SNHU Travel Sprint Review & Retrospective  
Andriana Djurdjevic  
CS-250: Software Development Lifecycle  
June 15, 2025

Introduction

Throughout the SNHU Travel project, the Scrum team depended on daily stand-ups to keep everyone aligned. We’d briefly recap what we accomplished yesterday, outline what we hope to tackle today, and flag any blockers that might get in our way (Ungvarsky, 2023). As the Scrum Master, I focused on making Scrum principles clear and actionable by explaining the “Definition of Done” and helping untangle process questions on the fly.

The development team owned the “how” of sprint work and decided how best to tackle backlog items and churned out working code for the Product Owner and testers to review. In my role as a developer, I took extra steps to smooth hand-offs: I drafted an email to the Product Owner and testers listing points of confusion around changing requirements, then asked to discuss them in our next meeting. That conversation helped clarify edge cases and prevented us from heading down the wrong path.

On the testing side, the testers were running checks at the end of a sprint and participated early to help define acceptance criteria and wrote test cases that spelled out exactly what “Done” meant for each story (Ungvarsky, 2023). Once a feature was received, they’d work through each step, reporting back on bugs or quality gaps and working with developers to get everything ship-shape.

One of the most beneficial things about our Scrum-Agile approach was the speed at which we could change gears. Since our Product Owner stayed in constant contact with the customer, we were able to bring in new feedback and add it to upcoming sprints without derailing our progress. Even when user-story details changed mid-course, we weren't forced to restart our backlog—we simply revised the stories, refocused our tasks, and continued on. Agile's embracement of change provided interruptions as chances to improve the product rather than emergencies to work through.

Communication was structured to include all team members in test-case development, user-story refinement, and task prioritization. This inclusive approach ensured clear visibility into project status and next-cycle priorities. Ultimately, the combination of regular check-ins, defined role responsibilities, and openness to feedback transformed what might otherwise have been a disjointed, one-time exercise into an efficient, iterative learning process.

Handling Interruptions

Within a week of starting the project, it was decided that SNHU Travel would shift entirely to detox/wellness getaways. Instead of stretching our sprint, we:

1. Conducted an emergency backlog refinement immediately after stand-up to re-estimate and re-prioritize stories.
2. Reprioritized away non-wellness items, concentrating on getting wellness filters and enhanced search done within the next two weeks.
3. Scope tuned based on developer capacity: We recalculated activity, making sure search refactoring and UI modifications are within our available velocity.

Being flexible with Agile, we maintained our sprint end date and delivered a minimum viable wellness booking flow.

Communication Samples

By asking precise, targeted questions to guide the process, I was able to produce the most thorough results. Here’s an example email that kept the project on track:

To: PO, QA  
Subject: Details Needed for Sprint 5 Features

Hi PO and QA,

For Sprint 5, I need:

1. US-210 (Search Filter):
   * Target response time (e.g., under 500 ms with 1,000 users)
   * Expected message if the API returns an error
2. US-212 (Error UI):
   * Exact steps and data to test the error popup
   * Any mockups or screenshots you have

Could you send these by Wednesday, June 11, so I can finish development on time? Let me know if you need more info.

Thanks,  
Andriana

Organizational Tools & Scrum Events

We kept everyone on the SNHU Travel team aligned with a Jira board and daily burndown chart. Our daily stand-up reviewed yesterday’s wins, today’s goals, and any blockers. After each sprint, the review let us demo the booking feature and new wellness filters to get early feedback. Finally, our retrospective distilled improvements like tighter refinement sessions and a clearer “Definition of Done” so we could keep getting better without losing momentum.

Conclusion & Evaluating the Agile Process

By adapting rapidly to changing requirements, like out mid-project switch to wellness filters, we were able to utilize Agile practices and continue our project without derailing ongoing work. Continuous sprint reviews kept stakeholders involved, so we caught issues early. Delivering in small increments reduced the risk of late-stage surprises. A Waterfall model would have locked in scope and deadlines, making our wellness pivot costly and time-consuming. Scrum’s iterative framework let us reprioritize on the fly and still deliver a working wellness booking tool on schedule.

References

Codington-Lacerte, C. (2024). Agile software development. *Salem Press Encyclopedia*.   
Ungvarsky, J. (2023). Scrum. *Salem Press Encyclopedia*.