

## - Layout issues

- General / specific for one issue

- very brief SPI techniques

+ Detailed improvement plan

+ Good analysis of guest leaf.

Final Report

ANONYMOUS AUTHOR

Examination in DIT347 Software Development Methodologies

Bachelor Program Software Engineering and Management

Department of Computer Science and Engineering

University of Gothenburg

Gothenburg, Sweden

*Abstract - Software process improvement is important for every organization in the industry in order to maintain competitiveness, but implementing an SPI framework is a challenging feat and there are no clear cut ways to guarantee success. In this report I will describe how me and my team implemented and later refined a process plan based on today's relevant research within the field applied to our specific needs.*

### I. INTRODUCTION

Software process improvement techniques (SPI's) are techniques used by organizations to maintain competitiveness by refining and optimizing their development practices. There are a wide variety of SPI's which all have their strengths and weaknesses in terms of applicability in any given organization. There are two main categories of SPI's, inductive frameworks and prescriptive frameworks[4]. In this report I will go through the usage of SPI tools, techniques and models to improve a development process carried out in a virtual workshop as well as discuss the differences and similarities between some of the most well known SPI's used in the industry.

### II. PROCESS APPLIED IN THE SCRUM WORKSHOP

The process that me and my team defined for the first SCRUM workshop consisted of six practices defined with limited knowledge of what the actual workshop would entail other than that we would be building a city in the Mine Test game. These practices were *Communication and Collaboration, Incremental Delivery, Iterative Planning, Review, Responding to Change* and lastly *User Stories* [8]. The actual process itself, based on the Scrum methodology, consisted of eight different activity spaces of which six were iterative while the first and last, containing the activities *Initial Meeting* and *Final Meeting*, were not. The Initial meeting was reserved for the workshop introduction, gathering information about the endeavor and making sure each team member was ready to begin working. The process' first activity space was called *Understand the Requirements*. The first iterative activity space in our defined process, *Coordinate Activity*, consisted of the activity *Pre-sprint Meeting* which in turn consisted of two activities, *Find and Prepare User Stories*. The next activity space, *Shape the System*, and it's accompanying activity *Design the System* was the activity where we

planned to decide on materials, size, technique etc. if any of these important aspects of the requirements were either missing or left to us to decide. Following activity space was *Implement the System*, consisting of only the activity *Build* which I in this report refer to

as the working phase. After the work phase came the activity space *Deploy the System*, which consisted of the two activities *Integrate-* and *Deploy Increment*. Once the work phase along with the deployment is finished we evaluate the increment in the *Use the System* activity space. Lastly, we enter our *Post-sprint Meeting* activity space, starting with the *Sprint Review* activity where we evaluate our work and the level of satisfaction regarding our work, planning and the resulting product. Following this the *Sprint Retrospective* meeting is held where we discuss the outcome of the sprint along with suggested changes to be made in order to more efficiently do the required work in coming sprints. As the final activity of the endeavor a *Final Meeting* is held allowing the teams to gather their documents and receive feedback on their work as well as the collaboratively created schedule, from the PO's. This activity was placed in the activity space *Stop the Work* [8].

Although the intended process was followed to an extent, the different activities during primarily the *Coordinate Activity* and *Post-sprint Meeting* activity spaces were blended together without clear and distinct separation. This can most likely be attributed to our lack of experience working in processes like these. This meant that trying to keep track of what was said for each activity, and maintaining any sort of documentation about this was next to impossible. In addition to this, the design phase of the process blended together with the *Coordinate Activity* activity space, meaning we started to discuss design and practical implementation before the requirements had even been discussed[9].

