On an Agile/Scrum team all roles are important and influence the team and project in different ways. The roles include Product Owner, Scrum Master, and Team Member/Developer. No role is more important than any other role as they are all important and needed for the Agile/Scrum framework to flow correctly and cleanly. All roles also rely on communication and collaboration throughout the project for it to run smoothly as well.

When It comes to the Product Owner role, they tend to be the bridge between the team and the Stakeholders and Users for the project. They determine with the Stakeholders and Users what the basic and minimum requirements for the project and what the minimum expectations for the team would be. The Product Owner also tends to get the final say on certain things pertaining to the project. One example would be that the Product owner tends to have the final say in the Sprint Review on accepting the code as “done” by determining if it meets previously set standards for the project. While working on the SNHU Travel project I had that exact experience while determining the initial requirements for the project from the Users and creating User Stories from the requirements and functionality requested by the Users. Another important task that was required while in the role of Product Owner is setting up and ordering the Product Backlog which was based on the User Stories created previously.

Another role on a Scrum Team is Scrum Master. They tend to be in the position of a servant-leader, in that they lead the team by helping them with problems that impede their work as well as help train them in Agile and Scrum practices. They are also responsible for facilitating many of the Scrum Meeting like the Daily Standup, Sprint Retrospective, etc. While in the role of Scrum Master for the SNHU Travel project, I experienced what it would be like to facilitate a Daily Standup and the expectations of a Scrum Master in that position. Keeping the discussion on topic for instance and taking not of any issues mentioned in the meetings that pertain to the team or team member and their work on the project. Another aspect in being a Scrum Master is helping with ordering the Product Backlog as the Scrum Master has a better understanding of the team and their velocity then the Product Owner.

The last role on a Scrum Team would be the Team Member or Developer. Everyone that is considered a Team Member or Developer is just that and no one taking priority over the others, giving everyone on the team a sense of equality. The main duty of a Team Member/Developer is to create code that lives up the set standards of “done” or releasable for the project. To create code that is also readable to others is also important for collaboration and future iterations or changes. It is also the Team Members responsibility to establish the amount of work they can do in a single sprint and to keep all boards and project information up to date to what they are working on. There is usually a Tester on the Development Team as well, that is still considered to be a Team Member but specializes in testing code.

When it came to completing User Stories there where a few Agile approaches that I took to help complete the project. One the first actions I took was breaking down the Users requests or User Stories into smaller Stories that are workable in a single sprint by a single person. This help greatly as it made the project seem less daunting and after ordering the Product Backlog by priority it made it even simpler because it became obvious as to what need to be done next and why. Another Agile practice that I took is being open and expecting change in the project. As the project progressed there was a change in the design or layout of part of the SNHU Travel project, but because it was expected in the Agile framework it was easy to initiate and complete the change.

This is also a great example of how Scrum supports change and makes it simpler and more possible then with the Waterfall Method. Expecting change was a huge in creating code as you know to make it clean and easy to read for future changes either by you or someone else depending on the change. Having the code created separately and in iterations that don’t affect the overall code as it is being made also helps with changes because a change in one part of the project that goes wrong wont mess up the entire project and instead only that one part.

Another part of the Agile/Scrum framework that is very important to the project is communication. Scrum is about taking responsibility for your work and sometimes that means you need to communicate with Team Members, the Product Owner, or the Scrum Master to have a better idea of what is expected of you. If there is something unclear or confusing, communication would be very important to not waste time on something that might not be needed or wanted which would result in a waste time for that person and the team as a whole. Some examples of this would be when I was acting out as a developer, and I had some questions about the changes that were made to the requirements and features on the SNHU Travel project. Though this was a pseudo e-mail this would have been very useful in changing the code closer what is wanted and expected.

The most useful organizational tool that was used during the SNHU Travel project would probably be the Product Backlog. This made everything very clear as to what was needed and about how much work each part of the project would take to complete. I feel like an overall project board like information radiators would have been useful if there was more then just me on the project as well as a communication tool such as Jira, but as it was it was not need as there was no one else to communicate with. With that I determined that the Product Backlog would have been the most useful “organizational tool” for this project as it was really the only one used.

When it came to the SNHU Travel project, the Agile/Scrum Framework would have been better for an actual team working together to create this product. As it was the Agile/Scrum Framework only seemed confusing seemed to have no actual goal. It seemed a bit scattered and there was no actual work being done on the project, it was a pseudo project after all. I feel if this project was to be fully completed as it is currently set up, the Waterfall Method would actually be better for one person working on the whole project with no one to actually communicate with. Without an actual team to communicate and collaborate with if seemed like an incomplete idea and project and a set-laid out plan would have been easer to follow. This would also mean much more planning in the beginning and less ability to change the functionality as it continues, but with only one person working on the project is seems like less of an issue as long as it is readable. If the project had been worked on by an actual Scrum team, I do believe that the Agile/Scrum framework would have been best as it would have required more communication and collaboration between the team members as well as the need to be able to change sections worked on by separate individuals. Making change more difficult if the Agile/Scrum framework was not being used.

**References**

Charles G. Cobb. (2015). *The Project Manager’s Guide to Mastering Agile : Principles and Practices for an Adaptive Approach*. Wiley.

Ambler, S., 2022. *Agile Requirements Change Management*. [online] Agilemodeling.com. Available at: <http://agilemodeling.com/essays/changeManagement.htm> [Accessed 17 June 2022].

Atlassian. 2022. How to Use Jira Software | Official Buyer & User Guide. [online] Available at: <https://www.atlassian.com/software/jira/guides> [Accessed 17 June 2022].