Ajay Colby

CS 250 -  Software Development Lifecycle

7-1 Final Project

08/17/2025

**Applying Roles**

In the SNHU Travel project, each Scrum role had its own responsibility, but as the tester my job

was to make sure everything we built actually worked the way it was supposed to. The Scrum

Master kept us on track with meetings and made sure we stayed focused on sprint goals. The

Product Owner explained what the client wanted and prioritized the backlog. Developers built

the features, and then I tested them against the acceptance criteria. For example, when the

developers finished the wellness travel slideshow, I noticed that the navigation didn’t work

correctly. I brought it up during testing, and they fixed it before the sprint ended. That made my

role important because it helped deliver a working feature to the client.

**Completing User Stories**

Agile made it easier to finish user stories because we worked on them in smaller chunks. Instead

of waiting until the end, I was able to test features as soon as they were ready. That meant issues

got caught early and could be fixed quickly. The Customized Top Destination List story is a good

example. The developers built it step by step, and I tested each version. By catching small

problems along the way, we were able to complete the story by the end of the sprint.

**Handling Interruptions**

As a tester, interruptions meant I had to be flexible with what I tested. One time the mobile app

story was too large and had to be broken into smaller ones. Instead of waiting for the full app, I

started testing smaller parts like the destination search. That way, we still had working features to

show even though the original plan changed. Agile made this possible because it encouraged us

to adapt instead of sticking to one rigid plan.

**Communication**

Communication was one of the most important parts of my role. I had to let the developers know

when something didn’t work and why. For example, I said, “The search filter doesn’t return

results for partial matches. Can we fix this before the sprint ends?” That type of direct

communication was effective because it showed the problem clearly, related to our sprint goal,

and gave developers the chance to fix it quickly. Daily stand-ups and sprint reviews gave me

chances to share my feedback so the whole team knew what was happening.

**Organizational Tools**

As a tester, tools like the backlog and burndown chart helped me know what to focus on. The

backlog told me which user stories to prepare for, while the burndown chart showed if we were

falling behind. Scrum events like sprint planning, reviews, and retrospectives also helped me

give input. For example, during planning I could point out testing needs for each story, and

during retrospectives I could share what went well and what didn’t. These tools and events kept

the team organized and made my role easier.

**Evaluating Agile Process**

From my perspective as a tester, Agile worked very well. The biggest benefit was that I could

test features as they were built instead of waiting until the end. This made it easier to catch

problems early and fix them before they got too big. The downside was that it required constant

involvement and quick turnarounds, which meant I always had to stay engaged and ready.

Overall, I think Agile was the best choice for this project. The client’s needs changed during

development, and Agile made it possible to adjust while still delivering working features. If we

had used waterfall, I would not have been able to test anything until the end, and fixing issues

would have been much harder.