**Sprint Review and Retrospective**

Justin Starr

Department of STEM, Southern New Hampshire University

CS 250 – Software Development Lifecycle

Professor Joseph Martinez

February 19, 2023

# Sprint Review and Retrospective

Each member of our agile team contributed in their various roles to help achieve specific goals/requirements, which became our definition of “done” for this sprint. To demonstrate how the various roles on our Scrum-agile Team specifically contributed to the success of the SNHU Travel project, the following is an outline of each role and the contributions they made.

**Scrum Team Roles and Responsibilities/Contributions**

* Product Owner – At the beginning of the project, as the product owner, I met with stakeholders to gain insight into the product's vision. I obtained as many details as possible to communicate this vision to the rest of my team and begin product development. I was given the opportunity to ask clarifying questions when uncertain how a specific feature would function, with my primary goal being to learn as much about the user requirements as possible. Once this meeting ended, I could begin the development of the product backlog, which includes all the user stories and the requirements for each story. The product backlog is a vital resource for our team that details where on the product backlog user stories are, what the priority level is of each story, its name, how long it is expected to be, as well as its value statement and its acceptance requirements. Some of this information will be more clearly defined once we begin our Scrum meetings, and I can gather more information from the entire team, such as how long user stories should take to complete. This information is essential to ensuring the team stays within the scope of each current sprint while working to achieve goals that have been defined for a “done” state.
* Scrum Master – As the Scrum Master, it was essential that a meeting first occurred between the Product Owner and me so that the product vision could be communicated. Based on this initial meeting, I developed a team charter which includes information such as the Business Case/Vision, the mission statement, a list of all team members and their roles, the success criteria, key project risks, rules of behavior, and the team’s communication guidelines. As the Scrum Master, it is vital for my team that I strictly enforce this charter and uphold the highest standards possible and me. I worked to ensure our team followed the agile manifesto, all while providing a positive and constructive work environment for every team member. I implemented daily scrum meetings, which only lasted about 15 minutes each so that all team members had the opportunity to speak about what they were going to work on that day, what they accomplished the day before, and to communicate any problems they encountered that hindered their ability to complete their tasks. I quickly resolved any issues during our daily standups and helped ensure the scrum meetings stayed on a focused track in the event conversation became sidetracked or off-topic. Another essential resource that I implemented was the sprint backlog. The sprint backlog is like the product backlog; however, it is more specific to the current sprint. Suppose problems arise during the sprint concerning a particular feature. In that case, we can look to modify the current sprint, or if it is determined that a specific feature may take us entirely out of the scope of the current sprint, we can work with the Product Owner to look at making changes to the product backlog, which would likely include a new sprint for the feature that was removed from the current sprint because of its scope. I want to note that while I started the sprint backlog, it is essentially maintained by the entire team during our daily standup meetings. I also helped to facilitate our sprint review and sprint retrospective.
* Tester – As the Tester for our agile team, I utilized the acceptance criteria for each user story to create measurable tests on a pass-or-fail basis. The acceptance criteria listed the essential attributes and functions that needed to be tested for each user story. At times, I found specific details, such as how the Top 5 Destinations should be presented (in a slideshow format or other) and even how specific information should be laid out to the user on each of the various destinations. Once I realized we were missing this critical information, I reached out to the Product Owner to determine how the team should proceed based on the needs of the stakeholders/users. Once the details I needed had been relayed back to the team, we could define and revise our user stories better. As the Tester, it was also crucial that in the beginning, I quickly implemented a policy for myself to work with everyone on the team so that together we could build quality and improve the product as early as possible.
* Developer – As a Developer for the SNHU Travel project, I was assigned tasks/requirements from the user stories that I was responsible for developing. I worked diligently to follow practices such as not changing functionality where possible, meaning that as the product was being developed, we only wanted to change the code necessary to incorporate new features or design elements. If the product was already working, we did not want to make changes that could drastically change the entire project. This could potentially cause a landslide of events, such as errors that did not previously exist. Therefore, it was critical that I only modify or implement new code where necessary that advances the project to its completed or “done” state. There were times when I was unsure of specific details; however, they were not significant enough to hinder my ability to produce working software. As the sprint progressed, I made sure to address my issues for clarification purposes. Some of these issues included the sizes of text boxes, as the current size limited how much information could be included in destination descriptions. I thought about incorporating a scroll pane but decided it was unnecessary and could lead to unwanted errors in the already functional software. It was also unclear to me the true definition of a destination that could be considered wellness/detox. I think this could be a very generalized and broad idea, and I wanted to make sure the terms were not used in too broad a sense. In instances where I did need specific information to move forward, I acted quickly by sending out communications to both the Product Owner and Scrum Master to receive a timely response where I would be able to continue product development without delay. This also helped keep my team members aware of my status, any problems I encountered, and critical information the team might need moving forward. A key ingredient to my success was agile and how adaptable and flexible it affords us not only as individuals but as a team altogether.

