Sprint Review and Retrospective

Jalexia Woolfork

Southern New Hampshire University

CS 250: Software Development Life Cycle

October 24, 2025

Applying Roles

Throughout the SNHU Travel App project, I rotated through multiple Scrum roles, which gave me a strong understanding of how each position contributes to successful Agile development. As the Product Owner, I defined the project vision by focusing on customer needs such as an intuitive interface for booking and exploring destinations. I translated these goals into clear user stories that guided development priorities. Acting as the Developer, I implemented core features like destination lists and user-friendly navigation. When I served as the Tester, I performed functional testing to ensure that each feature met acceptance criteria and worked across different devices. Finally, in the Scrum Master role, I facilitated sprint planning and daily stand-ups, helping maintain focus and resolve blockers quickly. Each role played a critical part in delivering value to the customer while reinforcing collaboration and accountability within the Scrum framework.

Completing User Stories

The Scrum-Agile approach was instrumental in helping user stories reach completion efficiently. Each sprint focused on manageable tasks, such as designing the main screen or improving user experience. Using iterative cycles made it easier to adapt when priorities changed or technical challenges appeared. For example, after receiving feedback on navigation layout, I quickly revised the prototype during the next sprint instead of waiting until the end of the project. This incremental approach allowed continuous progress and reduced the risk of major issues late in development. The emphasis on sprint goals and regular reviews kept development aligned with user needs, ensuring that every user story contributed directly to the overall project objective.

Handling Interruptions

One major advantage of the Scrum-Agile process was its ability to adapt when unexpected interruptions occurred. During development, I encountered technical difficulties importing image files for destination cards. Instead of halting progress entirely, I used the flexibility of Agile to re-prioritize tasks. The sprint backlog was updated, and I shifted focus to enhancing the destination descriptions and interface while troubleshooting the image issue separately. This flexibility allowed progress to continue without delaying the entire project. Agile’s iterative structure supported quick adjustments and prevented disruptions from derailing the project timeline.

Communication

Even though this project was an individual simulation, I practiced effective communication strategies as if I were working within a real Scrum team. I documented all updates in sprint summaries and reflected on progress during sprint reviews. My written communications were concise and transparent, emphasizing what was accomplished, what challenges arose, and what adjustments were made. In a real team setting, these updates would have encouraged collaboration by ensuring that all members were informed and aligned on goals. Clear communication is a cornerstone of Agile development because it fosters accountability, transparency, and trust across the team.

Organizational Tools

Using organizational tools such as Trello and Jira-like task boards made it easier to visualize workflow and manage sprint backlogs. These tools supported task prioritization, progress tracking, and sprint planning. Scrum events like daily stand-ups, sprint reviews, and retrospectives provided structured opportunities for reflection and continuous improvement. By consistently reviewing sprint outcomes, I was able to identify what worked well—such as early feature testing—and what could be improved, like refining backlog estimates. The combination of Agile tools and Scrum events significantly enhanced productivity and kept development organized and transparent.

Evaluating the Agile Process

The Scrum-Agile approach offered both advantages and challenges throughout the SNHU Travel App project.

Pros: The method encouraged flexibility, continuous feedback, and early issue detection. It allowed me to respond quickly to changes and maintain focus on user needs. Frequent reviews helped ensure that each feature delivered real value.

Cons: Working alone meant there was no real-time collaboration or feedback from teammates, which limited some of Agile’s benefits, such as diverse problem-solving perspectives. Additionally, time-boxed sprints required strict self-discipline to stay on schedule.

Overall, Agile was the best approach for this project. The incremental progress and adaptability suited the evolving design needs of the travel app much better than a rigid Waterfall model would have. While Waterfall might work for projects with fixed requirements, the SNHU Travel App benefited from the flexibility and responsiveness of Agile.

References

Beck, K., & Andres, C. (2020). Extreme Programming Explained: Embrace Change (3rd ed.). Addison-Wesley.

Cohn, M. (2019). Succeeding with Agile: Software Development Using Scrum. Addison-Wesley.

Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. Scrum.org.

Serrador, P., & Pinto, J. K. (2019). Does Agile work?—A quantitative analysis of Agile project success. International Journal of Project Management, 37(5), 599–609.