### III. EXPERIENCES IN THE SCRUM WORKSHOP

During the process' first activity space, *Understand the Requirements*, the team members were to gather the required information about the endeavor by participating in the introductory meeting to then discuss strategy as a team in the *Initial Meeting* and attempt to coordinate the coming work and scheduling with other teams. Already in this stage of the endeavor our team discovered difficulties communicating with the other teams as well as the product owners. This resulted in a poorly executed initial planning phase which I will get back to later under this heading. As for our practice relating to communication, we could clearly see that we had underestimated how important the inter-team communication is to this kind of endeavor and were already suffering the negative consequences[9].

why?  
how?

where?  
how?  
In the Coordinate Activity activity space our Scrum master joined the Scrum of Scrum meeting to assign the team to a user story in collaboration with the other teams and PO's. The user story was then reported back to the team where we were able to prepare for the upcoming work phase by dividing the user story into tasks that could be assigned to members of the team. In the first sprint's Scrum of Scrum everything seemed to work just fine on both a team and program level, however there was little to no actual collaborative planning done by the scrum masters which caused issues later in the workshop. Furthermore, the user story and the agreed upon tasks for said user story were documented but not properly updated as described by the first mentioned practice in accordance to their state. This really didn't affect the outcome of the work in this workshop but it was a problem nonetheless, and in a bigger setting this could cause a lot of confusion within the team [9]. The Design the System activity in the Shape the System activity space was very brief for two of the total of three sprints and required minimal effort from the team thanks to mostly well thought through user stories and requirements. During the second sprint however, our team was assigned to a user story with an impossible requirement, this meant that we needed to get in contact with the responsible PO which proved to be very difficult given the poorly planned ways of communicating both between teams and with the PO's [9]. The limited PO availability was also due to each PO being responsible for multiple user stories, requiring them to cater to many teams' inquiries which is a common problem in development processes [7]. This resulted in the team having to start work without having a clear DoD to match the schedule and therefore rely on luck that the PO did not reject our solution once we had gotten a hold of them. Fortunately this worked out and the work that had already been done did not have to be undone, but this problem still cost significant time[9].

After this we began the Build activity. The only notable problem that arose during this activity was the clear competency gap throughout the teams, to clarify it was obvious that our team in particular had far more experience with the game's mechanics than other teams [9]. Our team had no problem sharing competencies between members to ensure efficient work on all fronts within the team, but less fortunate groups missing the competencies we already had going into the workshop struggled far more. There was also a problem relating to resource-scarcity that inhibited some teams from fulfilling the originally listed requirements on the user story cards [9]. In some ways this can be seen as a failure to account for risks involved and the inevitable timeloss caused by this, but given that the teams had no previous knowledge of the gameworld before the start of the workshop this was almost impossible to avoid.

The two activities integrate- and deploy increment did not entail much actual work in the case of this workshop since the user stories were built in place, by that I mean there was no actual launch or

integration process following the completed build of a segment or module of the city since the builds were already in the city once they were finished. These activities were therefore unnecessary and unused in the process, an oversight on our part.

The Use the System activity space is where the evaluation was done, this meant having the PO look over the build, check for fulfillment of each requirement and either give a passing grade or demand further development to satisfy the definition of done. In this phase we could clearly see the aforementioned differences in competencies between teams and just how big of a difference there was in the progress that had been made in especially the first sprint. Furthermore the PO's review of the builds took far longer than the teams had anticipated during the scheduling phase, making it clear that the lack of inter team communication during scheduling had a serious impact on the endeavour as a whole [9]. Lastly we performed the sprint review, this was however not done in a structured way, meaning that we had no data to compare sprints to each other, other than simple notes, making it hard to improve our process [9].

#### IV. SOFTWARE PROCESS IMPROVEMENT TECHNIQUES

The inductive frameworks, such as QIP (quality improvement paradigm) and iFLAP (improvement framework utilizing light weight assessment and improvement planning) focuses on defining the critical problem areas within the process based on the current actual situation of the organization in a bottom-up fashion through inductive information gathering. Prescriptive frameworks on the other hand, such as CMMI and SPICE, are model based frameworks that rely on defined best practices that have proven successful in other organizations. These frameworks take a top-down approach in the sense that a defined sc. best practice process should and could be applied to all similar organizations and does not take into account organizational differences or specific needs [4]. Below are some of the pros and cons of each of the above mentioned frameworks.

