Jae Ridley

CS 250

Scrum Review and Retrospective

For our project, in which we were testing the switch from waterfall to agile, we took on developing a booking system for customers to view vacation packages based on their preferences. At the start of this project, our product owner meets up with the client to get a mission statement, and me (as the Scrum Master) developed a team of a developer and tester while the product owner created a product backlog for the project. From here we went into an agile stage known as “sprint planning”.

During the sprint planning stage, the product owner talked to customers that used SNHU travel to ask about things that they would like to see in an update to the booking system, and thus created something called “user stories”, or short goals needed to complete the project. This is one of the things that slightly deviate from the waterfall approach, as while waterfall can involve some customer input, the agile approach involves direct feedback from the customer, having a bigger part to play in the creation (and can impact some changes, as we will find out later). It was also important for communication and feedback from the developer and tester for this product backlog, as although the product owner figured out “what” needed to be done, since the user stories themselves are very vague it was up to the rest of the team to help define them. After defining features within each user stories, both the product owner and the rest of the team helped to update the product backlog, which is a good place to talk about daily standups.

Daily standups are 15-minute meetings we had that we would discuss what was done the day before and how it was completed, what a team member was going to work on that day and any help that they needed and discuss any hindrances that was keeping them back from achieving a goal. As the scrum master, I regularly facilitate these meetings by starting the meeting off answering these questions and keeping the meeting on topic about the project itself, moving any other discussions to a “sidebar” for team members to discuss outside of this daily standup. During the daily standup we also updated progress of the product backlog by using a information radiator, a board in which the team members use to check and update the user stories and it’s progression. Both the information radiator and daily standups are important because it allows the team (including the product manager) see the progress made, discuss anything that would slow down the sprints, and allows for team members to discuss any other needs that they have in order to be successful.

Daily standups also allowed the team to be more comfortable with communicating with each other, even outside of the meetings. During the sprint the developer and tester kept communication with each other, with the tester asking for specifics about how the UI should look and how each feature should function, while the developer and product owner also worked out any additional details having to do with the user stories during the sprint. Again, communication was key for the success of this project compared to taking a waterfall approach, as although things within this project was defined, it allowed for the team to also try out different solutions compared to a stricter waterfall approach, which came into play halfway through the project.

During the sprint, the product owner continuously evaluated feedback from SNHU travel management, and within this feedback they wanted to changed focus from the top destinations slideshow we had previously made to one that focused more on detox/wellness destinations. Obviously not part of our original plans, this is where the agile approach had the most significant impact. Normally if a change like this happened in waterfall, there would be a number of things that happens, first and foremost being: scrapping the project all the way back into the planning stage. This in turn would cause a delay for the project, and costs would also drive up because of having to start from the beginning in updating everything. Since this was an agile approach, we kept the same sprint timeline while the product owner and developer looked at what features was done and could help with changing over to the new goal, the whole team (including the product owner) then went over what was possible and changing the product backlog by changing the priorities of the other user stories within it. After this, since the tester and developer had direct communication with each other, the developer helped the tester’s test cases by sending over the changes of how the new features was supposed to react.

One of the reasons why this approach was successful was because things went as planned as feedback was given throughout the sprint. Even with some slight changes made in the middle of the sprint, the team was able to manage the project successfully. I bring all these things up because although they can serve as pros, these very things can also serve as cons if not careful. Take for instance the product owner getting feedback, if consumers or the clients wasn’t giving details about what they wanted to see, the project could have stalled because it would be very hard to define anything (take “a list of vacation destinations” vs “top ten list of vacation destinations” for example). It would also be hard to keep track with the project and update things without the daily standups and the communication lines that the team had to each other and would be unimaginable if someone would leave mid sprint. These are all “nightmare scenarios” within agile methodology, and while could easily be more solved in waterfall (as each step is done in stages rather than all done within a sprint time period), could kill an agile project. If you want more thought out planning and strictly defined plan, I would choose the waterfall methodology to avoid these, although you would run the risk of either delaying a project or scrapping it if any changes come along in these plans. As for agile, for projects where changes are inevitable, I would recommend taking the risk as it is more acceptable to changes, and even with drastic ones, with a short sprint period in between intervals you can update the planning before the next sprints.