

CSM Practicing Certification Renewal Assessment

Name: Mark Pushinsky email:icmbnar@yahoo.com

date:07/11/2005

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 – The project's purpose was to develop a system that would allow the existing internal business process of identifying and collecting on bankrupt and deceased customer accounts to be outsourced to a new third party vendor. The intention was to deliver the customer data necessary to process this work to a vendor who specializes in that industry
 - Length – The project was initially a plan driven project. Due to typical poor performance of the process and increasing date sensitivity to the business problem, the project was converted to a Scrum driven project. The waterfall portion of the project lasted 8 months and delivered an incomplete system requirements document. The Scrum portion of the project is still underway, but has been in effect for 4 months and is currently implementing its second release.
 - Cost – The original budget for the project was \$2 Million. The waterfall portion (see description of length) consumed \$3 Million. Since conversion, the project has consumed \$500,000. The planned spend for the Scrum driven portion of the project is \$2.5 Million. I expect that the project will finish under that target.
 - Value – The value from this project is derived in 3 ways. First, thorough outsourcing the business process the acceptance rate of bankruptcy and estate claims is increased, producing more revenue for the business. Second, the vendor processing these claims can do it with more efficiency, translating in to operational savings for the business unit. Finally, the project is intended to eliminate or significantly diminish the risks of improperly processing these claims, reducing the likelihood of law suits or other compliance related actions.
 - Size/Teams– There are two primary teams. The first team of 12 people is responsible for building and testing the system components to send the customer data to the vendor and process the responses. The second team of 10 people are collocated in a separate city and are responsible for building, enhancing, and testing the vendor components used to process the data and produce the claims. There was an extended team of 20 people that were brought in at various points to provide subject matter expertise and /or execute some tasks for the core team. While the team size is large, it works well given the collocation and commitment to use user stories, task tracking, and burndown charts in a consistent manner. Another technique that the team came up with to help manage the big team syndrome, was assigning owners to user stories. These team members were responsible for insuring that all the tasks were owned, updated, and the estimates rolled up appropriately for use on the burndown chart.
 - Initiation – The project was converted from a waterfall to a Scrum driven project after 8 months and \$3 Million were spent to produce an incomplete system requirements document. The IT and business leaders leveraged some Scrum pilot work that was being done across the enterprise to transition the project to an agile methodology. I worked with another ScrumMaster to select the core team from an existing team of 80 people. We then spent a week educating customers and stakeholders on the Scrum process, coaching them on what to expect from the team and understanding their expectations for delivery. We also tried to understand what far exceeding their expectations would look like. We then took the core team through a 3 day discovery process where we reviewed Scrum, began the team building exercises through discussion and play, and began to use the product backlog to understand the content of the first sprint.

- Reporting – I sat down with the stakeholders to understand what they needed to see to feel like they had the right level of insight into the progress of the team and the ability to help the team where necessary. I broke the status reporting into two main components. First, a weekly sprint status was communicated containing the following:
 - Team assessment – The sprint was given a color (red, yellow, green). The team decided how they felt about their progress towards the goals of the sprint.
 - Accomplishments for the week – The milestones (identified by the team during sprint planning) that were achieved that week
 - Work planned for next week – The milestones the team plans to complete the following week
 - Risk/Issues – High level risks/issues that could impact the project that the team does not feel it can solve on its own. These were typically organizational or cultural in nature and haven't changed much over the course of the project. I question the value of this.
 - Sprint Goals – The goals and changes (if any) to the goals that the team committed to in that sprint. These were established during sprint planning
 - Burndown Chart
 - Current Product Backlog
 Second, I sent out a summary after each sprint to keep the stakeholders updated on the following:
 - Review of sprint outcomes (recap of goals and how the team met them)
 - Current release plan
 - Lessons learned usually at a high level and organizational or cultural in nature, not the detailed view developed in the retrospective.
- Change – Each day has its successes and challenges and Scrum helps create more successes than challenges. I would say the two biggest challenges that Scrum elevated quickly were the issues with command and control driven culture and the inability for outside teams/processes to respond to change. I spent significant time with senior managers, communicating to them the importance of letting the team form and not to solve problems for them before they needed to be solved or before the team asked for help. The typical senior manager wants to 'help' the team through driving and directing and the self organization and unexpected efficiencies that Scrum promotes is damaged by this approach. After they saw how the team performed while being 'left alone' they understood the dynamic and the problem dissolved. The other issue is in dealing with teams outside of the core team. They are so used to doing work one way and using process and documentation to protect them from over committing or working on the wrong things. As a ScrumMaster, I found it beneficial to meet with all of these teams in advance of making requests of them to prepare them for the kind of interaction and response time that would allow us to remain as productive as possible.
- Management – I transitioned into the role of ScrumMaster as the project manager. I was working on a different (non agile) project before taking on that role. When putting the team together, we shared the importance and expectations of the Product Owner role with all of the senior stakeholders. My viewpoint was that if the customer really wanted the project to be successful they needed to put some skin in the game and dedicate a person to performing in the role of Product Owner. The PO has been a great asset to the team. The level of engagement was great and their ability to help articulate business value and drive external stakeholders to make key decisions quickly contributed significantly to the success of the team.
- Engineering – Early in the second sprint we began incorporating test driven development into the engineering practices. This was not only a technical shift in that it required new tools and new approaches to development and testing, but it was a mind set change for developers.
- Stabilization – After completing system testing and user acceptance testing, we would implement increments of the product and run them in production for 1-2 weeks before converting the operation to leverage the new functionality. This allowed the team to

