For the SNHU Travel project, each role contributed to the project’s success by furthering its goals at each step. The scrum master ensured that each step of the process was completed by providing discrete objectives for each week. They also gave the initial overview of the project to be completed during the sprint and the timeline. The product owner provided user stories from the interview with the company, which gave larger parts of the project that needed to be completed to meet the requirements of the company for the program. This helped the developers and testers with their portion of the work as well by providing important details for their work. The testers gave test cases with details on what functionalities the code needed to have to meet the needs of each user story, and the developers changed the code to fill those needs. When the client provided updates to their needs, the product owner updated the user stories, which allowed the tester to update their test cases, and the developers to update the code to meet the new requirements.

The scrum-agile approach helped each user story be completed by having each specified by the product owner, such as the top five destinations list. The tester then gave the test cases, which specified how the system needed to function, which allowed the developer to write the code to meet the requirements of the test cases. When the client changed their need from top five destinations to top five destinations for detox vacations, the product owner provided updated user stories, which let the tester give new test cases, so that the developer could change the code.

Communication with team members is an important part of the software development lifecycle to make sure that it runs smoothly. From the product owner, asking the client for details is necessary to make sure that they are satisfied with the product delivered. They then need to communicate these details to make sure that the team knows what the client wants, and if anything changes, they need to update the team.

From the tester, asking the product owner for use cases as shown below, is necessary to make sure that test cases can be written.

Product Owner,

Please provide any prerequisite functionality required for [list user stories], and if possible, user stories for those, so that we can ensure those are functioning before testing the user stories listed above so that testing is not confounded by a nonfunctional or nonexistent prerequisite.

Thank you,

* Tester

This example shows how the tester can make sure they have the details they need to write test cases and that the product owner knows why they are important to have for the project to move forward.

A communication sample from the developer to the product owner is shown below, which highlights the importance of continuing communication so that the team can continue meeting goals as they change:

Product owner,

I understand that there have been changes to the system’s requirements. Please send the details for the desired changes with any changes to user stories, as well as any that need to be removed, or new ones. If there is a change is scheduling for this, please send that as well.

Thank you,

Developer

This email shows that while it might be known that changes were made by the client and communicated, they might not immediately be sent to the team, especially in a form such as user stories which make it easier to break down the product into parts to build. It may be important to remind others to communicate updates, even if it is something small like changing the types of vacations, the new user stories may be easier to work with than outdated ones while needing to track one change to the client’s needs.

For scrum, breaking down the project into smaller tasks is important. The weekly modules in the project served as daily standups with the assignments serving as the daily tasks to complete, and the discussions gave opportunity to get a better idea of how things worked for completing that week’s assignment.

The scrum-agile approach was effective for the SNHU Travel project, since it allowed for changes to be made as they came up without sacrificing progress, as happened when the client changed the type of vacations to showcase. Some things could also be worked on without having every detail of the system, such as the database, which was not needed to test the frontend. In spite of these advantages, the whole system could have been outlined then built, since the only change that came up was minor and would have only affected the vacations listed, not the functionality. Scrum-agile was the best approach based on the changes that came up during the project, since it allowed things to be compartmentalized and worked on, and changes to be made when the client needed.