**CS 250 Final Project – Sprint Review and Retrospective**

Buddy Marcey

[buddy.marcey@snhu.edu](mailto:buddy.marcey@snhu.edu)

Southern New Hampshire University

August 13, 2023

**Sprint Review and Retrospective – Introduction**

The SNHU travel project has now come to a close. The following is a retrospective which will examine many aspects of the experience and discuss how using the agile methodology specifically aided in the successful completion of the project. Examples will be provided that show how an agile mindset can be used to solve problems and guide team members to the best possible outcomes. We will begin by examining the Scrum team members and their roles, followed by discussing how an agile-based approach helps to complete tasks and solve problems, and finally, we will discuss some good communication practices and organizational tools that contributed to the success of the SNHU Travel project.

**Scrum-agile Team Roles and Their Contributions**

**Product Owner:** Our product owner was responsible for handling our backlog and developing user stories to help convert information gathered in focus groups and client meetings into a format that could be readily implemented and tested by the team. The focus groups that she held allowed her to make the user stories that formed the basic aspects of the SNHU travel software; by asking the users what they would like to see from the program functionally, she could develop a strategy to implement those features. For example, one of the focus group members mentioned wanting to be able to sort vacations by type. The user story she generated read “As an end user, I would like the ability to sort recommendations by type, so that I can choose the type of vacation that best suits my needs.” Her meetings with the client allowed her to share our work in progress, which in turn allowed the client to make changes and guide the final touches they wanted to see in the product; in fact, developing the previously mentioned user story led the client to push for a change to emphasize the Detox and Wellness vacation types.

**Scrum Master:** Our Scrum master made sure the trains stayed on schedule, so to speak. In our daily Scrum events, we discussed what we were doing for the day, what we did the previous day, and what might get in the way of having a successful day today. The Scrum master’s main tasks were to make sure the rest of the team had everything we needed to successfully complete the project by leading and coaching the team.

**Testers:** Our testers took the user stories and built test cases for each of them so that the developers would have a clear definition of the requirements of each story. For example, the user story generated above by the Product Owner was then broken down into easily testable cases, such as “User clicks on vacation type: Passes if types are displayed and function correctly when clicked”, along with several other test cases that, when passing, lead to properly functioning software.

**Developers:** Our developers were responsible for writing the code that would pass the test cases that were handed off from the testers. Developers can fall into many categories, but for our project, they seemed to be well-rounded in back-end, front-end, UI-UX, etc. The developers created the software and wrote the functionality to pass tests based on the client feedback and focus group data gathered in earlier sessions. They had to work together in order to fulfill the requirements of the sprint within the sprint window at the highest quality.

**How Scrum-Agile Completes Tasks**

The Scrum-agile approach to user stories follows a pattern. First, the information is gathered through focus groups or client meetings, typically by the product owner. The user stories are then defined, at which point the testers usually turn those user stories into test cases. The test cases are then passed to the developers to write the code that can pass the test cases. All the while, the Scrum master is keeping everything organized and keeping everyone connected. For example, the client wished to see Detox and Wellness vacations emphasized. That was turned into a user story by the Product Owner, and test cases were made by the testers. The developers adjusted the existing program to reflect the emphasis on Detox and Wellness vacations according to the specifications laid out in the test cases. Every team member had a hand in meeting the client’s expectations quickly and efficiently.

**How Scrum-Agile Adapts to Changes**

One major strength of a Scrum-agile environment is adaptability. Because of the iteration cycle, where minimally functioning code is tested and then built upon and strengthened, changes can be made without having to start all over again. Having constant feedback from the client and users allows for changes to be made with much less interruption and heartache than could be achieved in the waterfall method. For example, during our project, the client wished to change from the “Show the top 5 destinations” type model we had already created into a slide show with a heavy concentration on Detox and Wellness type vacations.

Under the waterfall methodology, the client would not have had that opportunity to make changes on the fly, as they would not even see the product until it had been fully developed and tested. Going back and making changes at that point would have been prohibitively difficult. Under the Agile model, however, the client could request a change to the work in progress and those could be implemented with minimal effort. Overall, this leads to a much happier customer experience, as the final product will align better with the customer’s needs.

