**Sprint Review and Retrospective**

David Smith

Department of Computer Science, Southern New Hampshire University

CS 250: Software Development Lifecycle

Professor T. Morrison

April 18, 2021

Throughout the course of this experiment in Scrum-agile development, the development team has shown their ability to be flexible and efficient in their tasks which have enabled us to adapt to change while still meeting our goals. Overall, Scrum-agile has been more efficient than the waterfall method for this project and our team.

The Product Owner developed User Stories and prioritized them to ensure that they are ready for selection as we went through each Sprint Planning. They also made sure that the client’s and users’ input was passed along to the team. This enabled the development team to better understand the tasks on which they were working. This was especially noticeable during the changes being made based on user feedback on what changes would make the product more useful to them.

The Scrum Master acted as a facilitator for everything the development team needed to stay productive. They accomplished this by making sure that each event stayed on track, removing obstacles that hindered development or productivity, and being available to the team. This role was extremely important during this process as they lead the team in discovering what Scrum is and how it works. They also refined backlog items to ensure that they were correctly sized to be able to complete in a sprint and available for selection at the next sprint planning. The availability of the Scrum Master was helpful in keeping events on topic and within their time allotments. They were also helpful when the client was going directly to developers during a sprint and distracting them from completing their tasks.

The developers worked diligently on the tasks that they chose during each sprint planning while asking for help when needed and being flexible with changes that were asked of them. Having access to the Product Owner while changes were occurring enabled them to ask questions that helped refine the changes to tasks and avoid developing components that did not align with expectations. This enabled the developers to adjust to changing requirements as they occurred instead of finishing the product before realizing they needed to make changes that could have been more easily done earlier in the process. This saved a lot of time and effort put into code that would never be useful.

The testers created clear acceptance criteria for the User Stories. These requirements allowed the developers the ability to know what exactly their work needed to accomplish. The testers were also able to make quick changes to adapt to different user expectations. When the User Stories changed after the Product Owner had a chance to ask users what they would make the system more useful to them, the testers were able to quickly change what they were checking instead of having to wait for the developers to complete the task and hand the product over to them.

Having User Stories to define the tasks for the developers to take into each sprint allowed them to see how the part of the product they were working on would be used and useful to the users. This allowed the developers to understand what was being required and why. By having the User Stories prioritized, the developers were able to take the most important pieces and get them out to the client first. This allows the client and Product Owner the opportunity to decide that the product has reached a point where other User Stories are not going to add any real value to the product. Having specific results defined in the User Stories, the developers had the information they needed to complete their tasks.

By working in a Scrum-agile environment, the team avoided some of the pitfalls of the standard waterfall method. Once User Stories were available, testers and developers could start working on sprints to create useable pieces of software for the client. In the waterfall model, the client and project management spend a lot of time and effort on specifications and requirements before any actual development is done. This is a large amount of effort with no actual client useable results. In the Scrum-agile environment, the development team starts work with much less refined requirements assuming they will change. This allows the team to create value for the client more quickly.

The value of the Scrum-agile method to software development was most apparent when changes were being made to the requirements in the middle of the project. The users were asking for different ways that they could search and filter their results to see travel options that were more relevant to them. The client also decided that they wanted to focus their product on a different type of travel. The Product Owner discussed these changes with the rest of the development team and changes were able to be made almost immediately. Each team member was able to express their concerns with the changes, and the Product Owner lowered the priority of some items in the backlog to ensure that the team would have time to make the changes that were being requested. Having the ability to make these changes in the middle of the development process reduced the amount of code that would be changed, enabled the testers to change their acceptance criteria quickly, and allowed these changes to occur quickly and efficiently.

The agile tools that we used, including Scrum events and task tracking, were extremely helpful in making sure that we stayed on task and productive. The Sprint Planning allowed each person in the development team the opportunity to select the tasks that they would be able to complete during each sprint. The Daily Standup allowed each team member the ability to see how the team was progressing, whether anyone was having any difficulties, how to amend those difficulties, and brought the team together as a more cohesive group. During the sprints, having task tracking through a Kanban board or software tool allows each part of the team the opportunity to keep track of progress.

The Sprint Review gave the development team a chance to show what was accomplished during the sprint to each other and the stakeholders. This gave each the opportunity to see how things were progressing, what might need to be changed, and what tasks should be considered as next steps in the project. This gave the team a chance to show the work they have completed, explain any issues they encountered, and how they overcame those issues. The final Scrum event, the Sprint Retrospective, only involved the development team and gave them a chance to evaluate their progress. This was a great chance to explore how we are working both as individuals and together as a team. It also let us evaluate how our process was working and discuss any changes we needed to make. As the Scrum-agile process is iterative, we progressed through each of these stages multiple times throughout the development process with each iteration bringing the team to a better understanding of each other and the Scrum-agile process as a whole.

I believe that Scrum-agile was the best method to use for the SNHU Travel project. With the changes that were being made throughout the development process, the adaptiveness of Scrum-agile fit very well and made modifications to requirements easy and effectual. The only negative aspect that I could attribute to this process was that it was new to the team using it. The team, therefore, had to learn Scrum-agile while also working through the project. This worked well with this team, because they pulled together and made it work. Having a larger team with more people set in the waterfall method of development could have been a harder transition to make.

Communication is key in a Scrum-agile process. Being able to express how your work is progressing, what is holding you back, and if you need help determining what needs to be done or how to do it are all extremely important. The following are some examples of communicating with the Scrum team to clarify changes and work through issues.

To: Christy

CC: Brian

Subject: Changes to product requirements

Dear Christy,

I have been looking over the changes requested in our meeting about focusing on wellness/detox travel. Could I get clarification on the definition of detox/wellness travel? I just want to make sure that I am including the correct types of destinations and activities in the travel packages that will be included in these results. I am CCing Brian on this so that he can get the same definition for these types of travel for his test cases.

Brian, could you please send me your updated criteria for test cases so that I can make sure we are on the same page as to what we are expecting.

Thank you,

David Smith

To: Christy

Subject: User Story Clarifications

Dear Christy,

I have been going through the user stories that you submitted and am working on developing test cases to determine pass and fail scenarios for the acceptance criteria. Some of the information is not detailed enough for me to make a determination on how to evaluate the performance of the product with test cases. Would you please elaborate on the following for me?

**User Story 7**

When a user clicks the link to set a price limit for their displayed destinations, would the results appear on a new webpage or would the previous results be updated on the same webpage?

**User Story 3**

Should this filter option be the default interaction with the webpage or a link to this specific filter on a separate webpage?

**User Story 8**

Should the Hot Deals be displayed as a numbered list, descending percentage off of normal price list, or alphabetized list? Should the individual items on the list be displayed as an actual list, a slideshow, or some other format?

Thank you,

David