Aela Mason

CS250

Retrospective

The whole scrum-agile team had a part to play in the success of the SNHU Travel project. The Product Owner was vital to keeping the client, SNHU Travel themselves, happy with the direction of the project. From the initial intent of the project through to unexpected overhauls, the product owner had to maintain clear lines of communication and keep the product backlog organized. One of the primary moments for the product owner was SNHU Travel's decision to branch into health and wellness destinations/packages. While they were able to get the changes across to the team, they did so in a less than ideal way. They rolled out the changes straight to the entire team, before communicating them with the Scrum Master and before reorganizing the product backlog. This caused unnecessary uncertainty among the team that required additional work to bring back into line. However, this was countered by the very successful way in which the product owner gained feedback from prospective users. They used a focus group, and showed up with specific questions catered to get them pointed and actionable feedback.

The Scrum Master had a very limited scope within this project. Their primary function was to keep the team on the same page at all times. They achieved this with a sprint planning meeting and daily scrums. This retrospective is another piece in that puzzle. They also partnered with the product owner for several processes, chief among which was backlog refinement.

The developers were the one role whose actions remained largely the same in the change from waterfall to agile. The biggest changes were the move into time-boxed sprints, and the way in which tasks were assigned. The developers took user stories based on priorities that had been

pre-specified by the product owner and the scrum master. The team used the sprint planning meeting to commit to a limited number of user stories. The important part here was that the team felt confident in their ability to get through all of the stories they brought into the sprint.

The testers played a very important part in the SNHU Travel project. In waterfall, their role would mostly have come in at the end of time-boxes, doing QA testing on completed units. Instead, for this project, they worked directly with the user stories to develop test cases. These test cases were then used by the developers during sprints to know if their user stories were complete and functional. This was a form of Test Driven Development, and it was incredibly useful for this project.

The user stories themselves followed a very specific process. Many of them were created specifically from end-user feedback. For example, one of my user stories from feedback was, "I want to filter out options based on my budget so that I can avoid having to look through a bunch of options I can't afford." The stories were all given sizes and sorted by priorities. Then they were given to the testers to develop test cases, as mentioned above. The budget story was given test steps like, "Verify that budget button removes options from the list." These steps had corresponding expected results. For this test step, the expectation was that there would be no options that exceeded an entered budget after the button had been pressed. The development team used these prepped stories/test cases to add functionality during sprints, based on established priorities. All told, this process created a fluid process that took stories from design through to completion.

The agile approach also helped on a larger scale. One of the main purposes of an agile development cycle is to provide for flexibility within the project. The SNHU Travel project had a mid-development turn from general popular travel destinations to health and wellness retreats,

and it was very much because of the agile process that the team was able to roll with these changes. There were two possible routes within agile at that point. The first was, given no time constraints, the project would simply be refactored and development would progress unhindered. However, since SNHU Travel had an approximate intended launch, that changed things slightly. The product owner and scrum master, working with the client, had to determine what the core deliverables were. From there, the team was able to remove/cut additional deliverables to allow for the project as a whole to still be completed by the intended time. Waterfall would not have had the flexibility for this. Typically within waterfall, when cuts are required, they leave a noticeable impact on the project as a whole. They are cut based on time-constraints and immediate need, regardless of priority. Within agile, the team is able to anticipate the cuts and make sure that what gets cut will not have a negative or noticeable impact on the delivered product.

Communication within an agile team can take many forms. Some of it follows traditional patterns. For example, during one portion of the project, I, as a developer, had to reach out to the product owner. I did so through the email below:

To: Christy (product owner)

Subject: User Story Priorities

Christy,

I've been looking at our user stories categorization and classifications, and I think we need to revise how we denote priorities. The current system specifies the size of the story and its relative

priority, but nowhere in the system do we have absolute priorities. We don't need anything extreme. Just a low/medium/high system.

Would you be able to go through the user stories document and mark each story with one of the three priority levels for me? It would greatly help me in prioritizing test case development, especially as our user stories backlog gets bigger and more difficult to parse.

Thanks,

Aela

This email demonstrates a need that the developer has detected, information that would help with their job that is missing or incomplete in the current form of the user stories. It asks for specific deliverables from another member of the team.

The nice part about agile, though, is that it encourages transparency and open communication. The team is most successful when they're all on the same page. That's a large part of why the team has daily scrums. Each member of the team brings to the scrum their progress since the last meeting and potential roadblocks to their progress. Even outside of meetings, though, the team is able to use things like an information radiator to communicate. It's a highly visible resource that the team can reference as often as they need to, and that can and should be updated in real-time.

One tool that the team utilized for the SNHU Travel project was Jira. It's a board that seems to have been designed out of a kanban mindset. It breaks work down into a limited set of distinct states. Each user story starts out in the left-most column. As the team develops and tests

it, it moves across the board. It's done when it reaches the right-most column. In this way, it gives a burn-down sort of visual of what work remains to be done. Each column, in-turn, represents a specific portion of the process. The left-most column, "to do", is where all of the work for the sprint goes. The team chooses these stories in the sprint planning meeting. The stories move to "in progress" when the developers begin work on a story. When the developer believes their code is complete, usually when it meets all test-cases, it gets moved to "code review". Once confirmed as complete by another member of the team, it gets moved to "done". At the end of the sprint, all tasks should be "done", and the team should be able to reflect on them in the sprint retrospective. Then a new sprint is planned, and the process repeats.

If this project had been attempted with a waterfall development cycle, there is a good chance it would not have met with the same level of success. The switch to health and wellness travel would have presented a major challenge to the team. While it wasn't a fundamental switch to the architecture of the system, it still made a lot of user stories obsolete, including some work that had already been done. In a waterfall style with a pre-determined timeline and budget, that would almost certainly have meant a missed deadline, going over budget, or requiring some form of crunch from the team.

Agile had numerous benefits, as I've talked about throughout this retrospective. It wasn't without its own difficulties, though. One of the biggest ones is the flexibility inherent in agile. While it is one of agile's greatest strengths, it also asks a lot of the team. In the SNHU Travel project, the product owner rolled out big changes with a lot of uncertainty directly to the team. If waterfall provides anything, it's certainty. Agile asks the team to be comfortable being uncomfortable and uncertain. Still, this project definitely went better because it was agile.