In a Scrum-agile team, there are a few roles that operate cohesively to develop software efficiently. At the core of a Scrum-agile team are the developers. These individuals build the software and handle the low-level tasks that go into building the desired product. Within Scrum, developers are also expected to maintain transparency throughout the team. Due to their hands-on role with the low-level tasks during the sprint, they have the best view of how the sprint is going and what still needs to be done. Through daily Scrum meetings, developers will share their current tasks and what still needs to be done to ensure that everyone on the Scrum team is on the same page. For example, during the SNHU Travel Site sprint, the developers were responsible for implementing the original software solution in addition to the mid-development pivot to focus the site on wellness retreats.

Similar to the developers, testers in a Scrum-agile team are primarily concerned with the low-level tasks of the sprint. Testers will generally work alongside developers to design tests for the tasks the developers are working on. This requires testers to actively communicate with the developers to fully understand the key aspects of the project. For example, throughout the SNHU Travel Site sprint, testers designed test cases to coincide with the user stories. During the pivot, testers were also responsible for re-evaluating the existing test cases to check for inadequacies with the new software changes.

 At the top of the chain in a Scrum-agile team, the product owner attempts to keep the team moving in a unified direction. Product owners are the direct line of communication to the customer and are tasked with translating the customer's needs to the rest of the team. In addition, product owners are expected to maintain the backlog of tasks to ensure nothing falls through the cracks and ensure everything is moving forward at a pace in line with the release schedule. For example, the product owner on the SNHU Travel Site team communicated with SNHU Travel and used that information to create the user stories that would direct the development of the website. They were also responsible for maintaining the line of communication between the team and the customer, which directly led to the decision to pivot to a focus on wellness retreats.

Finally, the scrum master overlooks the scrum team. While the scrum team manages themselves regarding the distribution of tasks and maintaining transparency, the scrum master acts as insurance for this process. They will oversee the process to make sure adequate transparency is met while also putting the processes into place to make that transparency easier to achieve. In addition, the scrum master will coach the team on Scrum principles. It is not guaranteed that the entire team is educated on Scrum-agile processes, but it is expected that the scrum master will be certified and will be able to educate the team and support them in the process o creating a self-organizing environment. On the SNHU Travel site, the scrum master's primary role was assembling the original plan on how we would maintain Scrum-agile principles throughout development.

The Scrum-agile approach to the software development lifecycle directly contributed to the completion of the user stories. Due to assembling user stories based on customer input early on in the development lifecycle, breaking down the project into easily delegated tasks became much simpler. It allowed developers to check back to the user stories in order to implement the desired features and to ensure that they were staying in line with the original product vision. In addition, the user stores allowed testers to have a head start in developing test cases that were based on the original user stories. This streamlined the testing process and allowed these tests to be planned and designed before implementing the feature.

One event that happened during the development of the SNHU Travel Site was the desired change in direction from the customer. Originally the site was intended to show overall travel locations and popular vacation spots., but midway through development, the customer expressed interest in changing the focus to health and wellness retreats. While this didn't change the website's overall operation, it required the team to take a look at the user stores and test cases to make sure nothing would have to be significantly altered. Agile-scrum allowed this to be possible as, without it, the transparent systems showing the completed work and yet to be completed work wouldn't be readily available with as much detail as it was to the entire team.

I believe a core example of communication within the team came from the email after it was decided to change the project's focus to support health and wellness retreats. This was a key moment for communication as it required multiple team members to work together to decide what needed to be changed and what the scope of these changes would mean. Ultimately, this communication decided that the only changes that would need to be made are to the initial travel packages, and the site's core functionality could remain the same. This communication event also allowed the testers to make sure their test cases would still be relevant and make it clear if anything would have to change on their end.

The main tools that allowed the team to stay in line with Scrum and agile principles were the User stories and the test case documentation. These two tools went hand in hand, but they set the baseline for what was to be expected of the team throughout development. To begin with, the user stories took into account the customer's desires and organized them into development tasks, allowing the developers to focus on the work and not on the interpretation of the customer's needs. On the other hand, the test cases allowed testers to check the developers' work against tests built on user stories. This helps the implemented features function as the developer intended and check to make sure that what the developer intended was in line with the original product vision.

Overall, Scrum-Agile served the SNHU Travel Site project well. It allowed the team to change focus quickly and easily due to the modular nature of the development, and it maintained transparency through the Scrum team. This transparency ensured that unwanted downstream effects would be minimal, if at all. That being said, if the customer wanted a bigger change than what they had requested, I am unsure if it would have worked out so smoothly. For example, when the customer requested their change, the product owner assured the customer there would be no change in deadline despite the fact no other team members had investigated the impact of this change yet. If the customer had requested a change to core functionality, this could have backfired immensely and left us well behind the deadline. However, for our situation, the Scrum-agile approach worked very well, and I believe it was the best approach for this project.