HOGESCHOOL VAN AMSTERDAM

Advanced Scrum

Project Game Technology

Rosa Corstjens, 500702627

February 2017

Glenn Hoff, Wouter van de Hauw, Michael FuentsRodriguez,Rosa Corstjens,Guus van Gend,
Kevin de Leeuw, Frederick van der Meulen & Jasper Meier

Project Game Technology
Team 5

Under the guidance of
Alexander Mulder, Product Owner
Anders Bouwer, Scrum coach

Eric Klok, Technisch consultant

Advanced Scrum

Project Game Technology

Rosa Corstjens, 500702627
Game Technology
Hogeschool van Amsterdam
February 2017

ABSTRACT

This report gives insight into the effectiveness and applicability of different Scrum principles for a project team consisting of game development students. An advice of techniques to use for this team is given, which helps them to successfully carry out an agile project.

1. INTRODUCTION

Scrum is a framework, both iterative and incremental, which enables more flexibility (Wikipedia, n.d.). This framework is often used for software development projects, because of its flexibility. In Scrum, the team delivers a shippable product, every sprint, to enable direct feedback on results and the ability to adjust when needed.

In this report, different advanced Scrum principles and best practices will be explained and discussed to give insights in their effectiveness and applicability, with as goal to do recommendations for expansion of the use of Scrum in the target project team. The target project team for this report is a student's team at the Hogeschool van Amsterdam consisting of eight game developers. While this team already knows how to work with Scrum, improvements in their use of this framework can be made. After implementing the recommended principles and best practices in the team's implementation of Scrum, it is to be expected that they experience less discussions, indistinctness and loss of time because of unnecessary or misinterpreted work.
Since Scrum principles and best practices are extensive discussed an expanded, it's outside of the scope of this report to discuss all possible principles and best practices. The set of subjects to focus on is selected based on the knowledge and used practices in the team (discussed in chapter 2) and on the size and scope of the project, since some principles have been created especially for bigger projects. This report focuses on the following principles:

- Agile metrics;
- Continuous process improvement;
- Backlog refinement;
- Identifying and solving impediments.

To give advice about Scrum techniques, which are useful for the team, firstly we will discuss how the team currently is working with Scrum. Secondly, the different Scrum principles and best practices will be explained and discussed. Last of all an advice will be given about techniques that should be applied in the target project team. The report finishes with the conclusion, followed by the reference list.

2. CURRENT PRACTICES IN TEAM

This chapter briefly discusses the current practices, in the target project team, which are relevant to Scrum. The techniques already used will be discussed together with their implementation in the proceedings of the team.

The target project team, consisting of eight third year, Game Development students, learned how to work with Scrum two years ago. Since then, they have been using this framework in all their school projects and most of them also used it during their internship. In the current project, the team already makes use of a few Scrum techniques, namely the following:

- Stand-up meeting, a meeting with the team, in which each team member tells what he has done, what he is going to do and what possible/encountered problems are.
- Product backlog, a list with all user stories and epics that the product owner and the team think is needed to do to develop the complete product.

- Scrumboard, a task board consisting of the sections ‘to do’, ‘doing’ and ‘done’ with tickets for all the user stories of the product backlog that the product owner and team want to complete within the length this sprint.
- Retrospectives, a meeting with the team to reflect on the last sprint and to discuss how problems can be solved with action points.
- Scrumpoker, a part of the sprint planning, in which team members discuss the scope of the user stories that are chosen for the sprint and assigns story points to every story. These story points are a replacement of estimated working time in hours, often used in Scrum.
- Burn-down chart, a graphical representation of the amount of work to-do versus time, which is used to measure the performance of the team.
- User stories, a to-do ticket in the format ‘as a <user>, I want to <functionality>, so that I can <added value>’. This way, it’s clear to the team what they are making and why and for whom that functionality is important.
- Sprints, a defined timespan in which the team delivers a shippable product. In one sprint, the team has a goal and a direction that can be easily adjusted in the next sprint, which makes Scrum iterative and flexible.
- Sprint planning, a meeting with the product owner and the team in which the planning for next sprint is discussed. User stories are selected from the product backlog and a goal for the sprint is stated.

