# A Case Study On The Use Of SPI On A Minecraft Development Process

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*— In order to have a high quality software product, an organization needs to have a high quality software process and to achieve a high quality software process, we need to perform various software process improvement techniques in order to maximize process qualities such as productivity and time to market. This paper represents a concrete example of the use of software process improvement techniques performed on a scrum process for developing a product in the Minecraft environment.**

## I. Introduction

A software process is a scheme of tasks and practices that teams need to carry out for developing a high quality software [9]. In today's world where a lot of products are software driven, it is very important for organizations to have a detailed and mature software process [9]. Also the growing complexity of software has forced companies to ensure a highly efficient process plan for their software development and this as a result, has increased the importance of Software Process Improvement (SPI) techniques [9].

This paper explains various SPI methods and presents a concrete example of performing SPI in a Minecraft workshop.

This paper is structured as follows;

Section II describes the process planned to be carried out in the first Minecraft workshop and the actual process that was followed in the workshop. It mentions the differences of the actual process to the planned one and briefly states some of the underlying reasons for these differences.

Section III analyses the experiences our group encountered during the first workshop and concretely explains the exact causes of various issues we faced while following the process.

Section IV provides a general explanation of various SPI methodologies, some examples of concrete methods and their advantages and disadvantages.

Section V introduces two examples of how SPI was performed in industry and what barriers were encountered during the process. The companies mentioned in this section are Siemens and 1928diagnostics.

Section VI presents the concrete SPI steps that were planned to be taken in our second Minecraft workshop. It includes the use of various SPI techniques such as GQM and CMMI and provides metrics to measure the effectiveness of these approaches.

Section VII provides a deep analysis on which of these SPI methods were carried out in our second workshop, the benefits that they had for our team and the concrete measurements that were taken to analyze these benefits.

Section VIII summarizes this report and presents a conclusion on the findings of this report.

## II. Process Applied in the Scrum Workshop

We initially decided on following the most basic though complete scrum process in our workshop. Firstly, we decided on what roles to include in our process; the scrum team which consisted of the product owner who was the TA assigned to out group and the 6 members in our group with one being chosen as the scrum master, and the development team which due to the constraints of the course, also consisted of the same 6 members.

As described in our second assignment [2], we also decided to follow a number of practices in our process; the incremental delivery practice in order to implement the requirements in smaller chunks allowing us to get early feedback from the product owner and make necessary changes early on in the development, and the embedded customer practice which was chosen to ensure that the developers' views on the requirements were in sync with the customer's expectations by providing a way to get fast and continuous feedback on our development.

Another practice that we discussed before the workshop was the pair programming practice. This could benefit our team since some members were more experienced in Minecraft and other members could learn and be more beneficial for the team, but since all the members were working in distance, we decided that it would not help our team as it should. Also as described in The Essentials of Modern Software Engineering [1], pair programming practice is mostly used to "produce higher quality code" but since our workshop did not contain any coding tasks and also having a higher quality was not our team's main goal, we did not implement that practice.

Furthermore, we decided to have 4 sprints in the workshop each lasting 45 minutes. Our schedule for each sprint was to spend the first 5 minutes for the sprint planning, the last 5 minutes to perform a sprint retrospective and before the retrospective, to spend 10 minutes for the sprint review with the product owner, leaving us 25 minutes for developing one increment [2].

However, in the workshop, many parts of the process did not go according to how we had planned it. We ended up only having 3 sprints during which we only spent 1 minute for sprint planning, 5 minutes for sprint review and no sprint retrospective. We also failed to follow the embedded customer

practice as we had planned. I believe that the main reason for the inability to stick to our schedule was the lack of planning. Although we had made sufficient plans on how to approach the workshop and had a clear process plan in mind, we had not set any measures on how to stick to the schedule and to ensure that we are following it all the way. Moreover, we failed to plan for adjustments to our schedule if unpredicted situations arised.

For instance, two of our team members could not attend the workshop and our development team was reduced to 4 compared to the 6 we originally planned. This greatly affected our team's speed and skill set and put more work on each participating member which in turn impaired our development. Since we had no initial plan on changing the schedule to accommodate these situations, we ended up following the same schedule with less workforce leading to providing work that was not accepted by the product owner [2].

