During the project, I had many different roles. Each role played a part in the development process of the software being created. One of the roles that I took on was the Scrum Master. The Scrum Master is basically an event planner. They help establish rules, meetings, schedules, questions, and complaints for the team, among other things. One example from the SNHU Travel project’s initial meeting was that the Scrum Master created the team for the project, assembling a tester, developer, and a Product Owner. Another example would be the Scrum Master fixing the blinds on windows because the sun impeded the team members work. One very important thing that the Scrum Master does is facilitate the Daily Scrum meetings. These meetings provide the team with a ton of information about the project and the sprint. At these meetings each team member is asked an average of three questions: “What did you do yesterday?’; “What are you doing today?”; “What is impeding your progress?” These questions, as stated above, provide a lot of information about what is going on within the team. The Scrum Master, also, meets with the Product Owner and gathers input from them.

The Product Owner is another role for the agile process. They don’t interact with the team as much as the others, however, they still play an important role. As from the SNHU Travel project’s initial meeting, they communicate with the client, gathering wants and needs from them. Product Owners, also, provide updates to the stakeholders, asking if anything needs to be changed. As not everything the stakeholders say can be added to the project, the Product Owner decides what will go through to the backlog. All of this is important, because the Owner acts like a barrier between the client and the development team. Without it, it could cause a lot of stress and distractions among the team, leading to wasted time and unmet deadlines.

One important thing that the Product Owner and Scrum Master work together on is the product backlog and sprint stories, although, they are generally created by the Product Owner. The backlog is exactly what it sounds like, a backlog. It contains all the to-dos for the project and keeps track of their position in the timeline. A lot of companies use backlog related software to help them, like Trello. Most software can be made open to the public, read only of course, which will help the Owner with responding to clients. The sprint stories are short, descriptive pieces of text that detail what needs to be done for a specific piece of the backlog. It contains all the information a developer or team member might need to finish the piece.

One large issue in speaking with and keeping clients happy is adhering to their requests. This is mainly for the Product Owner to handle, and partly with the Scrum Master as event timelines may change. Thankfully, using the agile approach, direction can be swiftly changed to meet the client’s change. If it is a drastic change, the Product Owner will have to communicate with the client and stakeholders about any side affects that may occur. A prime example is the increase in work and the push back on the deadline of the project. The Scrum Master, also, may have to create new, temporary meetings to handle the change in backlog and user stories. All of the this is a good example of how the agile method can deal with these changes. A good way to think of it is creating a meal for dinner. Using the agile method, each ingredient is prepared separately, before adding it to the mixture. If the end result needs to be changed, the prepared ingredients, if not mixed together, can still be used.

Moving onto another role in the team is the Tester. The tester does exactly what the title states, they test the code. Testers work very closely with developers as the developers are the ones creating the code the testers are testing. They may even work together in teams creating the code as a second pair of eyes is always helpful when writing a program. While trying to break the code and test for any bugs, the Testers keep a log of what they did and what they did to cause the break or bug. Most of these are in response to Test Cases supplied by the team. Each Test Case goes over what needs to be tested and how it will be tested. Personally, I think something should be added to each Test Case that allows the Tester to be free in how they test it. This is because something could always break in a way you didn’t expect. The tester would be given a time limit as to not waste too much time on the task. After the test case and results are completed, it goes back to the developer(s) who fix it and send it back through again. This is another good example of the agile work method. Developers create code, Testers test and respond, and the developers go back to creating/fixing the code. It’s a small loop but an agile method, nonetheless.

The last role of the team is the developer. I chose this role in the project because I feel more comfortable with it than any other role. The developer’s job is to create, maintain, and fix the code for the program and project. The developer, also, works very closely with Testers to make sure the program and code works correctly. Another main job for the developer is to attend the Daily Scrum meeting. There, they choose what stories they’re working on, what they’re doing, and can ask questions if they need help. With the agile method, the developer(s) aren’t stuck doing one thing at a time if something happens. Developers can switch to another item in the backlog without wasting time. Because of the constant switching that may happen, having a software related backlog is good for the agile team. Developers can move things around and send notifications when something isn’t working right or if they have more questions.

The agile method, in my eyes, is by far, the best way for a development team to work. It allows team members to work on separate things and the same time. Comparing to the waterfall method, the agile method will get things done quicker. Developers wouldn’t have to wait for one portion of the project to be drafted before working on it. Nor would testers have to wait for the same portion to be created to test it. Using the change in trips from module five, the waterfall method would have a difficult time adjusting. New stories would have to be made, code would have to be thrown out, and the deadline would definitely be pushed back. With agile, the team can quickly switch without confusion, the code would be able to be adapted, and the deadline met. Therefore, I think the agile method would be the best choice for the SNHU Travel team.