- ensure that the new software functioned properly and nothing was lost in the conversion from the QA environment to the production environment.
- Success – While the project is still functioning, there has been early and consistent demonstration of success. After the first sprint, the customers realized that we had accomplished in 30 days what would have taken 5 months to begin if we had stayed on the waterfall path. The first release into production produced enough value to fund the team.
 - Scrum Process – Most of the out of the box process was implemented. The primary items of change were in the team size. The team needed to be large to deal with all of the various systems and subject matter expertise necessary to produce a viable product. The team felt that subdividing into a Scrum of Scrums would not be beneficial due to the tightly coupled processes that needed to be created and the level of resource ‘sharing’ they got by being only 2 large teams. As part of the inspection and adaptation process, there were modifications to the daily Scrum and ownership of user stories. Our daily Scrums took closer to 30 minutes to complete so that we could get everyone speaking about their work. The institution of ‘story owner’s’ helped ensure that tasks were not lost and that critical issues/risks/impediments were elevated to the team. Another adaptation we made was fully integrating all of the business components of the work into the core team’s work. As the project was a business process outsourcing project that needed extremely tight coupling between the movement of new process into production and the movement of business responsibility from the existing team to the vendor team, we had user stories and tasks associated with all of this work. We did not make extensive use of acceptance criteria on the user stories. While we did create test cases and perform acceptance testing, it took on a different form. We created a reference system that housed the expected results. The testers and the product owner worked together to ‘certify’ the reference system. This certification process accomplished two things. First, it made sure that the customer understood the outcome and bought into what done meant. It was acceptance testing for the system. Secondly, the system was built in a way that allowed for the use of test driven development concepts and automation of those tests. The results of the new system were compared against the reference system and automated to build a repeatable system test for the new product.

2. How do you cause the accuracy of Product Backlog estimates to improve? During planning sessions (sprint planning and release planning) we used historical data on user story level of effort to act as a heuristic for forming estimates for backlog items. As we learned more about what it took to build components we reflected that in the estimate, improving them over time. The accuracy helps in two regards. It helps the team frame up sprint contents more quickly, creating less stressful sprint planning. It also provides a way, coupled with the velocity of the team, to help frame up timing for various components/releases of the product and predict when the team might finish the current product backlog.

3. I don’t believe in driving the team to commit to anything. I believe philosophically that the team, given the right level of empowerment, will commit to what they can truly accomplish. This means a slight over commitment, which drives innovation, creativity, and high performance. I added historical data into their decision making process. I focus on items like the velocity of the team during the last sprint, calling out lessons learned developed from retrospectives, and ensuring the team understands what well exceeding customer expectations would look like. I think this is the ‘light touch’ that the team needs to commit to what it can accomplish plus enough to make it challenging.

4. The sprint burndown chart is the primary tool that the team used to track progress. It helps everyone know where they are relative to the time left in the sprint. We also used the white board with task cards moving from not started, to in progress, to complete. This was sufficient to understand the state of the team and provide enough incentive for the team to remain focused and motivated.

5. Spending time with the team and the stakeholders up front is key to having a successful project. Not only do teams have to understand, at least at a high level, the concepts and the process, but they have to have the groundwork laid to be a high performing team. I believe you

have to make sure the team has the tools to succeed and the stakeholders grant them enough independence to employ those tools and fail if necessary. Just in time training is a great way to get the team started on its journey; introducing enough concept detail to get them pointed in the right direction, but not so much content that they are overwhelmed.

6. When bringing XP engineering concepts into a Scrum environment you have to have considered a few things, in my opinion. First the team should be a reasonably functional Scrum team. They should have gone through at least one sprint and one retrospective. This allows for managing the level of change that you introduce at one time. Second, you have to have the technical environment ready for running an XP team. You have to have tools that facilitate Test Driven Development, a reasonable approach to dealing with continuous integration, and a team open to things like Paired Programming. A good way to bring these concepts to the team is to 'populate' the team with at least one engineer who has done it in the past and is passionate about what it can bring to the table.