

## CSM Practicing Certification Renewal Assessment

Name: George J. Schlitz

Email: [gjschlitz@comcast.net](mailto:gjschlitz@comcast.net)

Date: 2/6/2006

Certification: Where: Boston

When: 11/2004

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.

### Acronyms

BRUF Big Requirements Up Front

BDUF Big Design Up Front

CMS Content Management System

CSM Certified Scrum Master

PO Product Owner

SM Scrum Master (uncertified/untrained)

UCD Usage Centered Design

XP Extreme Programming

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?

Project goal was to develop an Enterprise Portal with integrated Content Management System for large enterprise. This included establishment of an enterprise information architecture and usability vision. The portal was to provide content management for the consolidation of what were previously hundreds of diverse, distributed web sites and information repositories. The project also included several portal applications to provide collaboration, application integration, and other features.

**Length** - what was the duration of the project?

6 months for the part of the project in which I was involved- to take the project from nothing to an enterprise portal release. Despite a 3-year prior effort for the same project yielding no tangible results and little progress, a 3-month deadline for a pilot release to a target audience was established, as well as a 6-month deadline for full enterprise rollout. The project (enhancements, new integrations, new applications, new business unit CMS incorporation, etc.) are still ongoing.

**Cost** - what were the budgeted and actual costs?

Our budget constraints were very specific, but can't be shared. We met our budget exactly (original goals were met under budget but additional was used to invest software that would help ongoing development, and additional feature development).

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

No specific value calculations can be shared but a major goal of the project was to provide value in the following ways:

- Reduce maintenance and infrastructure costs of the many independent internal web sites and information repositories by implementing a portal-integrated content management system that was both flexible and provided sufficient consistency
- Provide a usable portal that could be leveraged by all employees at any time for effective knowledge management
- Provide a powerful communication and collaboration platform incorporating instant messaging as well as targeted communication capabilities
- Integrate many internal systems and provide single sign on ability

By consolidating many knowledge repositories in the portal and CMS, operational efficiencies hopefully will be gained.

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

We had 2 teams:

1- The CMS-focused team focused on developing templates for the content management system, and specific customized business unit CMS web sites (that met common usability requirements and IA): 3 developers (Java/Javascript/web/CMS developer experience), 1 BA, 1 Tester, 1 CSM, 1-2 UCD experts

2- The Application/Portal-focused team worked on application development, application integration, and CMS integration: 5 developers (Java/J2EE), 1 BA, 1 Tester, 1 SM (not formally trained), 1-2 UCD experts

Shared Team members included 1 product owner for both teams (the program sponsor served in this capacity) and 1 Usage-Centered Design architect who worked with both teams to help the evolving information architecture and usability guidelines.

**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?

Teams were cross-functional and self-organizing (progressively more so after the first sprints as the teams gained familiarity). Team members were distributed between 2 countries, 3 total locations. Each team was distributed in 2 locations.

The first team location (1 of the 3 locations, where the majority of the team was located) was a semi-abandoned office building which our product owner commandeered when suitable space was not available in other buildings. It had trash everywhere, was in a poor state of repair, and had inadequate furnishings, no whiteboards or supplies, no computer equipment, and had various forms of animal life as occupants. The program sponsor (who acquired the space through resourcefulness when no other space was available) quickly had the rooms furnished with desks,

whiteboards, supplies and computer equipment. Team members scrounged throughout the buildings many rooms and were able to find and restore to working order a printer, telephones, and other equipment left behind by previous occupants. Though sparse, this location turned out to be great for working together- open spaces, quiet, lots of rooms available for working sessions. Unfortunately we had to leave for more typical accommodations in a new building after a few months.

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

After getting a brief intro to the project history, charter, and the already-signed SOW, I was sent to the client location and held a kick-off meeting of the entire team (distributed via Webex/Conference call). This included an introduction to Scrum mini-training and discussion. The organization (and all team members) had no Scrum or agile experience or exposure, and were used to various internal waterfall-based methods. All were eager to try Scrum immediately however, based on their past experiences and understanding of the aggressive deadlines.

