In a successful scrum, each role plays a crucial part in driving the team toward achieving its goals. The product owner was instrumental in defining a clear and prioritized backlog. For instance, while we were developing the SNHU Travel app the product owner engaged regularly with the stakeholders to understand their needs and adjust the backlog. This helped to ensure the team always focused on the most valuable work., reducing scope and increasing customer satisfaction.

As the scrum master, I facilitated all Scrum events and removed impediments that were blocking the team's progress. In a particular sprint, the development team faced an issue with integrating a third-party API. I coordinated with the external vendor to resolve the technical issue quickly, allowing the team to stay on track. Additionally, I fostered a culture of continuous improvement by encouraging the team to reflect on their processes during retrospectives and experiment with changes to enhance productivity.

The development team worked together to split off into different areas of development. Some of them were front-end developers and others were back-end developers. Many issues were brought up by this team in the daily-stand-ups and this allowed resources from other teams to get their hands and minds involved to push those problems into resolution.

The testers worked hand-in-hand with the product owner to develop accurate tests for the user stories. These tests were then shown to the developers to make sure the code they were developing would catch all the necessary requirements.

User stories are the lifeblood of agile development. They are the main way we steer the ship toward the goal of completion, so in a way, the stories and the Scrum-Agile workflow do the same thing for each other. The Scrum-Agile approach takes the user stories (usually acquired by the product owner) from the stakeholders and decides what needs to be done first. From the

stories, the testers can derive their test cases, and from those test cases, the developers can start developing code to suit them. This process goes around in a cycle where the other members of the team can help through backlog management and other processes to keep the user stories in focus.

When a project is interrupted or changes direction, a Scrum-Agile approach can be a lifesaver, enabling the team to adapt quickly and still deliver value. During one of the meetings with the stakeholders, the product owner noticed that they wanted to concentrate more on detoxification vacation destinations. Those are apparently becoming more popular. Agile allowed us to shift our work into developing this new angle for the SNHU travel app. If we had been using a traditional/waterfall development style, we would have had to wait until the other version was done, or completely scrap what we had already done and start over from square one. The fact that the stakeholders were allowed to even suggest changes midway through the development is a testament in and of itself about the effectiveness and flexibility of agile. In a waterfall development style, the stakeholders would have had no say in what was being developed after the initial meeting to discuss requirements, and there would have been no room for change at the point they received the software unless we were starting all over.

Collaboration between the teams in agile is one of the biggest responsibilities of everyone on the team. If one part of the team is not talking to another, there can quickly become a sizable discrepancy in what someone is expecting versus what is being worked on/delivered. When I was working as the developer, I wanted to make sure there was a system in place for us to do code reviews. There are many meetings scheduled throughout each sprint already and I wanted to make sure too much of everyone's time was not taken up by this, especially because many other people were suggesting types of meetings for us to take part in. I wanted to communicate

the fact that everyone should have familiarity with the code base and have a good understanding of where the code was toward completion. We decided to have the code reviews done weekly to make sure no one's work was continuously interrupted by things that are not specifically their job while still getting feedback from the team on a regular basis. On top of this, during the daily-stand-ups, I updated the team on everything I had done the day prior and was open to people's questions after those meetings to ensure communication was thorough.

The separation of work abilities of the software we used was vital in ensuring we stayed focused. Along with that, the kanban was instrumental in helping us visualize the progress we were making as the project progressed. As more user stories were added and progress was made, it could have easily become a clustered mess to try and figure out where we were exactly when it came to the overall development. Thanks to the software suite we used, all of that was not really an issue. Everyone had the tools their specific job needed whenever they needed it and the whole team also had access to certain assets that helped us all stay in constant communication with one another. This in turn let us stay focused on our jobs while not being overly insulated from what the other people on the team were doing and the progress we were making as a whole on the project.

There are many pros and cons of the Scrum-Agile workflow. The fact that the stakeholders have their finger on the pulse of the whole process is both one of its biggest upsides and one of its biggest downfalls. In a traditional workflow, the stakeholders have meetings with the team before the development begins where they decide the features they want and user stories are made from those alone. In agile, the stakeholders can change requirements as the project is going, and this can lead to a lot of changes if you are not careful. Scope creep can become an issue if the product owner does not do a good job of getting the base requirements

from the first meetings with the stakeholders. This can lead to an enormous backlog, slow progress, and unhappy customers. The stakeholders having their voice heard is also a big part that makes agile work so well. If the stories were not communicated exactly right the first time, and there needs to be slight changes, that is very possible with agile, whereas that would never happen in traditional workflow. This makes the stakeholders (hopefully) much more pleased with the final result, as the product can be more tailored to what they actually want. It is hard to come up with all the requirements for a project right at the start, even if you're a developer with years of experience. Agile allows the software to serve the stakeholders much better.

I think agile was the right choice for SNHU Travel. The slight changing requirements were not much of a hassle and the implementation of them would have been impossible in the same timeframe under a different structure.