

CSM Practicing Certification Renewal Assessment

Name: Brent Barton email: bbarton@solutionsiq.com date: 4/30/2005

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. Describe:

- Purpose - what business goal was the project intended to deliver? A realtor service provider aggregates Multiple Listing Service (MLS) feeds from many sources in many regions. The MLS services all implement their own standards, protocols and change notification procedures (usually no notifications). The client needed to replace its aging system with a more robust, scalable and extendable replacement that inspects data quality and could be modified to support a "developing" standard. The rate of change from these data feed sources, the volume of data and the limitations imposed by the consuming production applications required team flexibility and close communication.
- Length - what was the duration of the project? Approximately 9 months
- Cost - what were the budgeted and actual costs? \$100,000. Actual costs: \$250,000
- Value - what were the projected benefits and actual (if measured) actual benefits? Not disclosed due to IPO. All systems use this data, however, so it has mission critical value.
- Size - how many people were on the project team(s), how were they organized into teams? One Scrum team for most of the project. There was a second team for a short time but this proved to be less efficient and the people were moved onto other projects. The team size was at most eight people and at time of project closing, the team size was two people. The predominate team size was six. The variations represent changing throughput needs, schedule prioritization, budget adjustments and moving responsibilities to client teams.
- Teams - were the teams cross-functional and self-organizing? Yes. The team skill foci included analysis, dev, test and release.

Were the teams collocated in an open space? Mostly. The building floor plan had cubes and bays. The teams were in both. Maximum distance of people was 50 feet.

Were the teams physically separated within one location, or located in more than one physical location? The team members were all in one location for most of the project. The customer required a lot of transition help and we accommodated by placing team members on site as needed. The customer actively participated in the Daily Scrums and so these periods were pretty effective. The customer also considered recommendations for moving people around when our effectiveness was suffering. It worked very well.

- Initiation - how was the project initiated? This was started as a fixed bid project. This was a problem for 60% of the project duration. Even though the project ended up at much larger than the project budget, customer trust was continuously improved over the life of the project and eventually we worked on a time and materials basis.

How was the team trained to use the Scrum process? We used a training class with exercises. I mentored and eventually took over the project. We paired experienced team members with traditionally trained team members. We enforced daily builds, automated unit tests, automated integration tests and automated functional tests.

- Reporting - how did you report progress to management and the customers? We exposed a project wiki and bug database to the customer. We communicated daily in the Daily Scrum and the customer was very committed to this meeting so participation was 100% with at least one client representative and often three client teammates. We also placed a giant information radiator (6' long by 3' feet high: If it is written it must be true. If it is written big it must really be true!) at the client's site to track MLS feed implantations that had six attributes. We had regular meetings validating business perspectives from department heads to CTO.

- Change - what difficulties were surfaced by Scrum that had to be resolved? Some of the team was new to Agile and this was very painful for awhile. People angered each other off a lot. Code ownership was a big issue. Spaghetti code in one area forced reconciliation quickly.

How were these resolved? Let the team know they are in a new game. Focused on customer needs to introduce the team to the challenge at hand. Asked the team for commitment and an open mind. Tried inspiration: worked mostly. Then tried challenging the team to work together (it was like two factions in a team). In a few cases, I just got really pissed off and showed my alter-ego. Once in a while, it seems to work. When I happened, I tried to later reflect on the issues, that I got angry and ask them to reflect on whether it was worth it, despite my behavior.

- Management - what was the previous role of the ScrumMaster? Project Manager. Who took on the role of Product Owner? Our customer. To what degree were they successful in fulfilling their roles? The first Scrum master made a fatal flaw in my opinion. He refused training opportunities (ego, I think). When I took over, I used training techniques and retrospectives to reinforce pure Scrum and Agile practice.
- Engineering - what software engineering practices or environment had to be changed? Several team members needed to learn TDD. Frequent builds were unfamiliar to at least one tester. We recognized this systemic training problem in our organization and have taken steps to reduce the magnitude of this problem.
- Stabilization - for how long did the software have to be stabilized before it could be released? How did you structure this stabilization process? We started building into the customer staging environment at the sixth Sprint. We took two (2 week) Sprints and then we were releasing to production within 24 hours of release to staging. This averaged 3 times per week. Customer confidence was extremely high in our release management and Agile process at this point.
- Success - to what degree was the project successful? The customer considers us a key vendor for implementing best practices in their organization and also a primary go-to vendor. To what degree was the Scrum process instrumental in the success of the project? We could not have done this without Scrum. This is because the granularity of details in the requirements were never able to be fully identified due to the unusual, non-standard, ever-changing data sources constantly caused the requirements at the implementation level to change. Also, the agreed upon acceptance criteria for a go-live decision often had errors. These were because our system logged and identified issues better and we improved the data often. This would often result in false defects that ended up being defective tests.
- 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 the project with the following strict Scrum processes:

Events: Sprint planning, Sprint goal definition, Daily Scrum, Sprint Retrospective

Traditional Artifacts: Sprint backlog, Sprint burndown, Product Backlog. We posted a 40" printout of the Sprint backlog in the room where held our meetings. The updates became very revealing and when the handwriting became messy we would put up an updated printout (2x per week).

-We inspected and adapted the following:

We used 2 week sprints. We added an ideal time graph to the Sprint backlog/burndown to indicate variance (6 hrs per person per day), We stopped using the Product Backlog in the last third of the project because it wasn't helpful. The implementation information radiator worked better.

We released to production whenever we decided it was useful: the customer was always part of the decision and there was so much trust in our releases it was never a problem. The Product Owner was a Pig.

-Engineering: We had automated builds, unit tests and functional tests. There was very little UI so the focus was integration tests and data checks. We had to refactor some technical debt from earlier sprints and one time this was a long investment.

2. How do you cause the accuracy of Product Backlog estimates to improve? Accuracy is improved by familiarity with the project/product. I expect estimates on a Sprint to get much better by the third Sprint. This should translate into better backlog estimates. A bit (recommended 5% of Sprint) of pre-planning of high priority backlog items really helps jump start the next Sprint and helps project continuity. To what degree does their accuracy matter? The accuracy of backlog items matters to the extent that the Product Owner has confidence in the team. For example, if a consistent pattern emerges where half the items chosen for a Sprint end up on the backlog as dropped items this will cause concern. I wish I learned about seeing estimates as project budget sooner...very nice!

3. How do you cause the accuracy of what a team commits to for a Sprint to be what the team actually delivers? Scrum says these will improve naturally. I think just adding an available ideal time left line to the backlog causes this to improvement to occur more deliberately and enhances team productivity. Caution that the variance to the Burndown from this line is not used to interpret the team as "bad" but as subtle information that the team is trending a certain way.

4. What metrics do you use to track the development process? defect density, re-opened bugs, code growth, code complexity, check-ins
Which metrics have been changed, removed, or newly implemented as a result of using Scrum?
Forward looking estimates, Scrum scorecard, build pass/fail, unit test growth, number failing tests, number passing tests

5. What type of training, resources, or tools would best help you successfully employ Scrum in the future? Project startup checklist, implementation playbook, resourcing guide, visits from other community members

6. (Optional) Scrum and Extreme Programming are sometimes used together. What must be considered when this is done? Length of Sprints, Level of communication with the customer, to allow continuous change or fix iteration, managing documentation and other persistence requirements, resolve differences from team's experience: sometimes teams conflict a bit in terminology and approach and can confuse customers (e.g. 1 week vs. 4 week iterations, potentially shippable software and definition of a release).