### **Sprint Review and Retrospective**

**ChadaTech – SNHU Travel Project** *Scrum Master Perspective*

#### **Applying Roles**

Throughout this course, I studied and applied different Scrum roles—Product Owner, Developer, and Scrum Master—individually. While I didn’t work in a real team, I gained valuable insight into how these roles function in an Agile environment. As the Product Owner, I learned how to manage the Product Backlog, prioritize user stories, and define what value means for the customer. As the Developer, I focused on turning user stories into functioning code. And as the Scrum Master, I learned how to support a team’s productivity by encouraging communication, removing blockers, and guiding the process. Understanding each role helped me see how they work together to deliver continuous value.

#### **Completing User Stories**

Even though I worked alone, adopting an Agile mindset helped me break down complex features into manageable parts. For instance, when building the feature that displays the top five destinations, I treated each requirement (image, description, link, layout) as a separate story. This structure helped me focus on one task at a time, reflecting how user stories guide team priorities in Agile development. The iterative process of planning, building, and adjusting made it easier to handle changes without redoing everything from scratch.

#### **Handling Interruptions**

The SNHU Travel project underwent a pivot from general destinations to wellness-focused travel. This was a simulated change, but it gave me the opportunity to experience how Agile supports adapting to new requirements. Instead of being set back by the shift, I updated the backlog and reprioritized tasks accordingly—something that would have been much harder in a traditional Waterfall process. Agile's flexibility made the transition smooth and efficient, even in a solo learning environment.

#### **Communication**

Since there was no team communication, I simulated collaboration by documenting progress and using Agile-style tools to track tasks. I maintained a backlog, created sprint goals, and wrote status notes to reflect what I would share in a daily standup. This practice helped me understand how structured, transparent communication would benefit a real Agile team by improving alignment and responsiveness.

Example (simulated standup entry):

“Working on implementing destination cards. Ran into formatting issues with image sizing—will troubleshoot and adjust CSS after reviewing layout constraints.”

While no one else read these updates, they helped me stay focused and develop habits that are useful in collaborative work.

#### **Organizational Tools**

Agile organizational tools like digital Kanban boards and sprint backlogs were essential, even when working alone. I used checklists and task boards to mirror real Agile practices. Scrum events like sprint planning and retrospectives helped me assess what was going well and what needed to change. The structure of these events made it easier to stay organized, plan iteratively, and reflect regularly—key Agile principles.

#### **Evaluating Agile Process**

**Pros:**

* Encouraged flexibility and adaptation when project direction changed.
* Helped organize tasks in a manageable way through user stories.
* Promoted reflective thinking through sprint reviews and retrospectives.

**Cons:**

* Lacked real-time collaboration and feedback due to working solo.
* Some Scrum rituals (e.g., daily standups) felt less impactful without a team.

Despite being a solo experience, the Scrum-Agile approach worked well for this type of learning project. It gave me a hands-on understanding of how Agile development is structured and how roles, tools, and events come together. It also showed me why Agile is often better suited to evolving projects than the Waterfall method.