(Handwritten margin notes:)
G
- Language
- Poor justification of project plan
+ Good use of literature
- "Taxonomy" of SPI techniques unclear
- improvement plan hard to follow, goals not in GQM format, metrics unclear, improvement steps unclear
+ Good discussion of 2nd WS

Title

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 methodology that contains definitions of tasks, tools and techniques that can be used to improve the practices within the process plan. In this paper readers going to read about how our group implemented an agile process plan into a video game environment called Minetest. What issues we had with our process plan during the first workshop and how we solved them using framework for software process improvement. How we came up with goals for improvement, questions to be able to create metrics for seeing how well our improvement progression was going. Did we encounter the same issues as companies in software development industry? As see what could be done differently and what were the interesting finding from these workshop and others work.**

I. INTRODUCTION

Software process is a set of practices for implementing, designing and planning, this can be called a methodology used by an organization to help them structure their way of working in development. There are many ways of implementing a software process model. It can be waterfall model meaning its works as a waterfall with different phases that should be accomplished linear. Prototyping model is works by delivering small parts of the system to the customer of the software to check for further requirements. Incremental development model works by developing and incremental scale the product until it meets the acceptance criteria set by the customer. Iterative development model works by develop a downsized product with all the features to get feedback which helps build a better product. Agile model works by being flexible, dynamic and changeable. The model contains customer feedback and involvement, incremental delivery and strives to be maintain simplicity and changeability. All these models have its pros and cons and could also implemented in hybrids to fit the organizations need.

Software process improvement is a framework that have definitions of task, tools and techniques which can be used to help organizations improve their development process and change the way of working to be more effective for the organization. An effective way to improve process quality which leads to better product quality for companies. There are steps taken to assess how an improvement plan shall be structured. Mapping the process, analyze critical issues in the process, come up

with a new improved design for the process, relocate the right resources for the improvement, implement the improvement and communicate to everyone connected with that process about the change. After implementing analyze and measure the results from the improvement and then look at next step in the software process improvement framework for areas that could need improvement. Different frameworks can be applied but almost everyone follows these steps for assessment and implementation.

II. PROCESS APPLIED IN THE SCRUM WORKSHOP

As a group we created a process plan involving the scrum methodology divided into two levels, the program level and team level. Responsibility was specified for each role include in our process plan that clarified what each role main objective for the workshop should be. *(margin: And that would be?)* Our process plan was based on the scrum method with new integrated practices as pair programming practice and iterative planning practice. The reason for adding these practices was that we as group determined these practices would be beneficial for our process plan. *(margin: Why?)*

Being able to instantiate the process as a team we had to talk about key values for scrum methodology which are commitment, courage, focus and respect. This would help us as group to have better communication within the group and to create a healthy working environment for everyone involved. Rules were listed as a contract for members to *(margin: which rules?)* uphold during the workshop. These rules were following the time plan set for each activity as sprint planning, sprint, sprint retrospective. Don't make own decisions based on user story. Usage of a website that provide boards used for product backlog. Scrum master will be the communication link between team level and program level. [10]

The reason our decision be using scrum framework instead of other frameworks as Kanban and XP were that we decided that scrum would be better from the perspective of being able to adapt to changes and to improve our development and to create a plan for incremental improvement during each sprint. The first workshop was going to played within a game called Minecraft was also a reason for the using scrum instead of other frameworks. Meaning using Kanban and divided

(why?) user stories into smaller pieces could lead to confusion for members in our team. [10] We thought this method would give us best opportunity to adapt coupled with the practice pair programming, our team could let pairs of three worked simultaneously and to have one person with greater knowledge of the game within each pair.

III. EXPERIENCES IN THE SCRUM WORKSHOP

Difference between our plan and the actual development plan was that we had over exceed our time plan that made us not being able to deliver the user story in time. The reason of this problem was that we didn't have a plan for what and how we should gather requirements from the product owner. *(?)* Not a questionnaire was planned for streamlining requirements back to the team. That lead to double work of the outline of the building that was constructed. *(why?)* Pair programming practice didn't work out as intended and were too time inefficient for the time window inside the workshop, which made us abandon that practice in the second sprint.