During the course of the project, I held the following types of training:

- Scrum refresher training
- Agile Estimation and Planning (based on Mike Cohn's fine work of the same name)
- Continuous integration and build automation with ANT
- Product Owner training (based on Mike Cohn's fine Product Owner training course- the beta version)

I also mentored 2 other project managers (the other project team manager and my replacement- both new to Scrum)

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

Reporting included

- The usual meetings
- Sprint burndown charts
- Release burndown chart which included per-feature approximate spend. This tool allowed program sponsor make budget decisions at any time based on real time information
- I sent a bi-weekly status report to interested stakeholders via the PO which included snapshots of the above, as well as highest project risks and open project issues
- Product Backlog list was sent to team members, PO, and others in preparation of planning meeting

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

Immense amount of work to do, and not nearly enough people or time to do it. Scrum and agile estimation and planning provided a simple framework for prioritization and decisions so that the teams could get things done and break out of the rut. Our capability was easy to identify after

some 3-week sprints, then applying good, constant reprioritization of product backlog we were able to complete the subset of the backlog that was most important to the business. Project stakeholders were particularly impressed with the immediate and complete transparency the process provided into the goings on of the teams. This allows them to be part of the solution.

**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?

I had been a CSM on several previous projects, and was new to this client. The other SM was a project manager (no agile experience) on other software projects, mostly infrastructure-focused and all using waterfall methods. I trained a third SM (whose background was in application architecture and infrastructure) before moving to my next project. The new SM's, though experiencing challenges of breaking the "comfort" of traditional methods, were able to adapt pretty well to the new mindset. Still a ways to go there and I have recommended formal training.

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

"6-sigma" (in quotes as I believe that this was not a particularly effective implementation) influenced, in-house developed software development process that was highly waterfall-based, document-centric, and inflexible.

Waterfall familiarity and comfort. Environment included the need to interface with a new bureaucratic outsourced infrastructure organization that did not have its processes in place yet.

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

We held 2 stabilization sprints- the first was pre-pilot, focusing on UI refinement and the second was pre-launch, and focused on performance improvement and caching.

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

I consider this project a tremendous success. By all accounts, we were not a well-oiled machine. We were not able to incorporate more advanced engineering disciplines that may help these teams in time. We did not perfectly execute during any Sprint. What we did do was start with an understaffed team to revitalize a project stagnating for 3 years (thank you to the big-name consulting company whose prior efforts provided me with this opportunity!), with far too many things to do in far too little time, and create an environment of progress, where we helped the team produce increments of product, and helped stakeholders understand their true needs and prioritization. The result was accomplishing our objectives on time, under budget. There is still much work to do, but the teams should be able to repeat this success post go-live as more requirements emerge.

Management was able to apply prioritization and valuation to the backlog, get a realistic understanding of team capability, and help the team focus on the most important features and items.

This project's success was completely a result on the scrum process and the teams' efforts to accept the new ideas and approach. Without either, the project would have been an utter failure. We were messy, but we came together where needed to get the job done, and the teams, though still not perfect, are far more capable now.

An additional area of success was incorporating Usage Centered Design team members and practices. See "Scrum Process" section for details of how we complemented Scrum with some UCD processes.

**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 implemented most of the Scrum practices immediately. We had many challenges to overcome initially. Both teams were not skilled at estimating, and avoided regular estimate updates. On my team, I took a more hands-on approach to helping get estimate updates, and it had great results- useful burndown charts, a highly-accurate velocity, and story point estimations that provided the PO with a simple yet powerful release planning and prioritization capability.

Since we had distributed teams, we had to hold daily scrums over conference call. This proved not to be a problem at all (though collocation is ideal). We held many meetings that required distributed team member participation using conference calls and Webex. All review and retrospectives were held using Webex and conference call as well, using webex to collaborate during our retrospectives.