In addition, we originally planned to have the product owner embedded in our team, meaning that he is available most of the time and that we can clarify the requirements and to check our work continuously. This was however not the case since our product owner had to collaborate with many other teams and hence only managed to spend a short time with us. Similarly, we had not planned for this situation and hadn't decided on ways to approach the product owner and make the most of his short availability. This led to us implementing the requirements that were not in line with the expectations and had to redo parts of our work in the following sprints [2].

## III. Experiences in the Scrum Workshop

During the workshop, we started our process with 4 members as opposed to 6 as we had originally planned [4]. This was one of the main issues we faced and it led to many other problems and deficiencies we encountered in the process. As an example, we already felt short of workforce when trying to collect resources in Minecraft and building different parts of our structure resulting in us rushing for a completion rather than an acceptable work. Also, due to the shortage of members, we were unable to dedicate one person to leave the development team to seek product owners for requirement clarification.

Furthermore, we were not able to deliver an acceptable increment for the first two sprints. I believe that the main reason for this, was that the requirements were unclear and no acceptance criteria was provided by the product owner for each requirement. According to [9], "One of the major reasons causing projects to fail has been identified as problems concerning requirements engineering and the adjacent activities aiming at identifying goals, scope and requirements of systems". At the beginning of the workshop, we were given textual requirements with no explanation and no acceptance criteria. This caused us, the development team, to implement the requirements according to our own interpretations and not based on the product owner's expectation, resulting in increments that were not accepted and parts that had to be redone in consecutive sprints. Also as stated in [9], if requirements are not well elaborated, a lot of re-work has to be done by the team causing delays in the delivery of the project.

Although the direct cause of the above shortcomings were less human resources and weakly defined requirements, I believe that the underlying cause of both issues lies in the fact that we did not have a sufficiently detailed process plan on what actions to take in case of facing unexpected and unpredicted situations, or in other words, our plan had many unstated assumptions about the workshop such as the clarity of the requirements and the availability of all team members. Our team felt very lost when having to adjust for changes and this comes from the fact that we did not have a plan for adjusting to changes. For example, when two of our members couldn't show up for the workshop, we had no plan to change our team structure and modify our schedule, or when the requirements were unclear, we had no plan to seek the product owner for clarification. As pointed out by [9], a more thorough process plan leads to a higher quality product and faster delivery, and more importantly "is important to have a process capable of incorporating new demands, addressing all the development life cycle".

## IV. Software Process Improvement Techniques

With the software industry being one of the fastest growing industries in today's world, software companies in order to keep up with this pace and to hold their place in the competition, have to improve their software processes at an acceptable rate [5]. According to [7], most software process improvement(SPI) frameworks consist of 4 main steps: "an evaluation of the current practices, planning for improvements, implementation of the improvements and an evaluation of the effects of the improvements". They all work by measuring the current standards, process plans, way of working and how effective these are before any improvement. They then suggest improvement plans to replace the existing infrastructure and measure their effectiveness.

SPI frameworks generally categorize into inductive and prescriptive methods [7]. In an inductive SPI method, the initial concern is about understanding the current process and issues in the company, and setting improvement plans based on these issues. Since inductive frameworks are specific to the company, it is required to have an understanding of the maturity of the process in that company before providing any process improvement plan to ensure the plan is accurate, useful and specific to the situation [7].

Examples of inductive SPI methods are Quality Improvement Paradigm (QIP) and iFLAP [7]. QIP is based on first understanding the current issues in a process and then deciding on concrete improvements plans; for instance, Goals, Questions and Metrics (GQM) being one example of QIP, is a method where firstly, a goal has to be defined with the aim to improve an issue of an existing process area from an specific point of view, then a set of questions are asked to define the assessment of the goal often by asking about the existing issues, and finally metrics which are a set of quantitative measurements to answer each question [3]. iFLAP on the other hand, consists of 3 main steps; selection, assessment and improvement planning; in the selection step, selection of roles in different parts of the company takes place to find the people responsible for various process areas and furthermore, a good number of recently completed projects are chosen for assessment. In the assessment step, interviews with the selected roles happen to gather issues in various parts of the company, and finally improvement planning happens during which issues to be improved are prioritized and ordered based on the needs

of the company [7]. Advantages of iFLAP are that it is lightweight so it can easily be used in small organizations and also that it can be used for a single process area rather than for the whole process [7].

