Software Processes and Software Process Improvement

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 covers the process used at the Scrum workshop, experiences and problems in the Scrum workshop and SPI of the process used in the scrum workshop. It also covers tools and techniques used for SPI and provides a view of how SPI is used in the industry.**

I. Introduction

This report is an examination report in the course in the course DIT347 Software Development Methodologies. The course pertains to software processes and software process improvement (SPI). The purpose of this report is to describe and analyze a few aspects of SPI as well as my experiences with SPI.

A software process is a process used to structure the development of software. Software process improvement (SPI) is the endeavor of making a software process better in some desired way. There are many techniques which can be used for SPI, a few of which will be covered in section IV.

II. Process Applied in the Scrum Workshop

When planning the process for the scrum workshop my group and I had some misconceptions of how the scrum workshop would be structured. We believed that teams would be assigned a backlog each of buildings in their part of the city and be able to work on multiple user stories at once.

The group decided to base our process on Scrum. A requirement for the process was that it should resemble Scrum and the group thought it was easier and more efficient to base our process on scrum rather than to build a new process from the ground up. The three hours available for the workshop were split into three sprints with 45 minutes of work and a 15 minute review and reflection at the end. Daily Scrums were removed as the sprints would be too short. E.g. if a 15 minute daily Scrum is done four times per sprint and the sprint is an hour long, then there would be no time left for work. As there were no daily Scrums the group decided that the team should be in constant communication in order to quickly and efficiently resolve issues and collaborate. The group decided that planning poker should be used in order to estimate user-stories so that the optimal amount of developers could be assigned to each user-story.

Two roles where defined: developer and scrum master (SM). Developers were responsible for construction of the user-stories and the scrum master was responsible for communication with stakeholders and handling of the scrum board. Everyone in the team was a developer and one person was a developer and SM. Roles where used in practice as planned.

At the program level the group decided that product owners (PO) should be responsible for updating their teams with necessary information such as changes in requirements. The teams scrum master (SM) should be responsible for contacting the teams PO for questions or resolving conflicts between teams. The POs should be responsible for resolving conflicts between teams such as two teams attempting to build in the same location.

In practice the process used in the scrum workshop was not what we had planned for. When assigning user-stories the SMs from each team had a meeting were SMs called out which story their team wanted and was assigned one story, stories were not designated to teams or different parts of the city. Our team skipped planning poker since we only got one story so estimating its difficulty was unnecessary. We got no updates from our PO but contacting our PO for requirements went as planned. The sprint lengths and structure ended up after the first sprint as planned.

During the Scrum workshop the team forgot to use a scrum board, probably due to the user-story distribution working in an unexpected way. As the group had not planned for how tasks would be distributed or formalized, the team members chose their own tasks. Tasks were often half finished as team member switched tasks and there was no way to tell if a task was finished other than asking the other team members.

III. Experiences in the Scrum Workshop

The experiences of the Scrum workshop can be neatly summarized with one word, chaotic. The team often felt confused as there were issues in communication and changes to the process after the first sprint. There were also issues with organization and workflow for the team as tasks were not handled in a structured way. During the workshop there also appeared issues which were not planned for in the process, ownership of resources was the most pressing of these issues.

What about reviews? PO accessibility?

As mentioned in section II, how tasks should be handled was not planned for in the process. Tasks were not documented or managed, a scrum board was not used by the team. As there were no manager roles, no team member had the responsibility of managing the task distribution and workflow. While it was the SMs responsibility to see if the user-stories were finished, the SM was also a developer and did no have the time to see what was finished and what needed work. The unstructured tasks led to a lot of time being spent switching between tasks and ultimately to slow deliveries of the user-stories.

J. Bach claims that lack of collaboration and understanding leads to processes not being followed[7]. This might be the reason the team did not use a Scrum boards.

Other way around?

Ownership of resources was another area for which no established process or guidelines existed during the Scrum workshop. In the Minetest video game there are resources which can be collected, e.g. sand, and can be further processed, e.g. smelting sand to glass. A few of the groups took resources which other groups had collected and processed. Another group claimed in an interview that their resources had been stolen. The interviewed group said it was infuriating and led to slower delivery of user-stories as the resources had to be re-obtained. No software process literature found during a search of the internet mentioned the issue of resource ownership in a way comparable to the issue during the Scrum workshop. It can be assumed that this issue is not general.

