VG

Almost perfect

Final Report

ANONYMOUS AUTHOR
Examination in DIT347 Software Development Methodologies
Bachelor Program Software Engineering and Management
Department of Computer Science and Engineering
University of Gothenburg
Gothenburg, Sweden

Abstract—
This report looks at how a process created by students, in a course on software development processes, was applied and improved. It shows how the initial process, when applied in a workshop held within the course, was lacking in many ways. For example, the communication between the group and other participants hadn't been planned for and was inefficient as a result. Also, the group often didn't follow the process's activities as planned which often made the work disorganised and inefficient. By using some common software process improvement methods these issues were improved and the group applying the process performed better when using it in a second workshop. The report also gives examples of how SPI is used by organisations in the industry and describes what challenges they have faced while doing so, how they overcome those challenges, and how their experiences relate to those of the group.

I. Introduction

The purpose of this report is to show how a process created by a group of students was applied in two workshops and improved therebetween, and to show examples of how SPI is used in the industry. Section II will describe the process that was originally planned and the process that ended up being used in the first workshop, section III covers the issues encountered by the group in the first workshop, section IV describes some common SPI methods, section V looks at examples of how SPI is used in the industry, section VI lays out an SPI proposal to address the issues identified in section III, Section VII describes the results the SPI effort had, and section VIII summarises the results of this report and lays out a suggestion for another iteration of SPI.

II. Process Applied in the Scrum Workshop

The process the group had created for the first workshop included the following sequence of activities. The workshop would start with an *Initial meeting*, held before the first sprint, for instantiating the process (deciding on schedule and definition of done (DoD), looking through and understanding the requirements etc.). Every sprint would begin with a *sprint planning session* where we would choose user stories for the upcoming sprint and break them down into tasks that could then be assigned to team members. This would be followed by a *design* activity where the team would decide implementation details such as dimensions, building materials etc. (that were not already specified by the user story). The team would then move to a *build* activity where the user stories would be implemented by performing the created tasks. While performing the *build* activity, the team had planned to keep track of task progression and assignments by using a private team Scrum board. After the stories had been implemented, the results would be *integrated* into the city by being connected to other structures like roads and buildings, and, if the user stories were finished, the implementations would be *deployed* by marking them as ready and enabling other teams to use them (Deploying the increment was mistakenly planned for before the sprint review. This wouldn't have worked since we could not have known if the stories would be accepted before having them reviewed by the product owners (POs)). We would then have the *sprint review* with the PO to discuss if the acceptance criteria and the DoD hade been fulfilled, and finally the *sprint retrospective* where the team would discuss how the last sprint went and which changes needed to be made before the next iteration.

The process also included the standard Scrum roles. i.e. Scrum master (appointed to one of the team members), product owner (taken on by the teaching assistants), and Scrum team. We considered adding more roles such as secretary for documenting the team meetings and gatherer for being the main resource collector(s). For the role of secretary we opted not to have this as a permanent role and instead to rotate the responsibility by appointing a secretary for every meeting. We chose not to use the role of gatherer since we didn't know exactly how material collection would be handled in the workshop and thus not if mining for materials would be necessary. We also thought that, if mining did play a part in the workshop, it might be more efficient and flexible to treat this like any other aspect of the implementation of user stories and assign this task to as many team members as needed.

The process that was actually applied during the workshop ended up significantly different from the one we had planned. Some of the activities that we had prepared weren't performed. For example, integration of our implementations with the rest of the city was never performed as a separate activity. This instead happened in parallel to the build activity while implementing the structures. Similarly, we never performed a seperate design activity. Implementation details where instead discussed either while creating the tasks, or while building. There were also activities that, while they were performed, weren't performed as planned. For example, during the sprint planning sessions, the team usually started implementing the chosen user stories before the tasks had even been created. The little discussion that did occur regarding task creation happened in parallel to the implementations. Another activity that wasn't performed as planned was the build activity. In particular, the use of the team Scrum board during this activity was different from the way we had planned

why? impact?

it. Instead of people assigning themselves tasks and progressing these to different columns as they were being implemented, the Scrum board was often ignored. When team members did assign themselves tasks, this usually happened verbally, and the progression of tasks along the board was often overlooked until the tasks were completed at which point they were moved directly to the "finished" column.

why?