On the contrary, prescriptive frameworks do not always consider the current process of the company as their initial step; they extract process attributes that have shown successful results in other companies and researches, as a basis of coming up with an improvement plan of the company regardless of any specific issues the company is facing [7]. The downsides of this approach according to [7], are that the improvement plan is not specific to the company and it may address the issues that are not the most important in the specific case leading to limited improvement in the process of that company and also, since prescriptive methods usually take very long to implement, a large cost is often associated with following those methods.

Examples of prescriptive SPI methods are Capability maturity model integration (CMMI) and SPICE [7]. In CMMI, it is believed that there is a standard process with specific goals and practices that should work for all companies and more specifically, it states specific goals and practices for specific areas of the process [7]. SPICE also works similar to CMMI by having set goals and practices but allows for these goals and practices to become domain specific, addressing issues of the specific company [7].

I believe, both inductive and prescriptive SPI methods can be used to improve our process and have the desired effects. Following an inductive method to analyze the current process can be very beneficial since after the first workshop, we realized that our existing process plan was not good enough and lacked sufficient detail. We would also benefit from following the best practices of the CMMI model since as described in section III, we had issues in specific areas such as requirement clarification and predicting risks, and by following CMMI practices, we would minimize these issues and improve our process to be as effective as possible.

## V. SPI IN INDUSTRY

As mentioned by Dulce Goncalves [11], Siemens was a large company that was initially following a waterfall process. As their company grew and became more complex, they faced issues such as not being fast in delivering products and not being efficient in their work. This was when they felt the need to improve their process and follow an agile methodology with the aim of having small deliverables and continuous feedback. They started their agile transformation by including customers in their process; this had the benefit of customer satisfaction since customers could follow along and get involved with the development of their product and also benefited the development team as they could get continuous feedback along the production. As one step to improve their process, Dulce claimed that in Siemens, they created goals following GQM standards but without having any metrics for their goals, and this was because they wanted to make the teams focus more on the issue and the improvement rather than the measurements.

Some of the benefits of this process improvement in Siemens as highlighted by Dulce [11] are that cross disciplinary teams can work together and instead of developers making all the decisions as was the case previously, now the UX designers, testers and other teams can also decide on the product, also rather than having a hierarchical model where people are told what to do and everyone follows one large detailed process, now the organization is more flat, people are empowered and more honest and hence value teamwork more.

One barrier to performing their process improvement as mentioned by Dulce [11], is that agile culture, being very open and transparent, emphasizes on the team and individuals to make decisions and to be creative rather than following tasks set by their managers, and in Siemens this initially had the negative effect that since every member had to take responsibility for their work and their mistakes, and not being able to blame the management anymore, a lot of them felt very uncomfortable and no longer wanted to be a part of this new process. This is similar to the report by Niazi [8], that lack of support was a big issue in various companies in Vietnam and Australia; it is required for a successful SPI that everyone in the team shows support and has the will to make a change.

Furthermore, as described by Dulce [11], since Siemens was one of the first large companies to use agile, there was little experience on how to drive agile product development and what their company should do to implement that. This resulted in them facing another issue in this process improvement that their management had little skills and knowledge to drive the agile transformation and to train the teams of the agile culture. This issue is similar to an important issue in a number of companies in Vietnam as reported by Niazi in [8], that lack of project management and that their managers were not trained well about the suitable practices needed to perform SPI, was a big barrier to SPI.

In the lecture by Robert Engberg [12] about 1928diagnostics, he discusses the importance of improving the process through continuous monitoring and measuring the different aspects of the process to ensure that the process is as efficient as it can possible be and that it helps towards achieving the goals of the company such as fast delivery time to market or customer satisfaction. These aspects include organization and leadership, knowledge and resources, and size and number of projects. More concretely, Robert mentions that they use Key Performance Indicators (KPIs) such as burn down charts in order to measure the work completed in each sprint, and using this, they evaluate the effect of each of the above process areas on the productivity of the teams.

