**Sprint Review and Retrospective: SNHU Travel Agile Project**

CS 250: Software Development Life Cycle

Southern New Hampshire University

August 11, 2025

Professor Judy Mason

Kyle Dauk

**Sprint Review and Retrospective**

Throughout the SNHU Travel project, the Scrum-Agile approach allowed each team role to contribute uniquely to the success of the development process. Acting in multiple capacities during the simulated sprints provided an appreciation for how these roles interact. As the Product Owner, I helped define and prioritize user stories based on stakeholder needs, ensuring that the backlog reflected the most valuable features. In the Developer role, I implemented these stories into working software increments, adapting quickly when acceptance criteria required changes. As the Tester, I validated features against defined expectations, identifying issues early so they could be resolved within the sprint. Finally, in the Scrum Master role, I facilitated sprint planning, daily stand-ups, and retrospectives to keep the team focused on delivering value and removing impediments. These roles, while distinct, relied heavily on collaboration, shared accountability, and transparency to keep progress steady and aligned with project goals (Agile Alliance, 2022).

The Scrum-Agile approach to the software development life cycle (SDLC) was instrumental in bringing user stories to completion. The iterative nature of sprints allowed stories to be broken into manageable tasks that could be developed, tested, and reviewed quickly. For example, in Sprint Two, the addition of a destination filter feature for the SNHU Travel application moved smoothly from backlog prioritization into development and testing within a single sprint. Frequent sprint reviews ensured the product was always evolving in line with stakeholder expectations, and the definition of “done” was applied consistently to ensure quality. This was a clear contrast to a waterfall approach, in which all requirements might have been gathered up front, risking late discovery of gaps or misaligned functionality (Cobb, 2015).

Interruptions and changes in direction were managed effectively due to the flexibility of the Scrum-Agile framework. Midway through Sprint Three, a request was made to integrate a last-minute promotional discount feature. Instead of halting progress or derailing the sprint, the team reviewed the backlog, assessed the impact of the change, and adjusted priorities. The work was reorganized so that core deliverables remained on schedule, while the discount feature was developed in parallel. This adaptability demonstrated the value of iterative planning and the principle of responding to change over following a rigid plan (Scrum Guide, 2020).

Communication within the team was clear, concise, and effective, which supported collaboration and reduced misunderstandings. For example, when clarifying acceptance criteria for the booking confirmation page, a short message in the backlog item discussion thread ensured that both developers and testers shared the same understanding of what the completed feature would look like. This direct communication helped avoid rework and kept everyone aligned. Additionally, sprint retrospectives encouraged open dialogue about what was working well and what could be improved, fostering trust and shared ownership of the process.

Organizational tools and Scrum-Agile principles played a central role in the team’s success. The Kanban board was an effective visual management tool, allowing everyone to see the status of backlog items at a glance. Sprint planning meetings ensured that only a realistic amount of work was committed to each iteration, while sprint reviews provided valuable feedback from stakeholders. Retrospectives created a structured opportunity to discuss process improvements, leading to greater efficiency in subsequent sprints. These tools and events reinforced core Agile principles of transparency, inspection, and adaptation (Agile Alliance, 2022).

Evaluating the effectiveness of the Scrum-Agile process for the SNHU Travel project reveals both strengths and trade-offs. On the positive side, the approach promoted adaptability, rapid feedback, and continuous improvement. It allowed the team to adjust to evolving requirements without derailing the project and provided frequent opportunities to engage with stakeholders. However, the process also demanded consistent participation and discipline from all members; without active engagement, the benefits of the framework would be diminished. This mirrors lessons from Cobb’s Case Study B, where a waterfall approach led to delays and misaligned deliverables because adjustments could not be made until late in the process.

In conclusion, the SNHU Travel project demonstrates the value of applying Scrum-Agile principles to software development. By emphasizing collaboration, adaptability, and continuous feedback, the team was able to deliver incremental value, respond effectively to change, and engage stakeholders throughout the process. While the framework requires discipline and active participation to achieve its full potential, the benefits of transparency, early feedback, and iterative improvement clearly outweighed the drawbacks. These lessons affirm that Scrum-Agile was the most effective approach for this project and provide a foundation for its broader adoption within ChadaTech (Cobb, 2015).

**References**

Agile Alliance. (2022). *What is Agile?* https://www.agilealliance.org/agile101/

Cobb, C. G. (2015). *The project manager’s guide to mastering agile: Principles and practices for an adaptive approach*. Wiley.

Scrum.org. (2023). *The Scrum Guide*. https://scrumguides.org/