Sprint Review and Retrospective: SNHU Travel Project

Jennifer Joseph

Southern New Hampshire University

CS 250: Software Development Lifecycle

Applying Roles Across the Scrum-Agile Team

My journey through the SNHU Travel project was a learning experience that allowed me to step into each core Scrum role. I was a Scrum Master in Module 2, a Product Owner in Module 3, a Tester in Module 4, a Developer in Module 5, and a Scrum Master once more during the group discussion in Module 6. This hands-on experience of the software development life cycle (SDLC) from various perspectives has deepened my understanding of how Scrum values, transparency, and iteration are the driving forces behind success in Agile environments.

As the Scrum Master in Module 2, my primary focus was to facilitate Scrum events effectively and promote a culture of servant leadership. I guided the team through Sprint Planning, Daily Scrums, Reviews, and Retrospectives, ensuring that each event fulfilled its purpose and stayed within the designated time limits. My facilitation style was rooted in servant leadership, which emphasized supporting the team rather than directing it. I encouraged equal participation, established clear agendas, and promoted constructive communication while reinforcing Agile principles of transparency and continuous improvement (Schwaber & Sutherland, 2020). This experience taught me that disciplined facilitation and emotional intelligence are essential for maintaining morale and productivity in an Agile environment.

In Module 3, while serving as the Product Owner, I translated stakeholder needs into actionable user stories. Insights gathered from the User Focus Group animation highlighted what customers valued most: top destinations, customizable filters, and price-based searches. By writing clear and prioritized user stories and refining them collaboratively, I ensured that the Product Backlog accurately reflected both user intent and business goals. Managing stakeholder expectations while balancing technical feasibility deepened my appreciation for Agile's principle of "customer satisfaction through early and continuous delivery of valuable software" (Beck et al., 2001). This role emphasized the Product Owner's responsibility to continuously adapt the backlog as feedback evolved.

Transitioning to the role of Tester in Module 4 highlighted the crucial role of communication and shared understanding in quality assurance. While developing and revising test cases for the SNHU Travel application, I found that well-written user stories with clear acceptance criteria were the easiest to verify. In contrast, missing or ambiguous requirements often led to confusion, emphasizing the importance of direct collaboration with the Product Owner. Agile testing demands flexibility, critical thinking, and teamwork instead of isolation. By proactively asking clarifying questions and aligning tests with user value, I was able to contribute to both functional accuracy and customer satisfaction (Inayat et al., 2019).

In Module 5, as a Developer, I applied iterative development practices to convert backlog items into working code. This process involved interpreting changing requirements, implementing updates for features like the "Top 5 Destinations Slideshow," and collaborating closely with testers to resolve defects. I learned to request clarifications quickly and communicate technical challenges through structured updates and direct messaging tools. The Agile environment facilitated continuous refactoring and incremental improvements, turning

challenges into opportunities for adaptation. These experiences demonstrated that the Agile Software Development Life Cycle (SDLC) thrives on flexibility and shared accountability among team members.

In Module 6, I took on the Scrum Master role during our group discussion, with a focus on improving communication practices and team transparency. I highlighted Daily Stand-ups as the most impactful Scrum event because they turn abstract plans into visible, shared progress. Tools like burndown charts and digital boards further enhanced team alignment by keeping everyone informed about blockers and sprint goals (Rietze et al., 2025). My role as Scrum Master reinforced the idea that strong communication frameworks are essential for Agile teams to remain self-organizing and adaptive.

Completing User Stories Through the Agile SDLC

The SNHU Travel project showcased how user stories naturally evolve through Agile's iterative Software Development Life Cycle (SDLC). Our Agile Team Charter established a clear vision: to create an efficient, mobile-friendly booking platform for SNHU students, faculty, and staff. Each sprint transformed high-level goals into valuable increments, from initial destination listings to customizable search functions.

Rolling-wave planning (Cobb, 2015) allowed us to progressively refine details instead of trying to predict everything upfront. This approach proved especially critical when the management of SNHU Travel introduced a significant shift in focus toward detox and wellness travel. Rather than rewriting plans or extending deadlines, the Product Owner reprioritized the backlog while the team maintained their rhythm. This adaptability mirrored the real-world

lessons seen in the Vision Quest Software case study, where the transition from Waterfall to Scrum replaced rigid forecasting with continuous delivery and enhanced teamwork.

Each increment, including the slideshow prototype developed in Module 5, was assessed during a simulated Sprint Review process. In this stage, we gathered feedback on functionality and usability, which informed our priorities for the next iteration. The Tester verified the acceptance criteria, and the Definition of Done ensured alignment across all roles. Agile's iterative structure transformed uncertainty into progress, demonstrating that value delivery and learning are inseparable elements of a healthy Scrum process.