##### Inductive frameworks

###### QIP:

Helps improve specific processes through problem area elicitation and complement these by promoting acquisition of missing or poor competencies. Uses GQM to define goals to which fulfillment method can be chosen on a case by case basis. This framework is iterative and helps improve already existing processes by going through a loop of activities ended with evaluation of implemented improvement strategy [1].

###### iFLAP:

Similar to QIP, iFLAP focuses on the most critical problem areas but does so by first selecting the most important projects and roles needed for evaluating

how?

the process. Information is then gathered through mainly interviews, and triangulated as to not base findings on only one data source. Improvement steps are then taken based on available and needed resources as well as dependencies to other processes and projects [4].

#### Prescriptive frameworks

##### CMMI:

Facilitates mid to long term improvement through a model based approach [3] consisting of best practices divided into process areas such as Research and development and Project planning [2]. CMMI also allows for in-house assessment given that proper training has been conducted [4].

CMMI is however seen as resource heavy, requiring a long time of continuous work to achieve results. Since the framework is model based and uses best practices it does not cater to specific needs or situations [4].

##### SPICE:

Used as a best practice framework for domain specific enterprises, catering to the specific needs of e.g. automotive companies and streamlines their specific processes (automatic SPICE) [4].

Resource heavy and not flexible to specific needs of an individual organization, just like CMMI. SPICE also requires lots of resources and time to simply perform a single assessment [4].

#### V. SPI IN INDUSTRY

The approach to implementation of SPI methods and techniques in the industry vary based on the estimated needs and goals of any given organization. If we take a look at 1928 Diagnostics, a health tech company, their processes are regulated by international standards and laws restricting their ability to freely change their way of working and instead use what is referred to as the gold standard (IEC 62304). This standard is however adaptable to be able to fit most methodologies which allowed the company to apply a innovation process consisting of three main fields, exploring or conceptualizing, developing and implementing, and growing or improving. This along with a scrum-like framework for agile management has allowed the company to continuously adapt and innovate their processes to match company needs [10].

Siemens decided to restructure their management away from a waterfall model and venture into the world of agile process managing. Siemens were faced with high corporate complexity and stale development processes that were inefficient and expensive. Changes to development brought high costs due to managerial difficulties and overtime. As a result, the company looked into transitioning to an agile management structure, bringing development closer to the customer and ultimately cutting a whole line of middle management no longer needed due to this change. This was no easy feat due to the size of

the company, with over 1300 employees spanning 3 different time zones, making communication and collaboration during the transition more difficult than normal. Dulce Goncalves pointed out that the most valued part of the transition was in fact not the process itself but rather adopting the agile mindset, allowing for easier and more efficient changes to development improvements[10].

Although neither of these agile processes carry many similarities to the one we applied in the first workshop there are some minor details worth taking note of. Siemens applied a scrum-like management tactic to facilitate good communication, this entailed multiple differently-leveled scrum meetings ranging from only top management down to independent scrum teams. This allowed information to flow through the chains of command as well as allow for input and improvement ideas to be heard from every part of the process. This approach would have helped in our workshop where we had difficulties communicating across teams and with PO's who in our case could be referred to as upper management for the sake of the analogy. Further, the development at 1928 Diagnostics required highly detailed documentation required by the aforementioned regulations, but this also promotes improvement through backlogs and development history that lets the teams improve and suggest changes. As documentation was something our team struggled with during the workshop, such clear rules and regulations for how our work was to be performed could have helped us identify and document problems for increased chances at finding a good solution in future endeavors [10].

Good

#### VI. SPI PROPOSAL FOR FUTURE SCRUM DEVELOPMENT EFFORTS

In order to improve the process applied in the first workshop we start by focusing on the least well working practices and activities. The practices communication, planning and lastly competency management were the three most prominently problematic areas not only for our team but also for the endeavor on a program level. In order to increase performance and efficiency in these three activities we start by using GQM (Goal, Question, Metric) [1] to create goals for improvement.

The goals defined in the GQM are as followed [11].

GQM 1: "Improve the efficiency of team-PO communication from the viewpoint of the development teams."

