**CS 250: Software Development Lifecycle**  
**George Washima Usa**  
**April 17, 2025**

**Sprint Review and Retrospective**

### **Applying Roles**

Throughout the SNHU Travel project, I rotated through various roles on the Scrum-Agile team, which allowed me to gain a deep understanding of how each role uniquely contributes to project success. Each role provided valuable insight into the Agile development process and helped ensure that user needs were met with consistent quality.

As the **Product Owner**, I prioritized the backlog, clarified user stories, and ensured that each sprint delivered value to the client. For example, I refined vague user stories like “Build booking form” into more actionable items such as “As a customer, I want to enter traveler information on a form, so I can complete a booking.” This clarity allowed the development team to understand the scope and deliver precisely what was expected.

In the **Developer** role, I implemented key features of the SNHU Travel application, such as the destination search function and customer profile page. I adhered to user stories and collaborated with teammates to meet deadlines and acceptance criteria.

As the **Tester**, I created and executed test cases to validate whether features met their acceptance criteria. This helped catch issues early and maintain high quality. During one sprint, I found that the trip review form allowed submission without input. I documented the defect and collaborated with the developer to push a fix before the sprint ended.

Finally, in the **Scrum Master** role, I facilitated daily stand-ups, managed sprint events, and ensured team productivity and focus. I kept track of the sprint burn-down and removed blockers such as unclear user stories or communication gaps. This holistic experience across roles made me a better servant-leader during the final retrospective.

### **Completing User Stories**

The Scrum-Agile framework significantly streamlined the process of completing user stories. Instead of tackling the entire application at once, we focused on bite-sized, high-priority stories that were achievable within short sprints.

For instance, the user story: “As a traveler, I want to view available destinations, so I can choose where to go,” was approached during a sprint with defined acceptance criteria: the page must display at least five destinations, include images, and be responsive on mobile. Because the story was clear, the development team could break it down into tasks—such as backend integration, UI layout, and mobile testing—and complete it efficiently.

During sprint planning, the Product Owner prioritized stories based on value, and the development team committed to only what could realistically be completed within the sprint. The iterative nature of Scrum allowed us to inspect progress frequently and adapt as needed.

### **Handling Interruptions**

Midway through development, a request came in from the SNHU Travel client to add a “Traveler Reviews” section to destination pages. In a traditional Waterfall environment, this would have required extensive change requests and possibly delayed the entire project. But within the Agile Scrum framework, we simply treated it as a new backlog item and scheduled it for a future sprint.

This flexibility allowed the team to stay focused on the current sprint while accommodating the change in direction. It was added as a user story: “As a traveler, I want to read reviews of destinations, so I can make informed decisions.” By maintaining a prioritized and refined backlog, we remained adaptable without derailing our existing sprint goals.

### **Communication**

Effective communication was critical to our success. As Scrum Master, I encouraged open collaboration through various communication tools. We used simulated messaging tools like Slack and Jira comments for quick clarifications and updates.

For example, when a developer was unclear about whether the booking form required validation, I quickly facilitated a discussion between the Product Owner and Developer to clarify expectations. These interactions prevented delays and ensured that work met client expectations.

Additionally, I shared daily stand-up summaries that included progress updates, blockers, and upcoming tasks. These summaries kept everyone informed, especially when team members couldn’t attend a virtual meeting. By fostering a transparent communication environment, team cohesion and productivity improved significantly.

### **Organizational Tools and Scrum Events**

Several tools and Agile events played a major role in organizing and driving our progress. We used Trello to manage our backlog and track sprint tasks visually. The drag-and-drop interface helped clearly communicate task ownership and status.

Scrum events were foundational. **Sprint Planning** helped the team commit to achievable work; **Daily Stand-ups** provided real-time insights and issue resolution; **Sprint Reviews** allowed us to demo our product increment and gather feedback; and **Retrospectives** gave us a space to reflect and improve. For example, during a retrospective, we realized that tasks were poorly estimated, and adjusted our story pointing approach using team consensus (Planning Poker) in the following sprint.

These tools and rituals allowed our team to continuously improve, both technically and in how we collaborated.

### **Evaluating the Agile Process**

The Scrum-Agile approach proved highly effective for the SNHU Travel project.

**Pros included:**

* Rapid feedback and iterative delivery
* Easy adaptation to change
* Strong team collaboration and ownership
* Continuous improvement through retrospectives

**Cons included:**

* Learning curve for roles and events
* Team commitment and timeboxing pressure
* Risk of incomplete stories if poorly estimated

Despite some growing pains, Agile was clearly the best choice for this type of project. The evolving nature of client requirements, such as adding new features mid-sprint, fit perfectly within the flexible Scrum structure. In contrast, a Waterfall model would have delayed or denied such late changes and impacted customer satisfaction.

The structured feedback loops, visibility, and focus on incremental value made Agile not just a methodology—but a mindset that empowered us to build better software.