**Effective Communication in a Scrum-Agile Environment**

Communication is vital to the success of a Scrum-agile team. Face-to-face collaboration is probably the most effective way to communicate, but in cases where team members are geographically isolated from one another, face-to-face communication might be impossible. In the absence of physical presence, the second-best method would be via Zoom or some other video call platform, followed by instant messenger, and finally email. It is too easy to ignore an email, even if doing so is unintentional; maybe the recipient is busy and intends to get to it later, but it gets lost in the day. Instant messengers are slightly better because at least the sender can see the status of the recipient. Sometimes, however, we are at the mercy of the tools available, or in some cases the urgency of response is more relaxed. Here is an email provided from the SNHU Travel project requesting additional information:

buddy.marcey@snhu.edu

To: John Agileman (Product Owner) [john.agileman@SNHUTravel.com](mailto:john.agileman@SNHUTravel.com)

Dear Mr. Agileman,

I hope this email finds you well. I am working on the SNHU Travel project, and I need some additional information regarding the functionality. That user story is an epic and needs to have additional planning and backlog refinement for future sprint implementation.

Specifically:

1. We need to know the functional requirements of the app. Does it need to have all the same features as the online software? Should it have any additional features that are usually more mobile-friendly?
2. What should the app look like? Should the visual elements look the same as the online software, or have a unique look and feel?
3. Should the mobile app link to our existing booking and billing systems, or should we develop new mobile-specific solutions?

The answers to these questions can help us break the project up into manageable parts and help with sprint planning. There is likely a large difference in cost to develop a whole mobile app from the ground up, versus writing a mobile-friendly web experience.

Thanks,

Buddy Marcey

Requesting specific information will help the team to produce the desired output with the least amount of ambiguity. Communication is key; no one wants a project ground to halt because one part of the team needed additional information to proceed that could only be collected from one place. Careful planning can help a lot, but no amount of planning will ever remove the need for effective communication.

**Tools and Events**

**Events:** This section can be broken down into tools and events; first I will focus on the events and why they are important. First, daily Scrum events help to keep everyone on the team on track. Facilitated by the Scrum Master, these brief daily meetings allow the team members to see what each member is working on and identify any potential impediments. Prior to each sprint, typically every two weeks, is a Sprint Planning meeting where the tasks for the upcoming sprint are determined and story points are discussed. Several possible sprint estimation practices usually coincide with this event. And finally, Sprint Review and Retrospective events conclude the sprint; they evaluate the success of the sprint and identify any problems that came up, to prevent making the same mistakes in future sprints. This presentation is an example of how a retrospective might be presented.

**Tools:** There are several organizational tools that can be used to help keep the sprint on track and keep everyone engaged and working together. I will discuss a couple. Kanban boards show the tasks broken down and organized; in an environment where the whole team works together in the same space, this might be a physical board with something akin to post-it notes or index cards that can be moved around and modified to show who is working on which tasks, which tasks remain, and the completion stage each task is in. Communication tools, like messenger apps or discussion boards, are a common tool, whether the team is close together or not. Sometimes, a software solution makes the most sense; Jira software implements all the communication methods one could imagine, as well as a virtual Kanban board, timelines in Gantt format, swim lanes, and all manner of task management. It is especially useful for teams who are geographically separated or who work remotely but could also be a great solution for teams in the same space since it is extremely versatile and highly functional. Organizational tools are essential to efficiently managing projects and vary from team to team or across different work environments.

**Retrospective and Summary**

Now that all the details have been discussed, the final question remains: Is Scrum-Agile the best approach to the work we do here? For this project it almost certainly was. The defining characteristic of the methodology is in the name itself: Agile. The definition of agile is “having a quick resourceful and adaptable character” (Miriam-Webster, 2023). For our SNHU Travel project, the customer wanted to change directions with the program after work had already begun. Under the waterfall methodology, they would not have had the opportunity to do so, as they would not even be able to see a working product until all the work was done. The only real counterargument to completing this project using the agile methodology was that the whole process was new to us. Converting the whole company to an agile environment will be quite an undertaking; people are generally resistant to change, especially when things seem to be going well. Hopefully, this retrospective can assuage some fears and show how beneficial the change to agile can be.

**References**

Merriam-Webster. (n.d.). *Agile Definition & Meaning*. Merriam-Webster. https://www.merriam-webster.com/dictionary/agile

Overeem, B. (2016). *Characteristics of a great scrum team - amazon web services*. Characteristics of a Great Scrum Team. https://scrumorg-website-prod.s3.amazonaws.com/drupal/2016-08/Characteristics%20of%20a%20Great%20Scrum%20Team.pdf

Cobb, Charles G. (2015). *The Project Manager’s Guide to Mastering Agile: Principles and Practices for an Adaptive Approach.* Wiley.

Atlassian. (2023). *Jira: Issue & project tracking software*. Atlassian. <https://www.atlassian.com/software/jira>

Schwaber, K., & Sutherland, J. (2020). *The 2020 scrum GUIDETM*. Scrum Guide | Scrum Guides. https://scrumguides.org/scrum-guide.html#daily-scrum