VG

Evaluation of a Small Scale Agile Process Plan and SPI Effort

+ Excellent use of literature
- Very brief SPI techniques

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 paper is an analysis of the process plan and SPI effort of a small team (of which I, the author of this report, am a member) taking a course in Software Process Methodologies. The initial plan will be discussed, together with the subsequent SPI effort, which used iFLAP, CMMI and the Sidky & Arthur model. GQM was used to formulate goals and define metrics for the SPI effort. The resulting SPI effort ended up successful, although process problems still existed, in part due to not fully implementing the SCRUM framework.**

I. Introduction

A software development process consist of two components: a life cycle and method content. A life cycle is a way of dividing up a process into distinct phases. Examples of life cycles is the waterfall life cycle and the iterative incremental life cycle. By filling a life cycle with method content, one creates a process.

SCRUM, Kanban and Extreme Programming are sometimes seen as processes, but are rather frameworks for implementing one. They contain a suggested life cycle and suitable method content, but a team has to use these components to actually define a process that fits their purposes. According to Williams [1], this is often the way processes are formed.

Good!

When a team or organisation defines a process, it's rarely perfect. An SPI effort is likely necessary. SPI, or Software Process Improvement, is a way of continuously refining an already existing process. Ashrafi [2] states that SPI is performed in order to ensure work product quality, since this is closely tied to process quality. He also explains that SPI consists of a set of tools, practices and techniques used to elicit improvement needs and implement the required changes to address those needs.

This report will study the process of the team I, the author of this report, was a member of. My/our findings and experiences will be detailed, along with the process defined for a first SCRUM workshop, the SPI initiative which undoubtedly was required, and the results from a second SCRUM workshop (both workshops took place in a simulated development environment using the Minetest game). The report will also analyze whether this SPI effort was successful or not.

II. Process Applied in the Scrum Workshop

For the first workshop, an agile process based on SCRUM was planned. We studied typical SCRUM setups and analyzed common agile practices in order to structure our process. The essence language was used as a modelling tool. Here follows the planned process activities:

Initial meeting, find user stories (pre-sprint meeting), prepare user stories (pre-sprint meeting), design system, build, integrate increment, deploy increment, evaluate increment, review meeting (post sprint meeting), retrospective meeting (post-sprint meeting), final meeting.

Two of the practices we intended to implement were the Incremental Delivery practice and the Iterative planning practice. Hence, a repeating life cycle (which is essential to SCRUM, as detailed by A. G. Permana [3]) was used. Each iteration, or sprint, started at the pre-sprint meeting and ended at the post-sprint meeting.

The user story practice was also implemented, which was especially suiting since each team would be supplied with ready-made user stories by the Product Owners (PO's). We planned to use a Trello board for our team level SCRUM board, where we'd break up the user stories into tasks.

We also planned a potential SCRUM of SCRUM meeting structure for handling a scaled process. A set of questions which each SCRUM master should answer during these meetings was created. These were the questions:

- *What has your Team done since we last met?*
- *What will your Team do before we meet again?*
- *Is anything slowing your Team down or getting in their way?*
- *Are you about to put something in another Team's way?*

To instantiate the process, we studied the documents describing the workshop. The SCRUM master role was assigned to one of the team members, and a SCRUM board was created. A schedule suggestion and a definition of done was also designed. Any necessary tools were installed, and a shared drive was established.

When designing the process, we mostly considered common SCRUM practices and roles. Therefore, we picked practices like ”Incremental Delivery” (using sprints and review sessions) and ”Iterative Planning” (using sprints and sprint planning sessions), ”Review” (holding reviews with PO's after each sprint), ”Responding to change” (using retrospective meetings and the SCRUM of SCRUMs), ”User stories” (using the finished user stories and managing them using our own Trello board) and ”Communication and Collaboration” (communicating with other groups using zoom or slack). These practices (and many others) are detailed further by Williams [1].