GQM 2: "Improve the quality of sprint planning from the viewpoint of the development teams."

GQM 3: "Improve the competencies regarding the mechanics in Minetest from the viewpoint of the developers."

For these goals we then define questions that will help us determine if we improved, each question should then also be given metrics that allow us to gather actual data of the improvement [1]. The following questions and metrics are defined to do just

now?

that based on the above goals [11].

**GQM 1, Q 1:** "How much time is lost due to waiting for the PO?"

Metric 1: The average response to the statement: "I never have to wait for the PO to continue working on my current task."

Metric 2: The average response to the statement: "The requirements are well-defined."

**GQM 1, Q 2:** "How satisfactory is the team-PO communication?"

Metric 1: The average answer to the question "Is there a sufficient number of open communication channels for communication with the POs?"

Metric 2: The average response to the statement: "I am satisfied with the team-PO communication."

**GQM 2, Q 1:** "What is the current quality of the sprint planning?"

Metric 1: Time per sprint meeting actually spent planning divided by the time allocated for sprint planning.

Metric 2: The average response to the statement: "I'm satisfied with the time allocated to sprint planning."

Metric 3: The average response to the statement: "I am satisfied with the current quality of the sprint planning."

**GQM 2, Q 2:** "How high is the accuracy of estimated tasks in comparison to the tasks actually performed during the sprint?"

Metric 1: The number of tasks changed or added during the course of a sprint, divided by the number of tasks originally created.

Metric 2: The average answer to the question: "How often does your team make changes to the planned tasks during the sprint?"

**GQM 3, Q 1:** "What are the developers' competencies with the basic game mechanics?"

Metric 1: The average response to the statement: "I am comfortable with the game mechanics."

**GQM 3, Q 2:** "Were the developers able to fulfill the requirements of the user stories in the sprint backlog to the POs satisfaction?"

Metric 1: Number of user stories accepted by POs divided by the total number of stories taken on for the sprint.

Metric 2: Average response to the statement: "I'm satisfied with the implementation of the user stories."

Following this, we apply three generic practices from CMMI that support the reaching of the defined goals. The three chosen generic practices (GP) most suitable for the fulfillment of the goals are *GP 2.2 Establish and maintain the plan for performing the process*, meaning the old process plan will be reused and refined according to following segments as well as updated and maintained using proper agreed upon tools such as Trello. *GP 2.5 Train the people performing or supporting the process as needed*, discussing required competencies before the start of the workshop and distributing valuable information in the form of articles and wiki's to team members and other teams will help eliminate the loss of time caused by lack of skills and competencies. *GP 2.8 Monitor and control the process against the plan for*

*performing the process and take appropriate corrective action*, keeping track of the actual work in comparison to the planned process will help detect the need for changes. This could be done through assigning a member of the team to the role as process secretary who will monitor and report deviations. These deviations can then be used to discuss and implement changes to the process in order to better structure work [11].

We then apply specific goals and practices defined in the CMMI model to further elicit the concrete improvement steps that need to be taken in order to improve our process and reach our goals. These specific practices relate to the process areas *Project Planning* and *Requirements Development*, both of which we reached an estimated maturity level of zero in the first workshop. The specific goals to which the practices are chosen are *Estimates of project planning parameters are established and maintained* and *A project plan is established and maintained as the basis for managing the project for Project Planning and Stakeholder needs, expectations, constraints, and interfaces are collected and translated into customer requirements and Customer requirements are refined and elaborated to develop product and product component requirements* for [11], [2]. For the full list of the specific practices, please see [11].

In addition to this we expand upon our metrics defined in the GQM's in order to make sure we encapsulate the entirety of the actual work in the upcoming workshop as to make sure we collect enough data to create a realistic and fair comparison to the defined process.

In addition to these changes I recommend integrating the agile practices *planning at different levels* and *adaptive planning* [5]. These practices should be integrated as they enable us to impact and be part of the planning and scheduling of the endeavor in collaboration with the management and product owners, making it more likely that our proposed changes to the schedule in retrospective of how the last workshop went may be accepted and implemented. It's also a good way of increasing communication between teams and management by continuously adapting the plan and the schedule according to progress and resources. This would be useful should we run into a similar situation as with the review time constraint in workshop one [9].

