**Sprint Review and Retrospective**

Charles Mills

Southern New Hampshire University

CS 250: Software Development Lifecycle

Professor Deron Dantzler

2/19/2025

**Applying Roles**

Throughout the development of the SNHU Travel program, I have had the opportunity to take on multiple roles within the Scrum-Agile framework, and to experience how they each play such a crucial role in the success of a project.

As the Product Owner I was responsible for gathering the user requirements through interviews, surveys, and stakeholder meetings and took that feedback and used it to curate a product backlog for the development team to work through in sprints. My role was crucial to maintaining an environment of open communication and efficiency, ensuring that the team could quickly move through the backlog in a way that met the standards of the stakeholders and the product vision. When I needed the development team to make changes, such as the travel destination types, I communicated clearly and directly through an email, and then reviewed and confirmed the changes to be satisfactory.

As a developer I learned how important it is to maintain flexibility. Often changes need to be made mid-development, and as a developer it is my job to accommodate. Agile, being a dynamic project management framework allowed me to make those changes and to communicate with the rest of the team so that we could re-arrange the backlog and stay on target. Communication is the most important aspect of this framework, so daily interaction with the team and product owner was necessary to gain clarifications on any revisions made to the product.

As a tester I ensured that features in development were meeting the standards and quality expected of the owner. I validated user stories, tested features such as the travel destination filter, found bugs, did regression testing, and provided my feedback to the rest of the team. By working closely with the rest of the team, the product tester ensured that the final product met the technical specifications and quality standards that the user and owners expect of the product.

As a Scrum Master, I facilitated daily stand ups and communication between the team and the product owner. This position is held by a single member of the team and has the responsibility of creating an environment where the team shares feedback and new related to where we are at in our sprint. These meetings ensured that we were making progress as expected and gave the team an opportunity to raise any concerns, and to keep each other up to date on the progress of the product. In short, the Scrum Master role was crucial to maintaining momentum while ensuring that the product was staying aligned with the vision set forth by the owners and stakeholders.

**Completing User Stories**

The Scrum-Agile approach to SDLC allowed user stories to be developed incrementally, giving us the time and flexibility to ensure that each component met technical standards even as we made changes on the during the sprint. For instance, when we received a user story about wanting a feature to filter destinations by price, we were able to break that down into tasks that could be developed and tested before release rather than waiting for the final finished product to be tested. This example illustrates the way that Scrum-Agile allows us to complete user stories in the most efficient manner, so that by the time the product is finished, most of the stories have been well refined.

**Handling Interruptions**

This leads into another common occurrence while working on a project: interruptions. Interruptions and changing priorities are all but guaranteed during the software development lifecycle, and are a feature, not a bug. An example of this was the change in direction to “wellness vacations”. This was a change in directions that occurred mid-development, but rather than stopping our work, we just held a sprint review to reassess the requirements and adapt to the new directive. Sometimes interruptions can be avoided entirely by doing your due diligence and requesting clarification on a feature before implementing it into the program.

**Communication**

The backbone of our project was communication. All aspects of the development cycle were strengthened by communication practices such as daily stand-up meetings, emails between the team and product owner, and sprint reviews, alongside normal day to day communications. There are various ways for a team to communicate, whether through the daily scrum, around the office, over email (as we did with the product owner several times), or through information radiators such as message boards, and whiteboards. Most important to clear and effective communication is a team that feels comfortable with accepting different *tones* of communication. Sometimes the feedback someone gives you can make you feel like you didn’t do well enough, and that is why it is important for the team to be respectful and concise in how we talk to one another.

**Organizational Tools**

Various organizational tools contributed to our success. JIRA helped us manage the tasks during each sprint, as well as track progress and adjust on the fly. Azure Boards and Git repositories allowed for members of the team to have access to the project versions and to coordinate between the developers and testers. And as I discussed in the communication section of my retrospective, the information radiators such as white boards and burndown charts allowed the team to have locations with physical information for them to visualize at any time. Finally, our sprint reviews were a useful organizational tool for us to show the actual results to the owner and stakeholders, while retrospectives let us reflect on the work we had completed and discuss what useful tools to take forward into the next sprint.

**Conclusion**

The Scrum-Agile approach provided out team with a great many advantages that allowed us to move quickly and efficiently through our work. SNHU Travel needed a team that could adjust their product on the fly, and without moving the timeline too far back in the process, and the agile methods that we used were crucial in providing that. Another big pro was getting continuous feedback throughout the process. This allowed us to ensure that the product met the expectations of the owner before we got too far ahead and saving us from wasting time. Collaboration between had the same effect, allowing the team to be on the same page and not get out of alignment with various parts of the development. A con that I can think of would be that the constant requirement changes afforded by Scrum-Agile can sometimes feel overwhelming and might make you unsure of if what you are doing one day is going to be what the owner wants on the next day. Daily stand-ups can also feel somewhat redundant sometimes when there is not much news to share with the team. Despite these “inconveniences”, scrum-agile was the best approach to this project. The iterative nature of it allowed us to adapt and deliver results with the least possible resistance. Agile facilitated healthy collaboration, flexibility, and efficiency throughout the entire project. Having clearly defined responsibilities ensures that each member of the team knew their role, and how to communicate with others in their respective roles. This project reinforced the value of Agile principles for me.

**References**

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., … & Thomas, D. (2001). Manifesto for Agile Software Development. Agile Alliance. https://agilemanifesto.org/

Boehm, B., & Turner, R. (2003). Balancing agility and discipline: A guide for the perplexed. Addison-Wesley.