Because we as a team followed our process plan, we were able to manage and deal with a few of the critical issues. Our sprint retrospective worked as a continuous learning processes as we were able to find solutions and readjust our process plan for next coming sprint. Because we were strict with the process plan and did all practices, we were able to adapt ourselves. This led us to be able to deal with some processes that weren't working as intended as pair programming and sprint planning.

Time estimation of each practice of our process plan were wrong estimated. This is a problem that other researchers have pointed out as a problem when implementing agile methodology. A mistake done by the students participating in a study from 2002. This study points at wrong time estimation for a practice leads to other practice in the process suffer from not sufficient time spent on that practice. [2] When members haven't worked in scrum methodology, it's a common pitfall to not be able to estimate time usage for each practice in the framework. [8]

(how? why?) Coordination and communication with other groups that were linked with our user story was a problem and we didn't in our process plan tools to deal with this problem during the workshop. Past these issues my opinion was that we sticked to our process plan and did all our practices which also helped us improve our development throughout the workshop correcting problems and being more efficient with our time to be able to make our deadlines for our user stories. Through a study made in 2006 a group of researchers study a group of small software development companies to see how they adapted software process improvements into their development. By assessments of their maturity level and which area of improvements

should be done. Five companies answered on a questionnaire that issues which needed improvement was communication. [1]

IV. SOFTWARE PROCESS IMPROVEMENT TECHNIQUES

Inductive frameworks for SPI are a bottom-up approach to implement improvements which are picked based on the organization's current situation. Identifying issues in development then provide solutions that are tested in smaller pilot studies for a small section of the organization before launching these improvements to the whole organization. There aren't preexisting techniques and methods for assessment and comparing practices within the framework. Inductive framework is based on a goal that can be measured. [3] *(how?)* This makes inductive framework can be time consuming for companies to implement but provide personal tailored solutions for improvement where issues in development are found. They can more easily be implemented in smaller companies and leaves companies the choice for which areas should be assessed and improved. These frameworks are best suited for the problem that we encountered in the workshop.

Prescriptive framework is top-down approach, constructed with a collection of different practices that the whole organization shall adapt. *(Not necessarily)* Assessment doesn't account for different areas and only considered the whole organization. This assessment works by doing interviews and questionnaire to set which sets of methods should be chosen for the improvement plan. [3] By not seeing different areas of company and what these areas need as improvements to issues. This framework in my opinion is more towards bigger and larger development that need general solutions and improvement for coordination and communication between different areas. [4] By the time we had for implementing and assessment of our results it *(Did you not use CMMI?)* would only be possible to implement a lightweight method of this type of framework or only a few areas that a described in these frameworks.

IFlap is an inductive framework that works by doing a selection of critical issues that need improvement. Next step is to do an assessment of these issues by doing interviews and documentation of the situation. Creating a triangulation of these results. Last step is creating an improvement plan by prioritization in which order these improvements shall take. Calculating the cost and time for implementing for the organization. This framework can by implemented by all firms and not only suitable for large companies which is its advantage. [3] Our problems with communication and requirements problems could be chosen as critical problems and an improvement plan could be made for us in our group to improve. This makes an available framework to use for us.

The prescriptive framework CMMI stands for capability maturity model integration is a framework for integration and evolution through different maturity levels that can be reached by integrating methods for product development. It comes in two different ways staged or continuous representation. Staged representation considered the whole organization as one for its maturity while continuous representation considers parts of the process area for assessment of its maturity and improvements.

CMMI follows a roadmap for its consideration of which practices within the organization should be focused on and improved. Its advantage against other frameworks is that is clear state which areas an organization shall focus on to reach higher maturity levels and provide a concrete way for this. [3] By focusing on the roadmap companies can trying to improve areas that doesn't need improvement and aren't relevant for its development. Its time consuming for documentation and for writing new plans every time a level is reached. For our process plan we could use the continuous representation of CMMI. Meaning we could pick a few process areas that could be improved and use this framework.

