

Ethan Klukkert

Professor Coddington

CS-250

14 August, 2022

Sprint Review and Retrospective

Product Owner: The Product Owner contributed to the project by serving as the mediator between the client and the other team members. The Product Owner analyzed the client's requests for the project, and translated them into user stories and epics (Module 3). This person also helped clarify any questions that the other members had on the project and its direction.

Scrum Master: The Scrum Master contributed by organizing the developers and helping answer their questions. The Scrum Master also organized the Scrum events and laid out the rules of behavior and production. The Scrum Master led each of the Scrum events, and allowed developers to be as independent as possible, while guiding them when needed. This team member also facilitates the Sprint reviews and retrospectives in order for the team to learn about what they did and can do better (Module 2).

Development Team: The developers on the team contributed by creating the initial features of the project. As the project evolved, the developers made sure to make the software as flexible as possible. (Module 5) The developers also made an effort to design software according to what the testers are expecting. This way, the software developed was flexible and needed little refactoring in order to test and deploy it.

Tester (Development Team Continued): The tester contributed by providing testing software to ensure that the developers' software was working properly and fulfilled

requirements. When directions were unclear or missing, the testers made an effort to contact the right people and ask the right questions to get the answer. (Module 4) For example, the testers needed a better idea of what the user stories wanted as far as displaying the vacation and profile setting options. The developers contacted the Product Owner for clarifying questions.

The Scrum-agile approach helped the user stories reach completion by organizing the stories by priority, and making them as independent as possible. A team member could work on a story at any time due to its independence, and if the direction of the story changed, it did not affect the entire project. For example, (Module 5) the Product Owner wanted a slide show display for the vacation packages. This change was accommodated due to the flexibility of the code and its separate responsibility. The changed display was not tied to the way the profile settings would be processed, because they are done separately. The Scrum-agile approach also helped to simplify needs into smaller chunks, stories, instead of larger chunks called user epics. Breaking down the user/client requirements allows for more flexible and clear development that brings results to the client sooner.

There was a moment when the project had to change direction. (Module 5) The clients of SNHU Travel wanted to change the focus of the vacation packages to wellness and detox instead of the old options. When this happened, the team had any questions answered, and had the work they already did saved for later use. This was possible due to the agile approach, where user stories and software is completed in a way that allows it to be reusable and changed depending on the client's needs. This approach enabled the team to make these changes without much friction technically and collaboratively.

There were two instances where the team communicated effectively. First, the developers had a rough time finding out what the requirements were from the client, so they contacted the

Product Owner for questions. These questions were concise and clear, for example: (Module 4) “Does it matter if the user expects the destination list to automatically display on the screen? Do the filters for the destination locations have to be within the main website, or within a separate profile setting?” The questions get to the point and are specific in nature, helping the product owner answer the questions effectively. The second time the team communicated well was the testing team clarifying the testing requirements. Part of the communication asked questions like: (Module 5) “Would they want the software to be tested in units or by module, and how would they like the functions and classes to be set up for testing?” This communication was critical, as it would affect how the code is written in the first place, and makes the tester’s tasks easier, and the developer’s work useful.

Jira is a tool that uses simplicity and accessibility to increase the efficiency of the team. It had a simple interface for team members to be able to use the tool effectively, and understand what was happening in the project. It helped project managers see the process from a non-technical perspective, yet also helped developers communicate stories and tickets effectively. The subtopics and further-detail sections enabled developers and Scrum masters to look further into the ticket or task to see how it was supposed to be completed or resolved. The principles of agile that helped the team the most were the 6th, “the most efficient and effective method of conveying information to and within a development team is face-to-face conversation,” and the 7th, “working software is the primary measure of progress” (Agile Alliance 2022). The team had important meetings in person, and discussed any outlying confusion and change in direction. The team then went off to make software that worked first, without adding any extra functionality. These principles allowed the team to get work done at a clear and constant pace.

The benefits of the development team using a Scrum-agile approach were user stories and fast, incremental delivery. The user stories allowed for the development team to put themselves in the shoes of the user, and find out what they would want in the product. (Module 3) The Product Owner made user stories based on customer feedback, such as top vacations based on user history, as well as filters for types of vacations and pricing. These direct communications allowed for the development team to develop user interfaces that are curated to the customer's needs. The continuous delivery is another great benefit of the agile approach, where the client could review the results so far in the project, and give feedback. This enabled an environment for the developers to improve the project, and for the client to get a project that better suits their needs. The con of the agile approach for this project was the problem of estimating the project's progress. The agile approach makes it difficult to find the progress of the project while changes are made and business needs change. It was difficult to determine what the "final product" was later in the project.

The Scrum-agile approach was the best approach for the SNHU Travel project. The nature of the project, being about vacation packages, calls for an approach that welcomes many changes, especially for the UI/UX development. There was a change in vacation display, and so the agile approach helped the team accommodate it. This kind of project has to be user friendly for most customers, since it is non-technical. So the team must be able to adapt to changes in UI/UX trends and vacation trends, while sticking to a schedule based on vacation season. Because of the nature of changes, communication was crucial, and the team was encouraged to do so with the agile approach. Overall, the Scrum-agile approach was the best approach for the SNHU Travel project due to the nature of business change and software adaptability.

Sources Cited

12 Principles Behind the Agile Manifesto, Agile Alliance, 2022

<https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>