

My Experience with the Agile practices

Dennis Kwok

We were midway through a complicated project and had gained significant momentum, not to mention the team has just gotten accustomed to a fairly new process. At the time we were confident the team was doing our best job possible, yet a bit nervous as to whether or not we could meet a hard deadline. He had a relatively straight forward project plan to follow and was chugging along. All of a sudden our team gets thrown a curve ball. We were chosen as the pilot team to be trained in what sounded like a radical approach to rapid application development; so conveniently coined as the “Agile” methodology almost as if to imply our current processes were slow and sluggish. Being the Project Manager of the team, questions such as, and “why now? Why couldn’t we have started this training earlier, how am I going to meet my deadline, I don’t have time to go to training, wouldn’t this be extremely confusing at this stage of the project? Etc...etc...” quickly filled my head. After taking a deep breath I promised myself to keep an open mind and see what would happen. This now begins the first chapter of my Agile training.

In the Beginning, there was Chaos...

Is definitely was chaotic. We were following a relatively new process already and midway through the project we had to switch gears and try some Agile practices. There were lots of questions asked by the customer team. Questions on how much we should document, should we using wire frames, should we be using Mercury (our bug tracking system), how is status now reported, what criteria should I use to accept a completed story, can we meet our deadlines.

Some of the initial training sessions took almost half the day and occurred almost everyday the first week. Some of the team members are pooled resources or were borrowed by other teams, such as the Information Architect and Quality Assurance Analyst and they could not commit a lot of time into those meetings.

On top of that in order for Agile methodologies to be effective it seemed to require a lot of time and attention from the product manager. If the product manager would go on vacation or is not available for questions, I’m not entirely sure how this process would’ve work.

Sometimes Functionality could be complex; stories may not be enough to define specific functionality resulting in difficulty in the acceptance of stories. At first we would look at the iteration board and see that a story was checked off as “finished”. However sometimes the task listed didn’t provide enough detail so we didn’t know specifically what to check for. We had to keep constant communication with the developers to understand what exactly was completed (“what was it that you coded? or fixed?). I was secretly hoping that this was a tool that would encourage the team to communicate better as after all, Agile methods are supposed to be people-oriented rather than process-oriented.

Lastly, the word “Deadline” traditionally implied that you needed to get a product out by a certain date. In Agile, we’re encouraged to think differently. A “deadline” just means that the project team needs to communicate to management what can be realistically delivered by that date. It’s pretty much an honest, realistic approach to project management.

Midway, I Could See Some Light...

Our team was not fully Agile due to the time constraints of the product we were trying to deliver. As a result the team was only introduced to a few agile practices such as Pair Programming, Iteration Planning, Test First, Release Planning, and Retrospective. But even with these few practices I was beginning to see the light.

The team's spirit and moral increased. We were constantly working together; there was constant interaction, verification, and re-verification of the specifications. Creating stories and iteration planning was actually fun. I saw a lot more smiles on developer's faces. Everyone knew we were in it together, and there was a better sense of team.

The developers become more focused. We would take a story and the developers would break it down into smaller manageable tasks and post it on the wall. Developers then estimated the time needed to complete each task and signed up for the tasks themselves, which produced a sense of ownership. They knew exactly what tasks they needed to work on simply because they saw it every day on the wall.

The quality of the code was better. Developers were trained to test first then code. Before they wrote a function, they would write a unit test first. It seemed backwards at first but the result was better code. In addition before each demo to the stake holder, the customer did extensive testing against the tasks that were marked off as complete. The customer team was encouraged to test immediately when the tasks were marked off as "finished". So week after week we would catch defects before they formally got released to QA.

It was easier to manage change. In general we do not want to stray too far from the stories that are planned in iteration. However business decisions will change, that is inevitable in order to keep up with competition. Traditional processes usually have some sort of a change request form which requires approval from various parties. With Agile we add the change as a story and estimate the time it takes to complete that story. If the change requires 2 hrs, then an existing story in the current iteration that takes 2 hrs will have to be removed. The customer team is directly and efficiently managing the scope here, which relieves the pressure from the project manager to do so.

Estimates for tasks became more accurate. During Iteration planning, developers would discuss all the tasks involved for a particular story and come up with estimates. If the stories were broken down into reasonable and manageable chunks, the estimates became very accurate for that week.

Customers knew exactly what developers were working on and they knew why deadlines are missed. As the customers saw exactly what was on the wall and the progress for the task, they knew exactly what developers were working on. There were fewer questions such as, "what are you working on now?" If we could not deliver a story in iteration, we knew exactly why.

There was less time spent in documentation. Documentation was streamlined. Not to say there is no documentation, but we've were using more creative ways to document things. In our process before Agile training, if there was a change in the requirements, it required a change in the wire frame which had to be approved and re-released. Being agile, we just took pictures of the change that was scribbled or hand drawn on a wire frame and communicated the change verbally to team members during our daily standup. We would also note it as a story or a task we had to complete for iteration.

The End Looks Pretty Bright...

At the time that I'm writing this blog we have yet to launch our product. It is coming up very soon. However with just a few practices the team I can already see that the developers are enjoying their work more. I've also noticed that as a project manager I am no longer stressed about dead lines as it is now just an opportunity for me to realistically show management the great work the team can deliver. We may not deliver all the features we initially wanted to, but I at least know that the quality of the product we will deliver on that deadline will be nothing less than excellent.