Each roll in the Scrum-agile approach is important for its own reason. Without each role on their own, agile would not be nearly as effective as it is. The three general rolls are the Scrum Master, the Product Owner, and the Development Team. The Scrum Master ensures that the team stays on track with agile ideals in regards to both the project itself and with the interrelations and communications between the team members. They serve as a regulator for the agile environment as well as someone to make sure that the team stays on track. This can be shown in the interactions in the team through the SNHU Travel development project. When there was a change that was made to the project, the Scrum Master asked the Product Owner about how the schedule and deadlines would be affected by the change, making sure that everything remains on track and on time for completion.

The Product Owner’s job is to manage and optimize the Product Backlog to maximize the efficiency of the Development Team. Additionally, it is the Product Owner’s job to confer with the clients and stakeholders in addition to making sure that everything on the Product Backlog is clear to all members of the team. This is evident when looking back on the meeting about the change in direction of the SNHU Travel development project. Not only did the Product Owner meet with the client to make sure that the team is up to date with any changes that need to be made, but they also made sure that the new instructions were perfectly clear to all members of the team.

The last member(s) of the Scrum team is the Development Team. Their job is to take the Product Backlog from the Product Owner and make the user stories from it become a reality. It is their job and their job alone to determine the best way of tackling the Backlog and turning it into a deliverable. Likewise, when any changes are made to the project, it is their job to be able to fit them in and make them happen. This is another thing that is evident when reviewing the meeting where changes to the SNHU Travel development project were announced. All of the members of the team talked and discussed the changes. At the end of the meeting, the Developer present took everything said in the meeting and resolved that she would need to take a look at where the features currently stand and how they would make the new changes possible within the same deadline.

A Scrum-agile approach to the SDLC helped all the user stories to come to fruition. To begin, it is worthy of noting that if an agile approach was not being used, then the chance of all the user stories coming to completion would have been much lower or impossible, since they were changed halfway through the project. The flexibility of agile allows for this, but the waterfall method would not allow for that. Because of the close relationship that the team members have with one another and the simple meetings that the client has with the team and with the product owner, all the possible user stories were logged and created. Then, at the same time, test cases for each of the user stories were being developed. At the same time as that, the development for implementing those user stories was also being done. Because of this simultaneous operation and flexibility offered by the agile, all of the user stories were able to come to completion. Additionally, because agile is so flexible, the user stories were able to be prioritized and deprioritized at will, instead of having to follow a rigid path like with the waterfall model. Two specific user stories that were completed because of agile were creating a list and a slideshow for the SNHU Travel development project. Initially, a top five list was supposed to be created as a user story, and it was prioritized to be completed. After it was finished however, the client decided that they wanted a top five slideshow instead (or in addition to). Because this is agile and agile is flexible, the team was able to make it happen. They prioritized the slideshow and created the deliverable in a timely manner. The user stories were easily completed due to using the Scrum-agile approach to the SDLC.

The agile framework ensures flexibility, especially in situations where a project is interrupted and has to change directions. This exact situation occurred with the SNHU Travel development project and the agile approach supported the change in direction graciously, which facilitated the project completion even though there were changes to the project in the middle of its development. In the middle of the development, the requirements changed to focus on detox/wellness travel versus anything else. Instead of everything needing to be scrapped or all of the plans falling apart like it might have if we were following the waterfall model, the team met with the product owner, adjusted their test cases, and deprioritized other things in the Product Backlog so that we could not only make the changes happen, but also make them happen by the original deadline. This is the great flexibility that is offered by an agile approach that allows project completion even if things go wrong or if the project has to change direction.

All the members of my team communicated effectively with one another and encouraged synergistic collaboration with their words. As a team, we all moved to agile and learned it together. Instead of getting frustrated at the transition from waterfall to agile, we talked with one another about the principles of agile that we wanted to focus on and ensure were implemented correctly and smoothly. We also ensured that when someone had a disagreement or misunderstanding about something that they were corrected or talked to in a way that encouraged learning and progress instead of blaming or shaming them. A perfect example is the tester on the team. He said, “Our products should always be functional, even if we haven't reached the final goal of the product yet. By doing this, we can measure our progress and success through these working iterations of the product.” This notion is close to agile principles, but is a little misguided. He was corrected by another member of the team who made sure to emphasize that agile is guided by the “definition of Done” and not by the unequivocal state of functioning, especially when there are dependencies being worked on. Specifically, he said, “Validating each function is still working as expected at each stage of development is critical. It is also possible that a user story does not function when moved to Done. This is especially true in the event of a dependency.” The member that was corrected did not get offended by this correction, but rather corrected himself saying that he “should have been more specific with [his] words” and then affirmed the correct principles that the other team member had corrected him with. By not being offended by a correction and by taking it graciously, the team is communicating effectively. Since the team further understands that it is okay to correct each other and to be corrected, further following of agile principles is encouraged as well as collaboration among the team members. It is critical that the team members know there will not be any sort of retaliation if there is a disagreement or a correction between any members.

Organizational tools contributed to the success of my team greatly. We decided to use Jira’s software to plan and manage the user stories and duties of those on my team. By using something that is accessible to the entire team anywhere at any time, we were able to maximize our workflow. Since this board was visible all the time and made each member’s workings transparent to the rest of the team, we were able to minimize confusion on the team where one member was unsure of what they should be doing and how it interacts with the rest of the team. Additionally, since the board is easily able to be edited and it changes in real-time for all the members of the team, tasks and duties could be added to the board during Scrum meetings. Instead of trying to remember the duties to write down later, they were able to be added instantly to a board that all members can access. In Scrum meetings, it was also easy to see what was accomplished the previous day and what tasks needed to be further split and delegated to team members. Jira allows us to have a birdseye view of the project and more easily manage time, workloads, and the efficiency of the team.

By following Scrum-agile principles, my team was successful. Some of the principles that helped my team the most were having a small team with a high degree of autonomy, focusing on continual improvement through reflection time and mannerisms that support learning, and maximizing co-location and face-to-face communication. By doing these things, I was able to foster a positive environment that allowed for continual learning where criticism is valued as a positive asset instead of an attack on somebody and communications between team members can be entirely honest and transparent.

The Scrum-agile approach was definitely the right choice for the SNHU Travel development project. This is evident when just looking at the changing/evolving desires of the client during the course of the project. If a waterfall model approach was used for the SNHU Travel development project, then changes in the middle of the project would not be possible, as well as additions in the middle of the project. In the beginning of the project, there was an initial meeting with the client and the Scrum team. If we had taken a waterfall approach, then we would not have been able to have started working on the project yet, but rather would have had to wait until all the planning was done so that we could begin to code according to directions. Instead, the project used Agile, which allowed us to get a start on the project even before we knew each and every desire of the client. Additionally, if the client decided that they wanted an improvement or an additional feature, all that would need to be done is to meet with them and then code it into the system that we are actively developing. This would not be possible with a waterfall model for the SNHU project.

The only real downsides to a Scrum-agile approach comes around when a team is inexperienced or unmotivated. If they aren’t motivated, then their work would be much slower and the project could fail, especially since a Scrum-agile approach makes the team members take on a lot all at once. If they aren’t capable of this workload and the intensity of a Scrum-agile approach, then waterfall would be a much better option for the team. Additionally, if the SNHU Travel development project had bigger goals and needed more team members, then you would have to deal with scaling agile, which can be difficult even though there are frameworks that help it be adapted to large teams.