We had no practice for breaking down large user stories into smaller ones. One of the essential components of user stories is that they should be small, and if they are not, they should be broken down into more manageable user stories, as described by Trkman et al. [4]. We never considered this as an alternative when planning our process. We also didn’t implement the daily stand-up practice, however this practice was not viable in our case (the workshop took place during a single day). Similarly, we skipped practices involving continuous integration and writing tests, since this wasn’t feasible.

why not?

When applying the process, a lot didn’t go as planned. This might be due to limited experience and/or lack of concrete steps for implementing an activity or practice. We simply hadn’t planned precisely how we were to implement the process, who was to do what, how to interact with our tools, how to hold meetings, and so on.

Here follows a few significant differences to the planned process. We never performed the ”deploy increment” activity, since deployment in the case of the first workshop had no meaningful equivalent, which we should have realized during the planning phase. Our described process also had a big error: ”Evaluate Increment” was incorrectly placed in the ”Use the system” activity space, before the ”review” activity. This activity never took place either; activities from the ”Use the system” activity space can only be performed after the product has been rolled out to the customer or other stakeholders.

We also didn’t interact with the Trello board in the intended way. Tasks were mostly created by one or two people, people only verbally assigned themselves or switched tasks while in the middle of working on one. One possible reason for this is that we had no concrete plan for how to use the Trello board. In addition to this, the ”design system” activity wasn’t performed properly – implementation details were rarely added to the task description.

Although collaboration within the team worked well, the process was generally a mess. We didn’t interact with our tools properly, activities blended together, the design phase merged with the development phase, team members made ad hoc design and process decisions in the middle of sprints, etc.

Continuing, there was no actual SCRUM of SCRUMs. Instead, the SCRUM masters only met to pick user stories for their teams. The idea of a proper SCRUM of SCRUMs was never suggested, and the schedule didn’t seem to allow another meeting. We never proposed the usage of our prepared questions.

Finally, one team member (who also happens to be the author of this report) informally took on the role of secretary, and made some documentation and filled in meeting protocols. This was done haphazardly and inconsistently; we should have appointed a formal secretary and planned how to interact with our documents.

III. Experiences in the Scrum Workshop

The first SCRUM workshop was slightly chaotic, primarily due to improper planning. Some parts of our process wasn’t applied at all, and certain important SCRUM elements, such as estimates and velocity, wasn’t used. Throughout the workshop, it was also hard to reach other teams. No communication scheme had been agreed upon before hand, and therefore there were no free breakout rooms for team-team discussion. Because of this, two SCRUM masters sometimes had to discuss inside a team breakout room, thereby disturbing regular internal communication. It was also difficult to know which team was working on which user story, since there was no indication on the Trello board, and no team name appended to the names of the users in-game. Finding out who to contact was therefore based on trial-and-error alone.

Generally, team-team and team-PO communication was limited. One reason for this was the lack of a pre-determined communication scheme. We didn’t know if we were to contact other teams using zoom or slack or any other external tool. Communication was discussed briefly during the initial meeting, but due to poor participation and no framework for making collective decisions, lots of time was lost and nothing agreed upon.

This could have been foreseen if we’d conducted proper risk analysis. Since there was no specified channel for talking to the PO’s, we ought to have realized that they would jump between the zoom breakout rooms of different teams, making them difficult to find. This isn’t an uncommon problem according to Vijay Anand et al. [5] and Miller [6]. They explain how PO’s often are not present enough, sometimes due to handling a large set of requirements (in our case, some PO’s handled too many user stories). Teams at Google experienced similar issues, as described by Sutherland and Schwaber [7]. There, the solution was to let the teams make more important decisions. In the case of this workshop, the we could have asked the PO’s to trust us to take more of the important decisions ourselves.