ISO 15504 is a prescriptive framework for improvement and capability assessment for software development which is an international standard. It works in a similar way as CMMI with a continuous representation. ISO consists of different framework that can be applied to different companies like automotive industry or space industry. By providing a set of methods and assessing capability by reaching the necessary requirements. These frameworks are mostly used in domains that integrate security and need to be secure and always functional for being able to be sold to the market. Healthcare, automotive and space have strict guidelines that must be followed and the assessment for requiring these capabilities must be done by external firms.

?

This taxonomy is based on our situation from first workshop in this course.

1. IFlap

2. CMMI

3. ISO 15504

Reasoning for this taxonomy is the personal I think IFlap suites most companies and can be used in conjunction with ISO if the specific company must develop a product to a domain that require that type of framework. IFlap lets companies to specifically focus on critical areas that need improvement without loss efficiency. CMMI in my opinion focus on large scale operation that need a general improvement plan for all their areas of development even do it can be scaled down with continuous representation. ISO is a specific

Unclear what this means...

ISO is an organisation, the numbers indicate which standard we're talking about!

framework that only applies to companies that focusing on these areas that require them.

The first guest lecture was done by a representative from the company 1928diagnostics. It's a software development company focusing on developing software for healthcare. By developing for healthcare areas specific standards need to be followed for it to be approved for usage called for ISO, same as the prescriptive frameworks in section III. *No!* These standards are waterfall in their layout but doesn't make the software process have to follow a waterfall structure. Instead, they integrated these standards into an agile methodology called Scrum with contiguous improvements. This made them still fulfilling the standards without having waterfall structure which isn't necessary. By using an agile approach and methodology they thought by having a good process quality leads to better product quality. By having a good process quality, they can adapt to changes with standards and regulations which lead to a more efficient time and cost development.

Product development process was developed to try to find issues that can arise during core development in the company, they tried to find these in the earlier stages of development plan. Creating definitions of requirements, risk management, architecture updates for example. Problems they encountered was how to implement the standards and changes in regulation without have time lost and cost increased by large changes. Other issues were to have more measurements tools for assess performance indicators, capability/maturity, knowledge and resources. Linking their problems with ours were that it's hard to have requirements control in new environment as they had with unexperienced developers that haven't follow ISO.

The second guest lecture was from a teacher at university of Halmstad that worked at Siemens for many years with their process to implement an agile methodology. Siemens went from a traditional waterfall model to an agile methodology. Scrum was implemented on a large scale and went beside the note from others that scrum is used preferable for smaller companies. They felt required to make change for more agile environment to stay competitive with other companies.

Barriers where it required a lot of work to be able to change methods. External collaborators were still using waterfall methods and failed to meet new faster deadlines which lead to difficulty to meet these deadlines. Training people into this new methodology were costly. Difficulty to have everyone taking responsibility of their work and hard to change people's mindset. The positive feedback of this change was that organizational complexity decreased from five levels to three levels that lead to a more efficient organization. Agile thinking made people adapt to development and requirements need more efficient. The

company became more competitive on the market as agile methodology incorporates customer and from inside their own development.

Siemens solutions for gap in knowledge and capabilities concept of continuous learning with training. The same problem we encountered was lack of knowledge in the processes could be improved with training. Siemens had issues with merging code between different areas of the development which correlates with issues we had in the first workshop trying to integrate the user story service station with user story road. Communication between teams lacked and insufficient tools within the process plan could tackle this problem.

VI. SPI Proposal for future Scrum Development Efforts

We used a GQM to be able to formulate goals, questions and metrics. CMMI roadmap was used to assess what should be focused on what level our development. By the GQM model a goal is defined from a purpose, issue, object and viewpoint. Questions are based on the goal and for providing a structure that can be measured. [5] Two goals are chosen for improving critical issues that arise from scrum workshop. These are improving requirements understanding and improving communication in planning stage in scrum processes.

