**Sprint Review and Retrospective**  
 **Chada Tech | SNHU Travel Project**  
 **Lamberto Nunez**  
 **Southern New Hampshire University | CS 250**

### **Applying Roles**

In the SNHU Travel project, I took on the full array of roles typically found in a Scrum-Agile team. These included the Product Owner, Scrum Master, and Developer roles, each contributing to the simulated success of the project across different stages of the Software Development Life Cycle (SDLC).

As **Product Owner**, I was responsible for defining the product backlog and prioritizing user stories based on business value. For example, the feature allowing users to search for travel destinations by budget and location was prioritized early due to its relevance to SNHU Travel’s customer acquisition strategy. This aligns with Agile best practices, where the Product Owner ensures maximum value delivery by maintaining a well-organized backlog (Schwaber & Sutherland, 2020).

As a **Developer**, I executed the sprint backlog by implementing front-end features using a basic UI layout and integrated user input validation. I used time-boxed sprints and iterative development cycles to deliver working increments by the end of each sprint.

Lastly, in the role of **Scrum Master**, I facilitated all Scrum events: Sprint Planning, Daily Standups, Sprint Reviews, and Retrospectives. I used digital tools like Trello and Google Docs to manage workflow and remove impediments. Although I operated solo, adhering to these roles helped replicate the collaborative structure of an Agile team.

### **Completing User Stories**

User stories were completed by dividing the work into manageable tasks over two-week sprints. For example, the story **“As a traveler, I want to search for destinations by interest, so I can plan a personalized trip”** was broken into smaller deliverables: user interface for search input, filtering logic, and results display.

This approach reflects Scrum's ability to translate customer requirements into working software rapidly (Sommerville, 2016). By focusing on delivering potentially shippable increments, I was able to complete stories while receiving feedback in simulated review sessions. Agile’s focus on customer collaboration and iterative development made user stories more dynamic and adaptable to project needs.

### **Handling Interruptions**

During Sprint 2, a new requirement was introduced—adding a **real-time alert system for travel restrictions**. In a Waterfall project, this would have required going back to the planning stage. However, in Scrum, the feature was added to the product backlog and addressed during the next sprint planning session.

Scrum supports this kind of mid-project change by promoting adaptability and value-driven delivery (Highsmith, 2009). The sprint backlog was restructured, and the less critical feature (social sharing integration) was postponed. This ensured the sprint goal was still achievable while addressing the most current stakeholder needs. Agile’s built-in flexibility made it possible to pivot without derailing progress.

### **Communication**

Although this was a solo project, communication artifacts were still maintained to simulate team collaboration. These included:

* **Daily Standup Notes**: “Blocked by API delay, will use mock data to continue progress.”
* **Sprint Planning Documents**: Defined goals, tasks, and estimated hours.
* **Sprint Reviews**: Documented what was completed, what was pushed to backlog, and user feedback (simulated).

These artifacts mirror real Agile communication, which relies on transparency, visibility, and short feedback loops (Rigby, Sutherland, & Takeuchi, 2016). Even when working individually, this level of organization promoted clarity and improved self-accountability. In a team setting, these same practices would improve coordination and foster collaboration.

### **Organizational Tools**

The project utilized several Scrum-aligned organizational tools. A Trello board acted as the product and sprint backlog, tracking story status through columns like “To Do,” “In Progress,” and “Done.” A digital burndown chart created in Excel visualized remaining story points over time.

The Scrum events themselves served as powerful organizational anchors:

* **Sprint Planning** helped define clear, achievable goals.
* **Daily Standups** maintained momentum and identified obstacles.
* **Sprint Reviews** encouraged feedback and reflection.
* **Sprint Retrospectives** allowed adjustment of processes for the next sprint.

These tools and ceremonies ensured the project remained structured, transparent, and aligned with Agile principles (Schwaber & Sutherland, 2020).

### **Evaluating Agile Process**

The Scrum-Agile approach provided numerous advantages for the SNHU Travel project.

#### **Pros**

* **Flexibility**: Able to adjust priorities and introduce new features.
* **Early Feedback**: Delivered usable increments and iterated based on review.
* **Focus on Value**: Prioritized high-impact features over lower-value ones.

#### **Cons**

* **Role Overload**: One person managing all roles may dilute their effectiveness.
* **Scope Management**: Mid-sprint changes, even minor ones, can risk derailing momentum.

Despite the challenges of role compression, Scrum proved ideal for this project. It allowed fast feedback, continuous improvement, and close alignment with stakeholder goals. Compared to a Waterfall model—where all requirements must be fixed upfront—Scrum encouraged adaptability and delivered working software in each sprint, making it the superior choice for this dynamic project environment (Pressman & Maxim, 2015).

### **References**

Highsmith, J. (2009). *Agile project management: Creating innovative products* (2nd ed.). Addison-Wesley.  
 Pressman, R. S., & Maxim, B. R. (2015). *Software engineering: A practitioner’s approach* (8th ed.). McGraw-Hill Education.  
 Rigby, D. K., Sutherland, J., & Takeuchi, H. (2016). Embracing Agile. *Harvard Business Review*, 94(5), 40–50.  
 Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. Retrieved from [https://scrumguides.org](https://scrumguides.org/) Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.