Having differentiated roles allowed each team member to specialize and focus on only a small set of responsibilities. This enhances focus and ensures no team member is spread too thinly. When everyone is made to feel like they need to handle everything, the door is opened for mistakes, sloppiness, and stepping on each other’s toes.

The Product Owner’s responsibility was to communicate with the client, in this case, the SNHU Travel agency, and convey their needs to the rest of the team. In addition to this, they also handled the prioritization of tasks that needed to be completed. This means that not only does the rest of the team not need to worry about communication with the client, it centralizes the voice of the client. Rather than having competing views and interpretations of what the clients needs, the voice is centralized in the role of the Product Owner and the rest of the team simply need to worry about following the Project Owner’s lead. For example, when the client wanted to modify the Top 5 Destinations feature and change it from a list to a slide show, the client did not have to worry about finding out who the developer was, they just had a singular point of contact they needed to relay their needs toward. Furthermore, the benefits of this singular point of contact did not only benefit the client. When a change is made, it affects more than just the person implementing the change. It affects the whole team, including the person testing the changes and the person managing the timeline. If the client had only reached out the developer, the developer may have not taken the initiative to convey these details to the rest of the group in a timely manner which may have caused confusion. The developer’s main role is to write the code, and while they do need to be concerned with timelines and testing, that is not what will be at the forefront of their mind. The fact that we had a Project Owner alleviates this concern and facilitates the deployment of changes in a way the entire group will be able to manage. The separation of roles basically allows the software development team to work as efficiently as an assembly line at a factory.

To develop the user stories, the Product Owner took it upon herself to meet with not only the SNHU Travel agency reps but also, and arguably more importantly, customers of SNHU. She did this to ensure that the agile team did not waste their time or their efforts on features that a decent proportion of the customer base did not want or would not use. Once she concluded her meeting, she took her notes and met with the agile team and discussed how those notes of customer requests could be broken down and organized into cohesive tasks (user stories) that could be then ranked and prioritized. Once this was done, the developer can now step in and implement those user stories, after which the tester could verify the correctness and completion of each task that corresponded to each user story. This process was further facilitated by the nature of the user stories in that each user story had specific requirements that acted as a check list for evaluating whether or not the task was successfully implemented. This helped both the tester and developer do their roles.

When the client indicated that they wished to change the focus of the Top 5 Destinations feature from a general list to a list that primarily targeted customers in search of a Detox/Wellness retreat, the Scrum-agile approach was integral in in making sure that in the aftermath of the change, development proceeded smoothly. In general, during a change it is the Product Owner’s responsibility to retrieve the information and update the backlog as necessary. Once the changes are adequately documented, similar to the case in which the Product Owner gets the initial information for the commencement of development, the Product Owner will meet with the rest of the team and discuss the changes, resulting in the creation of new user stories that relate to the new tasks to be completed. One of the main aspects of the Scrum-agile approach is that the team members meet in person, if possible, often. These face-to-face meetings are the opportune time to bring up such changes; this is exactly what my team’s Project Manager did. In order to minimize the chance of misinterpretation, my team’s Project Owner relayed the information about the modifications in the plans to the team during the face-to-face meetings. This was proven to be the best choice, rather than doing so by email, because there were several concerns that the change raised and they were able to be dealt with in a prompt and concise manner.

Both inside and outside of in person meetings, communication among team members is very important. It optimizes the work flow, allows everyone to be appraised of all important aspects of the project each step, and provides additional clarity whenever necessary. A good example of my team doing this was during the original plans for the Top Destinations feature, which was a list of the 10 most popular destinations. The tester reached out to Christy, our Project Owner, for clarification about the aforementioned feature, and this interaction actually resulted in the transition from having a Top 10 Destinations List to having a Top 5 Destinations Slideshow. Without this seemingly small interaction, more time would have been wasted on a feature that was going to end up getting changed, but because of this interaction, the process was expedited.

The main aspects of the Scrum Events are: sprint planning, daily scrum, sprint review and sprint retrospective (which is the most important part of scrum). The main tool my team used was VersionOne Ultimate Edition. This project management tool was integral in assisting my team in executing the aforementioned events. Agile project management tools, such as VersionOne, facilitate the team’s ability to engage all members in the process. This means that the planning, meetings, and reflections will be much more fruitful than if all of those events were driven by a singular force such as the project manager. VersionOne also helped us consolidate the requirement backlog which in turn helped everyone stay up-to-date on what was going on. This, as I noted earlier, is an integral part of efficiency, especially in a project as fast moving as an agile project.

From my experiences on the SNHU Travel team, I would say that the biggest pro of the Scrum-agile development process is the ability to evolve with the client’s changing values and handle modifications to the plan efficiently. However, this pro can be a double-edged sword and leads me to what I feel is the biggest con of the agile development process, which is the potential for team morale to suffer in the event that changes are made, features that have already been completed are scrapped, and team members are made to feel as if their energy and time have been wasted. With all this being said, I do believe that the agile-development process was indeed the best choice for the SNHU Travel project. This is because, while the project did entail changes along the way, and thus benefited from the robustness of agile, the changes weren’t so numerous nor were they so ill-timed that anyone was likely to feel as though their talents were being misappropriated or wasted.