The reasons for these deviations come down to a lack of preparations. Since we hadn't prepared a way of identifying or dealing with deviations from the plan, we often didn't even reflect on the fact that these had occurred. During the rare occasions where we did realize this, no serious attempts were made to fix the deviations since we still managed to implement our user stories. We also hadn't been precise enough when planning how the activities should be performed. For example, while we had said that the tasks on the Scrum board should be progressed along the columns while they were being implemented, we hadn't described precisely when a task should be moved from one column to another. This meant that while performing the build activity, not all team members knew when a task should be in a specific column and thus couldn't use it correctly.

Overall, the process the team applied ended up feeling very ad-hoc since we did things based on what we felt was beneficial in the moment, and not because it was described by the plan.

III. Experiences in the Scrum Workshop

While the applied process deviated from the planned one significantly, there were parts of it that ended up working well. An example of this was the communication within the team. For most of the workshop, when all members of the team were present in the breakout room, we kept each other updated on what everyone was doing, what was left to do, what issues there were, and how we could solve these. Even when not all members were present in the breakout room, the communication still worked well. For example, when the team's Scrum master left to pick user stories for the next sprint he would keep in contact with the rest of the team through the group Discord server. There, the Scrum master would inform the group about what was happening while asking them for their preferences. To some extent, the effective communication within the team made up for the lack of preparedness in other areas. For example, while we hadn't been precise enough when planning how to use the team Scrum board which led to it mostly not being used, we partially made up for this by always assigning ourselves to tasks verbally and keeping the rest of the team updated on the progress of the tasks.

There were also, however, parts of our process that did not end up working well.

Team-PO communication was an issue throughout the entire workshop. Not knowing how POs should be reached when questions arose led to delays as the team had to spend time looking for them before continuing with the implementations. This was made worse by the workload between the POs being uneven at times which could lead to further waiting times. These communication difficulties led to our team losing a significant amount of time when implementing one of our user stories where the required building material had run out, and we needed to contact the PO before continuing. The reason for these issues was that we hadn't prepared a way of communicating with the POs before the start of the workshop. I.e. there had been no conversation among the groups on how the POs should be reached or how to make sure that the workload was evenly distributed among the POs.

While It is important for POs to be available to the development teams throughout endeavours to clarify implementation details, issues with POs not being sufficiently available are not unique to us and are quite common in projects where Scrum is used [1].

Another issue that occurred was that there were members of our group who weren't familiar with games like Minetest from before, and therefore weren't comfortable with the basic mechanics of the game. For example, not everyone knew how to craft basic tools and items. This caused us some delays initially since these team members had to familiarise themselves with the game before they could work efficiently. This issue was not unique to our group. The team we interviewed stated that unfamiliarity with the game was their biggest issue in the workshop and that not knowing how to do things like craft tools for example slowed their implementations down significantly. The reason for this issue also comes down to a lack of preparedness. While the team did discuss and come to the conclusion that everyone should have at least tried the game out before the workshop, we should also have discussed what game knowledge we needed in order to function efficiently, and made sure that team members lacking in these areas could fill the gaps.

While it is not uncommon for new software developers to be unfamiliar with the tools used in their projects and for this to lead to difficulties during the endeavours [2], situations like ours where a large part of developers have little to no experience with the development tools are most likely specific to our efforts.

