**Sprint Review & Retrospective – SNHU Travel (Scrum-Agile Pilot)**

As the Scrum Master for the SNHU Travel pilot at ChadaTech, I guided a Scrum-Agile process throughout a complete sprint cycle to create an increment of a travel discovery application. This retrospective outlines the application of roles, how user stories progressed to completion, the management of interruptions, our methods of communication, and which organizational tools and Scrum events supported our success. This retrospective ends with a conclusion of the value of Scrum-Agile for this project.

Applying Roles

Product Owner (PO): The PO worked with the team closely defining and clarifying user stories and acceptance criteria. For example, when refining the “Top 5 destinations” feature into a slideshow, the PO clarified navigation behavior, edge cases (e.g., the behavior of empty results).

Developers: Developers engaged in iterative delivery and continuous refactoring. When change occurred mid-sprint, they alerted the group and provided us with updated priorities and sought out explicit acceptance criteria to ensure their work was aligned following the latest direction. They also coordinated with testing in relation to impacts on coverage.

Tester/QA: The tester transformed user stories into specific tests, such as ensuring that a price filter doesn't show trips above a threshold, and validated UI behavior and empty-state messages. Working closely with the PO ensured no critical edge cases were left covered early.

Scrum Master: I cleared impediments, timeboxed ceremonies, and promoted transparency. I enforced the Sprint Planning goals and kept Daily Scrums focused on blockers - this allowed for flow and protected the team from scope noise.

Completing User Stories

The Scrum-Agile methodology improved the speed of user story completion by encouraging smaller, testable slices with unambiguous acceptance criteria and a definition of done. For instance, the transformation of the "filter destinations by price" story from the backlog to "Done" emerged from the clarification of (1) results displayed update correctly, (2) an empty-state message displays when there are no matches and (3) tests included both mobile and desktop expectations. The Sprint Review cycle provided immediate stakeholder feedback, so we could queue for small, follow-up issues for the next sprint, instead of condensing larger issues.

Handling Interruptions

We had prioritized features that changed mid-sprint. Scrum's adaptive planning and short iterations allowed us to change scope without losing velocity. The PO reordered the sprint backlog list and developers asked for updated acceptance criteria. We logged the changes in our tracking tool, talked about the changes at our Daily Scrums, and limited churn on the sprint by completing any items that were just about done before pulling any new work because we did not want to loose concurrency. This way we limited the context-switching, while still being adaptive.

Communication

Effective communication anchored our progress. Below are representative samples and why they worked:

• Developer → PO & Tester: A brief email requested which of the newly prioritized user stories should be included in the current sprint, a request for updated acceptance criteria, and any questions about test impacts. The brief, simple email included a clear ask, provided a timeframe, and offered a quick sync opportunity during the next stand-up. The email encouraged a timely collaborative response, resulting in a commitment from the PO and Tester to unblock development.

• Tester → PO: Quick notes clarified if the price filter updated in real time or only after pressing Apply, what message to display for no results, and whether mobile would lose any of the features of the desktop. The focused questions will remove ambiguity and help ensure that the tests reflect a real user experience.

• The Scrum Master facilitation included timeboxed Daily Scrums (≤15 minutes) with a focus on the work done yesterday, the work planned today, and blockers. Each action item was recorded on the board and reviewed at the following stand-up. This structure created a shared rhythm of ownership and visibility.

Organizational Tools & Scrum Events

Backlog & Board (e.g., Jira): Supported transparent prioritization, WIP limits, and cycle-time tracking.

Sprint Planning: Aligned scope with capacity and clarified acceptance criteria before commitment.

Daily Scrum: Surfaced blockers quickly; facilitated micro-adjustments to keep the sprint on track.

Sprint Review: Demonstrated working software; gathered stakeholder feedback to refine the backlog.

Sprint Retrospective: Drove continuous improvement, e.g., we decided to template acceptance criteria and empty-state behaviors, and to add a definition-of-ready checklist for stories entering the sprint.

Evaluating the Scrum-Agile Process

Pros:

• Responsiveness to change through short iterations and adaptive planning.  
• Faster feedback loops via reviews and frequent testing.  
• Stronger cross-functional collaboration and shared visibility.  
• Higher quality from early test design and definition-of-done discipline.

Cons:

• Requires disciplined backlog management and engaged stakeholders.  
• Mid-sprint changes can introduce context switching if not carefully controlled.  
• Ceremony overhead can grow if timeboxes and outcomes aren’t enforced.  
• Estimation variability can challenge forecasting in early sprints.

Was Scrum-Agile the best fit? For SNHU Travel, yes. What SNHU Travel needed was quick learning from the users, constant adaptations to the UI/UX, and incremental value delivery. All of which Scrum was very good at. A waterfall model would have led to delayed feedback cycles and more costly changes to accommodate late-breaking UX design discoveries. Scrum has a strong structure (very easily defined roles) with flexibility; scrum allowed for working increments and a method for continuous improvements.

Conclusion

Scrum-Agile practices, clear roles, small testable stories, disciplined ceremonies, and continuous collaboration, helped our team deliver value despite change. The pilot demonstrated improved transparency, quality, and adaptability and provides a strong case for ChadaTech to scale Scrum across teams.