Over the course of the term I took on several different roles of the team; all of them having different, but just as impactful effects on the project. The project my team was assigned was to spruce up the website for the SNHU Travel project. As the project owner it was my responsibility to gather and interpret the desires and goals of the stake holders. Starting with a meeting involving myself, as the product owner, the scrum master and a selected panel of stakeholders, where they shared there griefs of their current layout, and what changes they hope to implement. From there I took my notes from there and organized them into user stories to organize into a backlog. For example, from my notes the number one priority I created was the ability to see the top five travel destination when you open the website.

Next I took on the role of the developer, as the developer I received the backlog, already pruned and organized by the project owner and scrum master, and began work on modifying the existing code of the webpage to fit the needs the best I could. Unfortunately once my job as the developer was over, there was a meeting held involving the project leader, scrum master, developer, and the tester, to inform us that the desires of the stake holders has changed rather drastically. It then became my task as the tester to modify, not only the costumers code, but now ours as well to meet these new requirements the best we can without breaking it all. Thanks to the way the team was set up, and their work within their roles we were able to accurately deduce the stake holders desires, organize a sprint to create a prototype of our goal, and respond quickly to a drastic change without a change in our original deadline.

Back to the user stories, that helped in every step of the process, the format used to help determine and organize them was also a huge help. The table format that broke down every part of the process to its most basic and technical forms ensured they stayed relevant to each member of the team. For example, breaking down functions desired by the stake holders, such as having the ability to organize results based on price, to just that, the most basic and technical aspect, helped not only the scrim master determine how important it is compared to other goals, but also the developer to interpret what is needed without any need to try and interpret it themselves and guess, and do exactly what is needed, no more or less.

The major interruption of our project was the change in desire of what to show, it was something completely different than originally desired and that we had created. The agile approach was followed, and the project owner held a meeting containing not only them and the scrum master, but also the developer and the tester. From there the normal format for a scrum meeting was followed, and the scrum master determined that despite the major change in direction, the deadline would not change. The developer expressed her concern that all her work thus far would be scrapped and the previous work would be useless; then the tester said that he might be able to work with the code he had received to modify it similar to the new direction. This process, involving all layers of the project, made it so that a resolution to this new direction was swift and a solid direction was laid out, and every one knew their roles and responsibilities moving forward.

This scrum style of communication makes all levels and steps of communication between different departments very effective. The best example was during the interruption, when the direction of the project was changed pretty drastically; the meeting held was in the scrum style of containing everyone on the team. This direct line of communication between all levels streamlines the process, and made sure all the important questions could be answered quickly, especially involving the deadlines, and everyone's griefs where also able to be expressed. This is a major improvement had the project owner just sent an email to the scrum master about the change with no further explanation, and the scrum master passed it down to the team. All of which could lead to a lot of confusion and even more frustration. Instead having everyone present and in an in person meeting, things can be made more clear much quicker, and problems can be resolved, hopefully, just as swiftly.

There were several tools used that helped the project be as successful as it was, but I think the most useful was probably the tool that helped us break the user stores for the rest of the team. Having a tool that helps organize a backlog is essential to any project manager honestly; but I think the user stories, no matter how much I personally disliked breaking down and organizing, helps every step of the process afterwards. Other than obvious benefit of having an organized backlog for the scrum master to break into sprint goals, having a single document, that can be shared or passed down to every step of the project, without any editing, helps everyone, the developer can tell clearly what their working towards directly from the costumer, and the tester nows exactly the parameters needed to be met.

In the end the scrum-agile approach was the best for this project, evident by the drastic change in direction part way through the project, which is a very common occurrence in todays projects. I think that would be the biggest pro to using the agile process for this, other being the things such as transparency and organization. The cons for using it for this project, would be the same as the cons of using agile in every other project, agile is pretty fast and loose with everything for the sake of being flexible and right on time; though my gripes with that are the same with running a ‘right on time’ business strategy in manufacturing, if anything disrupts the flow, every step of the process is disrupted and damage control is harder to manage. The alternative would be a water fall type of approach, any mistakes or disruptions would be pretty local, but a major upset like the one experienced, would have a much bigger impact, and instead of being resolved in a single meeting it would be harder and take longer to fix, definitely effecting the deadline, rather than keeping it the same as we were able to do with agile.