**Sprint Review and Retrospective**

David Mingace Jr.  
Southern New Hampshire University  
CS 250 – Software Development Lifecycle  
Instructor: Joseph Rangitsch  
April 2025

**Applying Roles**  
Throughout the SNHU Travel project, I had the opportunity to take on multiple Scrum roles, and each one showed how important that position is in a team using Agile methodology. As a developer, I focused on sprint planning, writing and reviewing code, and making sure tasks aligned with the goals for each sprint. That role helped me understand how collaboration and smaller, achievable tasks helped us actually finish features instead of working endlessly on big chunks of code. As a tester, I made sure that feedback loops were constant and that quality checks happened early and often. I worked closely with developers to build test cases and address bugs as they came up, which helped us avoid surprises later on.

When I stepped into the Product Owner role, I realized how much planning and clarity matters. Writing detailed user stories, organizing the product backlog, and keeping the team aligned with business goals were big parts of that role. I also made sure that the acceptance criteria was clear and that developers and testers could actually use the stories to guide their work. Finally, as the Scrum Master, I helped keep everyone on track by facilitating daily stand-ups and resolving any blockers. Each role helped move the project forward in its own way, and having that perspective made me appreciate how Scrum encourages everyone to contribute equally.

**Completing User Stories**  
The Scrum-Agile approach played a huge role in making sure user stories got completed efficiently. Instead of dumping a bunch of requirements on the dev team all at once like in waterfall, we focused on just a few user stories each sprint. This helped keep the workload realistic and made it easier to focus on quality. For example, when I was the Product Owner, I made sure user stories followed the INVEST model: Independent, Negotiable, Valuable, Estimable, Small, and Testable. That helped the developers and testers know exactly what the goal was.

During one sprint, we worked on a story that involved booking a travel package. The story clearly stated what the user needed to do, what counted as success, and any important exceptions. Having that structure made the work easier to estimate and complete, and we were able to test it quickly. The user story format in Agile helped make things more predictable and team friendly.

**Handling Interruptions**  
One of the things that stood out to me in this project was how Scrum handled changes and interruptions without everything falling apart. In waterfall, changing direction mid-project usually causes delays or even restarts. With Agile, we were able to pivot quickly without losing momentum. For example, halfway through the project, we got feedback that customers wanted more customization options when booking travel.

Instead of panicking or pushing everything to a later release, we just added new user stories to the backlog and discussed them during sprint planning. This allowed the team to adapt and prioritize the most valuable features without breaking anything. It showed how Scrum supports flexibility, which is a huge benefit in real-world projects where things rarely go exactly as planned.

**Communication**  
Communication was a major part of making this project work. As a team, we used daily stand-ups to keep everyone updated, and it helped make sure we weren’t duplicating work or running into problems silently. Here’s a sample message I might send during a stand-up:

"Yesterday I finished the travel booking story and pushed the code for review. Today I’ll be starting on the payment validation story. No blockers right now, but I may need help testing edge cases later this afternoon."

This kind of clear, short update helped teammates know what was going on, and it encouraged quick collaboration. It was also helpful when I was the Scrum Master because it gave me visibility into who might need support without them having to ask directly.

**Organizational Tools**  
We used several organizational tools to stay on track, and most of them aligned well with Scrum principles. We used a task board to manage our backlog and sprint tasks. Each story or task was labeled with its status: To Do, In Progress, Testing, or Done. This visual organization helped the team see what needed attention and made sprint planning and retrospectives a lot more efficient.

Sprint planning meetings gave us a chance to review the top items in the backlog and decide what was achievable in the upcoming sprint. During sprint reviews and retrospectives, we discussed what went well, what needed improvement, and what to change moving forward. These events helped us fine-tune our process and build stronger teamwork. The structure kept us accountable and made sure we kept improving each sprint.

**Evaluating Agile Process**  
Overall, the Scrum-Agile approach worked well for the SNHU Travel project. One of the biggest pros was flexibility. We could shift priorities without starting over. It also created a more collaborative and transparent environment. Everyone knew what was going on, and work was divided into smaller, manageable chunks. We got feedback faster and improved faster because of it.

One con is that it does take time to get used to the rhythm of Agile, especially if people are coming from a more traditional project management background. There can also be some confusion when responsibilities overlap or when the backlog isn't clear. But even with those challenges, I think Agile was the right choice.

Waterfall would have made it harder to adapt to changes or customer feedback. Agile allowed us to learn and adjust as we went, which felt more realistic for a client-facing product like this. If ChadaTech is looking for more flexibility and faster delivery cycles, then I think moving to Scrum-Agile across teams is a smart move.