Many issues could have been avoided if we spent more time thinking about scaling and discussing this with other teams beforehand. The program level of the workshop was not handled properly. An example of this is the review meetings. Initially, these were held in the main zoom room, with all the teams present. Here, each team got a sense of what the others were doing, but this had the side effect of taking up way too much time. We never discussed how to conduct reviews efficiently, and this solution became the default. After the first sprint, this structure was changed to make sure we we could keep to the schedule. Reviews were now to be kept in private, with the PO’s joining the breakout rooms of different teams. Although time was saved and teams didn’t have to wait as much, we now knew nothing about the work of other teams. A proper SCRUM of SCRUM, or at least better participation on slack, would have alleviated this issue. (Another side effect of the new review structure was that reviews and retrospectives often were held simultaneously, or in reversed order. Separation of activities was hard to maintain.)

Poor time management was another major issue, which made it difficult to adhere to the planned process. During the initial meeting, the first schedule suggestion was accepted without much discussion, likely due to a lack of participation. This was a common problem: we rarely reached a true consensus since only a few teams weighed in during discussions. Most often, one team made a proposal, and if no one objected, the proposal was accepted.

Regarding the issues with using the Trello board, mentioned in the previous section: this might have occurred due to insufficient planning and a lack of defined roles. A lack of assigned roles, or insufficient experience with SCRUM roles, are common reason for SCRUM failure, according to Miller [6]. This could be solved by training the team, especially those with important roles. This is echoed by Raza and Majeed [8]: new teams often have difficulties implementing SCRUM properly, with more training being proposed as a possible solution.

We also had trouble following the chosen activities correctly. Activities blended together, and some weren't performed at all. One reason for this was the missing secretary role, and the fact that many team members didn't properly understand the intended process. A few reasons for this: poor time management and not enough concrete steps to take for implementing the process, in addition to a lack of experience. Raza and Majeed [8] says being too idealistic and not being used to SCRUM are common issues. Velocity should have been lowered (if it was actually used) to account for this. Roles should have been defined. According to Wagner et. al. [9], unclear responsibilities are common reason for SCRUM failures. Not having enough time is another one.

IV. Software Process Improvement Techniques

An SPI initiative has four concrete steps: 1) Evaluate current situation, 2) Plan for improvement, 3) Implement improvement, and 4) Evaluate the effects of improvement. These steps are repeated until the desired result is achieved.

SPI frameworks can be categorized into two groups: inductive (bottom-up) and prescriptive (top-down). Inductive frameworks work by studying the organisation/team, eliciting improvement needs, and evaluating the current state of a process. Hence, inductive frameworks are good for step 1) and step 4) of the SPI cycle. Conversely, prescriptive frameworks are model based and provide a collection of best practices which an organisation or team might want to apply. These frameworks are most efficient for step 2) and step 3) of the cycle, since they can provide a gold standard to strive towards, as well as concrete improvement ideas.

Examples of inductive frameworks are QIP, which acquires knowledge about an organisation and defines issues which then can be resolved on project level, and iFLAP, which is similar to QIP, but more lightweight. Some prescriptive frameworks are CMMI, which provide concrete process areas and concrete practices, as well as a way of thinking about the current state of a process, and ISO/IEC 15504 (SPICE) which is a generic process model that can be refined for specific domains. When designing an SPI initiative, it's sensible to combine both inductive and prescriptive frameworks.

For our purposes, QIP has too much overhead. Instead, iFLAP was used to elicit improvement needs. CMMI was used for planning concrete steps for improvement. However, some parts of CMMI, such as staged representation, wasn't feasible in this case – it's simply not possible to implement all the process areas and their related practices. Instead, continuous representation was used, which allowed us to focus on relevant process areas and instead use the more manageable capability levels. Using CMMI, we were able to study state of the art practices specific to those issues we've identified in our process, for example practices regarding project planning, requirement management, and competencies. A target profile and a CMMI roadmap was used to guide the SPI effort.

V. SPI in Industry

Before the second workshop, all teams listened to two guest lectures that in one way or another was connected to SPI.

Robert Engberg at 1928 Diagnostics described the use of SPI in the field of medical device software. In this area, continuous feedback and product improvements are essential, and hence a well-structured process which effectively handles changes in requirements is vital. One method for conducting SPI at 1928 was to compare their own process to the IEC 62304, the gold standard within medical device software. Their process was also tracked using various feedback mechanisms, performance indicators and capability measures. Continuous improvement methods were then use to ensure process quality, although Engberg didn't go into detail regarding what those entailed.

