**Sprint Review and Retrospective**

Andrew Jones

Southern New Hampshire University

CS-250: Software Development Lifecycle

Prof. Cindrell Harry

February 20, 2022

First off, for this initial sprint review and retrospective, I would like to detail how the role of Scrum Master contributed to the overall success of the SNHU Travel Project as they are the one that will be leading this meeting. Much like with this review and retrospective, the Scrum Master led the team meetings and ensured that each team member understood the Scrum framework and was able to work cohesively without the need for excessive input from the Scrum Master or the Product Owner. The most important part of this role was making sure that the team could become self-organizing rather than being reliant on leadership to offer input and guidance. The Scrum Master also facilitated some of the interactions between the different roles in the team including the Developer and Product Owner so that any additional changes by the client, such as the switch to the focus on wellness retreats, were understood by the entirety of the team and made known in a public place so that tasks could be added to the product backlog and discussed at the next daily standup.

Secondly, the Product Owner acted as a “hub” of information and communication throughout the project to make sure that the team was always on the same page and could be made aware of any changes requested by the client or request clarification by the client on any features. The Product Owner also organized the product backlog and ensured that features were delivered in a timely manner and in order of significance by analyzing the user stories they were able to create during meetings with the client and stakeholders. For instance, the Product Owner initially created the first set of user stories by analyzing the conversation that took place between themselves, the client, and several end-users and following the simple format of “As a <type of user>, I want <some goal> so that <some reason>”. They were then able to categorize these stories and present them to the team during the sprint planning so that the team could work on the features that the client and the Product Owner felt were most significant. Overall, the most important skill the Product Owner provides to the team is active listening. Without the ability to be a good listener, a Product Owner cannot effectively translate user stories, communicate between team members and stakeholders, or perform other necessary tasks.

The Tester is quite a difficult role to approach in the Scrum-agile approach as they must often extrapolate test cases based upon the user stories which may not always be complete or contain extraneous cases that were not initially considered by the Product Owner when the user stories were created. The Tester must ultimately consider the end goals of the stories in order to create the best possible test cases and also communicate effectively with the Product Owner so that they can clarify any extraneous cases with the client and ensure a clean, polished deliverable. In many instances, the Tester may actually be the one to find any issues or possible features that other team members may miss because they are not considering other possible inputs or asking the right questions of the client/stakeholders.

In the Scrum-agile approach, the Developer is ultimately responsible for developing a working deliverable for the client, however, they must also effectively estimate their completion times and communicate that to the Product Owner by using one or more agile approaches such as story points, burn-up/burn-down charts, etc. These methods allow the Product Owner to organize the product backlog and ensure a smooth development process rather than simply guessing at how much time a feature may take to develop. The Developer must also effectively communicate with the Product Owner so that they can clarify any questions that may arise, and also be available to changes presented to the Product Owner by the client, such as when the client requested the destinations be wellness-focused and presented in the form of a slide show instead.

The Scrum-agile approach to the SDLC helped each of the user stories come to completion in several ways. For instance, the user stories are initially created using the Scrum-suggested format of “As a <type of user>, I want <some goal> so that <some reason>”. The Developers are then able to look at this simplified format and conclude, using Scrum estimation techniques such as “planning poker”, the time it will take to complete the user story. Once the Development team has come to a consensus, the Product Owner can then organize the user stories into a clear and concise product backlog. This process then allows for the user story to be completed efficiently and with less confusion amongst team members. The Tester is also then able to develop their test cases based on what the end-user has requested the “some goal” to be. For example, in one instance, the end-user requested that they be able to view a “custom recommendation list” based on their travel habits and settings. The Tester knows then that in order to be successful the feature must include the acceptance criteria that the user is able to view their own custom recommended destination list amongst other things.

The Scrum-agile approach also made life significantly easier when there was a change requested by the client to shift towards a wellness/detox destination focus along with a slide show style presentation format. In a traditional waterfall approach, the project would have to have been completed in its entirety, then the SDLC would have had to begin again before the change could have been implemented. However, the Scrum-agile approach allowed the team to implement the change on the fly by simply adding the task to the product backlog and modifying or adding to the existing user stories. The Product Owner was able to effectively communicate the change from the client to the Development team who were then able to proceed with their tasks with relatively few obstacles.

As with most things in life, communication is key. Throughout the SNHU Travel project, there were several instances of effective communication. One such instance was during the group discussion in Module 6 where I took on the role of the Developer. Our Product Owner, Marques, neglected to mention how the team would be able to view the product backlog and other information related to the project, and, as such, my work as the Developer would be hindered by a lack of communication. Therefore, I suggested we integrate an information radiator or take advantage of software such as VersionOne to help keep the team on track and also improve time estimation for the future. Another instance of effective communication occurred during the Module 5 Journal in which I requested the Product Owner clarify a few points regarding the changes the client had requested, including what exactly a “wellness/detox retreat” was and if there was any additional information the end-user should be able to view on the slide.

Overall, the Scrum-agile approach was extremely effective for the SNHU Travel project primarily due to the fact that agile allows for on-the-fly changes and encourages effective communication skills to create a more cohesive and productive team. The structure of Scrum also ensures that team members do not stray off the path in the project and tasks are completed in an efficient manner. There are some downsides to Scrum, however. For instance, Scrum encourages code to be developed quickly and then refactored later on. This can lead to a lot of time spent refactoring and more effective solutions to be missed when they would have been found during the planning process of the waterfall approach. Scrum-agile is also less accurate for estimating timeframes when compared to the waterfall approach because things are done in short spurts rather than in one long fell swoop. With that being said, the Scrum-agile approach was the best choice for this project due to the fact that it was a relatively small development-wise so refactoring was minimalized and encouraged changes from the client which was a necessity.