**Sprint Review and Retrospective Continued**

A Scrum-agile approach to the Software Development Life Cycle (SDLC) helped each of the user stories come to competition by using small independent teams who utilize an adaptive approach instead of a plan-driven approach such as the waterfall approach (Cobb, 2015, p.4-p.5). This adaptive approach enabled us as a team to only have a small amount of information about the SNHU Travel project to begin work. As the project progressed, more information became available. By utilizing this adaptive approach, user stories could be worked on and modified as needed based on new information to continually move the product to a finished or “done” state. For example, when I previously spoke about the tester role, I included an example of how the tester realized critical pieces of information were missing. Once we received this information, we could re-evaluate each user story and make changes where necessary while incorporating the additional information that was missing.

A Scrum-agile approach supported project completion when the project was interrupted and changed direction when, at one point in the project, there was a shift in focus from a broad range of travel destinations to s more specific types of destinations, wellness/detox destinations. The Scrum-agile approach was beneficial because we could change a specific direction of the project without significantly impacting the work each team member was working on at that time. Because agile promotes a highly collaborative environment, the Scrum team was able to make this shift/modification to the user stories with very little impact/hardship on the team as a whole, which supported project competition.

Working in an agile environment means working within a team that communicates effectively. Without effective communication, the team would be working more independently, leading to significant problems as the project moved forward. For agile to work correctly, team members must be in continuous communication. An example of when our team needed to communicate with each other was during our daily scrum meetings. In these meetings, each team member hears essential information about each member individually and the team as a whole. What one member is doing can affect another member’s work, so this is a critical time when questions can be asked of each other if needed. An essential aspect of these meetings is face-to-face communication. This is preferred to be done in person; however, for teams that operate outside of an office environment, this can also be accomplished through video conferencing. Face-to-face communication is vital because so much context can be lost through electronic communications such as emails or even text/chat messaging. Much of our communication is given by body language, and the tone of a person’s voice and messages can get lost or poorly translated/interpreted when these forms of context are removed from a conversation. Face-to-face communication encourages collaboration in a group setting where individuals have a turn to speak and voice their opinions. Oftentimes, team members may have ideas about how a task should be completed, especially in situations where one team member may have a problem and is unable to determine the best solution. This can be an excellent opportunity to get feedback from other team members, especially when there could be more experienced members who may already have a great solution that can be implemented.

At some points in a project, there may be times when emails need to be used to get information faster, for example, rather than waiting for the next scrum meeting. During product development, there was a specific time when the tester needed to send an email to the Product Owner to determine specific requirements that had yet been previously discussed. The following is a sample of concerns that were mentioned in the email to the Product owner:

**User Story One**

* First, it appears that there are no design specifications for the application at all. What kind of look and feel does the customer wish to have? How should the top five destinations be presented when the appropriate link is clicked to view the destinations?
* Should the destinations be presented in a slideshow format, or would they simply like to see one bog list that the user is able to scroll through?
* Is the customer concerned about having specific profiles for each user, or should the user be setting specific characteristics to a travel search each time they visit the website, never actually storing any personal information about the users?
* If we are creating profiles, exactly what type of information will be pertinent to each profile? What information do you need to know about the customers? Name, Address, Telephone... etc. Do we want to have an email address or any other type of information they think they need to know about their customers? If so, what are they?
* If saving personal information, are there statements we should make available about how we handle this type of information? Should this information be available on a separate page, perhaps an about us page?

It can be seen in this communication that very specific questions were asked of the Product Owner, which helps ensure that the answers needed to move forward are provided. The communication was very direct, and it can be implied that the required information from the Product Owner should be considered time-sensitive. The sample is organized in a way so that the receiver can address all critical questions for each topic that is presented. In this example, collaboration is encouraged by having the ability to reach out to any team member as necessary. Also, when the tester receives an email back from the Product Owner, they can pass the information to all team members so that they can react accordingly to the responses. This helps keep everybody in the loop, and if anybody else has questions that may be related, they, too, can have a voice in the conversation. This, however, may become a conversation that needs to happen in the daily standup, depending on how many or what types of changes need to occur. This is an excellent example of why the Scrum Master should be a part of these types of communications, as they can help ensure the team stays within the scope of the current sprint/user story.