Based on CMMI these goals corresponded with maturity level 2 practices called project planning and requirements management. By doing this assessment specific practices and specific goals can be extracted from CMMI framework. [6] For requirements the specific goal chosen is requirements are managed and inconsistencies with plans and work products are identified. Specific practice is developing an understand with the requirements providers on the meaning of the requirements. For communication the specific goal is a project plan is established and maintained as the basis for managing the project and specific practice plan the involvement of identified stakeholders. *How is that implemented?*

Following the GQM model goals are defined and choosing questions based on these goals becomes for requirements: Do the defined tasks within the user story satisfy all the acceptance criteria? Are there any dependencies to other user stories? For communication the question is as following: Is communication between team sufficient for being able to meet acceptance criteria. Metrics for measurement is based on these questions that is scaled on Likert scale from very unsatisfied to very satisfied, done by each member of the team during sprint retrospective. For requirements a question is asked: How the perception of the understanding of the requirements were during the sprint? For communication a question is asked: How was the communication this sprint compared to the previous sprint with other teams? This gives our

team tools to see progression and being able to improve issues during each sprint. These measurements can be used to fill gaps in requirements and communication. [11]

The problem with coordination and communication could also be solved by implementing on a program level a scrum of scrum methodology. Meaning instead of having only daily scrum meeting within the team, scrum master of each team also has a meeting to discuss issues that can't be solved on the team level. A couple questions can be used as a structure to be able to get effective results from these meetings. What have your team done since last sprint that relates to other teams on program level? Here dependencies can be discussed of user stories and solutions can be found. What is the next task your team is developing that is related with other teams? Here teams could see if issues discussed can be solved by the next sprint. [9] Instead we in conjunction with other group that have dependencies together had our team leaders communicate with each other in sprint planning.

VII. Implementation of an SPI Initiative

Our goals were improving requirements understand and improving communication. Improving requirements meant to make sure that executed work follow them and meet acceptance criteria. In our team retrospective we had analysis and measurements being done to track improvement progression and find issues that can be fixed during the workshop. Improving communication between dependencies between our team and other team's user stories. Our Scrum Master will handle the communication and relay back this information regarding dependencies and if any requirements from product owner regarding our user story need communication with other team's user stories.

In the second workshop we handled the user story building a haybarn. We discussed before going into first sprint improvements that shall be done when developing these were to communicate more with other teams about conflicts and dependencies. Handling sprint planning differently by doing more planning with other teams which had a dependency with our user story. Integrating definition of done in sprint planning more efficient. From our retrospective from first sprint, we gather measurement data from our questionnaire. Our median score from this questionnaire was 2 of 5 for both requirements and communication seen in fig 1-2. Notes took from retrospective showed time ineffectiveness for fetching requirements, confusing requirements. Communication with other teams weren't sufficient, hard to handle multiple dependencies between user stories.

Question:
Vote on **how good the communication was** with other teams during the previous sprint
(Not good a1-5)

"I think the communication worked perfectly with other teams without any
complications during the past sprint"

1 Strongly disagree	2	3 Neutral	4	5 Strongly Agree

Answers 1-5:	first	second	third
Loe:	3	4	5
Elham:	3	4	5
Adam:	2	4	5
Anton:	2	4	5
Henry:	2	4	5
Hasan:	2	5	5
Median	**2**	**4**	**5**
Mean	**2.33**	**4.17**	**5**

Fig 1. Questionnaire from retrospective about
communication

Question
Vote on how the perception of the understanding of the requirements were during the sprint
(1-5)
"I think our team's understanding of the requirements were perfect during this past
sprint"

1 Strongly disagree	2	3 Neutral	4	5 Strongly Agree

Answers 1-5:	first	second	third
Loe:	2	5	4
Elham:	3	4	5
Adam:	3	4	5
Anton:	2	4	5
Henry:	2	5	5
Hasan:	2	5	5
Median	**2**	**4.5**	**5**
Mean	**2.33**	**4.5**	**4.83**

Fig 2. Questionnaire from retrospective about
requirements

Going into second sprint our team tried to improve
requirements developing structure and communication
with other team to get better results and better product
quality. In the second retrospective our median score for
communication was 4 of 5 and for requirements 4.5 of 5,
which is an improvement from the first one.
Communication was more effective, and requirements was
in place. Requirements improvements was done with
communication with the stakeholder in our case product
owner and we came up with a plan about how the
requirements should be prioritized.

Last sprint of the workshop we didn't experience any
communication and requirements issues, notes taken from
the retrospective showed that everyone was satisfied with
communication. Discussion with other teams help our
team structure our requirement plan more effective.