## VII. IMPLEMENTATION OF AN SPI INITIATIVE

In preparation for the second workshop, our team decided that it would be a good idea to start a discussion early about the schedule by which every group was to adhere to. By communicating over the courses' Slack channel we were able to do just that and better prepare for the upcoming workshop. To form this new schedule we used what we had learned from the first workshop to anticipate which activities should be allowed more time than previously. Although not every team took part in the discussion, every registered student does have access to this

Report needs to be self-contained how?



channel and was informed that relevant information pertaining to the workshop was to be published there. Also, in an attempt to better structure the communication primarily between PO's and the teams, we made suggestions as to how to handle any inquiries from teams to PO's during the workshop and successfully created a queue system and an updated structure and plan for breakout rooms and reviews. These changes may seem small, but they allowed us to much better manage our time and eliminate the need for ad hoc communication solutions during the workshop.

Further we also created a dedicated channel in the course' slack channel where we posted relevant information, guides and wiki links to help those who were lacking required competencies to acquire these before the workshop started and by doing so further reduce time loss caused by competency shortages. Although our group wasn't in need of this, we saw it as very important to the endeavour as a whole.

In order to better structure and maintain the process documentation we divided the team into roles creating responsibilities and authoritative levels in our team beyond the program level [6]. One member was assigned the role of secretary and another the role of process secretary. The secretary kept track of our documentation, mainly the Trello board and wrote down the created tasks and marked them in accordance with their status. The process secretary was responsible for keeping track of what people were doing during the working phase of each sprint, checking to make sure the tasks that were created in the planning phase were followed and documenting changes to the Trello board in regards to new task creation or the need for such. These two roles were critical for our ability to efficiently keep track of our documentation and in extension our metrics relating to our GQM's. The remaining team members, excluding our scrum master, were divided into resource gatherers and builders. Doing so allowed for clear division of labor during the workshop, with two members constantly gathering required materials for the builders and dropping these off at an agreed upon location the builders could focus on building according to requirements seamlessly.

The process was slightly changed in accordance with the specific practices mentioned under previous heading, this meant that each activity space and activity now had documentable practices that decreased the use for as hoc solutions during active work. This meant that we now had a detailed way of working, acting as a checklist for each activity and streamlining the process and allowing for less confusion and higher efficiency. For a detailed description of each and every single practice implemented in the improved process please refer to [11].

The metrics gathered from the second workshop points to the fact that these process improvements indeed had a great impact on the efficiency and quality of the process. Information regarding the

subjective metrics from the workshops were gathered through a questionnaire consisting of the statements in:

- All metrics in GQM 1
- Metric 2 and 3 in GQM 2, Q 1
- Metric 2 in GQM 2, Q 2
- Metric 1 in GQM 3, Q 1
- Metric 1 in GQM 3, Q 2

The questionnaire was answered retroactively for workshop one before the second workshop in order to determine a baseline to which we could compare our improvements. These metrics showed a massive improvement in the communication related practices, with a far less prominent dissatisfaction with the team-PO communication. For example, after workshop one 50% of the team answered the statement in GQM 1, Q1, M1 with Strongly Disagree and 33% answered Disagree, implying that the team-PO communication was in fact subpar and caused a loss of time as earlier described. In contrast to this, the same metrics' statement was answered with 60% Agree and 20% Strongly Agree after the second workshop, pointing to a significant increase in communication efficiency and less time lost due to this. Further solidifying this claim, the statement in GQM 1, Q2, M1 after the first workshop received 83,3% No answers and only 16,7% Yes answers in comparison to 100% Yes answers after the second workshop. The planning related practices were also perceived as highly improved, going from 50% disagreeing and 16,7% strongly disagreeing to the statement in GQM 2, Q1, M3 after workshop one, to 80% agreeing and 20% neither agreeing or disagreeing [13], [14].

