

CSM Practicing Certification Renewal Assessment

Name: Bartłomiej Ziółkowski

Email: bartlomiej.ziolkowski@gmail.com

Date: 23-Dec-05

Certification: Helsinki, Finland in October 2005

Scrum depends on the inspect and adapt mechanisms of process control to manage the complexity of projects. For inspection to work, everyone must know what is being made visible. To implement the Scrum process, such regulating mechanisms as defined roles, involvement versus commitment, time-boxes, and regular cycles are used.

1. Describe one project on which you have used Scrum over the last twelve months.

Purpose - what business goal was the project intended to deliver?

The project delivers a framework for modeling of managed networks. It is the core part of a new product that serves horizontal as well as vertical customers, and is an enabler for the company to take a position in a new market segment. Moreover, the project promotes renewal of architecture and technology in order to satisfy customer demands.

Length - what was the duration of the project?

The project was started in January 2005. The first time-boxed iteration was in April 2005. There will be 15 iterations in total divided into 3 increments. The first increment will be delivered to internal customers in the middle of February 2006. The first product release is planned at the end of the final iteration.

Cost - what were the budgeted and actual costs?

The project started small but due to high customer interest, its importance and size grew over time. It is expected that the project cost will be in range from 3 to 3,5 million €.

Value - what were the projected benefits and actual (if measured) benefits?

This is a fixed-time project. Due to the technology change the previous products were based on, the team must deliver on time so that company does not loose the market share. The major benefit, if the project delivers on time, is that the company will be able to enter new market area.

Size - how many people were on the project team(s), how were they organized into teams?

The project started with 7 people: 2 architects, 2 developers, lab engineer, and project and product managers. They were all in the same team having standing meetings every second day (not exceeding 15 minutes).

Currently, there are 21 team members (including 11 consultants): 1 Scrum Master, 1 Product Owner, 2 Architects, 2 Build/SCM Engineers, 2 Lab Engineers, and 13 Developers. There are 2 Scrum teams having daily Scrum meetings. There are people from 11 different nations in the project.

As of the next iteration there will be another 10 people joining the project, the reorganization is planned in order to have teams of 7 ± 2 people. Additionally, more Scrum Masters will join the team.

Teams - were the teams cross-functional and self-organizing? Were the teams collocated in an open space? Were the teams physically separated within one location, or located in more than one physical location?

The teams are cross-functional and consist of lab engineer, build/SCM responsible, DB specialist, architect, and developers with different skill level. Every developer is responsible for unit as well as automated acceptance testing. The team members are sitting in the same floor but in different rooms. There is no war room. The Scrum meetings are held in the corridor and the big whiteboard and walls are used for information sharing.

The self-organization was encouraged and facilitated by the Scrum Master but due to 'waterfall mentality' of senior project members and the constant inflow of new people with different background it has been reached only partially (i.e., there are still areas in the project where Scrum Master's actions are expected).

Initiation - how was the project initiated? how was the team trained to use the Scrum process?

The project was started as pure 'waterfall' with writing the specifications before writing any code. However, due to fixed-time constraint there was an approval for a change as, historically, most of the previous (waterfall) projects were late. The management approved the iterative and incremental development with time-boxed iterations and the program schedule was accordingly adjusted. There were trainings and info sessions organized for the program members to introduce and explain new practices.

Afterwards, it became clear that project management practices should be more agile. I chose Scrum and started to practice it based on the information from the Web and different books. The initial reception of Scrum was positive even though there were no well-organized briefings or info sessions for the team.

Reporting - how did you report progress to management and the customers?

The iteration length was set to be 4 weeks. At the end of every iteration, we have a Demo for stakeholders. That includes the end customers or their representatives (i.e., product managers) as well as upper management.

Additionally, the outcome of daily build, installation, and automated end-2-end tests execution is sent to all program members. The Sprint backlog is updated daily and is available in Wiki.

Change - what difficulties were surfaced by Scrum that had to be resolved? How were these resolved?

The biggest difficulty was to find out when some feature or scenario is ready and does not require further development. In order to solve this we defined what does it mean to have something 'done'. Additionally, we agreed that something can either be done or not (i.e., feature cannot be completed in 80%) and can only be marked as completed in the Product Backlog when done.

Management - what was the previous role of the Scrum Master? Who took on the role of Product Owner? To what degree were they successful in fulfilling their roles?

Previously, the Scrum Master (me) was a project manager and the Product Owner was a product manager. The Product Owner was able to work only 50% of his time for the project and was not trained to be a Scrum Product Owner. Moreover, he was faced with 'waterfall mentality' of upper product management and had to make predictive schedules and long-term commitments. On top of that, the initial fear of showing the real state of the produced software to the customers was an obstacle. Those factors limited the success of the Product Owner. However, the Product Owner worked much closer with the team in comparison with previous (waterfall) projects and that added to the overall success of the team.

The Scrum Master was in better situation as he could choose any process as long as every month progress was visible. Also, other projects managers were more supportive for processes that support visibility of the project state and require often (continuous) integration. The feedback received during regular reflection sessions with the team has proven the success of the Scrum Master.

