Sprint Review and Retrospective

The SNHU travel project provided me with the chance to experience Agile development through the Scrum framework while practicing different team roles. Over these past two months, I have acted in multiple positions including Scrum Master, Product Owner, Developer, and Tester. Each role gave me a different perspective on how Scrum functions and cow communication, flexibility, and organization come together to drive a project forward. In this retrospective, I will review how our team applied roles, completed user stories, handled interruptions, communicated effectively, and the importance of organizational tools. I will also reflect on whether Agile was the right process for this project and what lessons I will carry forward in the future.

Scrum is build on clearly defined roles, and each one came with its own responsibilities that contributed to the team’s progress. As Scrum Master, I facilitated events like Sprint Planning, Daily Stand-ups, and Retrospectives. My goal in this role was to keep the team aligned, ensure that our backlog was revised, and create a safe space for identifying improvements. Later, in my role as a Product Owner, I gained a stronger appreciation for the importance of stakeholders and users. Engaging directly with users helped us prioritize what mattered most, such as recommendations based on past trips and the ability to filter destinations by type. As a Tester, I relied heavily on acceptance criteria to write meaningful test cases, ensuring that user expectations were met in practice. Finally, in my Developer role, I experienced how critical it was to stay flexible and communicate frequently with both the Product Owner and testers when requirements shifted.

User stories formed the foundation of our team’s work. Each one captured a specific user goal in the format of “As a user, I want… so that…”. This structure kept our focus on solving real problems rather than building unnecessary features. For example, one story required displaying personalized top travel destinations with names, descriptions, and booking links. From that story I created test cases that confirmed that each piece of information appeared correctly. Another story centered on allowing users to filter trips by vacation type, which directly shaped development tasks and testing priorities. By breaking down the work into small increments and clear acceptance criteria, we could confidently say whether a story was complete. This made the review process transparent and meaningful for the team and stakeholders.

One of the biggest lessons came from adapting to interruptions and changing requirements. Midway through the project, the Product Owner shifted the focus of the destinations to detox and wellness trips. This change significantly affected the backlog, the user stories we prioritized, and the features that testers needed to validate. Instead of derailing the project, Scrum allowed us to adjust quickly. As a Developer, I communicated with the Product Owner to clarify new requirements, like whether specific wellness programs or booking options should be included, At the same time, testers updated their test cases to align with new filter and search functionality. These interruptions reinforced how Agile thrives in changing environments by encouraging collaboration and iteration rather than a rigid, linear framework.

Effective communication was the key to our success. Daily stand-ups helped us share our progress, raise blockers, and coordinate next steps. Backlog refinement sessions reduced confusion before Sprint Planning by ensuring that stories were well understood and prioritized. I also learned the value of direct communication outside of events. For example. As a Tester, I sent emails to the Product Owner asking specific questions about unclear requirements, such as whether results should display in a slideshow or as a single page. This clarified expectations early and prevented unnecessary delays. Peer code reviews during development was another helpful practice that caught issues early on and spread knowledge among team members. All of these communication methods kept us aligned and built trust, which was essential when the scope of the project changed.

Scrum ceremonies served as organizational tools by creating structure during development. Tools like Jira for backlog management and sprint tracking were also helpful, allowing us to visualize progress, while burndown charts gave insight into whether or not we were on track. Acceptance criteria also acted as an organization tool by aligning development and testing work with user expectations. For example, the test case assignment required me to check whether vacation filters returned accurate results and whether users received appropriate messages when no matches were found. Having these details documented game both developers and testers a shared understanding of what success looked like. These tools and processes ensured that the team stayed aligned and focused even when priorities shifted.

Scrum-Agile was the right choice for this project. Its iterative nature allowed us to respond to changing requirements without scrapping out prior work. The short cycles and continuous feedback made it easy to deliver meaningful increments and show progress at every review. The biggest benefits were flexibility, transparency, and stakeholder engagement. However, Scrum also required consistency. Backlog refinement had to be done regularly to avoid confusion, and communication needed to be consistent to prevent delays. If this project had been run using a Waterfall approach, the mid-project pivot toward detox and wellness destinations would have been much more disruptive, and we likely would have had to push back the timeline. Agile’s adaptability made this shift much more manageable.

The SNHU Travel project was both a learning experience and a practical demonstration of how Agile works. In each role as a Scrum Master, Product Owner, Tester, and Developer, I gained unique insights into what it takes to deliver value as a part of a Scrum Team. User stories and acceptance criteria helped us stay user-focused, whole communication and organizational tools like stand-ups, backlog refinement, and Jira kept us aligned. When requirements changed, Scrum’s iterative process allowed us to adapt and continue moving forward. Overall, I’ve learned that communication, flexibility, and structure are equally important to Agile success. These lessons will shape how I approach future projects, ensuring that I deliver valuable results and contribute to a collaborative and adaptable environment.