During the Scrum workshop it was not clear which communication channel should be used to contact stakeholders and other teams. Which communication channel to use was never explicitly defined in the process used. This became a problem when needing to contact POs for requirements on a user-story and when needing to contact another team to resolve a conflict. The communication issue was resolved a bit towards the end of the Scrum workshop as one communication channel slowly became the preferred one. Communication is an area of great concern in large organizations as well[8].

Production of the city went well despite the shortcomings in process quality. At the end of the Scrum workshop the product definitely looked like a city, which was the goal of the endeavour.

IV. Software Process Improvement Techniques

A. Basic cycle of SPI

Basic cycle of SPI, after [1], is a model for the SPI lifecycle. The cycle consists of four steps, each step leading in to the next with the forth leading back to the first.

1. Evaluate current situation.
2. Plan for improvement.
3. Implement the improvements.
4. Evaluate the effect of improvement.

Different types of SPI frameworks have their strengths in different steps of the cycle. Inductive, or bottom-up, frameworks are most useful in step one and four of the cycle. Prescriptive, or top-down, frameworks are most useful in steps two and three of the cycle. Prescriptive and Inductive frameworks can be combined using the cycle.

The basic cycle of SPI is applicable to any SPI effort including improving the issues noted in section III.

B. GQM

Goal/Question/Metric (GQM) is an SPI technique used to guide and measure an SPI endeavour. The technique can only be used once a process' improvement needs have been elicited. GQM consists of three levels, the first of which is the goal. A goal is created for an area of improvement in the process, e.g. decrease time needed per code review from the perspective of senior developers. Goals must follow a clear structure, they must have a purpose (decrease in the example), an issue (time needed in the example), an object (code review in the example) and a viewpoint (senior developers in the example). On the second level of GQM we have the questions. Questions are used to determine when the goal has been fulfilled and give useful information for how to fulfill the goal. E.g. What is the current average time per code review? The questions need to be answerable using measurements, the example question can be answered with a time measurement. On the third level of GQM are the metrics. Metrics are used to answer the questions from the second level through measurements. A metric can be seen as a function that takes some aspect of the world as input and gives a numerical measurement as output. E.g. Average time in minutes it takes a senior developer to complete a codereview, data collected though self reporting. Metrics need to have a data source and a unit of measurement, otherwise they cannot be used in practice.

Birk and colleges writes that GQM is in practice beneficial for meeting improvement goals, increasing stability of SPI programs and improve group synergy in an SPI effort[2]. How well GQM works is dependant on multiple factors including management commitment, measurement process and planning of measurement programme [2]. A notable disadvantage of GQM is that it is not useful unless the areas for process improvement have already been identified as it does not provide a way to identify them.

GQM can be a useful tool for improving the issues noted in section III. As the areas for improvement have been identified GQM can be applied. However, for other areas of improvement in the Scrum workshop process which have not been identified GQM cannot be used.

C. CMMI

Capability Maturity Model Integration (CMMI) is a prescriptive (top-down) SPI framework. A prescriptive framework defines a set of best practices for software development processes with the goal of SPI efforts being to get closer to the defined standard. CMMI in particular also defines maturity levels which indicate how mature an organizations processes are and how closely it follows CMMIs best-practices.

Significant improvements in developer productivity, customer satisfaction and cost has been noted by companies after adopting CMMI[3]. CMMI has a history of success and provides clear guidelines on how to adopt its practices. There

is also consultation help available for CMMI as it is well used in industry.

A notable disadvantage of CMMI is that all its practices may not apply to a particular organization such that the organization can never achieve a high maturity level. CMMI does account for this shortcoming by also defining an alternative to maturity levels which is useful if an organization does not want to apply certain CMMI practices. Another disadvantage of CMMI and prescriptive SPI frameworks in general is that its practices may not actually be the best for a certain organization, one size does not always fit all.

CMMI can be used to tackle the issues identified in section III by applying its practices to the relevant areas.