## VIII. SUMMARY AND LESSONS LEARNED

In this report I have described the most prominent problem areas that myself and my team identified in our attempts at implementing and improving development processes. These being difficulty communicating on both team and program level, difficulty planning an endeavor and in a structured and organized way adapting the plan when needed while also learning from these changes, and lastly difficulties surrounding competencies, facilitating training and promoting inter-team collaboration to reduce the effects of competency gaps. I have presented a structured way to improve processes suffering these weaknesses as well as described the implementation of these changes in the courses' second workshop. If i were to further improve upon the process improvement plan, given the same circumstances in the same environment, i would attempt to facilitate actual training opportunities for my course mates in order to decrease competency gaps. Furthermore I would dedicate more time to creating building standards in collaboration with the other teams, promoting a unified way of building and by doing so hopefully also decrease the need for inter-team communication in order to solve confusion

during active work. Much like how interfaces and software documentation work in actual software development processes where each component has to be compatible with either each other or a legacy system of some sort. By my estimation, the most valuable tool used in this course has been GQM which helped us develop specific goals to then use in order to shape and implement the specific practices needed to reach those goals. In addition to this, the metrics allow for a simple yet effective way to determine level of improvement from the adapted process.

Working with SPI has made me realize just how difficult it is to coordinate a development process composed of many teams. But at the same time I have been able to see first hand how applying proven practices to unstructured and chaotic activities and processes help increase understanding for the underlying cause of inefficiency as well as help optimize those processes.

#### REFERENCES

- [1] Basili, Victor, and Gianluigi Caldiera. "Improve Software Quality by Reusing Knowledge and Experience." *Sloan Management Review* 37.1 (1995): 55. Web.
- [2] CMMI Product Team. (2010). CMMI® for Development, Version 1.3. Hanscom: Carnegie Mellon.
- [3] Ogasawara, Hideto, Takumi Kusanagi, and Minoru Aizawa. "Proposal and Practice of Software Process Improvement Framework – Toshiba's Software Process Improvement History since 2000." *Journal of Software: Evolution and Process* 26.5 (2014): 521-29. Web.
- [4] Pettersson, F., M. Ivarsson, T. Gorschek, and P. Öhman. "A Practitioner's Guide to Light Weight Software Process Assessment and Improvement Planning." *The Journal of Systems and Software* 81.6 (2008): 972-95. Web.
- [5] Sidky, Ahmed, James Arthur, and Shawn Bohner. "A Disciplined Approach to Adopting Agile Practices: The Agile Adoption Framework." *Innovations in Systems and Software Engineering* 3.3 (2007): 203-16. Web.
- [6] Sutherland, J., C. Ruseng Jakobsen, and K. Johnson. "Scrum and CMMI Level 5: The Magic Potion for Code Warriors." *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)* (2008): 466. Web.
- [7] Vijay Anand, R., and M. Dinakaran. "Issues in Scrum Agile Development Principles and Practices in Software Development." *Indian Journal of Science and Technology* 8.35 (2015): Indian Journal of Science and Technology, December 01, 2015, Vol.8(35). Web.
- [8] [Undisclosed Authors], Assignment 2 in course DIT347 H20, Software Engineering Program, University of Gothenburg, Sweden, 2020
- [9] [Undisclosed Authors], Assignment 3 in course DIT347 H20, Software Engineering Program, University of Gothenburg, Sweden, 2020
- [10] [Undisclosed Authors], Assignment 5 in course DIT347 H20, Software Engineering Program, University of Gothenburg, Sweden, 2020
- [11] [Undisclosed Authors], Assignment 6 in course DIT347 H20, Software Engineering Program, University of Gothenburg, Sweden, 2020
- [14] [Undisclosed Authors], "Questionnaire - workshop 1", Workshop 2 in course DIT347 H20, Software Engineering Program, University of Gothenburg, Sweden, 2020
- [13] [Undisclosed Authors], "Questionnaire - workshop 2", Workshop 2 in course DIT347 H20, Software Engineering Program, University of Gothenburg, Sweden, 2020