As mentioned in section II, the team didn't follow the prepared process as planned. This turned out to be one of the bigger issues in the workshop since it often led to a worsened quality of the outcomes of deviated activities and a reduced efficiency of the team. The clearest examples of this were the sprint planning sessions and the build activities. The fact that the planning sessions weren't held as distinct and organized meetings, but instead occurred in parallel to the implementations of the user stories, led to the created tasks usually not being very thought through which sometimes meant that unnecessary tasks were performed or that the performed tasks could have been done in a more efficient way. The fact that the team, while performing the build activity, didn't use the team Scrum board as planned, sometimes led to a reduced efficiency of our implementations. For example, not seeing which team member was assigned to which task coupled with not always knowing which tasks were left to do sometimes led to people starting to work on tasks that were already being performed. Like mentioned earlier, these issues were partially mitigated due to the good communication we had within the team. They could, however, had been avoided completely had we used the Scrum board as intended. This issue also comes down to insufficient planning. We should have been more precise when describing how the activities should have been performed. For example, while planning for the sprint planning session, we should have been clearer about when the meeting should end and what we should, and should not do while holding the meeting. We should also have prepared a way of detecting deviations from the planned process. Had this been the case, we could have adjusted when noticing that what we were doing wasn't what was planned.

Issues with people not adhering to the process are, according to Overhage et al. [3], not uncommon in organizations that are adopting Scrum for the first time. In these cases the issues are often caused due to individuals being resistant to the changes that come with adopting a large framework like Scrum. Marchenko et al. [4] describe how a situation like this occured at a department of Nokia when it used Scrum for the first time. Some developers chose not to follow the process as planned (not participating in activities, or participating as planned) which led to difficulties in adopting the framework.

IV. Software Process Improvement Techniques

According to Pettersson et al. [5] SPI frameworks can be divided into two groups. Inductive frameworks and prescriptive ones. Inductive frameworks take the approach that process improvement efforts should be based on a thorough understanding of the current situation of the organization. Unlike the prescriptive frameworks, these SPI methods do not come with a predefined set of practices that can be used to solve the identified issues. They do however offer steps to analyze the current situation of the organization, and to identify the issues that it has. Two examples of inductive SPI frameworks are QIP and iFLAPS.

Quality improvement paradigm (QIP) is, according to its creators Basili et al. [6], an iterative SPI framework intended to acquire core competencies within the applying organization that can then be used within its projects to increase the quality of the created products. Basili et al. describe that this is achieved through six steps where the organization identifies issues within it, creates goals to address these issues, chooses techniques and tools to reach them, applies these in projects, analyzes the results, and distributes the gathered knowledge within the company. Basili et al. also state that QIP includes the Goal-Question-Metric (GQM) model, which gives the organization a structured way of defining the goals that it aims to achieve, together with questions that break the goals down into its major component and whose answers indicate whether progress towards the goal is being made, and metrics that can be used to answer the questions.

Petterson et al. [5] state that iFLAP consists of three main steps. In step 1 - selection, the projects that should be analysed are selected. Roles within that project are then chosen, followed by people that can represent each role. In step 2 - assessment, improvement issues are identified through elicitation techniques such as interviews held with people identified in step 1, and analysis of documents from the chosen projects. An important part of iFLAPS is, according to Peterson et al., the triangulation of issues. i.e. finding multiple sources that confirm that issues exist. This leads to fewer issues to tackle, as problems that can't be confirmed by other sources can be dismissed, and a greater certainty that the identified issues are real. In step 3 - improvement planning, the issues will be prioritized and analyzed for dependencies. This should result in a prioritized list of issues that can be addressed by the organization. Peterson et al. also say that iFLAP is a lightweight framework that can be adjusted to require less or more resources. This might make it preferable over other SPI frameworks like QIP. Particularly for smaller organizations.

While both inductive frameworks could be used to identify issues from the workshop, the time constraints that we had makes iFLAPS a more suitable choice since it is, as described by Petterson et al. [5], more lightweight than QIP. The GQM approach introduced by QIP could, however, still be useful when defining our goals.

According to Pettersson et al. [5], prescriptive frameworks will, unlike inductive ones, not provide steps to identify issues specific to an organization. They do, however, come with a set of best practices that should, with some modification, be applicable to all software development endeavors, and that can be used to improve the organization's processes. One example of a prescriptive framework is CMMI.

