Sydney A. Brown

Southern New Hampshire University

CS - 250 - Software Development Lifecycle

Professor Denise Washington

August 17, 2025

Applying Roles

Throughout the SNHU Travel project simulation, I rotated through all three major Scrum roles—Scrum Master, Product Owner, and Development Team member—just as outlined in the Chada Tech scenario. Even though I didn't work with a real team, each role gave me insight into how Agile teams collaborate.

As Product Owner, I used feedback from a simulated focus group to create user stories based on real user needs. Instead of just guessing features, I learned to phrase stories in the format "As a [user], I want to [do something], so that I can [get a benefit]." This helped clarify priorities and focus the project on delivering actual value, not just features for the sake of development.

When acting as a Developer, I created test cases from those user stories. I had to ask clarifying questions, like whether the "Top 10 Destinations" should be shown in a slideshow or all at once. This back-and-forth with the Product Owner role showed me how communication impacts development and quality assurance. It also taught me how to handle ambiguity, which is common in Agile projects.

As a Scrum Master, I focused on communication practices like stand-ups, visible task tracking, and shared documentation. I emphasized collaboration and made sure we stayed aligned on goals and priorities. I also explored tools like JIRA to visualize progress, reduce blockers, and keep everything transparent and organized.

Even though these roles were done separately, working through each of them helped me understand how Agile roles work together and why each one matters to a project's success.

Completing User Stories

Working with user stories helped me understand how Agile teams prioritize work and stay user-focused. Writing them out in the format of "As a user, I want to…" forced me to think about what users actually care about instead of just listing out technical tasks. These stories gave the team direction and helped tie every feature back to a user goal.

Later in the project, the user stories became essential when it came to testing. I used them to guide the development of test cases and to make sure we were meeting the original expectations. The stories also gave me a way to validate whether a feature was done or still needed more work. Without them, it would have been easy to lose sight of the bigger picture and just complete tasks without thinking about the user experience.

Handling Interruptions

Agile's flexibility really showed its value when changes came up. One example was when expectations about a feature weren't clear at first, like the slideshow display for the Top 10 Destinations. Rather than panic or start over, the Agile mindset encouraged me to reach out, ask questions, and adapt based on the new info.

Since Agile breaks work into short iterations, there was room to adjust without derailing everything. That iterative nature helped prevent wasted effort, and made it feel less overwhelming when changes came in. It was easier to pivot because not everything was locked in from the start.

Communication

Communication was one of the most important parts of this simulation. I learned how simple check-ins, like daily standups or quick emails, could help catch issues before they became big problems. Having shared documents and tools also made it easier to stay aligned, especially when people were working on different parts of the project.

When I played the Scrum Master role, I focused on creating space for open communication, encouraging the team to speak up, and checking in when something seemed unclear. Transparency made it easier to stay on track and avoid duplicated work. It also made the work feel more collaborative, even though we weren't on a live team.

Organizational Tools

Using JIRA helped me visualize the workflow and understand what needed to be done. The drag-and-drop interface made it easy to update task statuses and see the bigger picture. Having a visible backlog, sprint board, and clear assignment of tasks gave structure to the process.

Even in a simulation, seeing the work laid out in one place helped me prioritize better. It also made collaboration smoother because everything was accessible and updated in real time. Agile tools like this reduce confusion and make it easier to adjust when project needs shift.

Evaluating Agile Process

Overall, Scrum-Agile worked well for the SNHU Travel project. The biggest benefit was how it let the team stay flexible and adjust to changes without restarting the whole process. It also helped keep the user at the center of every decision, from planning to testing.

One downside was the lack of detailed documentation early on, which sometimes made it harder to fully understand a feature without asking questions. But in an Agile setting, that's expected—the process encourages ongoing clarification instead of trying to define everything upfront.

I think Scrum-Agile was the right fit for this project because of its adaptability and focus on collaboration. The structure of sprints, user stories, and defined roles helped guide the work, even in a simulated environment. It gave me a solid understanding of what it would be like to work on an Agile team in a real workplace.