Based off the work that has been accomplished through the past 7-8 weeks, I can wholeheartedly say that each member of the Scrum team has contributed significantly to the progress of the program we are developing for SNHU Travel. Reflecting on the strides the team has made, we can identify that without the user stories developed by the Product Owner we would have trouble finding proper direction and issues satisfying each of the hard requirements as well as soft requirements of the customers. Not only from the first concepts of the user stories, but also during the sprint the product owner kept good contact with the customer and allowed us to work towards exactly what the customer wanted the program to perform. One such case of additional information being obtained by the product owner would be during the print process we were told that at first the list of destinations would be the top 5 most popular destinations. Later in the sprint the product owner got together with the customers, and they relayed the information that instead of the top 5 most popular destinations, they wanted it to be focused more on detox and wellness retreats. Ontop of communication with the customer, the PO also managed our product backlog which allows us to keep track of the requirements of the software. Had we not had the product owner reaching out to the customer and obtaining more information on this shift in program focus, we would have delivered the original product that was requested rather than the final product that was required.

Now given the position of Scrum Master, the general structure of the sprint would not exist and would be thrown into disarray if we did not have one. Structuring the sprint and assigning each small team with their own tasks is a huge benefit. Not only from the organizational benefits, but our scrum master also helped lead the charge with the daily meetings which allowed them to assign each small team their tasks without having to run around and try to catch everyone when they are free. As for each of the teams of developers and testers. Each of these groups provided feedback that was crucial to the completion of the user stories.

Using our agile approach, we were able to more efficiently adapt the product to the shift in focus that occurred halfway through development. A major benefit of the agile approach will be the as the name suggests, agility of the methodology. Using it over the Waterfall method has allowed us to review, revize and in some cases completely restart our thought processes and work completed on certain portions of the software. One of these cases would be when we jumped from having the list sorted as a top 5 destinations list overall to a top 5 wellness retreats. Had we been using the waterfall method we would need to restart the entire program rather than adjust the content being shown in our framework. Had we needed to restart the program we would have needed to redesign the display around our new content, in the process scrapping our previous work. This would have caused a major set back. However, by using the agile method as we did, we were able to circumvent the setback and this allowed us to deliver our product on a shorter time scale rather than possibly having to push back the completion date.

During the lifecycle of our system, we traverse through many phases. Beginning this process is the planning phase in which we prepare ourselves for the trek ahead. This is done by identifying our problems that require solutions. On top of this we needed to determine what our eventual objective would be for the program to accomplish. Next we step into the Analysis Phase. During this phase we took what we had planned for the system in the planning phase and built off of it. This was accomplished within our daily scrums as well as our longer planned meetings. We also acquired critical information for the project from our customers, conducting a meeting/survey of our users lead by the Product Owner and myself the Scrum Master. We use this information in order to properly develop our prototypes in order to fulfill the requirements set by our users. This allowed us in our project to fine tune what our program accomplished. If we had gone with just what was planned for in our planning phase we would not have caught the shift in requirement from top 5 vacation destinations to the top 5 detox and wellness retreats. We then hit the Design Stage, which included the process of defining the different features of the program. This ranged from both our user and system interfaces as well as the databases and the network uses and requirements. Without getting a universal understanding of how we wanted to as the name suggests design these facets of our program, we would have a much more difficult time with producing a proper program causing the development time to take even longer. Finally we get to the Development stage. Which as the name suggests, is where the development of the program takes place. This is when the developers on the teams sit down and write out code to the specifications provided earlier in attempts to fulfill the program's requirements. Obviously without this phase we would not have a workable product to provide to the customers. Afterwards, we test the product to make sure it is up to both industry and personal standards in the Testing Phase. This is so we can ensure we are releasing a quality product and that there is no requirement that it does not fill. Next we come up to the Implementation and Integration phase, this is where the final product really starts to come together, during this we implement our smaller modules and code into our main code. We then use varying forms of testing in order to detect any more errors or defects before it is a workable program ready for our end-users. The combination of the testing phase and implementation/integration phases ensures that our product is polished before we ship it out. Lastly, we come to the Maintenance Phase consisting of us performing “repairs” on our product in order to fix any bug reports or concerns expressed by our end-users. If we look at the early stages of the game industry, where physical cartridges and discs were shipped out as finished products and there was no maintenance phase during these. This resulted in many older games being left as is with whatever game breaking bugs may have slipped through the testing phase. Thankfully, the much more adaptive environment of modern times has allowed us to make use of this phase in order to keep our program working as designed and as intended. Also allowing us to adapt it to the varying demands of our end-users through the agile method.

When deciding between waterfall and agile methodologies, I would personally use agile when dealing with products that don't have a specific clear vision of the final version. Where waterfall excels is when there is a very strict and clear road path that needs to be followed and has a strict immovable delivery date. This is because with the waterfall process we get strict specifications on the deliverables between each phase shift. While with agile, it is better suited for a project that has no clear end product, is a fast moving cycle and allows for a little bit of experimentation.