In an agile software development lifecycle framework, there are various roles that are required to have a successful team. The roles include a Product Owner, a Scrum Master, Developers, and Testers. The Product Owners are responsible for representing the business and their needs and making them clear to the team. They drive the success of the product by ensuring everyone is on the same page. Product managers also communicate with end users to try and see if they can help drive the product in the right direction. They also work hard on prioritizing work to be done. It is their job to look at the backlog of user stories and set them in an order with the highest priority one being worked on next. Product managers also ensure that the deliverable is something that is what the business is looking for. A Scrum Master is responsible for helping the internal team communicate effectively, leading cross team collaboration, and resolve pending blockers. They are to ensure that everyone on the team knows what they are doing and does everything they can to help the team be as efficient as possible. They are also the prime facilitator of all the agile events, making sure the team is attentive and on target for each one. They also own the agile tools used and work closely with the Product Owner to keep things like the backlog and sprint board in good order. Our team’s Scrum Master is in charge of the daily standup and always makes sure to help out with any blocker that comes up. The developers do exactly what you’d expect, they develop software. Depending on the type of project, there could be different types of developers including ux engineers, devops engineers, web developers and so on. The developer domain is not important though, their job is to deliver software that is derived by the business through the Product Owner. Developers are responsible for taking specific user stories and turning it into a deliverable piece of software. Lastly, but not of least importance is the tester. Their job is critical to delivering a solid product at the end of a project. They work closely with the developers creating test scenarios to ensure that the software meets the necessary requirements. They also provide a level of security to the project goal by developing tests that prevent regression. Often times when new things are introduced into software, it can silently break other things. A testers role is to prevent this as much as possible. They are the reason for bug-free software and are absolutely crucial for a software project. The testers in our team are incredibly transparent and do a great job communicating with the developers. This is crucial because as they give each other feedback, they make sure that the project is polished, and they don’t do anything to put each other down. There are a bunch of different roles, but they are all critical to the success of the softae.

The agile based approach was absolutely critical in getting through the user level stories. Because the stories were not feature long, but items that could be completed in a couple of days, it really helped us to prioritize. If something came up with a high priority it was okay for some developers to shift gears and stop what they were working on. And since the stories were small in scope, it was not difficult for the developers to pick up their work again, which would have been the case if we had done waterfall and were working on feature level tickets. Because our Scrum Master and Product Owner were great, they were always making sure that the user stories had great acceptance criteria and requirements. This allowed the developers and testers to work incredibly fast and effectively without having to chase down stakeholders.

We were able to communicate very effectively. At one point we had a retrospective, where it was brought up that we ought to do more discovery on user stories before putting them into the backlog. This then led the Product Owner and the Scrum Master to do a more thorough job on adding acceptance criterium to the ticket. We were able to be transparent and open about where there was a bottleneck and it really stare to make us more efficient. Because we had more descriptive acceptance criterium, there was a lot less back and forth between the developer/tester and the Product Owner. We all noticed a big difference after we put this practice into place on our team.

There were a lot of ceremonies and tools that helped us get our job done effectively. I think the most important one is the retrospective. It’s good for a couple different reasons. First because we get to look back at what went right and celebrate the small wins. This was great for team morale and helped us gel as a team. Secondly because it gave people who wouldn’t normally speak up the option to give back feedback. It wasn’t always bad feedback, a lot of times is was just small comments here or there, but I don’t think they would have been surfaced had we done a retrospective. Lastly because it gave us the opportunity to look back at the spring and honestly evaluate ourselves. There were a lot of times where we were thankfully able to admit something went wrong and were able to talk about a solution in our meeting. Similarly, the daily standup was great as well. Although it can feel redundant and mundane it allows us to make sure everyone is on the same page and helps us resolve blockers as fast as possible. Spring Refinement was also a good practice because the whole team had a chance to review and improve user stories, making sure that everyone was in sync and also ensuring that everything was technically possible.

The Agile framework made the SNHU travel project a breeze. It allowed us to iterate when things came up and made sure that every different team member was on the same page. Usually there isn’t a direct relationship for testers, developers, and product owners, so we really saw the benefit of having us all coordinate with each other. We learned new things, and also had a better understanding of the project as a whole, instead of certain parts being siloed into a particular group of people. Because there were a couple of times that our priority and plan change, I can confidently say that an agile based approach was definitely the best decision for the SNHU travel project.