Another example I wanted to provide is when our team initially started. It can be challenging and, at times, daunting to change the direction of projects from a plan-driven method to agile. The team was asked to collaborate and adopt at least one agile principle from each team role (Product Owner, Scrum Master, Tester, and Developer) to get the new Scrum team started and moving forward. To accomplish this goal, each team member wrote about the specific agile principles associated with their role and then asked questions to our team members for clarification about any questions they had about those roles. Once questions had been asked and answered, the team created a visible list of each role and all the principles mentioned, where then the team could agree upon at least one principle to adopt for each role. Again, this encourages collaboration by providing an opportunity for each other to ask questions about the various scrum team roles or interject when they believe a specific principle might not be appropriate at that time. One example of when this occurred was when the team was leaning on adopting two principals for the tester role, continuous integration and code refactoring, until later in the conversation, when the adverse effects of doing so became understood, the idea to implement code refactoring was decided to be held for future use. Again, here, it had yet to be thought of at the beginning how code-refactoring can significantly impact a project, both negatively and positively. Still, because refactoring can cause portions of code to stop functioning, it might be more beneficial to dedicate an entire spring to code refactoring. This way, if significant problems are discovered while cleaning up code, there is a whole sprint worth of time to address the issue.

While working on the SNHU Travel project, various organizational tools were implemented to help the agile team be successful. I think the most critical organizational/communication tools that an agile team can implement are the use of information radiators, face-to-face communications, daily standups, and various methods of communication among distributed teams (Cobb, 2015, p.139 – p.142). One communication method among distributed teams is using services such as Microsoft Azure Boards. This is an incredible tool that effectively uses critical aspects/components across agile teams. Azure Boards provides its users the ability to track various types of work, such as user stories, bugs, features, and epics; it offers interactive backlogs and can also track sprints for scrum events (Microsoft Corporation, 2023). Azure Boards effectively use an online information radiator so that information can be accurately communicated with minimal loss of context in messages. Concerning teams in an office environment where they meet daily, I believe the use of information radiators is highly critical as one tool that works alongside face-to-face communication and daily standups. Daily standups are essential to creating an environment that promotes collaboration. Simply having a scrum and making it a “standup” meeting invites team members to interact with each other. It allows members to have face-to-face communication and promotes the use of additional information radiators such as whiteboards where all team members have a chance to add information where they can see each other’s progress. One of the most significant benefits of all these concepts is that it promotes “one of the most important values behind agile,” openness and transparency (Cobb, 2015, p.139). Openness and transparency promote team collaboration and help to prevent any illusion of progress. This means that all team members are aware of the current status of each other’s tasks, not just the status of the overall project. If one person is having trouble with a task, another team member can jump in and help when they can provide solutions.

To assess the effectiveness of the Scrum-agile approach for the SNHU Travel Project, it is essential to address the pros and cons that the Scrum-agile approach presented during the project and to determine whether a Scrum-agile approach was the best approach for the SNHU Travel development project. The pros presented during the project were primarily based on how agile is adaptable and flexible, as well as the collaborative environment it supports. When changes needed to be implemented into the project, there was a small amount of re-work required, and the shift in focus from all types of travel destinations to a more narrowed type of destination was easily incorporated. Similarly, when information such as layout design and how information should be presented needed to be included, work could still be conducted while the team awaited a response from the Product Owner. On the flip side of this same pro, I would say it was also the project’s biggest con. While it was beneficial to begin working with only a small number of details, it could have resulted in significant changes needing to happen to the code later had the users not liked the direction the team decided to take. Fortunately for this project, the information missing was insignificant enough to cause a considerable amount of code refactoring. I do believe that a Scrum-agile approach was the best approach for the SNHU Travel development project because the final product was not dependent on any variable. In contrast, if an error occurred after the final product was delivered, it would not severely impact the project. This was not a project that needed to be plan-driven due to the types of errors or things that could go wrong, and if something did go wrong, the impacts would be minimal and easily re-workable with nominal costs.

**References**

Charles G. Cobb. (2015). The Project Manager’s Guide to Mastering Agile : Principles and

Practices for an Adaptive Approach. Wiley.

Microsoft Corporation. (2023). *What is Azure Boards?*. [https://learn.microsoft.com/en-](https://learn.microsoft.com/en-us/azure/devops/boards/get-started/what-is-azure-boards?view=azure-devops&tabs=agile-process)

[us/azure/devops/boards/get-started/what-is-azure-boards?view=azure-devops&tabs=agile-](https://learn.microsoft.com/en-us/azure/devops/boards/get-started/what-is-azure-boards?view=azure-devops&tabs=agile-process)

[process](https://learn.microsoft.com/en-us/azure/devops/boards/get-started/what-is-azure-boards?view=azure-devops&tabs=agile-process)