**Sprint Review and Retrospective: SNHU Travel Project**

Jordan Krueger

Southern New Hampshire University

CS-250 Software Development Lifecycle

June 23, 2023

**Introduction**

This semester in CS 250 Software Development Lifecycle, I did my first ever Agile run of the mill experience along with writing test cases for a project. I was hired by a company called ChadaTech to which I tried Scrum for the SNHU Travel app. I got to display my skills through all the roles on the team for the project, and now, as a Scrum Master, I get to share exactly what those roles were and how this project came to be.

**Applying Roles**

Everyone on the team (Me) took turns doing different jobs throughout the semester. The Product Owner had a to-do list and told us what priority was and what was not. As a developer, I played a part in building the Top Five Destination list in Java, which included me figuring out how to show both the names and the pictures of the destination. Sometimes I had to ask for help if the code decided it did not want to work. When I became the Tester, I used my own test cases to test the features by breaking it and finding bugs so we as a team could move forward. When working on this semester long project it became known that doing all the roles myself made it easier to see how each job compliments one another. If I as a Developer built something I got to see as a Tester and make sure everything could move forward without any headaches.

**Completing User Stories and Testing**

So, for every feature we had a user story, which was new to me. One of my biggest stories involved letting a user click a link that got to view the top five destinations and see the most popular places for travel. When I got done coding this, I used my Module Four test cases to check if it really worked. The first thing I did was open the app and click the link to see if the list would show up. Then, I double checked to see if the names and images were correct. I (in a tester mindset) tried breaking it on purpose like removing an image file to see what happens. I did this quite frequently and would check each case off in my spreadsheet.

**Handling Interruptions and Changes**

The project had its ups and downs. At first the client had a clear and concise idea of what was expected to be done. Midway through they had changed their mind, which is ok because as a team we must be agile in terms of the conditions brought upon us by the client. First the project was about wellness and detox trips instead of regular travel. This meant I had to change the pictures around and update the links. Instead of starting over completely from scratch, we decided to write new user stories and add them to the to-do list. For these new test cases I made sure that the right images were loaded, links worked, and that the images lined up correctly.

**Communication**

Communication is key, always. During the development lifecycle surrounding this project we had constant meetings where we would gauge where everyone was at in the process and what they worked on (i.e. me). If I was running my test cases and something broke, I would just say, “Hey, my app crashes when the image is missing,” and someone would suggest a fix. We had a group chat too, so I could send screenshots or copy and paste error messages. This kind of teamwork and fast help made testing and fixing bugs a lot less stressful.

**Organizational Tools and Scrum Events**

We tracked all our work on a digital Scrum board, which I’d never used before this class. Each user story, test case, and bug went on its own card. At first, I forgot to keep it updated, but I got better about it as the project went on. We did meetings for sprint planning, daily standups, reviews, and retrospectives. These helped keep things organized and made sure we knew what needed to be fixed before moving on. My test case spreadsheet was useful too, because I could check off what I had tested and see what still needed to be tried out.

**Was Agile a Good Fit?**

Now that I’ve done the whole project, I think using Scrum-Agile worked out for us. My test cases found bugs and fixing them right away was easier than trying to fix everything at the end. It was nice being able to change the app without starting over. The meetings and the Scrum board made things less confusing. Sometimes we forgot to update the board, and sometimes it was hard to tell when the project would really be finished, but overall, being able to test as I went made everything more manageable.

**Conclusion**

This project was my first time doing Agile and writing my own test cases for software. I tried out every role, and now as Scrum Master, I can see how all the pieces fit together. Actually, running my Module Four test cases made my code better and helped my team fix problems early. I think Agile is a good way to do projects like this, especially for someone learning for the first time.