The discussed principles and best practices are already applied within the team, but the use of some could still be improved. Of course many, if not all, of these techniques can be further improved in a student’s team, but there is a difference between improving by learning to use it and improving by being driven in the right direction.

Since the team is a student’s team, they don’t have a standup meeting daily, although having daily standup meetings is seen as an essential part of Scrum, so that everybody is up-to-date and problems can be addressed quickly. It’s a possible improvement to find a way to have daily standup meetings.

Another improvement the team could make is stating of the sprint goal. So far, this goal was only vaguely described by the topics of the user stories, instead of a concrete and neatly formulated sentence.

Finally the team should create a concrete focus, a goal of a sprint. This way, everyone will have the same direction and idea which avoids discussions or misinterpretations.

3. SCRUM PRINCIPLES

Scrum offers a framework consisting of multiple techniques, artifacts and a workflow. Far not all of this is used to the maximum extent in the target project team. Note that not all of these techniques should be used in a team and project of this size. Since this concerns only one team, techniques such as scrum of scrums are irrelevant (Agile Alliance, 2001), since this project doesn’t have stakeholders or limits in terms of costs, metrics such as net promoter sideare unnecessary.

The principles will be discussed, ordered in four aspects:

- Agile metrics: qualitative and quantitative ways to measure e.g. the team or the product. This information can be used as input for improvement;
- Continuous process improvement: active engagement of all team members in the improvement of their process. This will mainly be discussed in retrospectives;
- Backlog refinement: the process of getting the user stories in the ‘ready-state’ to improve understanding of the goal and tasks at hand;
- Identifying and solving impediments: the main task of the Scrum Master to help the team focus on the work, not on the problems.

3.1. AGILE METRICS

There is a wide scale of agile metrics available to teams adopting Scrum. It’s important that metrics are relevant and add value to the team (Wolpers, 2016). The team should not use a metric just for the sake of having metrics, only use metrics to measure aspects that are important to or improve the team. A technique that can be used to determine the right metrics is the Goal Question Metric (Prowareness +, n.d.). This is an top-down approach in which the team sets certain goals for itself. These goals are split in several questions and metrics, that help answer these questions.

Still, there is a need for a set of metrics that are useful, so that the team can select their metrics from a pool of options after the Goal Question Metric has been used. The following set of metrics is not inexhaustible, but is selected based on the size and scope of the project and on the fact that no stakeholders or money is involved in this project. The metrics are ordered in categories to ease the selection of metrics.

Satisfaction of both the team and the stakeholders is important to keep the team motivated. Since there aren't any stakeholders for the project at hand, only the satisfaction of the team is considered. This can be measured with the metric 'team member happiness' (Prowareness +, n.d.). Each team member gives a number between one and five at the end of a sprint. When these grades are compared, over time, conclusions can be drawn about, aspects that improved the team's happiness.

Since the considered team is a student's team, the improvements in their maturity are an important aspect. Maturity will grow if knowledge is created and shared along the members, this way dependencies on individuals are resolved. The degree of maturity can be measured with the 'focus factor' or the 'estimation accuracy' (Prowareness +, n.d.). The focus factor indicates how much time is spend on committed work, calculated by the work capacity of that sprint divided by the summed story points for all of the user stories planned that sprint. In short: it gives you the ratio between planned work and completed work. Assuming the team has accurately estimated its stories (which is the next metric) this metric shows how much time was lost on other, possibly irrelevant, tasks.

The estimation accuracy indicates exactly what it says: the ratio between the actual story points for completed user stories and the estimated story points for those stories. It is not strange at all if the team has trouble with estimating in the first sprint, but in the second sprint the estimation should improve. This metric is a great way to give insight in whether the team underestimated or overestimated the work.

Another important aspect of Scrum is guarding the product quality and quantity. The team should produce as much as possible i.e. have a high velocity, as long as they deliver high quality work.