According to the SEI CMMI Production Team [7], Capability maturity model integration (CMMI) comes with 22 process areas. These are all accompanied by a set of specific goals that should be fulfilled to improve the process in that specific area, and by specific practices that can be implemented in order to reach the goals. CMMI also has generic goals that, instead of being related to a specific process area, are relevant to many. These are also accompanied by generic practices that can be implemented to achieve the generic goals. The SEI CMMI Production team states that CMMI can be used in two ways. Through continuous representation which allows organizations to focus on some process areas that are specific to their issues, or through staged representation where the organization attempts to increase its overall maturity level by implementing process areas related to one of five levels. Petterson et al. [5], mention that a disadvantage to CMMI is that it can be costly compared to other frameworks, as SPI cycles can be lengthy and resource demanding. For this reason, smaller organizations often choose against using the framework [8].

CMMI could have been used to guide our team when making the concrete changes needed to improve the process. For example, in order to improve our issues with team-PO communication, the team could have implemented process area *project planning* which, according to the SEI CMMI Production team [7], includes the specific practices *plan for the involvement of stakeholders* and *plan for resources to perform the project*.

V. SPI in Industry

The processes used by medical device companies such as 1928 diagnostics are, according to the company's QA/RA director Robert Engberg, limited by the laws and regulations of their targeted markets, as well as by standards that they are expected to follow. For these types of companies, SPI becomes an important tool to have in order to deal with eventual shortcomings to these standards and regulations. Engberg stated that the company, when performing SPI, uses performance indicators such as burndown rates to measure the capability of their process and applies continuous improvement methods to improve where it is lacking. They also compare their process to the one described by the standard they're following. While the standard doesn't need to be followed exactly as prescribed, it does say what needs to be included in the process used by the company, and can thus be used to see where it is lacking. One challenge, described by Engberg, that medical device companies face when creating and improving their processes is in deciding to what extent the standard should be followed. To some people within the company, such as the people who will be present in auditing meetings, it might seem beneficial to follow it exactly as described, as it will then be clear how all the necessary steps are being taken. For others this might not seem beneficial as it could mean a drastic deviation from the process already in use and a lot of changes which could be met by resistance. For 1928 diagnostics, what worked better was to begin with the process already in use and what they consider to be an optimal way of working. From there, they consider what the standard says and add what is missing to fulfill its requirements. This leads to a smaller deviation from the process already used and to less resistance from developers and management.

In 2003, the software development process used by a division of Siemens followed a waterfall approach. According to Dulce Goncalves, who was a part of the division at the time, the choice to start following agile principles and adopt an iterative process was initially made to combat its growing organizational complexity and the issues that came with it. Goncalves stated that they wanted to flip the organization by having the managers support the teams which would bring the developers closer to the end users and the product closer to the market. They also wanted to flatten the organization and reduce the layers of management that existed within it. This was achieved through the use of cross-functional and self-managing teams that could function well on their own and didn't need to rely on managers. Performing these changes worked well within the division which, eventually, could reduce the numbers of middle managers because of it.

Going from a traditional waterfall process, to an agile one, can mean a lot of changes for people within an organization. For example, while following agile principles, transparency is emphasized much more than in traditional processes and the frequency of meetings is significantly increased [3]. While these changes might have positive effects, they may not be preferred by everyone in the organization. Goncalves mentioned that the tension and resistance that this can lead to was one of the challenges that the division of Siemens faced when changing its process. To combat this tension, it was important to make sure that the people within the organization, besides adopting the new way of working, also adopted the agile culture. This means that employees have to understand why it is beneficial to follow an agile process, and agree with the way of working that it entails. Attempting to follow these principles without also implementing the corresponding culture wouldn't work, according to Goncalves, as this would lead to resistance and "checking of boxes" i.e. doing things only for the sake of doing them without knowing why.

One challenge both organizations had to face when changing their processes was how to do so without creating resistance among the employees. This was an issue that we also faced in the workshop where the process prepared by the team, in many ways, wasn't followed. It might also be the case that some of the solutions proposed by the organizations could have reduced these deviations. For example, only a few members of the team had used a Scrum board in previous projects. If these members had made it clear to the rest of the group why we should do this and what benefits it could bring, then the board might have been used more.