For the second guest lecture, Dulce Goncalves helped the company Siemens with an SPI initiative which completely transformed the process of the company. Siemens went from a waterfall-like process to an agile one, with the purpose of reducing organizational complexity and various overhead. Since the company didn't have much experience with agile methods, they opted for a slow transition. One initial step was to connect traditional roles with more agile ones. For example, the System architect would work closely together with the Product Owner. They also set up various SCRUM meetings, ranging from the daily SCRUM, to a SCRUM of SCRUMs, to a "mega SCRUM" on the top level.

Initially, the teams resisted the change, which is common when changing a process, often due to limited experience, as described by Miller [6]. However, the SPI effort was ultimately successful, and Siemens managed to completely cut a layer of middle managers while making the teams more empowered and self-managing.

My team also had issues when implementing agile, likely due to limited experience with agile methods; we were all used to other ways of working. However, thanks to our SPI effort, our process for the second workshop was significantly improved, as will be detailed in the upcoming sections.

One major realisation we got from the transformation at Siemens was that the agile culture is extremely important. The culture, or the mindset, inspires creativity, independence and motivation, and is vital to fully implementing agile. Using a SCRUM-like process is not enough.

VI. SPI Proposal for future Scrum Development Efforts

Based on the experiences of the first workshop, we came up with three major improvement areas. We wanted to improve the communication between the teams and the PO's, improve the quality of sprint planning, and finally improve the competencies of the development team.

We defined three improvement goals using GQM, which according to Caldiera et al. [10] is a robust method for eliciting improvement needs and evaluating the quality of a process, as well as determining if an SPI initiative was successful. Our GQM's followed the structure Caldeiera et al. [10] describe in their paper. However, for the purpose of brevity, the questions are excluded and the metrics are simply summarized. Our goals:

- G1: "Improve the efficiency of team-PO communication from the viewpoint of the development teams."
- G2: "Improve the quality of sprint planning from the viewpoint of the development teams."
- G3: "Improve the competencies regarding the mechanics in Minetest from the viewpoint of the developers."

The metrics for G1 were mostly subjective, formulated as questions for a questionnaire, pertaining to time waiting for PO's, clarity of requirements, and fluency of team-PO communication.

The metrics for G2 consisted of a mix of subjective and objective measurements. They regarded time allocated for and quality of sprint planning.

Finally, the metrics for G3 were also mixed, consisting of questions regarding how comfortable the developers are with the game mechanics, the time spent doing basic crafting, and how well the user stories were implemented.

In order to implement our goals, a CMMI roadmap was defined. In a set of case studies performed by Goldenson and Gibson [11], CMMI proved useful for improving quality of work, accuracy of scheduling and satisfaction of customers, and hence it seemed useful for our purposes. Some generic practices we decided to implement were GP 2.2 (*Establish and maintain the plan for performing the process*) which relates to G2, GP 2.5 (*Train the people performing or supporting the process as needed*) which relates to G3, and GP2.8 (*Monitor and control the process against the plan for performing the process and take appropriate corrective action*) which again relates to G2.

Some of the more specific practices from the CMMI process areas were also studied, mainly those from Project planning (PP) which relates to G2, and Requirements Development (RD) which relates to G1 (many RD practices regards the elicitation of stakeholder needs and conducting communication).

In addition to this, a few practices from the Sidky & Arthur model were also implemented, such as ”User Stories”, ”Customer commitment to work with developing team” and ”Knowledge sharing tools”.

