

## Bryan Zarnett

1. Describe one project on which you have used Scrum over the last twelve months. Describe:

a) Purpose - Scrum was used as the process to manage the Pre-production testing (PPT) cycle within the Retail Branch technologies department of a major Canadian bank. Scrum was to provide process improvement, repeatability, and general project management within the existing RUP-oriented project structure. The PPT section is used repeatability in numerous projects and is a core requirement for release.

b) Length - each project's PPT cycle lasts approximately three months in total. I am working on the third project using our new approach.

c) Cost - I am not privy to this information. Since the inception of Scrum as the management process, the projects have been brought in on time, or slightly under time with no overtime required by the team members. The process has been identified as repeatable.

d) Value - reduction of overtime required by team members, the ability to have a released produced on time, repeatable process that can be executed by other members of the team. We have reached each of these goals.

e) Size - the average team is 6 individuals, only two of which are directly part of the RBT group. Other team members managed under the project were "service providers". All individuals reported into the PTE Architect (aka ScrumMaster aka Me).

f) Teams - the teams often worked on several projects but under the context of the management process where self-organizing and responsible for their tasks. Myself and the performance analysts will be located in the same open space. All other individuals were distributed around Toronto. Communication was done by phone and e-mail.

g) Initiation - I was brought in by the director to implement process improvement. The team had no Scrum training nor were they knowledgeable about Agile approaches to management and engineering. Aspects of Scrum were introduced at a slow pace using terminology that already existed in the organization to ease transition. New terms and concepts were introduced slowly over a period of two months.

h) Reporting - progress to management and customers was executed through daily status reports (via e-mail) which contained the details of our sprint signature log developed through our daily scrums. This included any impediments and their resolutions. Management was not interested in other reports so we did not provide them - no expectation.

i) Change - no real difficulties occurred. To negate issues, we used terminology that was already spoken and slowly changed the context or provided additional structure. Most terminology was used but never documented. We documented the terminology and presented the initial stake in the ground. Step one was to analyze the current processes and

map them to similar Scrum practices. From this mapping, we created a documented and working structure.

j) Management - no ownership existed for this process prior to my arrival.

Projects on average where 6 months late. The product owner role was the project manager. This was never successfully performed.

k) Engineering - we created standards in which performance and operation testing was to be performed, and provide standard documentation and formats surrounding this. The core practices involved implementing standard conventions and idioms, checking test scripts into version control, and continuous validation of the test scripts.

l) Stabilization - we provide an approach to accurately manage expectations and requirements. Our approach allowed the products to be released on time .

m) Success - the projects have been released on time based on the visibility and management approach of Scrum. Furthermore, because of the approach (underlying philosophy of Scrum) all members in the department have a better understanding of what is expected, what PTE means, and how it is executed against. Additional departments outside of Retail Branch Technologies are requesting details on our approach.

n) Scrum Process - the Scrum management process was implemented out of the book with the following alterations: 1) we used the terminology that already existed and used the Scrum practices to provide a structure; 2) No one knows we are doing Scrum; 3) E-mails with one or two slides where consistently used to present and document changes; 4) the only modification was in terminology; 5) we did not include development engineering practices.

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

As work is executed on a task in the sprint backlog or a requirement in the product backlog, the actual versus the estimated is tracked. In our sprint retrospective we examine these discrepancies and discuss initial assumptions and what impediments caused the issue. This works in conjunction with the sprint signature log. When a particular backlog entry is being created the individuals responsible are always asked to compare this with previously executed work of a similar nature. By comparing to previous work, the estimates are more accurate - the adjustment factor aids in the final estimate scope. The retrospective meeting helps team members understand how estimates where created and what assumptions are made - positive or negative to the situation.

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

I use the same method as described above - the key is the process of analysis - having team members thing in terms of similar tasks previously accomplished. The tool is "three points of view" - I ask the team members to think in regards of their perspective, the perspective of the business advocate, and the perspective of a 3rd party (such as an end user) - this provides them a simple process to think through the

problem. As a member of the team, I sometime acts as the 3rd party. I also ask the team to provide their assumptions or conditions in regards to their estimate - this way we can better judge the adjustment factor. The adjustment factor is very important.

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?

I use a burndown chart - the amount of work completed in regards to the amount of work that needs to be completed. This is the core metric. The sprint signature log provides a perspective on the accuracy of the backlog numbers. Based on the projects I am working with, I do not use any other metrics. I look at metrics in terms of useful, unuseful, and useless. The burndown chart I find very useful while metrics such as "number of tests created each sprint" to be useless and provide information that has no relevance.

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

Based on the organizations I am working with, integration with Microsoft Project seems the most viable tool - a series of views that allow me to manage backlogs through Microsoft Project outside of a Gantt chart view. I think targeted training in the future will be important - training that provides individuals with practical information on the development of backlogs and how to ensure their accuracy; or how to integrate engineering practices.

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

I believe in examine things on their general principle. General principles work under pressure and allow for easy adaptation to the environment, predicament, and team. In using XP practices, the twelve core practices I look at as being "extremes". I initially look at the essence of the requirement. For example, for pair programming the basic principle or general requirement is code reviews. Based on return on investment and return on expectations for the project, given the team, environment, and predicament I will determine if the general engineering practice brings a return on investment and expectation to the project. If it does, I look at my three variables (environment, predicament, team) and determine the best approach. For team A, weekly code review for a period of 2 hours might be the best approach in conjunction with code review tools that check for coding conventions and idioms (expectations are provided). In other circumstances, the variables might say that pair programming is the best approach. The twelve engineering practices I use as basics - fundamental building blocks of engineering practices. With each basic, I see which is useful, unuseful, or useless to the project. From there I determine the appropriate implementation given return on expectations and return on investment.