D. QIP

Quality Improvement Paradigm (QIP) is an inductive (bottom-up) SPI framework. Inductive frameworks are based on analyzing the current software development process and determining what to improve and how. While using QIP does not require using GQM or any other measurement tool, GQM is quite compatible with QIP as both are based on setting goals and working to achieve them[4]. QIP proposes using pilot projects to test and analyze process improvements before rolling out the process improvements for the entire organization.

The main advantage of QIP and other inductive SPI frameworks is that the SPI found using them are tailored to the organizations needs. QIP is also focused on organizational learning so that only measurable improvements are implemented in the whole organization.

The main disadvantage with inductive SPI frameworks is that they do not provide improvements out of the box unlike their prescriptive counterparts. Also, when using QIP, the analysis of the current processes must be thorough which is very time consuming.

QIP can be used for SPI of the issues in section III, however as issues are already identified it may be more efficient to use a prescriptive framework.

V. SPI in Industry

A. Robert Engberg at 1928 Diagnostics

Robert Engberg gave a guest lecture about SPI at the medical technology company 1928 Diagnostics. In the medical industry there is heavy regulation from both state and international agencies. Not complying to regulations can cost a lot in fines and lead to injury or death for patients. Checking for compliance with regulations can be very time and resource consuming. So for it is critical for a medical technology company like 1928 Diagnostics to have processes which lets them follow regulations while also working efficiently.

At 1928 Diagnostics multiple SPI tools are used. The plan-do-study-act (PSDA) model is used as the basis for SPI. PSDA is similar to the basic cycle of SPI, described in section IV. The company also uses metrics such as performance indicators and maturity measures. These metrics are used to determine success of SPI efforts and areas for improvement in future SPI efforts.

The process used at 1928 Diagnostics is a combination of the waterfall structure and an agile process. Agile methods such as definition of done are used to ensure software quality and regulation compliance. The product development process resembles the waterfall approach with the distinction that development is done iteratively in sprints.

Agile methods of ensuring code quality such as definition of done (DoD) can be very useful when working with safety critical code. However such a method also has its disadvantages. In the course DIT257 Agile software project management our team added a lot of constraints to the DoD in order to produce high quality code. These constrains where not applicable to all user-stories and in some cases where very time consuming to fulfill. Our DoD managed in some cases to make our process less efficient as time was spent fulfilling constraints which contributed little to quality. In my experience it is very important to have a good configuration of agile methods in order to get their benefits.

B. Dulce Goncalves as Siemens

Dulce Goncalves gave a guest lecture about transitioning health care division of Siemens from a traditional software process to an agile software process. Growth in the health care division had been causing problems with the current waterfall based software process. The division, before transitioning to agile, had attempted to switch to an incremental version of waterfall without success. As SPI was needed and agile software processes were a promising new alternative to waterfall, the department decided to transition to agile. Goncalves said that there were many potential benefits from switching to agile, e.g. better management of growth, reducing costs and reducing time to market.

There were multiple obstacles along the way in the transition to agile. A major issue was the established company culture. The opaque and hierarchical established culture was not suited to an agile process, which requires a more open and flexible culture. Getting multiple levels of management involved was critical to transition the company culture to one more compatible with agile processes.

Initially when the division transitioned to agile, four levels of Scrum where used to manage the development. Later the four levels where cut down to three and then two in order to reduce complexity of managing development and allow developer to manage themselves more.

In the end, the transition was a success. The scalability of an agile process allowed the division to more easily manage its growth. Software quality increased and once the transition was complete, deadlines were met more often.

When working with an agile process in the course DIT257 Agile software project management, the experience with culture was similar to what Goncalves described. Initially the team had no experience working with an agile process. The culture did not have the flexibility that was required for an agile process to work efficiently. After the third sprint the

No questions

culture had changed to one more focused on collaboration and openness and the work progressed more efficiently.

VI. SPI Proposal for future Scrum Development Efforts

There where a few issues with the process (and lack of following the process) as mentioned in section III. The goals should then be to improve the most pressing of those issues which are planning and communication.

The first goal (1) is to improve task planning and task execution. A lot of time was wasted on developers switching tasks and not knowing which tasks had been finished or where worked on by the team.

VP?

