# SNHU Travel Sprint Review and Retrospective

Malcolm Marberry

Southern New Hampshire University

CS 250 Software Development Lifecycle

Professor Zhen Li

April 20, 2025

## Applying Roles

### Scrum Master

The Scrum Master helped educate the team on Agile principles and facilitate agile development of SNHU Travel. They created the Agile Team Charter which outlined the project: business case/vision, mission statement, communication guidelines, etc. Before development, they also researched Agile principles and development strategies that the team could use. The Scrum Master assisted the team's incorporation of the Agile methodology.

### Product Owner

The Product Owner contributed by translating the client's demands into specific features of the SNHU Travel application. They created prioritized user stories with acceptance criteria that could be added to the Product Backlog—stories which could be implemented and tested by the development team and testers respectively. The size of each item was estimated to help with prioritization. When the client's requirements changed, the Product Owner clearly communicated with testers and developers to meet their new demands. This included a demo application to help testers update their test cases. By communicating the business, functional, and non-functional requirements of the application, the Product Owner provided testers and the development team the information they needed to ensure the agility of their contributions.

## Testers

Our team of testers communicated with the Product Owner and development team to create test cases for each functional requirement of the application. They responded to the Product Owner's feedback to create cases that had clear pass criteria for each step. Testers also updated their test cases to match the changing user interface of the app. They questioned the Product Owner and development team to better understand the acceptance criteria of certain features and user stories. The testers helped verify that each user story was fulfilled according to our client's vision.

## Development Team

The development team implemented the features requested by each user story. In the early phases of the project, they created an app with simple list-view control which testers could create test cases for. Later in development, when the client's requirements, explained by the Product Owner, shifted towards focusing on detox/wellness vacations, the development team made the relevant adjustments. They also changed the app from a list-view control to a slideshow at the request of the Product Owner. Our team of developers have repeatedly delivered features matching the user stories of the project.

## Completing User Stories

Scrum-Agile development centers around user stories—helping with their timely completion. Stories are created, backlogged, interpreted as app features, then tested based on predetermined acceptance criteria. This process follows the first principle of Agile: “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software” (Bailey et al., 2024). Development focuses on one story at a time to completion, and cross-story dependencies, such as not being able to complete one story until another is finished, are identified beforehand so as to avoid reworking on completed stories. For example, when I was working as a developer, my task was to adjust the application so that its Top 5 Destinations list focused on detox/wellness vacations. I worked on that one item/story until it was completed; when I believed the app reflected requested changes at least. The next step would be for a tester to create a test case to verify that the story was completed. That would conclude the story’s development, allowing me to start working on another. This strategy of Scrum-Agile increases the speed in which stories are completed by simply prioritizing items one-at-a-time.

## Handling Interruptions

As previously mentioned, the project experienced a sudden change in requirements when the client wanted to focus the Top 5 Destinations List around detox/wellness vacations. If the project were managed using something like the waterfall model, then this adaptation would've significantly interrupted development. In Scrum-Agile however, the change could be added to the user-story-to-feature pipeline like any other. After a meeting between testers, developers, the Scrum Master, and the Product Owner, the team pivoted to the new, high priority user story. I explained what this meant for the development team in the previous section. One of the concerns in the meeting was that the team's previous work would have to be "scrapped", but the change was implemented after a few relatively small adjustments to the original program thanks to the code's modularity—another strategy of agile development. After the tester makes their adjustments to the test cases, the interruption would be accounted for.

## Communication

Communication and collaboration are core parts of Agile development, especially between the Product Owner, testers, and development team. These three must work together to efficiently deliver on user stories. As an example, here is an email sent to the Product Owner by one of our testers:

"I have some questions regarding three of the user stories added to the backlog as I'm developing test cases and wanted to confirm the acceptance criteria:

'Destination Recommendations'

- Does this include a user preferences feature where users can influence the destinations that are recommended to them?

'Searching by Vacation Type':

- What kind of form should the selection of vacation type be? Should it be a drop-down menu, checkbox, flag-based system. etc.?
- How many search results should be returned? And should there be a unique message if there arent any available destinations matching the desired vacation type?

