For the SNHU Travel project, we utilized an agile approach using a Scrum-agile team. The team included a project owner, scrum master, developers, and testers. They all played a huge role during SNHU Travel’s project. The project owner provided an outline of the development by creating a backlog. The project owner worked to keep the development process transparent for the team, client, and stakeholders. They kept in close contact with the client to confirm the development was going in the right direction. Then, I, the scrum master, aided the developers with their role by running sprint meetings and daily standups. I also helped the team if any of the developers had a concern or an issue. The developers helped bring the product to life by coding it. The developers were also responsible for organizing the project development. Lastly, the testers aided the team by providing quality assurance for the product. They created test cases for each user story and made sure they worked properly. The testers also collaborated with the project owner to clarify how the client wanted their product to run.

With the agile approach, we used the phases of the Software Development Life Cycle (SDLC) to help complete the user stories. The SDLC phases consist of requirement analysis, design, coding, testing, deployment, and maintenance. All these phases have aided in developing SNHU Travel’s product. In the first phase, requirement analysis, the project owner spoke with the client about what they wanted for their product. Then, the product owner interviewed the end-users to ask what they wanted in the SNHU Travel application. Using the end-user’s suggestions and the client’s needs, the product owner developed a backlog of user stories. During the design phase, the developers prioritized the backlog's user stories for development. The developers and product owner also went over what development tools they would need to create the user stories. After they prioritized the user stories, the scrum master had a meeting with the developers to estimate the user stories and plan for the first sprint. To estimate the user stories, the developers thought about the workload for each of them. After the sprint meeting, it was time for the developers to start working on the coding phase. The developers coded the user stories they chose to work on for that sprint. During the sprint, the developers and scrum master had daily standups to go over the project progress, any issues or concerns, and the plan for that day. Once the sprint finished, the testing phase started. The testers used the backlog to develop test cases for the user stories. They also worked with the project owner to help clarify what the result of a task should be. If any issues or bugs had to be corrected, the testers let the developers know about the error. If the user story worked as requested, it was time for the deployment phase. The development team and product owner had a meeting with the client and stakeholders. During this meeting, the team showcased the product’s progress. If the client wanted to change or add anything, their request was added to the backlog. Afterward, the development team went over the backlog with the product owner to refine it. After that, the next sprint started, and the project went back to the requirement analysis phase. Once the product was complete, then the final deployment went live. The last SDLC phase, maintenance, allowed the developers to make any tweaks or update the live product if needed.

Using an agile approach allowed us to go back and make changes when necessary. An example of this is when the developers created the top destination list. In the middle of the development, the client wanted a change. They wanted the destination list changed to only include wellness and detox trips versus a general destination list. Since we used an agile method, the team could edit the code to facilitate that change. If the waterfall method was used, we would not have been able to change the project at all. Not only that, but the transparency allowed the client to vocalize that they wanted a product change. The developers were understandable distraught about the change. There was a concern that the developers would have to throw away all the code they had completed to that point. With communication, the developers and I were able to talk about the change and their concerns. The team collaborated on how they could implement it without throwing away the code. This communication resulted in a happy team and client.

During development, I wanted to focus on the transparency and communication of the agile method. I had the team start the sprint by holding a sprint meeting. This meeting allowed the developers to collaborate on what user stories they were going to prioritize. During the sprint meeting, I also had them estimate the user stories, allowing them to think about the workload. Once the sprint started, the developers and I had daily stand-ups at 9:00 am every morning. These stand-ups allowed the developers a chance to talk to each other. They went over if there were any issues or successes the day prior. Then, we talked about the plan for that day. During the daily stand-ups, we used a progress board so the team could visualize the project. On the board, we had sticky notes of all the user stories. The developers took the sticky notes and placed them under “not started,” “in progress,” or “finished.” We also used a virtual progress board, Azure, that allowed the client and stakeholders to view the project progress. Transparency with the client about the project development is essential when using an agile method. After finishing a sprint, the development team had a meeting with the client. This meeting allowed the developers to showcase their work and the product. It also allowed the client to voice any of their concerns with the development. The developers and I also held a retrospective meeting after every sprint. During the retrospective meeting, we talked about the successes and failures of the last sprint. Then, we went over what could improve for next time.

There are always going to be advantages and disadvantages regarding product development, and agile is no exception. Advantages to using a Scrum-agile approach include time management, client transparency, less workload in one sitting, flexibility, encourages team building and collaboration. On the other hand, the Scrum methodology has a few setbacks. For example, it can be catastrophic if a developer leaves, time limits are usually unknown, bigger teams have a more difficult time adapting, and not everyone likes to work as a team. Though, when it comes to the SNHU Travel project, I think the Scrum-agile approach worked perfectly. We were working with a small team that needed to get the project done quickly. Also, the client wanted to be involved and even asked for a product change. They were unsure of what they wanted the result to be. These reasons are why scrum was best suited for the SNHU Travel project.