Furthermore, in the lecture, Robert talks about the regulations that apply to them as a medical device manufacturer. These regulations apply to the process, the product and the marketing of medical devices with the intention of ensuring patient safety. However, I believe these regulations can act as a barrier for SPI; meaning that they cannot perform SPI as freely and to its full potential, and are bound by adhering to these regulations. Moreover, Robert [12] discusses the importance of maintaining the privacy and confidentiality of patients in their software and that since nowadays all the information is digitalized, more people have access to this information and the rate of stealing these information has increased. As a result, their security team had to implement encryption in their systems, improve their password management, improve their login system and control access to hardware to prevent any data breach. This can also be a challenge for performing SPI since similar to having to

follow many regulations, having to follow strict security measures can prevent SPI being carried out as it should and limits how various data can be analyzed for improvement.

In connection to our process, I believe our issues were not similar to the issues mentioned by Robert. This is partly because our process was very short compared to a software process that happens in the industry, and also that we did not have as many regulations to adhere to as in 1928diagnostics. However, one barrier that Siemens faced about their management not being highly skilled in performing SPI, can relate to our situation. Although this was not one of our main issues, it is needless to say that since all of our members were new to software processes, our response to issues has likely been weak and insufficient.

## VI. SPI Proposal for future Scrum Development Efforts

For the second workshop, we planned on achieving the following goals set according to GQM standards; "1 - Improve the understandability of the user stories from the perspective of the developers, 2 - Improve the quality of the sprint review from the perspective of the team, 3 - Decrease the time to solve disagreements about product implementations within the team from the view of the project management" [6]. According to [4], the first goal was chosen to eliminate any ambiguity the developers might have about the customer's requirements by ensuring that they understand each requirement thoroughly. The second goal was chosen because our team felt that the sprint reviews were not effective enough and we were not getting enough feedback from the product owner, and lastly, the third goal was chosen because at times of disagreement, our team spent a long time discussing what to do in that situation and this limited the valuable time dedicated for our development.

We also decided on various specific goals and practices according to CMMI models for various areas of our process. For our project planning, we chose "SG 2 A project plan is established and maintained as the basis for managing the project - SP-2.2 Identify and analyze project risks" [10], to identify and analyze the risks in our project, helping us achieve our third goal. More concretely, we decided to create a risk identification matrix before the workshop to help us quickly identify risks during the workshop and come up with solutions [6]. For example, we had identified risks such as members not attending or requirements not being clear, and set plans to mitigate the effects by negotiating product backlog with the product owner based on the number of developers and also assign one member to contact the product owner and discuss the requirements in case of ambiguity. To reinforce the aforementioned plan, other goals and practices such as, "SG 1 Actual project performance and progress are monitored against the project plan - SP - 1.3 Monitor risks against those risks identified in the project plan - SP - 1.5 Monitor stakeholder involvement against the project plan - SP - 1.6 Periodically review the project's progress, performance, and issues - SP - 1.7 Review the project's accomplishments and results at selected project milestones" [10], were chosen. Here, we decided that the scrum master identifies project risks that affect our speed, efficiency and the quality of our process every 10 minutes, and records them on a specific document, again working towards achieving our third goal [6].

The above practices also help us in achieving our first and second goals; the scrum master was responsible to ask the development team about the issues and checks the progress of the team every 10 minutes, recording it on the same document [6]. The scrum master also records the amount of time and the detail of feedback the product owner provides at each sprint review to monitor his involvement, helping to achieve our second goal. Furthermore, we also chose a number of specific goals and practices for the requirement management aspect of our process; "SG1 - Requirements are managed and inconsistencies with plans and work products are identified - SP 1.1 - Develop an understanding with the requirements providers on the meaning of the requirements - SP 1.3 - Manage changes to requirements as they evolve during the project - SP 1.4 - Maintain bidirectional traceability among requirements and work products - SP 1.5 - Ensure that project plans and work products remain aligned with the requirements" [10]. During the sprint planning meeting, we planned to discuss and negotiate the requirements with the product owner and clear any ambiguity for the developers. We also decided that the scrum master should create a document with all the requirements in each sprint and manage any changes that might arise during the development. This ensures that all the work that has been done is documented allowing the team to provide a complete overview of the product to the product owner in each sprint review and getting his exact feedback and areas of concern, benefiting our second goal [6].