Further discussion within our team found that we lacked
tools for further progression in our development and we
didn't measure these areas. Estimation, team velocity,
other teams' feedback on communication was other areas
that would help us improve further in our development.

VIII.   SUMMARY AND LESSONS LEARNED

In this report the reader has gotten information about
my experience implementing an agile methodology and
how software process improvement can help improve
critical issues within development. The interesting part
about this work is that the original process plans our group
had how we should handle the first workshop didn't
counted issues we would encounter which would be hard
for any group to foresee. The positive outcome of our plan
was that we strictly followed all procedures during each
sprint even if it meant missing a deadline of user story. It
led us to be able to improve throughout the workshop even
do we didn't have all the tools necessary to fix them all.

CMMI would be a great tool to see were our group
level where and which steps could be taken to improve
areas for that level. Instead of focusing on the current level
and the progression to further levels of CMMI,
organizations shall focus on the journey of software
process improvement. Original goals can be changed
during the journey and shouldn't be about reaching the
highest level. [7] A different approach would be to a
training hour inside Minetest pre workshop and then do
assessment of our maturity level and capabilities. This
could help us see which areas will likely be a problem in
the workshop and tools could be developed to help us close
the gap in these areas.

To be able to get good results when implementing a
agile methodology and by improving it with software
process improvements members of that organization need
to adapt the right mindset for getting effective and positive
results. Being able to adapt to new working methods and
being open minded with improvements and full follow the
general process plan was one of positive outcomes from
the first workshop which then carried into the second
workshop. Even if we didn't meet all the deadlines, our
group improve our process instead and we able to meet
deadlines because we were able to improve throughout the
workshops.

A consequence that came up under a group activity
during a lecture in the course were when we as group
interviewed another group about their experiences during
the workshop. They had better knowledge about the
Minetest and were pressured to start gathering, crafting
and building without having their process plan in mind but
delivered their user story in time. As our group didn't
delivered in time our user story but followed our process
plan throughout the workshop. The same experience can
be seen in other organization when adapting scrum

methodology. Deadlines can't be meet but aren't because scrum is failing it's because the organization don't have a general process plan and guidelines to follow. [8]

My own opinion would be that I would read more articles about software processes and software process improvement to have a more knowledge about general pitfalls and issues that could be fixed before the workshops started.

REFERENCES

[1] A. Cater-Steel, M. Toleman, and T. Rout, "Process Improvement for Small Firms: An Evaluation of the RAPID Assessment-Based Method," Information and Software Technology, vol. 48, no. 5, pp. 323- 334, 2006.

[2] P. Abrahamsson and K. Kautz, "Personal Software Process: Classroom Experiences from Finland," Proc. Software Quality- ESCQ '02, pp. 175-85, Springer, 2002.

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

[4] B. Sharma, N. Sharma and N. Sharma, "Software Process Improvement: A Comparative Analysis of SPI models," 2009 Second International Conference on Emerging Trends in Engineering & Technology, Nagpur, 2009, pp. 1019-1024

[5] Basili, V.R., Caldiera, G. and Rombach, H.D. (1994), The Goal Question Metric Approach, in Encyclopedia of Software Engineering volume 1, John Wiley & Sons, pp. 528–532,

[6] J, J. Cannegieter, A. Hejjstek, A, B, Linders. and R van Solingen. "CMMI roadmaps" Software engineering process management, Technical Note. 2008.

[7] Bill C. Hardgrave and Deborah J. Armstrong. Software process improvement: it's a journey, not a destination. Commun. ACM 48, 11 (November 2005), 93–96

[8] R, P. Maranzato, M Neubert, and P Herculano. 2011. Moving back to scrum and scaling to scrum of scrums in less than one year. In Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion (OOPSLA '11). Association for Computing Machinery, 125–130

[9] M. Paasivaara, C.Lassenius, and V, T. Heikkilä. 2012. Inter-team coordination in large-scale globally distributed scrum: do scrum-of-scrums really work? In Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement (ESEM '12). Association for Computing Machinery, 235–238.

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

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