The various roles within a scrum-agile team consist of: the product owner, the scrum master, and the development team.

The product owner is responsible for taking care of the details that the project needs to achieve. In this course we were working on the SNHU Travel application. This resulted in being patient and listening to what they originally wanted, which was primarily a list of the top places to travel. They didn’t have too much they wanted in the beginning other than top places to travel. Later in the life cycle it was changed that the project needed to now relate to health and wellness vacation spots. As the product owner, I was the first one to know and had to relate this information to the scrum master.

The product owner is the back bone of the project information to get the scrum master to relate the information to the development team. The two roles work back a forwards to make sure the whole SDLC works to the end. The scrum master leads the development team and primarily helps the team rather than working on the project themselves. They need the proper time to make sure the project is done and step in whenever needed.

In the project lists the scrum master has multiple meetings with the team and helps lead everything to the end. I primarily pushed my expectations with my team to make sure the project was being done. I even “tested” the project with code and helped small sections to make sure the project is being completed. The scrum master also had to change the project later with the top 5 vacations being only health and wellness now. This push and change was communicated and the team even asked if the date of release was changing, which it didn’t. This minor set back was 100% perfect and well communicated to the team and the scrum master was direct with the information. To the team, this can be a huge motivation for them to continue pushing the project itself.

The development team consist of the 2 sides of tester and developer. The tester tests the projects and relate information back to the developers about the project in the field and the developers make the project it self to be sent out and used. The team had to work together to make the project first, which was code and tests on the app itself in a “final” version. This was tested and mentioned back to the team with each other. I mentioned some of the traded information in the email test about the tester email. I said, “When testing I would like you to be as detailed as possible to tell me what steps you took during testing so if an issue arises, I can attempt to replicate it and find the correct code causing the issue.” This type of communication is common and results in proper test of the code and keeps the project in line.

The scrum agile approach helped the completion of the project by in the examples of the product owner mentioning the change to a health and wellness vacation spots. This was told to the scrum master and then the scrum master pushed this information to the team to be worked on and changed, The team asked if the time was going to be changed and asked if this was a hard change to force the project to change. The scrum master confirmed the time would stay the same and pushed the team that this wasn’t going to hurt them. This enforced the team to stay motivated and kept the project on track.

As I mentioned above when I said, “When testing I would like you to be as detailed as possible to tell me what steps you took during testing so if an issue arises, I can attempt to replicate it and find the correct code causing the issue.” This was a message to the tester from the developer and I wanted to make sure the code problems could be replicated to make sure I can accurately grab the problem.

The tools and scrum agile principles that helped my team be successful was the push of the information to the whole team as quickly as possible to ensure accuracy of the project in the end state so the team could work on the final section. This worked in our favor to ensure that information was pushed down and mentioned so we didn’t go far into the project life cycle and have to change more than needed. This is a direct pro to the scrum agile approach and allowed for a perfect project.

The only con to the project I can think of is that the customer changed midway through the project and resulted in a little bit of a time delay for the team. While this is not a big deal for the example, in the real world the change can be a huge issue for teams. It could result in a huge set back in the final stages of the life cycle if the customer changes their mind. But this also shows another pro of the approach. It is the direct communication from the customer to the rest of team without having to need the customer to tell them. It is a fast communication process. I think the approach is perfect for this due to the app needed more information for certain sections of the project. The customer to the rest of the team and the rest of the team to the top people.