Moreover, we chose a number of generic goals and practices such as "GG2 -Institutionalize a Managed Process - GP 2.4 Assign responsibility and authority for performing the process, developing the work products, and providing the services of the process - GP 2.5 Train the people performing or supporting the process as needed" [10]. We decided to assign roles and responsibilities to members of the team during the first sprint planning. This explicitly was designed to happen during the workshop to know which members have attended, eliminating the issue mentioned in part III of this report where few members could not attend the first workshop. We decided that we would randomly choose a scrum master who will be responsible only for documentation, negotiation with the product owner about the requirements, monitoring and preparing questions for the sprint retrospective, and the rest to be the developers. Although this reduces the number of developers, it greatly enhances the productivity of the development process as opposed to all members having to discuss requirements with the product owner and monitor the process individually. We also decided to assign one member to spend 5 minutes explaining the basics of Minecraft to other developers making sure they have the right skill set in order to help the development process to go smoother and more efficient [6].

Lastly, the measurements we decided to take, classified by each goal, are represented below. For the first goal we decided to measure the "Satisfaction of each developer of the details of each user story provided to them" where a higher satisfaction score represents little ambiguity and hence more understandable requirements, and also the "satisfaction of the team on the support from the product owner" where a higher

score means more involvement of the product owner and again measures how close we are to achieving our first goal [4]. For the second goal, we decided to measure the "satisfaction of the developers with the feedback provided by the product owner" where a higher score would indicate that the feedback given in the sprint review was detailed and valuable ensuring a higher quality sprint review [4]. The above measurements were planned to be taken through questionnaires during each sprint retrospective. For the third goal, we decided to measure the "Average time in minutes to solve a disagreement about product implementation" recorded by the scrum master before and after performing SPI, where a longer time indicates that the improvement plan was not successful in decreasing the time to solve disagreements [4].

## VII. Implementation of an SPI Initiative

The second workshop went much smoother than the first workshop. Our team managed to follow the plan and managed to perform almost all the practices we had decided on. During the workshop, we managed to have 3 complete sprints each consisting of a sprint planning, sprint review and a sprint retrospective. We managed to perform a risk assessment before the first sprint, noting down the unexpected risks such as having vague or changing requirements, shortage of members and unavailable product owners. This risk assessment significantly helped to reduce disagreement times since we were more prepared for conflicts and what to do in those situations. This was clearly visible through the results of the questionnaires the scrum master took during the sprint retrospectives.

During the first sprint planning, we assigned one member to be the scrum master and one person to explain the basic gameplay for the team. With the members having sufficient knowledge about the game, they no longer had to spend time searching for ways on how to perform certain tasks, and hence had a more rapid and coherent development. The scrum master firstly wrote the requirements on the group documentation and since we were 4 members in total, only 3 members were assigned to perform development tasks, and as a result our scrum master contacted the product owner negotiating on reducing these requirements.

Furthermore, we could manage to increase the understandability of the requirements provided to us by asking our scrum master to discuss them one by one with the product owner and ask for rigid acceptance criteria. This led to our increment being accepted in the first sprint review according to the product owner's expectations. This clearly shows that we have improved the understandability of the requirements and have achieved our goal. To further emphasize, the results of the questionnaire shown in Appendix A done by the development team about their satisfaction on the details of the requirements also support this point. The results show an average satisfaction score of 4.3 out of 5 meaning that general opinion of our group is that we 'agree' that the requirements were sufficiently detailed. With the requirements being clear and the scrum master documenting every aspect of the process, our team could benefit from the sprint reviews more than before. This was because the feedback was more about how well we had implemented our increment and moving towards improving the work by implementing further requirements and not about an alternative view of the requirement that we missed to implement. This led to members asking better questions and getting better answers and overall having a more beneficial sprint review. The result of the questionnaire shown in Appendix B about developers being satisfied with the product owner's feedback further proves this point. Here, the results also represent a relatively high average score of 4.2 out of 5 meaning that the general consensus of our group is that the feedback from the product owner was satisfactory.

