

# CSM Practicing Certification Renewal Assessment

Name: Bas Vodde      email: bas.vodde@odd-e.com      date: 17 Feb 2006

Certification:

Helsinki, May 19-20 2006, Boris Gloger and Jens Stergaard

Helsinki, Oct 05-05 2006, Ken Schwaber

Helsinki, Dec 15-16 2006, Ken Schwaber

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.  
Describe:

We've used Scrum in the last year in several projects. I'll try to describe several instead of one.

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

The purpose of all the projects are to deliver network elements or network element management system to multiple customers. One of the interesting aspects of the projects using Scrum (and all projects in our company) is that the amount of customers is more than one and still limited to a small amount (thus every single customer's requirements are still essential).

- Length - what was the duration of the project?

The length of all of one release is between half a year and two years. The length of the development on the same product is still much longer

- Cost - what were the budgeted and actual costs?

Depends on the projects

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

Depends very much on the products.

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

The smallest team is around 4 persons. The biggest project is around 100 people. Most of them somewhere in between and thus use several sub-teams. The organization of these sub-teams are different in different projects. One project (the 100 person one) has organized the sub-teams per component and are fixed over a long time-period. This however brings challenges related to how the requirements are coming in (typically not evenly distributed over the components). Other teams have divided their team per features and thus change almost every sprint. Other teams are in between these two mode of organization.

- 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?

Most of the teams are cross-functional and somewhat self-organizing. Some of them are located in an open space and some are even in more than one physical location. Some teams have not

been cross-functional yet since the testing persons were busy testing some other release. This is highly undesirable and has already led to problems (since there starts to be a backlog of testing work, and developers didn't pick up these tasks). Traditional development is happening in cubicles and it's been difficult for people to "give up their personal space". A few teams have done that and moved into a common room with most of the time positive results. Very few teams have been not located on one physical place (two cities). This has always led to communication problems and has never been really solved.

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

Almost all usage of scrum has been triggered by big one day training by Craig. This is organized by a small project which works cross-department (which I'm in). Normally this happens since some people in a project got interested in agile development (while not knowing much about it). They request a training, we give the training to the whole project (so that everyone understands the terms and concepts the same). After this the project has the choice of using scrum or not. If they want to know more then I give short trainings on scrum basics and concepts. After which some of the people participate to the scrum master course. After this it's their responsibility to teach scrum to their teams.

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

End-customers are still far away from the R&D and the product management (product owner) is in the way. Product owners always participate in the sprint review, which gives them some visible progress. Around 2 teams have been able to use release burn downs, others have not been able to add the estimates yet. Also all traditional progress measures are in place since management is hesitant to remove them (like progress based on time-tracking, actuals vs estimates). This is slowly on-going and will take some time still.

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

The amount of difficulties surfaces are too many to mention. Two of the biggest ones have been the stage-gate process which requests unreasonable accurate estimates and huge amounts of documentation in the beginning of the project. This is solved on gradually since it's used to manage all projects and not all projects have been using scrum and management does not want to treat scrum projects different than others. Currently already some changes to the gate-criteria have been made to make it more light and flexible. Another problem has been multiple people working on multiple projects. This is because traditionally testing and development (and architecture) have been so much separated that testing is one release project behind. When forming a cross-functional team and involving testing early this becomes impossible. A proposal for not running multiple releases in parallel (company-wide) has been created and will now be considered. Unfortunately these changes take time though.

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

Most of the scrum masters have been former project managers. Most have been successful but a common theme is that they do have difficulties not to control the team. This is actually simply because they want to help and they have difficulties of staying out. Also most of the team members are used to having a project manager around which tells them what to do. They actually have difficulty when there is not such a person.

All of the product owners come from the product management department. They are doing quite well but are not used to working close with R&D (and R&D is not used to work close with them). One of the problems has been the prioritization and splitting of features into smaller features. Also they are often not able to add the estimates to the backlog. (they are not asking the team for the

estimate!). None of them has been able to estimate value.

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

Most teams have moved to continuous integration. Some teams adopted test-driven development. All teams have removed some of the documents from their normal way of working since otherwise they could not deliver anything within a month.

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

Three projects have entered a stabilization phase and they planned it to last 3 sprints. No concrete experience from these sprints yet though.

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

All of the projects are currently going well and no team has had second thoughts or changed back, while all of them know it's still going to be a lot of work. One project delivered to their customer after 5 sprints which they say would have been impossible to do without the usage of scrum. One of the bigger projects was able to demonstrate running software to management in early project phase which triggered more interest in using scrum in the future. One project demonstrated their functionality on a user conference and for very positive feedback from the end users.

- 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?

Basically all of the projects use scrum out of the box with only slight modifications. Not all projects have been able to get everything implemented (yet). Like I mentioned, product backlogs often miss their estimated (and value). One project lengthened the iteration to 5 weeks so that one week could be used for workshops, requirement clarifications and planning (100 persons).

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

To start with the last question, the accuracy of the estimation depends very much on the situation and expectations of the customers (and maybe management). When multiple products need to cooperate and be delivered at the same time then the estimates might become extremely important. The longer term estimates can be less accurate than the near term estimates, also the further away the feature is from implementing, the more difficult it is to estimate and the most likely it is that there will be some change within that feature or related to that feature. Improving the accuracy of the estimates will be done by the team and will take time. If the product owner feel this investment is needed then he can request better estimates.

Estimates will be improved when the team gets a better understanding of the exact meaning of the features. This can be done by having the product owner and the team discuss the features needed in the future. Estimation itself can then be done with normal estimation techniques.

For all estimated, the product owner and the team should keep in mind that it's always an estimate and that it's not possible to create 100% accurate estimates.

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

First the team needs to make estimates which they believe in. This can be done by having a good and realistic sprint planning meeting in which the team really makes serious estimates (not because "management wants to see estimates"). Sprint planning can be done using "ideal engineering hours" so that the team realistically knows the time they can spend. The task planning then needs to be concrete and in not too big tasks. The team can also use the previous sprints velocity to do the next sprint planning. This is sometimes better, however it's more difficult to use when the team members change between the sprints (e.g. when the teams are based on features).

Then the way to improve this over time is to try to discuss in the team the estimates in the sprint review (retrospective) and see if the team can come up with improvements. Typically the team learn to improve their estimates over time anyway.

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?

All teams use sprint backlogs and burn down charts. We're using "virtual lava lamps" which also include the amount of test cases and coverage to also make that progress visible to everyone.

Most of the teams used to use a "project management tool" with Gantt charts in the past. They have been removed when a team changes to scrum.

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

Scrum Master trainings have been very good and have always received extremely positive feedback. The product owner training could be more concrete, as I already mentioned earlier.

On resources side, it would be nice to have more ideas about implementing scrum in very large projects. That is typically the question which people ask, they can see scrum working in a small project but not in a large project.

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

Most of the XP engineering practices can be used inside Scrum without much consideration. Both XP and scrum have slightly different recommendations for iteration length and iteration planning, thus these need to be looked at. Personally I would recommend teams to go for scrum by the book with engineering practices from XP (TDD, continuous integration, pair programming). If needed the iteration length can be shortened to 2 weeks. If user stories are used, they can be used on the product backlog. The two approaches are fairly compatible.