1 major area of adjustment to the process related to Usage Centered Design. Before each sprint, with a basic idea of upcoming stories, a UCD expert/lead would use various UCD analysis tools (e.g. card sorting, observation, many others) to define and refine an overall "Information architecture" and usability guidelines/vision that helped guide the work of the development teams during their iterations. In parallel, another UCD expert was part of each cross-functional Scrum team, and would work daily with the teams to resolve usability issues and help analysts, testers, and developers refine the product. Feedback between the UCD team members resulted in learning from the sprints being incorporated (when appropriate) into the overall IA and usability guidelines. This combination of "just enough" up front analysis and within-sprint refinement with the help of a UCD expert was incredibly helpful in the evolution of the product- without this, it would have taken months more sprints of iteration and adaptation. I was/am not a fan of BRUF/BDUF, but the UCD experts have a set of tools based on psychology and human behavior that can truly help get accurate user information any time, and I intend to try to get UCD participation on any project I can in the future..

Another area we modified the process was in the role of the Scrum Master. I took on additional roles with the approval of my team:

- Build automation. I helped the team develop build scripts that provided 1-step builds, automated deployments across environments, and complete 1-step environment refresh ability.
- Developer. I signed up for a story or two, with the approval of the team- including

development of a quartz-scheduled database purge utility.

Finally, we did not keep a consistent sprint length. We tried out various lengths to determine which the teams were most comfortable with- the result being sprints of 3 weeks. Consistent sprint length will help the teams achieve better rhythm in the future, though a decent rhythm had been achieved by sprint 5 by my team. Adopting some more advanced engineering practices (such as those recommended by XP, TDD, etc.) will help the teams improve dramatically in the future.

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

Initial estimates were terrible. After attending Mike Cohn's Agile Estimation and Planning and Product Owner training, and reading his book, I trained the team and product owner in these practices, and we quickly gained control over our product backlog. We used planning poker techniques for story point estimates, and used actual results from previous sprints to determine a very accurate velocity which was used for subsequent release planning. Accuracy is fairly important for the current and next sprints, but decreasingly important for subsequent sprints- having a general velocity and reasonable story point estimates is more than enough to develop and maintain a very useful release plan (more so than any release planning technique I've witnessed)

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

I think this question is not worded properly, so hopefully I am answering the correct question. Ensuring that team members sign up for tasks that can be completed within a day or two, and ensuring that completed tasks are informally demoed or made available for acceptance testing immediately. The result is immediate identification of changes needed, problems, and other issues, as well as early completion of tasks so that they can be moved into the "accepted" column. Techniques from the agile estimation and planning training (and book) are also helpful, such as story splitting and relative sizing of stories.

## 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?

Burndown chart initially- for the Sprint backlog as well as for the release (releases identified with supersets of stories that were planned for several iterations. Use of these two burndowns was completely new to the organization. I am starting to believe that a burnup chart might be more useful- broken down by story "status" at the release level and iteration level, and have started to use it instead. I believe it gives a clearer view of things like scope increases/change.

Our project was weak on metrics. We had so many obstacles to overcome and still deliver a system that our process improvement goals were focused on implementing the basic process of Scrum. Only after 4 sprints (averaging 2 weeks each) did we start doing good release planning, and it improved each sprint. We were all impressed, however, with how much we were capable of by just implementing the core principles and practices.

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

Product owner training for new PO's. I attended Mike Cohn's beta- product owner training at the Fall 2005 Scrum Gathering, and it helped us much- I was able to mentor our PO in the basics of the role such as product backlog ownership, prioritization, and estimation and planning techniques. This PO will be attending the first formal PO course in Florida in 2006.

CSM training for new scrum masters. Having to run a full, extremely aggressive project in this environment, mentor teams, product owner, stakeholders, and 2 new scrum masters had me stretched far too thinly to do an excellent job mentoring the 2 SM's. If I had to do it over, I would spend less time taking on automation and build tasks and more time mentoring closely (though the former proved useful as there was no in-house experience).

In addition to a basic Scrum training presentation, discussion, etc, I will try to incorporate more participative activities such as "59 Minute Scrum" (thanks Jean T/etc.!).

Use of a big, visible task board is more effective (in my opinion) than software tracking tools, though a challenge for distributed teams...

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

N/A for this project. XP was not used- though XP would have provided the teams with incredible productivity gains over time.