Two metrics for measuring quantity are the sprint burn-down and release burn-up charts (Hayes, 2014). These do show the quantity of work the team finished over time in relation to the amount of work that would have been finished in the ideal case. Since the target team for this report already uses a burn-down chart, they can supplement it with a burn-up chart. Another metric, useful for quantity monitoring, is 'process efficiency' (Prowareness +, n.d.). The problem with this metric is that it assumes the team is relatively good at estimating; it calculates the same as estimation accuracy, but when the team knows that the estimation was pretty accurate, you can measure whether some stories took more time due to inefficiency. Before this metric becomes useful, the target team should improve their estimation, this is why this metric doesn't seem useful for the team at this time.

To assess quality of the product the team can use the metrics 'defect count' and 'fault severity' (Prowareness +, n.d.). The defect count should be as low as possible and is simply measured by counting all defects in the software. Note that this metric can be misleading: a bug that is never found is still a bug. The fault severity is used to differentiate known defects, since not all are just as bad. This metric could be combined with the defect count to create three counters of minor, major and critical defects.

Finally the team might want to measure the alignment of their efforts with the goals and vision of the project. This helps the team to stay on track and focus on the things that really matter. To measure this alignment a few metrics exist, but most require input from stakeholders and the organization. Metrics that can be measured by only the team are 'earned business value' and the 'end-to-end metric' (Prowareness +, n.d.). The earned business value uses points similar to story points. For each user story, value points are assigned. This is often done by the Product Owner and only communicated to the team. To determine the meaning of a value point, a base user story is chosen in the same way as the base story is chosen for story points. By summing the total value points from completed user stories, the team can determine how valuable their work was. If the team has the same velocity with a different earned business value in two sprints, the focus should be improved.

The team can also use the end-to-end metric. To use this metric, the product is split in different components or aspects. When a user story is finished, it's added to one of the components. After a few sprints it will be clear on which components the team tends to focus and the focus can be adjusted if needed.

3.2. CONTINUOUS PROCESS IMPROVEMENT

Scrum states that an organization should be a learning organization, meaning that it learns what should be improved and make the right changes in order to achieve those improvements (Villanueva Jr., 2014).Continuous improvement of the process is implied in the retrospective and supported by the metrics in the last chapter. It is therefore essential that the Scrum Master offers the team a structured retrospective, focused on improvement. The Scrum Master should be alert for common mistakes, including (Van Solingen, 2016):

- Don’t make the retrospective a routine meeting. This can be caused by repetition of the same problems that are mentioned, but also by repetition of the same technique for the retrospective itself.
- Don’t try to fix all problems in the same sprint, since this causes none of the problems to really be fixed: there is too little time. Another consequence is lower energy in the team since the same problems keep being repeated.
- Stop focusing on problems that are not possible to fix inside the team, otherwise the team will only talk about problems and not about solutions. After fifteen minutes the focus should move from identifying problems to solving the most critical problems.

Based on these common mistakes, the retrospective should vary in format and should focus on solving a core set of problems. The Scrum Master should also include the metrics in the retrospective to give insight in different aspects. Some problems are hard to identify when you only collect opinions of the team members. These problems can be easier found when data i.e. the metrics are visualized over time.

Furthermore, improvement is easier when a clear direction and goal are defined (The LeSS Company B.V., n.d.). This helps the team measure themselves and strife forward. Practices that help define this direction and goal are:

- Formulate one concrete sentence at the beginning of a sprint defining the goal;
- Talk often about the direction of the team to reach clarification and agreement;
- Set a perfect vision. This vision will probably be unreachable, but it has been proven that it guides the team.

3.3. BACKLOG REFINEMENT

The product backlog defines the work the team should finish before the product is completed. In a more traditional development setting, the design of the program would be created before actual development starts, but in agile projects, these phases exist side-by-side (Huizer, 2013). This means that the design is made while that part of the system is developed and that planning far ahead is not possible. On the other hand, starting to work on something that is too vague and not at all defined, will lead to many discussions and a loss of time. The team must find a balance between too little and too much preparation and preliminary research. This balance can be found by a good execution of backlog refinement.