Another goal (2) is to improve communication both within the team and with stakeholders and other teams. Getting in contact with other teams and POs was troublesome during the Scrum workshop. Communication within the team was also a problem as the status of different tasks was not communicated to the whole team.

A third goal (3) is to reduce the amount of ownership conflicts between teams, issue described in section III.

These goals will be attempted to be achieved though applying relevant practices as prescribed in CMMI[5].

A. Goal 1

Following generic practice (GP) 2.4 "Assign responsibility and authority for performing the process, developing the work products, and providing the services of the process".

Following Project Planning (PP) specific practice (SP) 1.2 "Establish and maintain estimates of work product and task attributes"[5]. A scrum board will be used, which was planned for in the Scrum workshop but never used in practice. User-stories will be broken down into tasks and estimated during the sprint planning phase. The SM will be responsible for managing workload during the sprint, so that the SMs estimate of required developers are working on each task. The SM will be also be responsible for contacting POs for requirements, updating requirements for each tasks and verifying requirement fulfillment following REQM SP 1.3[5]. This also ties into generic practice (GP) 2.4 "Assign responsibility and authority for performing the process, developing the work products, and providing the services of the process"[5].

As the team now has a member responsible for managing tasks, requirements and workflow, it should be easy for developers to ask SM what they should be doing. This would, in theory, remove the time spent for developers between tasks.

B. Goal 2

Following Organizational Process Definition (OPD) SP 1.1 "Establish and maintain the organization's set of standard processes"[5]. Establishing defined communication channels between the different parties would remove the uncertainty of which communication channel should be used. Slack will be used for communication between teams as well as for getting in contact with POs. As Slack appeared to be the most widely used communication channel by the end of the Scrum workshop it is a good candidate. Zoom will be used for communication within teams as in the Scrum workshop as it worked well.

C. Goal 3

Following again, ODP SP 1.1, a definition of resource ownership is needed. Any resource collected, placed and or refined by a team is owned by that team and cannot be taken by any other team. Any area of land which has been marked by a team is also owned by that team and no other team can build there. These definitions should clarify which team owns what so that conflicts over ownership between teams should occur less.

D. Measuring SPI success

Each developer should record, during the sprint retrospective, how much time was spent in between tasks that sprint. Each developer should also record how efficiently they worked on their tasks on a scale from one to five that sprint. These measurements are used to validate the success of goal 1.

Each team member should record, during the sprint retrospective, if there was any communication problems during the sprint. If a communication problem is recorded it should be breifly described and its description recorded. This will be used to validate the success of goal 2 as well as give information for future improvement in communication.

The team should record how may conflict with other teams occurred during the workshop, this will be used to validate the success of goal 3. For the team not to miss recording any conflicts, when a conflict occurs the team should increment a counter.

VII. Implementation of an SPI Initiative

A. SPI in small organizations

Small organizations consisting of 10 people or less make up the vast majority of IT companies in Europe[6]. These organizations have different challenges in SPI than the tech giants people typically think of when thinking of IT companies.

Common challenges for small organizations include: everyday work and deliveries being affected by SPI efforts, high costs, personnel lacking SPI skills and SPI models being too complex[6].

SPI for large organizations may be a lot easier than for small organizations. Large organizations have vast resources compared so small organizations and can afford to use large prescriptive frameworks such as CMMI and SPICE. Small organizations opt to not use these frameworks as they are seen as too costly, require too much documentation and bureaucracy[6]. There is an SPI framework, ISO/IEC 29110, which is designed specifically for small organizations of 25 people or less. ISO/IEC 29110 is becoming the standard framework used by small organizations[6].

Larrucea and colleagues[6] are working on creating an "experience factory" to help small organizations in their SPI efforts. The aim is to use their experience and research as well as the collaboration of small organizations to make SPI efforts

easier for small organizations who participate. While I think this is a good idea in theory, I have a hard time seeing it being widely used as organizations will be unaware of its existence unless it get a lot of exposure.

Unless they look for it ...

In my experience using CMMI and learning other SPI frameworks has felt overwhelming. As the organization, my team in the Scrum workshop, was small I encountered some of the problems previously noted for small organizations. CMMI felt like it was too complex for the basic SPI which was done after the Scrum workshop. CMMI also felt like it required too much resourses, my time, to be used efficiently in the short time-span given. Maybe ISO/IEC 29110 would have been a better fit for the Scrum workshop SPI.