VI. SPI Proposal for future Scrum Development Efforts

The goal of the SPI effort will be to improve the major issues identified in section III, i.e. to improve the efficiency of team-PO communication, improve the developers knowledge of Minetest mechanics, and to improve the quality of the sprint planning sessions (As described in section III, the unstructured nature of these sessions had a particularly negative impact on the teams performance). The following GQM model could help us by concretizing the goals (G) (making point of view and what should be achieved clearer), giving us questions (Q) that indicate if the goals are achieved, and providing metrics (M) that answer these questions.

G 1 - Improve the efficiency of team-PO communication from the viewpoint of the development teams.

Q 1.1 - How much time is lost waiting for the PO?

M 1.1.1 - Average agreement to the statement: "I never have to wait for the PO in order to continue on my current task."

M 1.1.2 - The average agreement to the statement: "The requirements are well defined" (having poorly defined requirements could increase the need to contact POs which would give them an increased workload and lead to more time lost trying to reach them).

Q 1.2 - How satisfactory is the efficiency of the team-PO communication?

M 1.2.1 - The average agreement to the statement: "I am satisfied with the team-PO communication."

M 1.2.2 - The average answer to the question: "Is the POs workload reasonably even?" (if the workload between the POs wasn't even then it would make sense for the efficiency of team-PO communication to decrease).

G 2 - Improve the quality of the sprint planning sessions from the viewpoint of the development teams.

Q 2.1 - What is the current quality of the sprint planning sessions?

M 2.1.1 - Time per planning session not spent planning.

M 2.1.2 - The average developer agreement to the statement: "I am satisfied with the time allocated for sprint planning." (If

No measurement plan

the time allocated is too low, it would make sense for the quality to decrease.)

Q 2.2 - How often are tasks from the sprint planning sessions changed during the sprints?

M 2.2.1 - The number of tasks changed or added during a sprint divided by the number of initially created tasks.

M 2.2.2 - The average agreement to the statement: "The created tasks rarely change."

G 3 - Improve the knowledge of Minetest mechanics from the viewpoint of the developers.

Q 3.1 - How knowledgeable are the developers currently of the game mechanics?

M 3.1.1 - Average developer agreement to the statement: "I am comfortable with the game mechanics."

M 3.1.2 - The number of times an item was needed in the workshop that the team didn't know how to create.

Q 3.2 - Did the developers' lack of knowledge of the mechanics hinder them from implementing the user stories?

M 3.2.1 - Number of user stories accepted by the POs divided by user stories taken on.

M 3.2.2 - Average developer agreement with the statement: "I am satisfied with the implementation of the user stories." [9]

when? For the subjective metrics, data will be gathered through the use of a questionnaire given to the team. For the metrics measuring average agreement, the questions will be answered using the Likert scale. For the other metrics, yes/no questions will suffice. The questionnaire will be taken once before the start of the second workshop, to answer with the first workshop in mind, and then after the conclusion of the second workshop to see if progress was made. For the objective measurements, we will designate one team member who will, while participating, observe the group and collect the necessary data. The data gathered will be analyzed after the conclusion of the workshop and compared to the results of the previous one to see if we have achieved our goals. Since we didn't collect data during the first workshop, estimations will be made for the metrics that weren't answered by the questionnaire.

CMMI could be used when making the changes to our process. Two process areas that might be implemented are *project planning* (PP) and *requirements development* (RD). Before the start of the improvement effort, the group's capability levels in both of these process areas are 0 since the specific goals of neither one are satisfied. The goal for both process areas should be to reach capability level 1, i.e. to satisfy all specific goals of those process areas as well as generic goal 1.

