

CSM Practicing Certification Renewal Assessment

Name: Alan Atlas email: aatlas@sprynet.com date: Oct. 31, 2005
Certification: where Seattle when Sept. 2004

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

The project was intended to deliver a generalized object storage system for use by other internal services within the company. The storage system was to be scalable to thousands of terabytes, with essentially 100% availability.

- Length - what was the duration of the project?

The implementation of the project (design and code) took 9 months.

- Cost - what were the budgeted and actual costs?

Costs were budgeted as headcount over time. Originally budgeted at 16 heads over 9 months. Actual was 13 heads over 9 months.

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

The benefits are to provide a scalable answer to the storage needs of many internal services. Since many of these services directly support a high-traffic ecommerce site and there are no other storage alternatives available that meet the availability, scalability, and durability requirements, the benefit is arguably many tens of millions of dollars yearly.

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

The entire project team consisted of Dev Manager/Scrum Master, PM/Product Owner, 8 developers, and two test developers.

- 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 entire team was located in one large space divided into individual and two-person high-wall cubes. We used a large adjacent hallway for our daily scrums.

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

The Scrum Master is a CSM, and has developed a 90 minute scrum training presentation that has been used to train about 150 people in the company to date. This presentation was used to train the team. The project was initiated after the basic architecture was determined by senior architects at a multi-day offsite meeting. The team determined that significant design time was required for this particular project, so the Scrum Master reverted to Dev Manager for a 60 day design period before switching over to scrum.

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

Biweekly status reports to immediate management were the basic reporting mechanism, followed by an in-person review with Scrum Master, Product Owner, and management. Sprint reviews also became a form of status reporting to potential internal customers and management. Status reports were not exceedingly detailed, basically covering major issues, progress, and impediments. At the beginning, they were couched in waterfall terms, but over time, began to use SCRUM vocabulary, such as publicizing the goals of the next sprint.

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

The project was developed inside a Service Oriented Architecture environment, and depended on other services being built by other groups. One problem that surfaced was that we were ready to "integrate" with the other services much earlier than they were, because we had built a continuous integration environment. The team addressed this by creating two sets of acceptance tests, where one set only exercised only our service, and the other exercised our service in the context of the outside services. This worked very well. We were able to run both sets of tests and provide feedback to the other service groups, but still concentrate on test failures occurring within our own project when the other projects were still behind. We were often "Green" for our own tests, but "Red" for the tests that failed when other services were invoked.

Numerous difficulties were raised around detailed requirements and our novel use of corporate IT infrastructure as the project progressed. The Product Owner led the efforts around requirements issues in conjunction with the team (team input was carefully considered and often determined the final requirement definition). Infrastructure issues were treated as technical issues, addressed by the team where possible. When coordination with other groups was required, the Scrum Master drove the discussions, keeping the team burden to a minimum.

- 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?

Scrum Master had been, and continues to be, the traditional manager of all of the developers on the team. Product Owner was, and continues to be, the Product Manager for the product. Based on feedback from the team just prior to release, and also to team comments during a project post mortem, both were judged to have done a good job.

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

We had to create our own acceptance test framework so we could run tests 24x7 against the continuous integration environment, and we also created a unit test framework and a continuous build capability.

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

At approximately Sprint #3 (out of 6) we began to run 24x7 acceptance tests, so stabilization began early. It was painful and enormously beneficial. The team is very positive about that aspect of the project.

We literally did not have final versions of the associated services required for launch until the day of scheduled launch. We took three more days to stabilize past the planned launch date.

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

The project shipped 3 days late on a 9 month schedule. We have all of the features we need to support internal customers for production (revenue-generating) use. It appears to have been successful, but has only been in release for a few days, so we don't know what will happen when the customers actually start to stress it. We have been running acceptance tests for months, so we're not too worried.

There were many success factors in this project, and Scrum is considered to be one of them. For a system of this size and complexity, we could easily have spent a year doing detailed design work, but we chose to use the Agile iterative approach, which worked well. We never could have predicted how the actual implementation turned out, and any prescriptive plan would have been out of date before it was done printing.

- 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?

We did sprint reviews for the first four of 6 sprints, but because this is primarily a back-end system that is used via an API, it got difficult to "show" anything once the entire API was in place. The final two sprints were all back end work that was invisible to our customers, so we couldn't figure out how to do a good sprint review, so we didn't do that for the last two sprints. During the post mortem, the team decided that in retrospect this was a mistake and that we should have figured out a way to continue with the reviews.

We changed the planning process a bit. After the first sprint, the Scrum Master prepared a sprint backlog for the next spring planning session. This was done by collecting tasks dumped during the previous sprint, and also ones that were discovered during the previous sprint. This made the planning sessions much more efficient and didn't seem to have any negative effect.

We also, based on a team decision, allowed more specialization than scrum recommends. Scrum Master encouraged team members to venture out of their specialization area with varying success. This tension between specialization and "anybody can do any task" is a constant, but not huge, issue.

Prioritization of tasks within the sprint was not particularly useful, since this is a bug new system being built from scratch, and there was a huge number of tasks that were required to merely get the whole thing working. Nobody paid attention to the prioritization because it all had to get done. This made the Scrum Master privately crazy, but didn't really matter. Now that we are in a different situation, with a basic system in place, the Scrum Master is adding emphasis to the prioritization of tasks.

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

My experience to date is that accuracy is not nearly as important as consistency in both Product Backlog and Sprint Backlog estimates. If you have consistently "wrong" estimates, then the velocity reflects it and everything works fine. The team was interested in accuracy, so we collected "actual time spent" against each task, and showed the accuracy using a histogram chart. We did this for a few sprints, then it got uninteresting so we stopped. As long as the team was consistent in its estimates, predictable things occurred. If we always were estimating twice too much, for example, then the velocity reflected that, and the actual time to complete tasks simply dropped the burndown twice as fast as it would have if we had been perfectly accurate. There is some desire in the team to move to estimates like "Tiny, Small, Medium, Extra Medium,

Big, and Huge" rather than numbers. We haven't had a chance to discuss that yet.

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

It was dead-on accurate by itself, to the extent of measuring "Ideal Estimated Time Completed". We completed 65 Ideal Team Days of tasks each and every sprint, within plus or minus 10%. This was a very surprising and unexpected result. The only issue became managing new tasks that wanted to be added during each sprint that sometimes kept us from meeting sprint goals, even though they were necessary.

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?

We're not much on metrics of this sort. There will be a plethora of metrics against the deployed system, however.

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

We have not found the right Scrum Management software yet. We tried ScrumWiki and Scrumworks and disliked both. I now use Excel-based backlogs published to an internal website in html because I can make it do exactly what I want, but it's hardly turbocharged. The tool has to be web-based, very flexible in handling Product and Sprint Backlogs with drag and drop between them, support flexible printing of backlogs and burndown charts, show estimate history, and a bunch of other things.

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

We have stayed away from Extreme Programming so far, on purpose, so I can't really address this.