The various roles on our Scrum-Agile Team contributed to the success of the project in multiple ways. The Product Owner (PO) did a lot of work for us on this project. They were the person who spent the most time with the client, SNHU Travel, and helped them figure out what their vision was for the application. The PO also organized a focus group to gather information about what the end users want from the application, and they created the product backlog with prioritized User Stories. They did a great job communicating with the client and the team. They were the ones to inform the rest of the team about the changes made to the project by the client. The Scrum Master (me) did not have to do as much work for this project compared to the other roles on our team. I was there to support the Developers, Testers, and PO when they needed help or guidance. For example, I helped the PO to build the team based on the given project parameters. Once constructed, there was a decent amount of work at the start of the project since our team was unfamiliar with Agile and I had to guide them through the structure of the Scrum events, but they caught on very quickly which allowed me to step back and let the team be self-organizing. The Developers did a great job with constructing the application. They were able to follow the well-defined user stories to push the project in a direction that provided the most value to the client. They did well in adapting to change when the client decided that they wanted to change their vision of the end product to focus on wellness/detox vacations. Like I said above, by the end of the project they were able to self-organize and hold their own Daily Stand-Ups to communicate where they were at and what needed to be done. The Testers did a great job as well with adapting to change and providing the Developers with tremendous feedback. They created well-defined test cases that thoroughly checked for the acceptance criteria and were efficient at updating them when the project faced changes. They also did a good job at communicating with the PO via email when they had questions about the User Stories.

The Scrum-Agile approach to the software development life cycle helped user stories come to completion by focusing on iterative development, teamwork, accepting changes, and having a well-defined Definition of Done. Once the User Stories were created and prioritized into a Product Backlog by the PO, the team was ready to get to work. The iterative nature of Scrum allowed the Developers and Testers to work on small chunks of the project at a time focusing on the most important and valuable first. The iterative approach improved team morale because there was a sense of completion after each iteration instead of the entire project looming over their heads; this in-turn promoted a good attitude heading into the next week and motivated them to tackle the next set of User Stories and tasks. Another reason the team had good morale was due to them finding solutions together instead of dealing with issues individually; this was amplified by the team’s choice to use Pair Programming which also promoted faster execution of User Stories. Another reason the Scrum-Agile approach helps to complete user stories as the project evolves is because it understands that changes are inherent to a project. It sets clear expectations for the current iteration of what should be accomplished and if changes occur then they will be addressed in the next iteration. This is important because it helps to keep the progress moving forward instead of the team trying to change everything in the middle of other tasks. For example, when the client decided that they wanted to change the vision of the project to focus on wellness/detox vacations, the team was able to finish what they were currently working on before addressing the changes. This provided sustainable progress for the project to build on its current version instead of scrapping all of their work to start over. This is an important part of ensuring that value is being delivered after each iteration and that User Stories are being completed. One of the biggest benefits of the Scrum-Agile approach to completing User Stories is having a well-defined Definition of Done (DoD). By having a unified DoD, the team understood exactly how much needed to be accomplished in order to complete each task for the User Stories.

The Scrum-Agile approach supported project completion when the project was interrupted and changed direction by focusing on being flexible, communicating, and utilizing Backlog Refinement. Like I stated above, the Scrum-Agile approach understands that change is an inherent part of the development process and it promotes being flexible. When this flexibility is baked into the culture and processes, it helps to reduce issues with the team that could arise otherwise like arguments, reduced morale, and poor productivity. Our Testers showed excellent flexibility during this project when the client decided that they wanted to change the format of the top destinations from a list to a slide show. They updated their test cases to accommodate the changes and communicated with the necessary parties to resolve any questions that they had. Communication was a huge part of the team handling the changes that were given to us. The PO did a great job with letting the team know whenever any changes were being made which allowed the team to stay on top of the issues instead of allowing them to pile up. Another huge part of dealing with changes when utilizing a Scrum-Agile approach is Backlog Refinement. Our PO did a great job of managing, reprioritizing, and updating the Product Backlog whenever changes occurred. The reason this is so important is because it gives the team clear guidance on what the most important User Stories are for providing the maximum value to the client after each iteration.

With regards to communication, I have already touched on a few different examples of how our team communicated and why it has been so helpful, but I will expand on them more. As the Scrum Master, my communications mostly were about the Scrum process and developing the team’s skills. I mainly worked with them on the Scrum Events since this is the first time the company has decided to implement Scrum. This communication was instrumental in providing them with the necessary information to become a self-organizing unit in the Agile landscape. The Scrum events inherently promote collaboration amongst the team. One of the examples that I already talked about was our PO’s communication. When the client decided to make changes to the project requirements, the PO called a meeting in order to cascade this information. By calling a meeting and utilizing face-to-face communication the team was able to engage in a much more collaborative environment than they would have if the information was passed via email. Agile focuses on personal interactions over informative messages for this very reason, collaboration. The team was able to discuss the changes with the PO in real time and get immediate answers which allowed them to continue with their progress right away instead of dealing with delayed and unclear answers that might arise through written communication. Although, face-to-face communication wasn’t always possible; a good example of written communication is when our Testers addressed their questions regarding the top destinations’ format changing from a list to a slide show. They promptly emailed the PO to address these questions. When the Testers emailed the PO they showed that they bought in to providing value to the client as efficiently as possible. Instead of just guessing what they believe to be the answers to their questions, they went through the proper channels to collaborate with the PO in order to find the correct answers.

The organizational tool that I am going to focus on is Azure Boards and how it was integrated into our Scrum events and utilized Scrum-Agile principles. The use of Azure Boards was incredibly beneficial to our team’s success. During the Sprint, Azure Boards was able to act as an information radiator for the team and promoted the principle of self-organizing teams by allowing each member of the team to interact and update it as progress was made in the Sprint. During Sprint Planning, Azure Boards was able to promote the principle of iterative development through its analysis by providing the team with helpful insights into metrics like Velocity which ensured that the team was planning an appropriate amount of work for the upcoming Sprint. During the Daily Stand-Ups, Azure Boards promoted transparency by acting as an information radiator and providing the team with information in regards to project progress. During Sprint Reviews and Retrospectives, Azure Boards promoted transparency via its reporting and acting as an information radiator. Stakeholders were able to see exactly what had been accomplished and performance metrics such as Velocity and Lead Times.

There were multiple pros and cons to the Scrum-Agile approach with regards to this project. Some pros were flexibility, transparency, and customer satisfaction. The Scrum-Agile approach allowed the team to be flexible and adapt to changes as they surfaced. Its focus on transparency provided the client and stakeholders with the necessary information with regards to the project’s progress. By being transparent with SNHU Travel and collaborating with them during the development process, the Scrum-Agile approach was able to provide the most value to them by adapting to their changing needs. Some cons of this approach were its learning curve and time management requirements. Its learning curve required the team to spend extra time learning how to properly implement each aspect/event of Scrum which had an impact at the start of the project. The other con to this approach was its heavy dependency on the team being able to effectively manage their time due to all of the Scrum events and meetings. This also was a responsibility of the managers to push the team in the right direction, but this approach also focuses on the team being self-organizing and limiting manager involvement so it took a little time to find a balance. I believe that the Scrum-Agile approach was the correct decision for this project since it did not have exact requirements defined at the start and the client had evolving needs; the Scrum-Agile approach was able to provide flexibility and adaptability which proved to be a great asset for this project.