To implement these practices, a table was created containing all the concrete steps required. Most important changes are summarized here:

- Assigning additional formal roles, such as secretary (responsible for meeting protocols) and process secretary (responsible for taking measurements and monitoring the process).
- Estimating user stories using story points and handling workload using velocity. Also estimate the time each task would take.
- Deciding on a communication scheme for team-PO communication before the workshop.
- Request the creation of spare breakout rooms for team-team communication.
- Decide on a new iterative incremental schedule together with the other teams, before the workshop.
- Risk management in the form of collective discussions and backup plans (in the case of, for example, a lost connection). Also extend risk discussions to collective discussions with other teams. Finally, create a template document for risk assessment.
- Managing competency gaps by discussion, practice and knowledge sharing (using new communication channels for sharing knowledge about Minetest).
- Create documents for taking measurements (a questionnaire for subjective measurements and a spreadsheet for objective measurements).
- Discuss resource management within team, including Trello board and shared documents.
- Update meeting protocol templates to reflect new process.

We planned to use the role of the process secretary to handle the objective measurements. The process secretary was to use a spreadsheet which had been prepared beforehand, containing the appropriate rows and columns for each metric. The measurements would then be filled in by the process secretary during the second workshop.

In addition to this, a questionnaire was created containing questions for all the subjective measurements. A copy of this questionnaire was taken once before the second workshop, with respect to the first workshop, and once after the second workshop (with respect to the second workshop). The answers would then be compared in order to make conclusions regarding the success of the SPI effort.

VII. Implementation of an SPI Initiative

Most of the planned changes were actually implemented, however with varying impact.

The formal secretary role greatly helped with managing meetings and going through the intended meeting points, and filling out the protocols. This, in turn, made the entire process progress more as intended. Similarly, the process secretary helped making sure the measurements were taken, however, the person assigned to this role seemingly didn't monitor the overall process, and never raised any issues during the retrospectives.

The effectiveness of our newly implemented estimates and use of velocity was difficult to gauge. Since we only worked on a single user story throughout the workshop, we only made one user story estimate and we never used our current velocity to decide how much work to take on. Still, for the purpose of adhering to the process, we adjusted our velocity based on the workload of the previous sprint. The velocity stayed mostly the same, but was changed when we unexpectedly gained another team member.

Similarly, we did break down our user story into tasks, however many tasks were too big. Several tasks, such as those involving gathering materials, spanned over the entire sprint (or several sprints). Our task estimates (in the form of the number of minutes the developer(s) working on it thought the task would take to complete) was rarely used and mostly inaccurate. Also, the task estimates were sometimes performed by the secretary, and not by the developers themselves.

Risks were discussed, but only barely. A plan for what to do if someone lost connection the the Minetest server was defined, but was never used. We also planned for resource management, for example by deciding who would do what, where the documents would be stored, how the Trello board would be managed, etc. However, we never discussed potential risks with other groups on slack. This was a conscious choice on our part: participation on slack was so low we deemed it not worth it. We also never made a template for risk assessment.

It's unclear if the creation of the collective knowledge sharing channel had any effect, since, as mentioned, participation was low. At least our team felt more comfortable with the game mechanics, which some of our metrics (detailed below) will indicate.

Team-PO communication was significantly improved thanks to a proposal done by another group, which we sided with. In contrast with the first workshop, PO's would now have breakout rooms of their own, which teams could join for reviews or to ask questions. This made the reviews faster and simplified the process of eliciting clarifications regarding requirements. However, this was less frequently necessary, since the requirements were more well formed, thanks to a second proposal by another group. It was clear that the general waiting times were reduced. Thanks to this, work never had to stop completely while waiting for clarifications from the PO's.

The iterative incremental schedule was also discussed before the workshop, which made it less time consuming to come up with one at the initial meeting.

The results of our changes was a more well-defined process which the members of our team more or less followed. People knew what to do most of the time, and it was clear how we were supposed to divide up the work. The improved usage of the SCRUM board was probably one reason for this. For this workshop, everyone knew how to interact with the board, and properly assigned themselves to tasks. People seldom switched tasks and hence we always knew what the other's in the team were doing. This enabled us to have designated builders and gatherers who established a drop-off point in-game, where resources could be distributed within the team. This greatly increased the efficiency of the building phase. Meeting protocols were filled in and measurements were made simultaneously, and documentation was made after the last sprint, all indicating an improved process.