Finally, we conducted short sprint retrospectives at the end of each sprint during which the scrum master asked the developers to answer the previously mentioned questionnaires and also to note down the progress, the issues and the identified risks of the previous sprint, to be able to make necessary changes to the following sprints. Although risk assessments happened at each sprint retrospective, our scrum master failed to perform assessments along the process. This is partly because the discussions and requirement clarifications with the product owner took longer than expected. However, due to the nature of the workshop and that the sprints were already very short, this did not result in major problems for our team and any issue could successfully be recorded in the sprint retrospectives.

## VIII. Summary and Lessons Learned

This paper represents an example of a software process that our team followed in a workshop together with improving our process using SPI methods for a second iteration of the workshop. It also discusses various cases of SPI in industry and where possible, links their challenges to some of the issues we faced in our process.

As a summary of this report, I believe accurate and mature planning is the most important aspect of a software process. A plan should not only contain a well-defined schedule of the process, but also to outline a plan for following the plan, a detailed risk assessment of the process and a procedure on how to change the plan if unexpected situations occurred. Furthermore, I believe that most SPI methods are beneficial for any organization regardless of their domain and their specific issues, as was the case in our situation. We benefited from applying a domain specific inductive approach as well as applying a more general approach based on good practices as in the prescriptive approach.

Lastly, as mentioned in section VII, the SPI techniques we used, had a positive effect on our process, productivity and efficiency of our development, and with all of our team members being satisfied, we will continue to use the same techniques for future process improvements.

## Appendix A

| Metric | Sprint 1 | Sprint 2 | Sprint 3 | Average |
|---|---|---|---|---|
| A Likert scale from 1-5 of satisfaction of each developer of the details of each user story provided to them | 3.7 | 4.5 | 4.7 | 4.3 |

## APPENDIX B

| Metric | Sprint 1 | Sprint 2 | Sprint 3 | Average |
|---|---|---|---|---|
| A Likert scale from 1-5 of satisfaction of the team on the feedback from the product owner | 3.9 | 4.1 | 4.6 | 4.2 |

## REFERENCES


[1] P. E. McMahon, P.-W. Ng, H. B. Lawson, I. Jacobson and M. Goedicke, "The Essentials of Modern Software Engineering: Free the Practices from the Method Prisons!," Association for Computing Machinery and Morgan & Claypool, 2019.

[2] [Undisclosed Authors], "Process Plan", Assignment 2 in course DIT347 H20, Software Engineering Program, University of Gothenburg, Sweden, 2020

[3] V. R. Basili, G. Caldiera and H. D. Rombach, "THE GOAL QUESTION METRIC APPROACH," FB Informatik, Kaiserslautern.

[4] [Undisclosed Authors], "Process Analysis and Improvement Goal Formulation", Assignment 4 in course DIT347 H20, Software Engineering Program, University of Gothenburg, Sweden, 2020

[5] M. Y. a.-T. M. S. Abdullah and A. B. M. Ali, "A proposed methodology for establishing software process development improvement for small software development firms," *Procedia Computer Science,* vol. 3, pp. 893-897, 2011.

[6] [Undisclosed Authors], "Concrete process improvement", Assignment 6 in course DIT347 H20, Software Engineering Program, University of Gothenburg, Sweden, 2020

[7] F. Pettersson, M. Ivarsson, T. Gorschek, and P. Öman, "A practitioner's guide to light weight software process assessment and improvement planning" Journal of Systems and Software, 81(6), pp. 972-995, 2008.

[8] M. Niazi, M. A. Babar and J. M. Verner, "Software Process Improvement barriers: A cross-cultural comparison," *Information and Software Technology,* vol. 52, p. 1204–1216, 2010.

[9] A. Ianzen, E. C. Mauda, M. A. Paludo, S. Reinehr and A. Malucelli, "Software process improvement in a financial organization: An action research approach," *Computer Standards & Interfaces,* vol. 36, no. 1, pp. 54-65, 2013.

[10] CMMI Product Team, "CMMI for Development, Version 1.3," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-2010-TR-033, 2010.

[11] Dulce Goncalves, "Lecture 8 – Agile Transformation", Lecture Slides, DIT347 H20 Software Development Methodologies, Software Engineering and Management Program, University of Gothenburg, 2020.

[12] Robert Engberg, "Lecture 9 – Medical Device Software", Lecture Slides, DIT347 H20 Software Development Methodologies, Software Engineering and Management Program, University of Gothenburg, 2020.