Backlog refinement is about getting just enough user stories ready, just in time (Agile Cockpit, n.d.). Since many decisions are made and can change during development, it’s not necessary to only have elaborated user stories on the product backlog. Only the stories that are needed for this and the next sprint should be ‘ready’. A user story is ready, when it is clear and small enough. The other user stories can be vague and serve to provide insight in the next sprints. The team will save a lot of time when they leave these user stories vague for the time being. With the use of backlog refinement, the stories will be ready when the team needs them. Therefore, the goal of backlog refinement is to get user stories, step-by-step, in the ‘ready’ state.

Backlog refinement can be done in many ways, but it often consists of the following parts (Van Rooden, n.d.):

- The Product Owner prepares the user stories before they are discussed in a backlog refinement meeting;
- Big user stories or epics are sliced;
- User stories are estimated at the backlog refinement meeting;
- The definition of done is written for user stories for the next sprint.

Before a user story is discussed in a backlog refinement meeting, the Product Owner makes sure that the story is as clear as possible. This will save discussion time at the meeting itself. The Product Owner does this by finding out what the story implies and why it is needed. User stories that are unclear or don’t add value will often be neglected by the team. After it’s clear what the story is and why it’s needed, the Product Owner determines whether it contributes to the long-term vision of the product and if it is valuable to create. If not, the story should be discarded or reformulated.

The Product Owner brings stories that are ready for refinement to the meeting. It’s important that every story gets enough attention, but not too much, set a time-box of 10 minutes per story. For each story a set of questions has to be asked and steps have to be taken to refine the story:

- The Product Owner explains the ‘what’ and ‘why’ of the story. If this is not clear to the team, the story should be prepared by the Product Owner again for the next meeting;

- Give the team some time to figure out if they know how this story could be created. If they don't know, the story should be further discussed at a further moment or a spike could be created;
- Estimate the story on the given level. Vague stories and epics can be estimated with T-shirt sizes (i.e. s, m, l, etc), more concrete stories with Magic Estimation and stories for next sprint with Planning Poker;
- Determine the value of the story. If the value is lower than the required effort for the story, discard the story;
- Put the story back on the backlog, once it has been refined.

Often a user story will go through three backlog refinement meetings before it reaches the 'ready' state. When a backlog refinement meeting is organized occasionally, the team gets the chance to have their sprint backlog ready before the next sprint starts and so the team gets a good understanding of the upcoming work.

3.4. IMPEDIMENTS

An impediment is anything that keeps the team from getting work done (Scrum Methodology, n.d.). They make the team slower and they kill the motivation and energy. It's the role of the Scrum Master to identify and solve impediments as quick as possible.

The problem of solving impediments is versatile. This means there is no template for problem solving. The Scrum Master must be creative and ask for input from the team in order to solve the impediments in time. Often an impediment should be solved in the time span of a day, but sometimes this is just not realistic.

The same goes for identifying impediments, there isn't such a thing as a template, but best practices describe the importance of the communication between Scrum Master and the team (Bland, 2011). Impediments can become clear at a stand-up meeting, when a team member tells about a struggle or problem he encountered, but some impediments are harder to identify. The Scrum Master should urge the team to think about different aspects of the project which can have impediments (Dufeu, 2016). After all, different kinds of impediments can be identified and not all are obvious. Different types are, for example, impediments with technology, with team members, with the process or with the team itself. It's important that the Scrum Master makes the team aware of these different possible types of impediments. This makes it easier for the team to notice impediments and communicate them to the Scrum Master.

4. ADVICED IMPROVEMENTS

To improve the current implementation of Scrum in the target project team, a set of Scrum principles and best practices have been discussed in the last chapter. This chapter will discuss concretely how the team can use these principles and best practices to become a more mature agile team.

To choose a set of metrics that help the team monitor themselves and their work, it's advised to use the Goal Question Metric, discussed in chapter 3.1. It's important to remember that documenting the metrics shouldn't become a task on its own. Also, metrics shouldn't only be numbers; always provide a context.

