**Sprint Retrospective**

Adrienne Sturgeon

CS-250: Software Development Lifecycle

Professor Vianelis Martinez

February 24, 2024

During the transition to Agile methodology, I was able to observe how adaptable and reactive Scrum can be for software development. In Scrum, there is a strong emphasis on having deliverable products at the end of short development cycles called sprints (Cobb, 2015, p. 12-13). As a Scrum Master I made a team charter in order to get an overhead view of what practices would help my team. I was able to define the success criteria for a small project, what the final deliverable product should be, and analyzed potential risks that might be encountered. As Scrum Master I set up the initial rules for conduct and other guidelines to ensure my team understood exactly how Scrum methods would be applied to our team.

A good product owner should be “responsible for maximizing the value of the product and the work of the Development Team” (Cobb, 2015, p. 35). The user stories that I created were invaluable for helping the Development Team envision the product from a user standpoint. By setting clear acceptance criteria I also made sure there was no ambiguity in when a feature could be considered completed.

These user stories would be the backbone of my work as a tester in the next module. Using those user stories I made test cases to determine if each feature was completed, not just from a technical standpoint but from a design perspective as well. Lastly, working as a developer I used feedback from the Product Owner to change an existing program. The ability to adapt to changes is fundamental to Agile, and implementing these changes helped me understand how software evolves alongside project requirements. Though the project initially displayed search results as a list, the final deliverable required a slide show format. Because there were detailed guidelines from the product owner, I was able to quickly transition the previously completed work to this new format and meet the acceptance criteria.

Each member of the scrum team interacted with the user stories differently, but they all worked together to help the final product be realized. The product owner first developed a product backlog of each potential feature, its level of priority, and an explanation of what user would find these features helpful. After reviewing this backlog three features were identified as most important: a vacation type search filter, a customized destination list, and a price limit search filter. As a tester I then used these user stories to build test cases. These test cases described what steps should be taken to use each of these features as a user, and what the expected results should be.

As the project shifted direction, I did find it important to reevaluate my previous work. Feedback from the product owner asked for search results to be returned in a slideshow format, which was a significant change from the previous list format. The first step to make sure that the project wasn’t derailed was to update the test cases. These updated test cases were still consistent with the user stories, but reflected how information should be presented to the user. An important step as both a developer and as a tester was to reevaluate the previous work. Understanding exactly what needed to be changed minimized unnecessary work, and much of the groundwork previously laid remained the same. Ultimately, these changes were easy to incorporate, and the general timeline of the project remained the same. This ability to change is one of the biggest advantages of Agile, as under waterfall each step in software development is unable to be revisited once completed (Cobb, 2015, p. 4).

During the restructuring of the project, I made sure to direct my questions to the appropriate team members. For instance, I had concerns about how the shift in focus would change our timelines. I wrote an email asking “How has this change affected the current ongoing user stories? Is there anything that’s been moved to the backlog that I should be aware of?” (Sturgeon, 2024). This direct communication prompted the product owner to respond, and helped provide clarification as to how this change would affect the project. I also made sure to specify that by answering these questions I would be able to do my work more confidently. This helped them understand my perspective better and gave us a better understanding of how my role used information.

At the beginning of the project a team charter was made as a method of sprint planning. This made sure that everyone on the team had similar expectations, and rules around communication were established. The team members and their roles were listed, and the project’s deadline was established. During the active development user stories were used to design features. This gave the team a better idea of not just how the product would work, but also how it would feel to use. This clarification allowed for rapid development during the sprint, and eventually test cases were developed to check that the work was sufficiently finished. As the product changed so did the test cases, and these were incredibly useful for ensuring that as the product evolved it was still meeting user expectations.

Ultimately, I think the Scrum-agile approach increased productivity, and made the project more flexible and adaptable. Under waterfall approach once a step in the development cycle is finished, there is no returning (Cobb, 2015, p. 4). When the product owner shifted the results of searches to be visually presenting as a slideshow, the project went back into the hands of the developers. This could have potentially increased the time needed to complete the project, but the result was a more robust program that better served the user’s needs. This sort of change can make developing general timelines more difficult, but there are ways to manage this risk. A burndown chart can visually show the rate that work is being completed and give a general idea of when the project could be done (Cobb, 2015, p. 110). I think the communication between the teams also enhanced both the product and the team’s experience. Being able to quickly get clarification made challenges like reevaluating test cases much easier to handle and empowered the team to work collaboratively. Agile was very successfully applied to the SNHU Travel project, and as the team continues to get more experience using Scrum, I believe that they will be able to use these skills in all other projects moving forward.

References:

Cobb, C. G. (2015). *The Project Manager’s Guide to Mastering Agile: Principles and Practices for an Adaptive Approach*. John Wiley & Sons. February 24, 2024, <https://web-p-ebscohost-com.ezproxy.snhu.edu/ehost/ebookviewer/ebook/bmxlYmtfXzkzNzAwOV9fQU41?sid=075b5d5b-9d00-40c2-a43a-598ef1432922%40redis&vid=0&format=EB&lpid=lp_iv&rid=0>

Sturgeon, A. (2024). Week 5 Journal [Unpublished paper]. Computer Science, Southern New Hampshire University.