Sprint Review and Retrospective

Scrum Master: Rachel Siminski

During this sprint the team worked on a project for SNHU Travel. The goal was to create a niche travel booking system which would display a list of the top five destination options for the user, based on the user’s previous travel and given a set of filter options. Each member of the team contributed to the success of the project in clear ways. Our Product Owner, Christy, did an excellent job acting as a liaison between the client and the team. She demonstrated superb foresight in seeking additional information from the client before any issues came up, allowing us to work as expected and meet our deadlines. I, as your Scrum Master, made sure to aid where I was needed and led our daily Scrums. I communicated with Christy about the team’s progress and I was able to inform her that some of the team had additional questions for the client, allowing her to get those answers in a timely manner. The development team was able to stay on target with each task and delivered an excellent product for SNHU Travel. Finally, the testers were able to create test cases for each user story and provide in depth feedback based on each test case for the development team to make adjustments as needed.

The Scrum-agile approach allowed the team to stay in sync with one another and stay flexible. Initially the team was operating based solely on the first set of user stories, but with the additional information requested by Christy, the team was able to address any questions that started to pop up and make adjustments in some areas. The original set up for the Top Five list was a vertical scroller, however it came up that a slideshow format might be the way to go. The team’s flexibility as individuals and within the flexible nature of agile, allowed for this switch to happen with no real impact on the deadlines. Additionally, Scrum allows for development and testing to occur simultaneously, meaning there is always new feedback from both the developers and the testers. This allows each task to be perfected before other tasks are completed, meaning each error or bug is caught in real-time, rather than waiting until the entire project is completed.

As mentioned above, there was a moment when it was decided the formatting for the list should change. The Scrum-agile approach allowed this issue to be addressed immediately, new test cases to be created in response, and the team was able to make the changes without losing much time. Along with this, there was a shift towards a detox and wellness focus in the travel destinations. This shift came relatively later, though luckily did not require a substantial change to the product.

It was in response to this change that the team was able to communicate effectively. The developers were able to send emails to the testers and the Product Owner requesting additional information on how the product should perform with this change. In the email to the testers the developer requested to know how the test cases would change, and if there were any issues that needed special attention. In their email to the Product Owner, the developer asked about the new priorities and product backlog. They wanted to know how the priorities would change with the focus on detox and wellness destinations.

Among the most useful tools, the use of a visible product backlog and user stories was the best. Having an organized Product Backlog helped the team see which tasks were the most important to work on first and which tasks were upcoming. This in conjunction with an Information Radiator helped facilitate the Daily Scrum meetings, by providing a visual reference guide on what each task was being worked on and by who, what had been completed already and what still needed to be addressed. This also helps during the Sprint as each person could physically move tasks along as they were being addressed then completed. Anyone could see the board and get an idea of where the project was at, in any given moment. The user stories helped in a similar way, by providing an easy access reference for what the task is addressing and why. The user stories were used to build every component of the product, so it was necessary to see them at all times as well.

The pros of the Scrum-agile approach for this project are clear: flexibility and organization. Scrum is rigidly organized and the timeboxes are hard set, providing a clearly laid out schedule and plan for the entire process. This made it easier to understand time expectations and required set times for retrospectives which encourages progress in subsequent sprints. It is due to this rigidity that Scrum is able to be so flexible. The strict schedule allows for tasks to be completed and tested, then adjusted, and tested, and adjusted, and so on until it is perfect. This constant cycle of feedback means that every task will be tweaked to perfection before a new task is started by each individual, but the staggered nature of having one person work on one task means that tasks are always being worked on and completed. The Daily Scrums are short 15 minute meetings which is a strict timebox, but encourages succinct discussion about progress and issues and allows for the team to remain up-to-date on everything. This also allows the team to bring up any issues or possible impediments to their work, giving the Scrum Master a chance to address the issues before they become too severe.

Based on my, honestly, limited knowledge of the various frameworks I would have to say that it is the best approach for this type of project. I can concede that there are likely other types of projects for which Scrum-agile may not be appropriate or at least, the best option, but for me and my team this approach worked seamlessly. Taking the prompt, “a niche travel booking system”, and the focus group responses, and turning these into user stories, then test cases, and implementing the program based on all of this foundation was very helpful for me. I had a clear understanding of what was expected, then I was able to work based on this. I would like additional experience in software development frameworks, but I did find Scrum-agile to be easy to understand and follow.