**Looking Back on the Sprint: Sprint Retrospective**

Dominic Clapper

Southern New Hampshire University

CS 250: Software Development Lifecycle

April 17, 2022

Now that the Sprint has reached its end, it is very important for us to look back at what we have done. Throughout this Sprint, we worked a lot on the SNHU Travel Project as a whole, but most of the time was spent on improving the user experience for the project. We also started testing out the scrum-agile methodology to facilitate the project development. There was a lot of new information to take into account with the change in software development style, but it has been very insightful. The first topic of major importance to discuss in regard to the Sprint is what everybody on the team did to contribute to the project development.

Since we were asked to utilize the scrum-agile methodology for the development of this project, roles had to be assigned to each of our team members. These roles were, Scrum Master, Product Owner, and Developer/Tester. Each team member was in charge of certain tasks which would help the team complete the project development. The Scrum Master for the team was tasked to essentially manage and/or facilitate the team in order to keep things organized. A good example of something specific the Scrum Master did for our team during this project sprint was managing the many meetings that were held amongst the team. Meetings such as the daily standup meetings and a sprint retrospective meeting, which took place at the end of the sprint, were facilitated by the Scrum Master.

The Product Owner for our team was mostly in charge of maintaining/managing the relationship between our project development team and the customers/users. In terms of specific work they did to contribute to the project as a whole, they were the connection between the development team and the customer/users. The product owner held a meeting with the users early on in the project, where key information was learned about what the users wanted to see in the software. Then, the Product Owner organized these requests into stories and placed them into the product backlog, while also accounting for which requests were of higher priority. The tasks that the Product Owner completed, such as managing the product backlog, were insanely helpful for the development/testing team because it made it much easier to understand what the users wanted. The development/testing team for this project were basically in charge of developing the actual software for the project, as well as testing the software for any problems. For our team, we had a developer who was in charge of creating the actual software, and a tester who was in charge of making test cases to ensure the software worked accordingly. One of the tasks that the developer was in charge of was developing the slide show feature that the customers wished to see. This required changing the current list format of destinations to a controllable slideshow format. The reason the developer had to make such a change was because the customer wanted it instead, which was communicated to the Product Owner, and then to the developer. Like mentioned above, the Tester on our team was in charge of creating/managing test cases to check that the software worked correctly. One specific task that the Tester was in charge of was developing the test cases for the user stories provided by the Product Owner. These test cases were then revised in order to more properly check the software for vulnerabilities.

The next major topic of discussion is how useful the scrum-agile methodology has been in regard to completing the user stories provided to our team. One way scrum-agile helped the team complete the user stories was by having different roles handle certain tasks involving the customers/users. The Product Owner’s main task in the scrum framework was to handle communications with the customer/user, and to also handle archiving any changes that the customer/user wants. Having one person focus on managing what needs to be worked on in regard to the users wants/needs took a lot of weight off of the development team’s shoulders. Basically, instead of having to focus on figuring out what needed to be worked on first, and what should be worked on at all, the development team was able to solely focus on actually developing the software. Another way scrum-agile helped the team complete the user stories was by providing open communication with the customers/users. Through this open communication, it was easy to discern what the customers/users actually wanted in terms of features or aspects of the software. This made it much easier for the development team to understand what was being asked of them, which in turn, helped the team create high value software that was very appealing and specific to the customers/users wishes.

At one point in the Sprint, the customers/stakeholders asked for a change to be made in the software to accommodate for a predicted trend in vacation destinations. This change involved turning the list view for top destinations into a slideshow for top destinations. Even though this customer/stakeholder requirement was given to the team out of nowhere, the team was able to adapt and handle the situation perfectly. The scrum-agile approach really helped in this situation because of how flexible the approach is in dealing with customer/user requirements. Instead of waiting until the end of the major development for the project to change the software, the team was able to make the change much earlier. This was because agile allows for such customer/user wants and needs to be communicated faster to the project team.

Throughout most of the Sprint, there was not much official communication involved. By this, I mean that throughout the duration of this course, there was not much official communication that I actually did/happened among the team in the module assignments. Most of the official communication made was through a few meetings, such as the one in module 5 discussing changes the stakeholders wanted developed. These meetings were definitely useful for the team as they allowed for information to be shared and discussed in an open manner. All the team members had a voice in the discussion which encouraged everybody to give their thoughts on the matter. This led to the proper changes and improvements being made for the project. Other than the few meetings, most of the communication amongst our team was theoretical, such as the sample emails I developed in the module 4 journal and the module 5 journal. Even though the emails were only theoretical, they still encapsulated the aspects of proper, effective communication. The emails were worded in a way that ensured I would get a timely response, as well as an informative response. This was accomplished by giving a time frame of when I would need a response by and providing very specific questions in order to get detailed answers.

Overall, I feel that the scrum-principles and organizational tools we utilized for this project helped us create high value content and helped our team grow closer as a unit. The Scrum events such as the planning meeting and daily scrums were insanely useful because they allowed our team to meet and discuss topics related to the project in a calm, open, and productive environment. This helped our team succeed massively because our team members were more focused on work inside a comfortable environment, where everybody was on the same page. Also, some of the Scrum-Agile principles such as improved adaptation and openness made our communication channels much more organized, which made completing the work an easier task. Being able to adapt to changes that the customer/user wants was much easier compared to the older SDLC method since communication was more streamlined. As a result, we were able to complete work faster, communicate better, and succeed more efficiently, which resulted in a much better, higher value product.

The major question of this retrospective is whether the scrum-agile methodology was the right model/methodology to use for the SNHU Travel project. In my own personal opinion, I believe the scrum-agile methodology was the better option for the SNHU Travel Project. I say this mainly because of the level of uncertainty related to the project. The details for what the user wanted with the project were decently defined prior to the start of the Sprint, but there was still a lot of room for possible changes. Such as with the list view of the destinations and even some of the other user stories such as the profile management settings. Even though it did take a bit more time and effort to add the changes the users/customers asked for, it was worth it because the value of the project was more relative to users/customers wishes. Another reason why the scrum-agile methodology made sense to use for this project was because the scale of the project was not all that large or complex. There were quite a few user stories to accomplish for the project to be considered completed, but compared to other, much larger projects, the requirements were not as complex. In reality, both agile-scrum and waterfall methodologies can be used for any situation. But, for this scenario, it felt much better to use the agile-scrum methodology.