CS 250 Final Project: Sprint Review and Retrospective
Student Name: Christopher Medeiros
Date: May 10, 2025

Applying Roles in the Scrum-Agile Team

Throughout the development of the SNHU Travel project, I assumed various Scrum-Agile roles including Product Owner, Developer, Tester, and Scrum Master. Each role contributed uniquely to the project's success. As the Product Owner, I clarified user stories and prioritized backlog items to ensure the product met business goals. For example, I specified that the app should focus on "top 5 destinations," then later adapted that focus to "detox and wellness" packages per management's change in vision.

As the Developer, I implemented a slide show application using Java and Swing. I integrated user story requirements like displaying images, navigation through Next/Previous buttons, and updating slide content. My work was informed by collaboration with the Tester and Product Owner.

In my Tester role, I created detailed initial and revised test cases using acceptance criteria derived from user stories. These cases guided development and helped validate that changes matched expectations after each sprint.

As the Scrum Master, I facilitated smooth team communication and made sure sprint goals were achievable. I also orchestrated the Sprint Review and Retrospective, coordinating feedback loops and leading retrospection on our process effectiveness.

Completing User Stories Through Agile

The Scrum-Agile methodology allowed us to break down the SNHU Travel application into manageable user stories. Each story had defined acceptance criteria and was completed incrementally. For instance, the user story "As a traveler, I want to view the top 5 destinations" was completed in Sprint 1 by creating a slideshow that displayed travel locations with descriptions.

When management introduced a shift to wellness destinations mid-sprint, Agile allowed us to quickly reassess and revise stories. New tasks like changing slide text and loading updated images were added without disrupting the overall project momentum. Agile's iterative nature enabled constant feedback and story refinement through backlog grooming and sprint planning.

Handling Interruptions With Flexibility

A pivotal interruption occurred when the Product Owner communicated a company-wide pivot to focus on wellness travel. Under a Waterfall model, this would have required a complete overhaul of the requirements document and halted progress. In contrast, the Scrum-Agile model allowed us to adapt immediately by re-prioritizing the product backlog.

Our stand-ups and sprint review gave us opportunities to communicate the change clearly. Instead of scrapping existing code, we swapped in new image assets and text while retaining functionality. This minimized wasted work and preserved sprint velocity.

Effective Team Communication

Agile's success relies on open communication. Our use of emails, retrospectives, and stand-ups allowed real-time collaboration. For instance, I emailed the Product Owner and Tester during the wellness change to confirm final destination names and test case updates. This helped ensure accuracy and reduced confusion.

Another effective communication tool was using shared test case documents. By organizing them in a template with clear pass/fail conditions, both Developers and Testers could verify software behavior without needing multiple clarifications. This created accountability and improved team trust.

Organizational Tools and Scrum Events

The Scrum events—Sprint Planning, Daily Stand-Ups, Sprint Review, and Retrospective—kept our team aligned and efficient. Sprint Planning enabled us to break stories into tasks. Daily stand-ups identified blockers early. The Sprint Review showcased our product increments and gathered feedback, while the Retrospective allowed us to reflect on what worked and what to improve.

Tools like shared templates (test plans, user stories, slide deck frameworks) served as lightweight versions of JIRA or Azure DevOps. While we didn't use those tools in this course, the templates simulated how teams track progress, manage tickets, and streamline collaboration.

Evaluating the Scrum-Agile Approach

Pros:

- Allowed rapid adjustment when project priorities shifted
- Broke down development into achievable, testable chunks

- Encouraged regular team communication and feedback
- Increased visibility into progress and blockers

Cons:

- Required more frequent team communication and coordination
- Test case updates and backlog grooming added overhead

Despite some overhead, Agile was the best fit for the SNHU Travel project. The ability to adapt quickly, test frequently, and demonstrate working code in sprints aligned perfectly with our evolving requirements. The iterative nature and team collaboration ensured we remained on track despite mid-project changes.

Conclusion

The Scrum-Agile methodology provided the flexibility, structure, and collaborative rhythm necessary to build the SNHU Travel app efficiently. Each role I performed reinforced how valuable Agile practices are when handling evolving priorities. With clearly defined roles, ceremonies, and lightweight tools, our team delivered a working prototype and adapted smoothly to change.

The success of this pilot supports ChadaTech’s decision to expand Scrum-Agile across its development teams.