'Setting a Price Limit':

- How many destinations should be in the search results?
- Does this include deals and discounted prices?

I also had a general question regarding whether the lists produced through these features alter the Top 5 Destinations list or if they are separate? Thank you for your time."

In their message, the tester asked the Product Owner clarifying questions about specific user stories. While this communication encourages collaboration, it's also an example of how to get an effective response. They used a bulleted list to organize their questions—making them easier to respond to—and they kept their email brief. Another example is one of our developer's emails to the test team:

"After the latest meeting, I had some requests regarding the test cases for the newest requirement of theming SNHU Travel's destination to detox/wellness retreats:

- Could you please add a test case for verifying that each item in the Top 5 Destinations list fits the criteria of being a detox/wellness-centric vacation?
    - Also please verify that the text descriptions for each destination explain why it's a detox/wellness retreat or include a summary of the vacation"

This email was in response to the project's shift to detox/wellness-themed vacations. The developer wanted to request that certain changes be made to test cases. It uses the same communication strategies as the tester's email: bulleted list, brevity, and direct questions. That kind of efficiency encourages collaboration by making it easier to send and respond to emails or other forms of written communication. It would also improve the effectiveness of in-person meetings if similar strategies were used to mark the key talking points. Verbose communication is anti-Agile. To follow the fourth Agile principle, daily collaboration (Bailey et al., 2024), communication needs to be efficient and effective.

## Organizational Tools

Tools like Gantt Charts, Kanban boards, and Azure boards can be used for the “efficient and effective” communication mentioned in the previous section. Our team organizes the project around a Product Backlog and uses the principle of “face-to-face" (Bailey et al., 2024) conversation to delegate tasks. The backlog is referenced during Sprint Planning and Daily Scrum to schedule who and when user stories are worked on. The size and dependencies of each story also influence its priority in the schedule. For example, the schedule was influenced by the new, high priority user story when the new detox/wellness requirements were introduced. During an in-person meeting, the Product Owner de-prioritized other items in the backlog so the team could complete the new story. The Product Backlog was used to represent, and in this case reorganize, the project’s schedule.

Evaluating Agile Process

Pros

- User stories were completed efficiently
- The interruption of a new, high priority story didn't have a large impact on the project's development
- The team was able to communicate and collaborate more frequently
- The development team had clear direction: user stories directly translated to app features
- Test cases could be adjusted to new requirements

Cons

- Agile introduces scope-creep—meaning the application can easily continue to grow and become more complex as more user stories and features are implemented
- Time constraints: in the meeting where the detox/wellness change was discussed, the Product Owner said that the schedule couldn't change to account for the change "since this is agile"
- Because agile "welcome[s] changing requirements" (Bailey et al., 2024), it can put strain on the team and reduces the predictability of the project; i.e. the tester needed to change test cases several times according to the Product Owner and development team's requests

I believe that Scrum-Agile was the best approach for the SNHU Travel development project because of the small team, relatively small project, and the adaptability it created. Agile works best with small teams according to the "Two-Pizza Team" that says that teams in agile development should have the amount of people it would take to eat two pizzas. The project is relatively small but could involve long-term maintenance and development as the client determines new requirements. As such, Scrum-Agile would be better than another strategy like the waterfall model because it supports long-term development and change. SNHU Travel, as an application, centers around its users—adapting to their preferences and recent trends—so it's likely that changes like the shift to detox/wellness-themed vacations will happen again in the future. Scrum-Agile, as discussed previously, allows those kinds of interruptions to be seamlessly integrated into a project. For those reasons, I believe Scrum-Agile was the best choice for this project.

## References

Bailey, D., Rathod, Y., & Alliance, A. (2024, September 12). *12 principles behind the Agile Manifesto: Agile Alliance*. Agile Alliance |. https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/

Kristensen, S. H., & Paasivaara, M. (2021). What Added Value Does a Scrum Master Bring to the Organisation? — A Case Study at Nordea. *2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Software Engineering and Advanced Applications (SEAA), 2021 47th Euromicro Conference on, SEAA*, 270–278. https://doi.org/10.1109/SEAA53835.2021.00041