**CS 250 Final Project — Sprint Review & Retrospective**  
Author: Adil Patel  
Date: August 17, 2025

**Overview**

This Sprint Review and Retrospective summarizes the Scrum-Agile work completed for the SNHU Travel application pilot at ChadaTech. Throughout the course, I rotated through different Scrum roles (Scrum Master, Product Owner, Developer, and Tester) and delivered artifacts that moved the product toward a working travel booking tool. This reflection explains how each role contributed to delivery, how Agile practices helped user stories reach “Done,” how interruptions were handled, what communication supported collaboration, which tools made the team effective, and whether Scrum was the best approach for this project.

**Applying Roles**

* **Scrum Master:** I facilitated sprint planning and daily scrums by keeping discussions focused on the three core questions: What did I do yesterday? What will I do today? What impediments are in my way? For example, when a teammate exported the JAR file in Java 21 instead of the required Java 8, I coordinated a short spike with the developer to fix the compatibility issue quickly.
* **Product Owner:** In this role, I managed the product backlog. When stakeholders pivoted to emphasize detox/wellness travel after the SNHU Travel animation, I rewrote stories and re-prioritized the backlog to reflect the new theme, ensuring we still delivered the highest value.
* **Developer:** I implemented a “Top 10 Destinations List” feature. Initially the list was static, but feedback required the list icons to be clickable and open the photos. I made the UI adjustment and retested the feature.
* **Tester:** I translated user stories into structured test cases with preconditions, steps, and expected results. For example, one case ensured clickable icons opened the correct image and link. A failed test revealed the missing click action, which was fixed in the next sprint.

Each role directly supported progress: Scrum Master kept the process on track, the Product Owner ensured value alignment, Developers built working increments, and Testers confirmed quality.

**Completing User Stories**

Agile practices helped stories reach completion because each story had clear **acceptance criteria** and a shared **Definition of Done (DoD)**. For example, one story stated: As a traveler, I want to filter trips by wellness themes so that I can discover detox retreats. Acceptance criteria included visible filters, selectable tags (detox, spa, yoga), and correct update of results.  
By breaking work into small slices—UI filter chips, filter logic, and result display—the story could be tested and marked “Done” within a sprint. Short sprints, backlog refinement, and constant review encouraged consistent progress.

**Handling Interruptions**

The biggest interruption came when management shifted focus mid-project to wellness travel. Instead of restarting, Agile practices helped us adapt:

1. We held an extra backlog refinement to turn the new direction into user stories.
2. Lower-value items were de-scoped.
3. Stories in progress were re-estimated and updated.  
   This prevented wasted work and kept the team aligned with stakeholder needs. Waterfall would have delayed this pivot until much later, making rework costly.

**Communication**

Two practices stood out:

* **Information radiators:** A visible sprint board and burndown chart made work and blockers transparent to everyone.
* **Concise written updates:** For example: “Looks great to me. I uploaded the post. Waiting on one more tester confirmation before we submit.”  
  This message was effective because it clearly stated the status, the next step, and responsibility. Such clarity avoided confusion and promoted collaboration.

**Organizational Tools**

We modeled our work in a digital Scrum board (like Jira):

* **Sprint Planning:** Selected backlog items based on capacity and set a sprint goal.
* **Daily Scrum:** Used the board to highlight blockers (e.g., export version mismatch).
* **Backlog Refinement:** Adjusted stories after the wellness pivot.
* **Sprint Review:** Demoed working increments (destination icons, filters).
* **Retrospective:** Identified improvements such as including environment requirements in the DoD.

The tools and events supported transparency, focus, and adaptability.

**Evaluating the Agile Process**

**Pros:**

* Quick adaptation to change.
* Early risk detection (e.g., Java version issue).
* Shared understanding through acceptance criteria.
* Higher morale from seeing progress often.

**Cons:**

* Mid-sprint re-estimations took extra time.
* Coordination overhead from frequent meetings.
* Learning curve with tools like Jira.

**Conclusion:** Scrum was the right choice for SNHU Travel. It delivered usable increments early, adapted to shifting business goals, and reduced risk. Waterfall would not have offered the same flexibility or speed.

**Lessons Learned**

1. Always include compatibility/environment requirements in the Definition of Done.
2. Break stories into thin, user-visible slices.
3. Keep work visible with updated boards.
4. Maintain feedback loops through demos and reviews.