B. Cost versus benefit of SPI

Resource constraints is an obvious problem in SPI efforts. Every organization has resource constraints, time and money are two such constrained resources. For a rational organization to consider SPI, the SPI efforts must cost less resources than are created or saved by the improvements provided. Where to apply SPI is also an area of concern as some areas may be in much greater need of SPI than others. Solingen proposes calculation of return on investment of SPI as way to quantify the benefits of an SPI endeavor[9].

Measuring return on investment (ROI) is easy as the exact number is not that important, just if it is positive and of what magnitude it is[9]. Costs can be estimated by e.g. working hours spent, release delays and purchases. Benefits can be estimated by e.g. working hours saved, perceived value of improvements by stakeholders.

A criticism of using ROI as a basis for benefits of SPI is that many benefits (e.g. less stress, better management) do not directly translate into money. Solingen disagrees, claiming their monetary value can be measured[10]. The method he proposes is to ask an affected party e.g. a manager, what they would be willing to pay for such a benefit e.g. less stressed employees.

In section V-A, one of the benefits of SPI considered was better compliance with regulation as the fines for non-compliance are vast. Here the monetary benefits of ROI are easily measurable as the fines are already monetary. Maybe once 1928 Diagnostics have improved their processes enough, ROI measurements can be used to determine that a focus on regulation compliance is no longer profitable.

I believe that ROI is a useful tool for determining the benefit of SPI. Money is an easy way for people to understand value so ROI gives a good indication of the benefits of SPI. Though I do not think that ROI is everything that should be considered when considering SPI efforts as value can become visible long after ROI has been determined.

VIII. Summary and Lessons Learned

Having a good software process is critical so the success of an organization developing software. SPI is not easy however there are a lot of good tools which can help in an SPI endeavour. When working with SPI it is important to follow a process just like in software development.

If more SPI where to be done to the process from the Scrum workshop I do not know what I would have done differently. The results of the SPI workshop would be needed to see if the improvements worked or not. If the improvements did not work I may try using other SPI tools to get a better idea of what needs to be done. If the improvements were a success I would use the same methods but look for other areas of improvement.

References

[1] 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, 81(6), pp. 972-995, 2008

[2] Birk, Andreas & Solingen, Rini & Järvinen, Janne. (1999). Business Impact, Benefit, and Cost of Applying GQM in Industry: An In-Depth, Long-Term Investigation at Schlumberger RPS.

[3] Gibson. Diane, Goldenson. Dennis, and Kost. Keith, ”Performance Results of CMMI-Based Process Improvement,” Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-2006-TR-004, 2006. http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=8065

[4] T. Gorschek, ‘Software Process Assessment & Improvement in Industrial Requirements Engineering’, Licentiate dissertation, Blekinge Institute of Technology, Karlskrona, 2004.

[5] Software Enginering Institute, “CMMI for Development, Version 1.3”, 2010.

[6] X. Larrucea, R. V. O’Connor, R. Colomo-Palacios and C. Y. Laporte, ”Software Process Improvement in Very Small Organizations,” in IEEE Software, vol. 33, no. 2, pp. 85-89, Mar.-Apr. 2016, doi: 10.1109/MS.2016.42.

[7] J. Bach, ”What software reality is really about,” in Computer, vol. 32, no. 12, pp. 148-149, Dec. 1999, doi: 10.1109/2.809258.

[8] Beecham, S., Hall, T. & Rainer, A. Software Process Improvement Problems in Twelve Software Companies: An Empirical Analysis. Empirical Software Engineering 8, 7–42 (2003). https://doi.org/10.1023/A:1021764731148

[9] R. van Solingen, ”Measuring the ROI of software process improvement,” in IEEE Software, vol. 21, no. 3, pp. 32-38, May-June 2004, doi: 10.1109/MS.2004.1293070.

[10] R. van Solingen, ”A Follow-Up Reflection on Software Process Improvement ROI,” in IEEE Software, vol. 26, no. 5, pp. 77-79, Sept.-Oct. 2009, doi: 10.1109/MS.2009.120.