However, as mentioned, our process still had some issue. The reason for the limited impact of estimates and velocity was that we mistakenly assumed that the second workshop would be very similar to the first one. We assumed the user stories to be small, which many of them weren't. Our team ought to have implemented a practice of breaking up big user stories into smaller ones – an essential part of SCRUM which we had neglected. Our team got tasked with building the castle walls, a huge user story which we worked on during the entire workshop. Because we never broke it up, we could never apply estimates and velocity as intended. If we had broken down the user story more, we could have used our improved process to actually manage the workload and implement SCRUM more fully.

Regarding our measurements: the process secretary handled the objective measurements during the workshop. For the purpose of creating a baseline, we also made an attempt at reconstructing measurements for the first workshop. This reconstruction, however, should simply be regarded as a crude estimate. These are the objective metrics used, followed by the measurements:

I. Time allocated for sprint planning divided by time actually spent planning:

For the first workshop, only 5 minutes were allocated. We used most of this time, with the exception of sprint 2. For the second workshop, only 5 minutes were allocated as well. For the first sprint we utilized 3 of those minutes, 4 for the second sprint, and 4 for the third sprint. An improvement, but slight.

II. Number of tasks changed or added during the sprint, divided by the number of tasks originally created:

The metrics are written on the form (tasks added or changed)/(tasks originally created). For the first workshop, these are the results: 1/6 (first sprint), 2/3 (second sprint), and 0/2 (third sprint). During the second workshop, these are the results: 1/4 (first sprint), 1/4 (second sprint), and 0/3 (third sprint). In total, we got 4/11 during the first workshop and 2/11 during the second. This suggests our planning phase produced a more accurate plan.

III. Number of user stories accepted by POs divided by total number of stories taken on for the sprint:

For the first workshop, the result looked like the following. 1/1 (first sprint), 0/1 (second sprint), and 2/1 (third sprint). For the second workshop, the results looked like this: 0/1 (first sprint), 0/1 (second sprint), 1/1 (third sprint). These measurements are a bit unclear. I think the process secretary didn't fully understand how to enter this measurement. 0/1 indicates that 0 user stories was accepted during a sprint, and one was taken on. However, only during the first sprint did we take on a new user story, which means the metrics for the second sprint should be 0/0 and for the third one be 1/0. Either way, it's difficult to make any conclusion using these measurements.

Here follows the results of the questionnaires. ”Q” indicates a question, ”A1” indicates answers for taking the questionnaire the first time and ”A2” the second time. Six team members took the first questionnaire and only five the second one. For the purpose of keeping this short, the answers are summarized. Most answer options were formulated using a Likert scale ranging from strongly disagree to strongly agree.

Q: I never have to wait for the PO to continue working on my current task. A1: 5/6 people answered disagree or strongly disagree. A2: 4/5 agree or strongly agree.

Q: The requirements are well-defined. A1: 4/6 answered disagree, the rest agree. A2: 4 agree, 1 neither agree nor disagree.

Q: Is there a sufficient number of open communication channels for communication with the POs? A1: 5/6 answered no. A2: everyone answered yes.

Q: I am satisfied with the team-PO communication. A1: 5/6 answered strongly disagree or disagree. A2: everyone answered strongly agree or agree.

Q: I am satisfied with the time allocated for sprint planning. A1: everyone answered either agree or neither disagree nor agree. A2: everyone answered agree or strongly agree.

Q: I am satisfied with the current quality of sprint planning. A1: 4 disagree or strongly disagree, rest neither agree nor disagree or agree. A2: 4/5 agree, 1 neither agree nor disagree.

Q: How often does your team make changes to the planned tasks during the sprint? A1: 3/6 answered sometimes, 2 frequently and 1 rarely. A2: 3/5 answered rarely and 2 sometimes.

Q: I am comfortable with the game mechanics. A1: 5/6 agree or strongly agree, 1 neither disagree nor agree. A2: Everyone strongly agree or agree.

Q: I am satisfied with the implementation of the user stories. A1: 5/6 agree, 1 disagree. A2: 4/5 agree and 1 neither agree nor disagree.