Handling Interruptions and Change

Mid-project interruptions provided valuable opportunities to analyze how Agile teams respond to change. In the course simulation, the project pivoted to focus on wellness travel, illustrating how a Product Owner can reprioritize the backlog while maintaining sprint cadence. As the Scrum Master observing this scenario, I reflected on how Developers and Testers would adapt their work and update related artifacts. Rather than viewing change as a disruption, I learned that it is an essential part of Agile's rhythm of inspection and adaptation (Schwaber & Sutherland, 2020).

This concept mirrored lessons from the Valpak case study, where effective coordination across multiple Agile teams required breaking down silos and using visual task boards to sustain workflow. Similarly, applying these ideas through our coursework and use of digital Agile tools showed me that adaptability succeeds only when supported by open communication and transparency.

Communication and Collaboration Practices

Effective communication is the driving force behind every sprint. Daily stand-ups ensure that each team member shares updates and identifies any blockers, which creates real-time alignment. This practice not only improves situational awareness but also increases psychological safety and engagement, both of which are key predictors of high team performance (Rietze et al., 2025).

Information radiators, such as digital Scrum boards and sprint burndown charts, enhance visibility between meetings. These tools reduce the need for redundant status updates and empower the team to self-correct without waiting for managerial input (Lippert & Guzman, 2022). As a Developer and Tester, I experienced firsthand how accessible visual data minimized confusion and increased focus.

Written communication was equally important. For instance, as a Developer, I crafted concise emails requesting clarification from the Product Owner and the Tester. These messages summarized the context, listed specific questions, and set deadlines, demonstrating how asynchronous communication complements Agile's synchronous events. Learning about the Sprint Retrospective emphasized how structured reflection fosters continuous improvement in Agile teams. Through discussions and analysis of coursework, I gained insight into how identifying even small adjustments, such as clarifying the Definition of Done or enhancing team communication, can improve collaboration and efficiency in future sprints. Although our retrospectives were simulated rather than conducted with a live team, the experience demonstrated the importance of transparency and reflection in maintaining Agile progress.

Evaluation of Tools and Organizational Practices

Studying Agile project management tools such as Jira and Azure Boards has deepened my understanding of how teams can maintain transparency and alignment. These platforms serve as digital information radiators, displaying backlog items, sprint progress, and testing results in real time. They integrate key Scrum elements like burndown charts, sprint backlogs, and task traceability (Atlassian, 2024; Microsoft, 2025).

However, as highlighted in the Vision Quest Software case study, tools alone do not create agility, mindset does. A culture of openness, adaptability, and servant leadership ultimately determines success. Through coursework discussions, I learned that while digital platforms can enhance collaboration, they cannot replace the interpersonal trust that is built through consistent communication and mutual respect.

Lessons Learned and Recommendations

Rotating through every Scrum role has shown me that Agile success relies on shared ownership, flexibility, and continuous reflection. As a Scrum Master, I learned that effective facilitation and empathy are key drivers of team performance. In my role as a Product Owner, I discovered the importance of establishing clear priorities and feedback loops to define value. As both a Tester and Developer, I observed how vital communication is in bridging the gap between technical and business objectives. These interconnected roles provided a comprehensive view of the modern software development life cycle.

For future sprints, I recommend the following: maintain disciplined Daily Scrums, integrate automated testing for continuous feedback, and keep an updated Definition of Done. Retrospectives should serve as a catalyst for improvement rather than just a formality. Most

importantly, Agile should be seen not merely as a methodology but as a culture of learning, adaptation, and collaboration.

Through the SNHU Travel project, I learned that true agility is not about having no structure; rather, it is about having a structure that embraces change. Scrum's empirical framework helped us transform uncertainty into insight, enabling our team to deliver functioning software while continuously improving our collaboration.

References

Atlassian. (2024–2025). *Jira Software features and backlogs.* https://www.atlassian.com/software/jira

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

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

Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2019). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior, 51*(1), 915–929.

Lippert, S., & Guzman, G. (2022). Operationalizing the benefit of information radiators from an awareness perspective. *SN Computer Science, 3*(4).

Microsoft. (2025). *Azure Boards—What it is and Agile workflow guidance.* https://learn.microsoft.com/azure/devops/boards/

Rietze, S., et al. (2025). Relations between daily stand-up meetings, psychological safety, and team effectiveness. *European Journal of Work and Organizational Psychology.*

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The definitive guide to Scrum—the rules of the game.* Scrum.org.

Vision Quest Software Case Study. (2017). *A software success story.*