agile Maintenance”, *XXIV International Conference on Information, Communication and Automation Technologies (ICAT)*, IEEE.
- [7] Yin, R. (1994) *Case Study Research: Design and Methods*, Sage Publications, Unites States of America.
- [8] Lee, S., Yong, H.-S. (2010) “Distributed agile: project management in a global environment”, *Empir. Software Eng.* 15, pp204-217.
- [9] Pries-Heje, L., Pries-Heje, J. (2011) “Why Scrum works: A case study from an agile distributed project in Denmark and India”, *Agile Conference*, pp20-28.
- [10] Sutherland, J., Viktorov, A., Blount, J., Puntikov, N. (2007) “Distributed Scrum: Agile Project Management with Outsourced Development Teams”, *Proceedings of the 40th Hawaii International Conference on System Sciences*, IEEE.
- [11] Sutherland, J., Schoonheim, G., Rijk, M. (2009) “Fully Distributed Scrum: Replicating Local Productivity and Quality with Offshore Teams”, *Proceedings of the 42nd Hawaii International Conference on System Sciences*, IEEE.
- [12] Sutherland, J., Schoonheim, G., Rustenburg, E., Rijk, M. (2008) “Fully Distributed Scrum: The Secret Sauce for Hyperproductive Offshored Development Teams”, *Agile Conference*, pp339-344.
- [13] Paasivaara, M., Durasiewicz, S., Lassenius, C. (2009) “Using Scrum in Distributed Agile Development: A Multiple Case Study”, *Fourth IEEE International Conference on Global Software Engineering*, IEEE, pp195-204.
- [14] Estler, H.-C., Nordio, M., Furia, C. A., Maye, B., Schneider, J. (2014) “Agile vs. structured distributed software development: A case study”, *Empir. Software Eng.* 19, pp1197-1224.
- [15] Heeager, L., Rose, J. (2014) “Optimising agile development practices for the maintenance operation: nine heuristics”, *Empir. Software Eng., Springer Science+Business Media*, New York
- [16] Hanssen, G., Yamashita, A. F., Conradi, R., Moonen, L. (2009) “Maintenance and Agile Development: Challenges, Opportunities and Future Directions”, *Proc. ICSM 2009*, Edmonton, Canada, IEEE, pp487-490.
- [17] Bannerman, P., Hossain, E., Jeffery, R. (2012) “Scrum Practice Mitigation of Global Software Development Coordination Challenges - A Distinctive Advantage?”, *45th Hawaii International Conference on System Sciences*, IEEE; pp5309-5318.

AUTHORS

Minna Kilpala has ten year experience of project management and she works as senior project manager in Tieto. She has IPMA-C project management certificate and she is certificated Scrum Master and Scrum Product Owner. She has been working last 5 years with globally distributed agile projects. She has master degree in Computer Science and she is doing postgraduate studies at Jyväskylä University.



Tommi Kärkkäinen serves as a professor in the Department of Mathematical Information Technology, University of Jyväskylä. He is vice head and head of a PhD study line in COMAS (Graduate School in Computing and Mathematical Sciences) and principal professor of Software Engineering and Educational Technology MSc study lines. His research interests are related (but not limited) to computational sciences, software engineering, and computing education research. He has supervised more than one hundred MSc theses, over 20 PhD theses, and co-authored more than one hundred scientific publications.

