As the Scrum Master I made sure our team was on track to finish the project before the end of this sprint. I talked to the product owner, who had some changes the customer wanted to make mid-sprint, and we were able to fit those changes in immediately without compromising our timeline. I also made sure that the daily scrum meetings went smooth, keeping the meeting on track and short. We made sure to only talk about the necessary topics at those meetings, what we did yesterday, what we plan on doing today, and any impediments in our progress (Cobb 42). I was able to sideline any extra topics for the people that wanted to talk about them and helped address some questions or impediments our team had along the way.

Our product owner was able to provide a great product backlog, which allowed us to focus on the important parts of this project first, saving less important items for later. Their organization of this allowed major changes to be made mid-sprint which did not affect our timeline at all. They knew when to say yes to the customer and when to say no, allowing only important changes to be made and not overwhelming the team with tons of changes. Having the product owner be transparent with the customer, working with me along the way, allowed the customer to have the necessary components within their product without any extra fluff. They were able to help us keep the project “just barely good enough” in order to meet the deadlines and requirements.

Our developers did an awesome job following the product backlog, working together if necessary. During our daily scrum meetings, we were able to assess how much progress was being made and moved some of our team to work together after a certain point, as some developers were having some glitches with creating executable jar files for production. Having another team member assess this with the dev in question, we were able to fix this problem quickly by the next week allowing no time to be lost. It turns out a virtual environment for programming does not follow the same file path rules as a native run IDE, which caused crashes when trying to load files within the virtual environment. Having pair-programming and our daily scrum allowed this to be found and solved within a day.

The tester on our team had some tough changes to make when the customer and product owner came to us with major changes in the middle of the sprint. They ended up working with the developers directly as they made their changes in the programming side of the project, allowing all tests to be created quickly. If this was a project that was done within the waterfall method, well first those changes would not have been allowed, but if they were the testers would not have been able to make quick changes, since they would not get the project until the very end of the development lifecycle.

The scrum-agile approach helped us complete our user stories efficiently by giving a clear statement on what the feature or end goal is. Having the user statement be in easy to understand English gives a very clear goal to look for. Also having the requirements of each user story clearly written makes it very clear what needs to be done in order to complete that user story. For example, the first user story we were using titled Top Five Destination List, had a few very clear steps as to what was needed to complete that. It was clear from that user story that it was a link that needed to be clicked, an ordered list was needed, and it was laid out what was needed in each element of that list. Having each user story give a clear expectation allowed us to efficiently complete each one without many, if any, questions regarding the expectation.

When the customer came to us and wanted to pivot the direction of our project, obviously it was a bit of a surprise, but we were able to respond and implement the change right away. Having our product owner change the priority of our backlog allowed us to focus on that change and not fall behind the timeline we had. Since the changes made were building on some of the progress we had already made, it was not a complete throwaway of all our progress. If this was a waterfall approach, our testers would not have seen this change coming and that would have slowed the process. Since the testers were working in tandem with the developers, we were able to make the desired changes right away.

Our team was able to communicate via email and meetings to get ahead of any problems that could have slowed us down. I know our developer had to write an email to the product owner and tester to help clarify the changes the customer presented in week 5. Although I would have rather the developer reach out to me first, I apricate the initiative to just get the issue figured out. That is an advantage of the agile approach, there needs to be an open communication between all members of the team. This way we can all just work together to get the project finished. It was also very helpful that our product owner told us about the changes made to the project right away, this allowed everyone in the team to pivot on those changes that day, giving us plenty of time to change everything necessary.

Our biggest assets in this process have been the use of user stories. Having a very clear backlog with user stories that are easy to understand and provide the expectations of each feature allowed us to complete each feature as necessary. Having them organized logically so that each built on the other also allowed us to get work done without waiting on other team members to finish their parts. Our daily standup was another great asset in getting this project finished on time. We were able to see what each team member was doing on a day to day basis and was able to assess if anyone was falling behind or needed help in completing their part of the project.

The SNHU Travel project was a great project to use the scrum-agile approach on. Some pros of using this method were that user stories allowed us to have a clear expectation on what needed to get done and when major changes to the project were made, it allowed us to change right away without losing time on completion. Another pro was that the communication of our team got rid of any issues that a member might be having right away. No one was waiting on another team member to finish something and just sitting idly. The only con to using an agile approach was that there was a lot of setup to this process. Since it was our team’s first time using this method, we had a lot to learn and did not have a good estimation on how well we worked together or how long a user story might take us as a team. On our next project I could see this process being much easier. Overall, I do see that the scrum-agile approach was the best way to approach this project.

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