The resulting set of metrics can be used as support or base for the retrospective, which will reinforce the continuous process improvement. The retrospective should focus on solving a core set of problems, instead of identifying everything that is wrong. For further reinforcement of the continuous process improvement the Scrum Master must help the team state a concrete goal for the current sprint, formulate a perfect vision and form a general direction.

Another important part of the improvements the team should make is focusing on backlog refinement. The delegated Product Owner must gather information about user stories for the upcoming sprints and prepare them for backlog refinement meetings. These meetings must be organized by the Product Owner at least three times per sprint, to keep improving the backlog and getting new stories ready in time. Different activities, such as slicing, estimating and writing definitions of done, are part of these meetings. They will give the team a clear idea of upcoming work and unexpected impediment, making discussions less common.

Finally, identifying and solving impediments is an essential task of the Scrum Master to keep the team from getting stuck. A newly discovered impediment must be solved within 24 hours or a solution must be proposed if it isn't solvable in such a timespan. The Scrum Master must be creative and use input from the team to solve impediments. For identifying impediments the Scrum Master must be aware of the different aspects of the project and talk to the different members of the team. This way he ensures that less obvious impediments aren't overlooked.

CONCLUSION

With the advised improvements in the last chapter the target project team is assumed to increase their efficiency and maturity. In the first sprint the team struggled with a certain amount of discussions about misunderstandings and uncertainties. The improvements drive the team towards a general direction and a shared, concrete goal. Nevertheless, the team must keep monitoring itself and discuss further improvements, since this report is only a stepping stone.

REFERENCES

Agile Alliance.(2001). Scrum of Scrums. Retrieved from https://www.agilealliance.org/glossary/scrum-of-scrums/

Agile Cockpit.(n.d.).Product Backlog Refinement. Retrieved from https://www.agilecockpit.com/nl/backlog-refinement-uitgelegd/

Bland, D. (2011, November 19). Agile Coaching Tip: ScrumMasters Beware of Impediment-Colored Glasses. Retrieved from http://www.solutionsiq.com/impediment-colored-glasses/

Dufeu, M. (2016, January 28). Different Types of Impediment to Identify. Retrieved from http://mattdufeu.co.uk/different-types-of-impediment-to-identify/

Hayes, W. (2014, September 22). Agile Metrics: Seven Categories. Retrieved from https://insights.sei.cmu.edu/sei_blog/2014/09/agile-metrics-seven-categories.html

Huizer, H. J. (2013, September 16). Backlog refinement. Retrieved from http://www.scrum.nl/prowareness/website/scrumblog.nsf/dx/backlog-refinement

Prowareness+.(n.d.). Agile Metrics, let the numbers tell the tale. Retrieved from http://www.scrum.nl/media/Agile_Metrics/$FILE/whitepaper_agile_metrics.pdf

Van Rooden, S. (n.d.). Product Backlog Refinement Explained. Retrieved from https://tinyurl.com/huabh5t

Scrum Methodology. (n.d.). Scrum Impediments. Retrieved from http://scrummethodology.com/scrum-impediments/
Van Solingen, R. (2016, February 27). Hoe verbeter je de Scrum Sprint Retrospective. Retrieved from http://rinivansolingen.nl/hoe-verbeter-je-de-scrum-sprint-retrospective/

The LeSS Company B.V..(n.d.).Continuous Improvement. Retrieved from https://less.works/less/adoption/continuous-improvement.html

Villanueva Jr., G. (2014, January 14). Scrum and Continuous Process Improvement. Retrieved from https://www.scrumalliance.org/community/articles/2014/january/scrum-and-continous-process-improvement-(1)

Wikipedia.(n.d.). Scrum. Retrieved February 28, 2017, from https://en.wikipedia.org/wiki/Scrum_(software_development)

Wolpers, S. (2016, December 11). Agile Metrics—The Good, the Bad, and the Ugly. Retrieved from https://age-of-product.com/agile-metrics-good-bad-ugly/