When describing process area PP, the SEI CMMI Production Team [7] states that its purpose is to establish and maintain plans that define project activities. By implementing this process area, we will improve most of the issues we faced during the workshop since most of them came down to insufficient planning. Some specific practices that might help how? us reach the goals are *plan for the involvement of identified stakeholders* which will help us plan for how communication with the POs should be handled, *identify and analyze project risks* which might help us deal with deviations from the plan, and *plan for the knowledge and skills needed to perform the project* which will help us deal with the competency gaps that existed in the first workshop. Process area RD will be particularly helpful in improving team-PO communication since it contains practices for developing and analyzing requirements. For example, it includes practices like *analyze requirements to ensure that they are necessary and sufficient* and *analyze requirements to balance stakeholder needs and constraints*. This will benefit the team-PO communication since improved requirements and better understanding of the requirements will lead to having to contact POs less often, giving them a reduced workload.

VII. Implementation of an SPI Initiative

The CMMI process areas described in section VI had the following effects on the teams process.

As a result of implementing the practices from process area PP, the team spent more time on planning than before the first workshop. For example, the team planned for how to communicate with POs by first discussing this among themselves, and then extending the conversation to the other groups. These conversations resulted in an agreement on how POs should be reached when questions arose and how to make sure that the workload between them was reasonably even. The team also planned for how to deal with competency gaps by having a discussion on what we needed to know to function efficiently during the workshop. This included how to use the basic game mechanics like mining and crafting. In order to deal with eventual competency gaps, we also created a knowledge sharing channel, available to all groups, where people could share resources like guides and video tutorials. We also planned for how to deal with deviations from the plan by appointing one of the team members to, while participating, monitor our applied process, compare it to the plan, and inform the rest of the group if deviation occurs. The planning that was done also ended up being more concrete than before the first workshop. For example, we were more precise when planning how the team Scrum board should be used, how task breakdown should occur, and when sprint planning sessions should end.

By implementing the process area RD, the team ended up spending more time on analysing and understanding the requirements. This happened during the sprint planning sessions where we would, for example, discuss if the acceptance criteria of our chosen user stories were clear, sufficient, and if there were any constraints hindering us from fulfilling them. In order to improve the quality of the user story acceptance criteria, there was also a discussion among the groups that resulted in an agreement to not choose user stories whose acceptance criteria were unreasonably undescriptive before these had been changed.

The changes to the process had a positive impact on many of the issues encountered in the first workshop and led to an improved performance of the team. Most importantly, the three major issues identified in section III were all improved, and the goals laid out in section VI were reached.

The efficiency of team-PO communication was improved as a result of the teams discussing and establishing a way of reaching the POs. This removed the issue of not knowing where the POs were when questions arose which, previously, had been a cause of lost time for the group. Also, as a result of further analysing and understanding the requirements, as well as our attempt to improve the quality of user story acceptance criteria, we ended up having to contact the POs less often which reduced their workload and further improved the efficiency with which these could be contacted. The positive impact the changes had on the team-PO communication is made clear by the questionnaires taken by the team before and after the second workshop. Before the second workshop, when asked to rate their agreement to the statement: "I never have to wait for the PO to continue working on my current task" 16.7% of the respondents agreed, 50% disagreed, and 33.3% strongly disagreed. After the second workshop, 20% of the respondents strongly agreed, 60% agreed, and 20% strongly disagreed. Similarly, when asked to rate their agreement with the statement: "I am satisfied with the team-PO communication", before the second workshop 16.7% of respondents agreed, 33.3% disagreed, and 50% strongly disagreed. After the second workshop, 20% of respondents strongly agreed, and 80% agreed.

As a result of more concrete planning and of having planned for how to deal with deviations from the plan, we ended up adhering to the planned process much more than we had in the previous workshop. One of the activities that saw improvement because of this was the sprint planning session. These ended up being more organized which resulted in tasks being more thought through. This meant that fewer tasks had to be changed or added after the conclusion of the planning sessions. From looking at the team Scrum board from the first workshop, the team estimated that, roughly, 36% of tasks were changed or added after the planning session per sprint. In the second workshop, this number was 16.6% which indicates that the tasks created during the sprint planning sessions were more sufficient in the second workshop. From the questionnaire, it also seems like most members of the group agree that the quality of the planning sessions improved. Before the start of the second workshop, when asked to rate their agreement with the statement: "I am satisfied with the current quality of the sprint planning sessions", 16.7% of respondents strongly agreed, 16.7% agreed, 16.7% neither agreed nor disagreed, 50% disagreed, and 16.7% strongly disagreed. After the second workshop, 80% agreed, and 20% neither agreed nor disagreed.

