**7-1 Final Project**

Joshua Brown

Department of Computer Science, Southern New Hampshire University

CS – 250: Software Development Life Cycle

Professor Jimmy Farley, Web Development (M.S., M.Ed.)

April 13, 2023

**7-1 Final Project: Sprint Review and Retrospective**

**As would normally happen at the end of a Sprint or an incremental release, the Scrum Master will put together a Sprint Review and Retrospective. For this deliverable, you will take on the role of the Scrum Master and create a Sprint Review and Retrospective to summarize, analyze, and draw conclusions on the work you completed during the course of the development.**

**Sprint Review**

Upon completion of the current Sprint, I have found that the shift to an agile methodology has been a stunning success. Starting the project, we met with the client where the product owner was able to gain meaningful insight into the customer’s needs for the project. From this, she was able to come up with a detailed epic of creating a customizable top-five list that could be derived from popular destinations and/or user preferences. The product owner then broke down this epic into smaller user stories and added them to the product backlog where they could later be organized and divided into Sprints accordingly by the development team during the Sprint Planning.

The user stories included points such as: having the user be able to click on the link from the homepage that would take the user to a top-5 list of vacation destinations; the functionality to navigate through the list and find all of the requirements for each listed destination; and the ability to click on each list and be taken to the appropriate destination package. Though there were other user stories that focused on other parts of the epic, these were the ones that the Scrum Team decided to address in the Sprint Planning for the current Sprint.

After the Sprint Planning, the development team successfully negotiated with the product owner what stories would be completed in the coming Sprint, and the product was underway. The testers immediately began to make test cases using the user stories from the product backlog. These test cases considered what was needed for the completion of each story and from here the developer team got to work creating code to meet these standards. It was around this stage that there was a slight shift in direction for the SNHU Travel project, voiced by the product owner on the behalf of the clients. This change shifted the focus of the SNHU Travel project on health and detox and would ultimately affect the type of vacation packages offered in the top-5 list; however, due to our new agile approach, the testers had no problem adjusting the test cases to match the new requirements in the user stories. From here the development team got to work on refactoring the code to meet the test cases and bring the Sprint to a successful end.

**Sprint Retrospective**

Looking back on the previous Sprint, it seems that the shift to an agile methodology suited my team quite well. Though we did not have the amount of documentation and project requirements that would normally be necessary for a traditional waterfall method, we found that we were able to swiftly move forward with the project based on the specifications that we had, adapting to handle any changes that came along the way.

During the course of our Sprint, the client requested a change in the direction of the project. After discussing the changes with the product owner, they ultimately decided that it was in the company’s best interest to shift their focus towards health and detox for the project and had the product owner adjust the user stories accordingly. Though the change was indeed different from what was originally discussed with the client, an agile methodology gave us the flexibility to adjust mid-Sprint to tackle these new specifications without having to go back to the beginning to recalculate cost due to changes in product specifications that would normally cause a waterfall method to come to a grinding halt.

Breaking this down even further, I found that the factors of an agile methodology that made this shift easy were: because we had divided the project into Sprints based on user stories, we could assess the changes that were required and determine how they would affect the current Sprint and from there decide if the changes could be made in this sprint or if they should be added to the backlog of a future sprint; and because we focused only on work by Sprint, the Scrum Team was not exceedingly ahead in the project and did not have to go back to do a lot of rework of the project.

We also found during the course of the Sprint that elements of Scrum along with maintaining swift and proper communications with each other through the use of emails, meeting with the team, and the utilization of the Daily Standup to communicate the teams’ daily efforts that we were able to efficiently stay on task and rapidly adjust to changes in the project. During the initial part of the Sprint, the testers showed excellent foresight by emailing the product owner to confirm details of the project in order to make test cases. The fast response of the product owner allowed work to proceed as scheduled without any downtime in the process. It was also imperative to know what communications required an email such as the case with the testers needing more information, and which communications required an actual meeting with all parties, such as the shift in direction with the whole project. The product owner did an excellent job, requesting a meeting to discuss these changes with the team; it was through face-to-face contact and interactions that she was able to accurately confer with the group the changes that needed to be made, and we were adequately able to respond to those changes leaving no room for confusion amongst the team.

Overall, I believe the move to a Scrum-agile approach was the best for this particular project. A challenging point that might have made us a little uncomfortable at the beginning of the project, was the fact that an agile approach did not require as many product specifications as what would normally be required in a waterfall project. Despite this, the product owner did an excellent job pulling user stories from the clients' initial interactions, which allowed for a strong and swift start to the project, also attesting to the strength of an agile methodology. Where this method truly shined was when the project shifted at the behest of the clients. If this were to happen in a waterfall approach, it could have very well been a make-or-break event for the entire project that would cause the project to go back into the planning stages, require a great deal of rework, and inevitably lead to a giant push to be done by the product deadline. However, agile is much more adaptive in nature; with it, we were able to account for the changes and even decide if they needed to be addressed now or if they could be put off until later. With that being said, I highly endorse the use of the Scrum-agile methodology; although, I would ultimately recommend considering the cases in which to apply this kind of methodology by examining the benefits of agile and from there decide if a project would profit from its adaptive nature.