Engineering - what software engineering practices or environment had to be changed?

Most of the developers in the team used to work in projects where iterations were 3 or more months long, there was 'enough' time for up-front specifications, everybody worked alone, and the integration was done at the end of the iteration. It had to be changed and the following practices were implemented: continuous integration (CruiseControl), automated testing (unit testing as well as FitNesse-like framework for automated acceptance testing), test-driven development, pair programming, buddy code reviews, and agile modeling and software development.

Stabilization - for how long did the software have to be stabilized before it could be released? How did you structure this stabilization process?

In the early phase of the project the team used to under-estimate the efforts by at least factor of two. Moreover, the effort of setting up the environment and the cost of introducing new technologies and new team members was heavily under-estimated. That led to the situation where software that worked only once was demonstrated to the customers. So far, we needed two stabilization Sprints during which only testing, documenting, and bug fixing was done.

Success - to what degree was the project successful? To what degree was the Scrum process instrumental in the success of the project?

As the project is only halfway through it is hard to estimate the success. So far, we have managed to build a lot of working software that really impressed our customers. We also demonstrated the most important use case to the top management on time and got their further support.

The Scrum process added a lot to this early success by empowering the individuals in order to get their commitment. The Scrum Master encouraged the self-organization and let the team decide (even though it has been recognized as heresy by other 'command-and-control' project managers). Such high commitment level has not been observed before.

Scrum Process - to what degree was the Scrum process implemented "out of the box?" To what degree did you have to modify the Scrum process for this project? For each modification, how did you formulate the modification so that the basic inspect/adapt mechanisms continued to function? What parts of Scrum couldn't be implemented, or failed, and why?

Originally, the iteration length was 4 weeks and there was one extra week between iterations used for planning. It was a part of the learning process. Currently, the iteration length is 5 weeks and there is Sprint Planning session at the beginning of iteration. There is a Sanity Check meeting in the middle. And at the end there is a Demo followed by Reflections session. And, of course, there are two daily Scrum meetings.

The Sprint backlog contains the tasks grouped under use cases or features they implement. The backlog is updated daily by the Scrum Master. This opportunity is used to follow up problems and have process-related discussion with every team member.

The definition of 'done' has been introduced only recently and was already overruled by the upper management ('You must show something!'). Additionally, the team was very optimistic about what can be done within one iteration.

2. How do you cause the accuracy of Product Backlog estimates to improve? To what degree does their accuracy matter?

The Product Backlog was not in a very good shape. The project started with an initial list of features for which there were initial effort estimates. Due to the little experience with the new architecture and technology the product is based on, those estimates were not accurate at all. After a number of design sessions with product managers, the initial list of use cases and

scenarios was created. Those are used as input for Sprint Planning sessions and the effort estimate accuracy has improved due to increased experience with technology as well as better understanding of the product vision.

3. How do you cause the accuracy of what a team commits to for a Sprint to what the team actually delivers?

During the Sprint Planning session the capacity of the team is estimated (in Ideal Engineering Hours) and the backlog is filled with the initial list of tasks needed to complete a use case. The estimates are summed up and the sum must not exceed 50% of the team capacity.

The product is use-case driven. That means there is a list of use cases and scenarios that should be implemented. For every iteration there are few use cases or scenarios chosen from the Product Backlog to be implemented. For every scenario there is an automated acceptance test implemented. When the test passes it is marked as regression. The number of regression tests versus the number of planned scenarios is used as a measure of iteration outcome.

As the project deals with new architecture and technology almost every estimate is wrong. Based on the previous backlogs I calculated a deviation factor for each developer and I usually adjust the effort estimates by this factor.

4. What metrics do you use to track the development process? Which metrics have been changed, removed, or newly implemented as a result of using Scrum?

The burndown chart is used to track the remaining effort. The definition of 'done' is used as ultimate metrics to find out which scenarios are completed. The automated coverage report (including complexity) is generated daily. The number of test code lines is compared with the number of production code lines. The number of unit tests and acceptance tests is tracked. The percentage of successful builds is tracked by CruiseControl.

5. What type of training, resources, or tools would best help you successfully employ Scrum in the future?

The ultimate solution for linking of the Product Backlog with Sprint Backlog that can be edited via Web independently by all the developers (some kind of Wiki-based spreadsheet that supports drawing the burndown charts and uses RSS for change notifications).

The Product Owner training that includes various product management approaches (use cases, features, user stories, etc.) as well as effort estimating techniques.

6. (Optional) Scrum and Extreme Programming are sometimes used together. What must be considered when this is done?

We are using elements of extreme programming like continuous integration, test-driven development, and pair programming. There is certain reluctance in the team while using these techniques as the code ownership and the need for complete design (as opposite to do only what's absolutely needed) are strongly rooted in the developers. Currently, we have employed few XP practitioners who are coaching the team and leading by example.

So far, I haven't noticed any incompatibility between Scrum and the elements of XP that we are implementing.