Project: Sprint Review and Retrospective

Daniel Dyall

Southern New Hampshire University

CS 250: Software Development Lifecycle

Instructor Maryann Krupa

October 19, 2025

**Sprint Review and Retrospective**

**Applying Roles**

Throughout the SNHU Travel project, I served in multiple Scrum Agile roles, each contributing uniquely to project success:

Scrum Master: I facilitated all Scrum events and emphasized timeboxing. For example, when our team encountered a significant integration blocker, I encouraged transparent discussion during Daily Scrums and used Retrospectives to collect process improvement suggestions. By shifting our daily meeting time (based on team feedback), attendance and participation immediately improved, which accelerated problem resolution.

Product Owner: In this role, ongoing dialogue with stakeholders was crucial. After conducting a user interview, I realized that our user stories were too ambiguous (“as a traveler, I want recommendations…”). I rewrote them to be more actionable (e.g., “As a traveler, I want the ability to filter trips by rating so that I can quickly compare my options”), which helped the development team work with precision and achieve successful Sprint Reviews.

Developer: I prioritized direct communication. When a requirement for the booking interface became unclear, I reached out to both the Product Owner and Tester for clarification on acceptance criteria. Their prompt feedback allowed me to fix bugs early, keeping the sprint on track and reinforcing the importance of raising blockers quickly.

Tester: Defining acceptance criteria directly from user stories enabled clear, measurable test cases. When the wireframe for our interface shifted to a SlideShow format mid-sprint, I contacted the Product Owner to clarify requirements immediately. As a result, new acceptance tests were rapidly aligned with the revised vision.

**Completing User Stories**

The Scrum Agile approach allowed us to deliver user stories consistently through continuous backlog refinement and stakeholder involvement. After observing recurring misunderstandings in Sprint Planning, the Product Owner began running backlog grooming sessions where each story was rewritten with input from both the development and QA teams. For example, the story “As a traveler, I want detailed pricing on trip packages” directly resulted in the addition of a pricing comparison feature, which the client immediately approved during the Sprint Review.

**Handling Interruptions**

A notable interruption occurred when the client requested, mid-sprint, that we add mobile browser support for the main vacation search feature. As a team, we discussed the change at the next stand-up, split the requirement into smaller stories, and prioritized accordingly in the backlog. This enabled us to accommodate the urgent new requirement within the Agile framework, demonstrating the value of our iterative, adaptable process without losing momentum or morale.

**Communication**

Effective and transparent communication was essential throughout the project:

Sample email (to Product Owner):

“Hi Christy, I’m working on test cases for the Top Destinations feature. Should those destinations be presented as a list or as a SlideShow? Do you want destinations to be removed permanently when rejected? A quick reply will help me make sure our acceptance tests match your expectations.”

Stand-up message (as Developer):

“If you find any issues during testing, please ping me right away fixing them now will prevent delays at the end of the sprint.”

These communications resolved ambiguities and encouraged real-time collaboration, making our teamwork more effective and reducing cycle time for both bug fixes and feature validation.

**Organizational Tools and Scrum Principles**

Jira boards and Kanban visualization made workflow transparent and helped all team members track story progress. Scrum events (Sprint Planning, Daily Scrums, Reviews, Retrospectives) enforced discipline while allowing ongoing improvements. For example, retention of a digital Kanban helped us quickly spot and resolve bottlenecks, while regularly scheduled Retrospectives led to refinements in daily check in timing and reporting.

**Evaluating Agile Process**

Pros:

The iterative approach enabled rapid adaptation to changing client needs and continuous delivery of value.

Regular feedback loops caught specification problems early.

Transparency and team morale improved with daily stand-ups and retrospectives.

Cons:

The team experienced a learning curve, especially in establishing consistent communication habits.

There was a risk of scope creep if user stories were not tightly managed.

Given the frequent client feedback and requirement changes on SNHU Travel, Scrum-Agile was unquestionably the best fit, enabling a responsive and collaborative development process that waterfall would have struggled to support.

The SNHU Travel project proved that Scrum Agile when practiced with role clarity, open communication, and adaptive planning delivers superior results in environments where client needs can change quickly. The experience not only reinforced textbook Agile principles but also highlighted the importance of direct, real-world team collaboration.

References

Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf

Harness. (2025). The Seven Phases of the Software Development Life Cycle. https://www.harness.io/blog/software-development-life-cycle-phases

Mendix. (2025). The 5 Stages of the Agile Software Development Lifecycle. https://www.mendix.com/blog/agile-software-development-lifecycle-stages/

Slideshare. (2024). Waterfall vs Agile Methodology Explained.pptx. https://www.slideshare.net/slideshow/waterfall-vs-agile-methodology-explainedpptx/265596850

Wezom. (2025). Agile vs Waterfall: Pros, Cons & Which One to Choose in 2025. https://wezom.com/blog/agile-vs-waterfall-which-software-development-approach-works-best-for-you