**Sprint Review and Retrospective**

Thomas Dang

SNHU

CS-250: Software Development Lifecycle

Prof Konishi

Aug 14, 2022

**Sprint Review and Retrospective**

Over the past few weeks, we have been working on the SNHU travel project. Utilizing agile methodology, we were able to deliver a finished product to the customer. In addition to the challenge to implementing agile, we also had to overcome a shift in the customer’s vision for the product. We accomplished by organizing as a scrum team, participating in scrum ceremonies, and applying the fundamentals of agile to our work and interactions to complete the sprint. Our team was composed of a product owner, scrum master, developer, and a tester.

The product owner is the primary point of contact with the company and is responsible for the backlog. During the sprint, our product owner was able to meet with the customer and get information about what the end users wanted to see in the finished product. Using this information, they worked to create the initial backlog for the sprint. When the customer requested a change to focus on health and wellness, the product owner was able to bring the customer’s new vision to the team and update the backlog to reflect the new requirements.

The scrum master is responsible for organizing scrum events, and helping the team have the tools and knowledge to perform their roles. During this sprint the scrum master was able to coordinate with the product owner and development team to begin the planning process after the product owner had met with the customer and prepared a backlog. The scrum master helped to establish when the daily stand up would take place and set an example of how the meetings should go. This helped ensure the meetings were beneficial for the team and did not get off topic. The scrum master could have done a better job explaining the daily standup before the sprint began, as there was some initial confusion as to whether the product owner should be expected at the standup. However, they did quickly explain that the product owner was not required and get the standup moving again.

The developer was responsible for turning the user stories into working code. The user stories created for the backlog helped the developer to understand both the requirements and the intent of each item. Understanding these helped to build the product in a way that would satisfy the end user. The stories being broken down into tasks that could be accomplished in a few days helped when the project’s requirements changed, as the developer was not too tied into one task and could change direction easily.

The tester developed test cases for each user story and was also able to assist in backlog refinement. By considering both the value statement and acceptance criteria, the tester could find areas where the story needed more clarification. The open communication agile promotes, and the known point of contact with the product owner meant it was easy for the tester to ask questions and refine both the test case and the user story.

Each role and its responsibilities played a part in the completion of the user stories, which in turn completed the project. The product owner began by meeting with the customer and end users to establish the requirements of the project and begin defining them as user stories. Once there was enough of a backlog to begin sprint planning, the scrum master organized the planning meeting, where the team could examine each story and determine how difficult it would be to implement or if the story needed to be broken down further. With the backlog now created, the developer could begin to create code, while the tester could create unit tests. The team could work closely together in part due to daily standups and in part the emphasis on communication the scrum master guided the team towards.

When the customer requested the shift in the project, the product owner was able to gather the required information and update the backlog. The scrum master was informed and set a meeting so the product owner could explain the change. After being told about the change and seeing the updated requirements, the tester was able to begin updating unit tests, while the developer could get the new backlog item and begin working on it. While the sprint was planned, the self-organizing nature of the scrum team and small size of user stories ensured no one was so heavily invested in one task that they could not change focus.

A key component of this was open communication. Agile suggests face-to-face is preferable, and we were able to do this in the product owner’s meetings with the customers and team, as well as the team’s daily standup meetings. Meetings in person gave a better opportunity to ask questions in the moment, and all team members were able to provide input for any question. Between meetings email and messaging apps were utilized. This was useful for questions needing to be directed to the customer, as well letting work continue while asking a question that might not need an immediate answer.

To keep the sprint organized we utilized several tools in addition to communication applications. Jira was used for the scrum board. This allowed the team and customer to see the progress of user stories during the sprint. The board displays the user stories, and each user story was able to be assigned precedence. As work begins the story can be moved across the board until it is complete, and the team member can select the next story to work on. In the stories, the team was able to make notes about any questions or issues they might see and those were visible to anyone with access to the board or create check lists for the implementation of the story.

Agile allowed the team to adapt to changes in the customers vision in the allotted time. With other management systems implementing the change in requirements may have meant delaying or even restarting the project. Compared to a waterfall approach, the team spent less time on documentation. While this is beneficial to completing code, some projects would benefit from heavy documentation. Adopting agile on a larger scale will mean multiple small teams can be working on any one project. This results in completing a project faster but could make tracking overall progress difficult. For the scope of the SNHU travel project, agile was an excellent choice.