Every question indicated improvement, especially regarding team-PO communication and sprint planning. Judging from this, as well as the objective measurements and discussions with the rest of the team, it can be concluded that the SPI effort yielded a clear improvement, although some major issues were still withstanding (as was described previously).

VIII. Summary and Lessons Learned

This report has described our planned process for the first SCRUM workshop. After the workshop, we elicited the most pressing issues using iFLAP, for example the difficulties regarding team-PO communication. An SPI initiative was then formulated and later implemented using GQM, CMMI and the Sidky & Arthur model, with the aid of guest lectures about SPI in the industry.

A second workshop was held, during which we took measurements based on our metrics (formulated using GQM) to evaluate if the SPI initiative was successful. We observed that team-team and team-PO communication had improved, as well as the sprint planning phase and the competencies within our team.

One of the main realisations, at least personally, is the importance of the agile mindset. After the guest lectures, and after studying agile and SCRUM more thoroughly for the second workshop, I think this is what really helped us improve our process. We ought to have focused even more on this aspect, for example by studying the pillars of agile and properly incorporating this mindset into our way of thinking.

We also learned that pre-workshop coordination (with other teams) goes a long way – when working on a large project, a well-designed project-level process is essential. This should have been considered even more, and we ought to have determined a way to make sure every team participates in the discussions. One issue was the fact that only a few teams participated in the slack channels. One possible solution is having a pre-workshop SCRUM of SCRUM meeting where all SCRUM masters could meet and discuss team-team communication.

Also, we should have focused more on core issues. For example, team-PO communication was useful to improve, but the core issue was the unclear requirements – this was the main reason for reaching out to the PO's. Our goals should have reflected this, and aimed to improve the requirements. We should also have implemented the split-user-story practice, which would have allowed us to more properly use many of the other SCRUM practices we aimed to implement.

In general, the SPI initiative was successful. Using state of the art SPI frameworks, we were able to significantly improve our process.

References

[1] L. Williams, “Agile Software Development Methodologies and Practices”. Advances in Computers, vol. 80, pp. 1-10, 2010.

[2] N. Ashrafi, “The impact of software process improvement on quality: in theory and practice”. Information & Management, vol 40, no. 7, pp. 677-690, 2003.

[3] P. A. G. Permana, “Scrum Method Implementation in a Software Development Project Management”. (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 6, no. 9, pp. 198-200, 2015.

[4] M. Trkman, J. Mendling, M. Krisper, “Using business process models to better understand the dependencies among user stories”. Information and Software Technology, vol. 71, pp. 58-76, 2016.

[5] R. Vijay Anand, M. Dinakaran, “Issues in Scrum Agile Development Principles and Practices in Software Development”, Indian Journal of Science and Technology, vol. 8, no. 35, pp. 1-5, 2015.

[6] G. J. Miller, “Agile problems, challenges, & failures”. Paper presented at PMI® Global Congress 2013—North America, New Orleans, LA. Newtown Square, PA: Project Management Institute, 2013.

[7] J. Sutherland, K. Schwaber, “The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework”. Draft 29/1/2011, Paris, Scrum Inc, pp. 191-196, 2011.

[8] A. Raza, M. Majeed, “Issues and Challenges In Scrum Implementation”. International Journal of Scientific and Engineering Research, vol. 3, no. 8, pp. 1-4, 2012.

[9] S. Wagner, D. M. Fernández, M. Kalinowski, M. Felderer, “Agile Requirements Engineering in Practice: Status Quo and Critical Problems”. CLEI Electronic Journal, vol. 21, no. 1, paper 6, pp. 1-15, 2018.

[10] V. R. Caldiera, G. Basili, and H. D. Rombach, “The goal question metric approach”. Encyclopedia of software engineering, pp. 528-532, 1994.

[11] D. Goldenson and D. L. Gibsonm “Demonstrating the impact and benefits of CMMI: an update and preliminary results”. SEI Joint Program Office, pp. 1.27, 2003.