The team's knowledge of game mechanics was improved as a result of discussing what should be known before the start of the workshop. Because of this discussion, members who had previously not known how to do things like craft basic items or tools closed these competency gaps by looking up guides and tutorials before the projects start. The improved knowledge of the mechanics can be seen in the questionnaire taken by our team where, before the second workshop, when asked to rate their agreement with the statement: "I am comfortable with the game mechanics" 33.3% of the respondents strongly agreed, 50% agreed, and 16.7% neither agreed nor disagreed. After the second workshop, 80% strongly agreed, and 20% agreed. This did not, however, seem to have led to a clear increase in the quality of our implementations as we expected. Before the second workshop, when asked to rate their agreement with the statement: "I am satisfied with our implementations of the user stories" 83.3% of respondents agreed and 16.7% disagreed. After the second workshop, 80% agreed and 20% neither agreed nor disagreed.

VIII. Summary and Lessons Learned

This report has looked at how a process created by a group of students was applied, improved, and what the results of those improvements were. One of the main findings of the report is the importance of concrete planning. The absence of this was one of the causing factors for all of the major issues identified in section III. Increasing the amount and concreteness of our planning also improved all of these issues. Another finding was how, even for students that have no previous experience with SPI, these methods are still very effective as they led to a much improved process in the second workshop compared to the one initially applied.

If another iteration of SPI was to be performed, one of the issues to address should be the team-team communication. Although we did see some improvement in this regard in the second workshop, it was still far from optimal. This was particularly evident when discussing schedule and DoD for the endeavour. Even though we had made it known beforehand where these discussions were to take place, not all groups participated. Information about why the groups didn't take part in these discussions should be elicited by holding more interviews with other teams. To make concrete changes to improve this issue, one of the CMMI process areas to implement could be *integrated project management* which contains practices related to managing the involvement of, and issues with, relevant stakeholders.

References

[1] R. Vijay Anand and M. Dinakaran, "Issues in Scrum Agile Development Principles and Practices in Software Development", *Indian Journal of Science and Technology*, nr 35, s. 1–5, dec. 2015.

[2] A. Begel and B. Simon, "Struggles of new college graduates in their first software development job", *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, New York, NY, USA, mar. 2008, s. 226–230.

[3] S. Overhage, S. Schlauderer, D. Birkmeier, and J. Miller, "What Makes IT Personnel Adopt Scrum? A Framework of Drivers and Inhibitors to Developer Acceptance", *2011 44th Hawaii International Conference on System Sciences*, jan. 2011, s. 1–10.

[4] A. Marchenko and P. Abrahamsson, "Scrum in a Multiproject Environment: An Ethnographically-Inspired Case Study on the Adoption Challenges", *Agile 2008 Conference*, aug. 2008, s. 15–26.

[5] F. Pettersson, M. Ivarsson, T. Gorschek, and P. Öhman, "A practitioner's guide to light weight software process assessment and improvement planning", *Journal of Systems and Software*, vol. 81, nr 6, s. 972–995, juni 2008.

[6] V. R. Basili and G. Caldiera, "Improve soft-ware quality by reusing knowledge and experience", *Sloan management review*, vol. 37, s. 55–64, 1995.

[7] SEI CMMI Production Team, *CMMI for Development v1.3*. 2010.

[8] M. Staples, M. Niazi, R. Jeffery, A. Abrahams, P. Byatt, and R. Murphy, "An exploratory study of why organizations do not adopt CMMI", *Journal of Systems and Software*, vol. 80, nr 6, s. 883–895, juni 2007.

[9] [Undisclosed Authors], Assignment 4 in course DIT347 H20. Software Engineering Program, University of Gothenburg, Sweden, 2020.