Even though it’s my job to make sure the team stays on task while also adhering to Scrum principles while developing, I wouldn’t be able to get this far without each part of my team. Their combined help makes the development process go by a lot smoother and can make it fun! Firstly, there’s the **Product Owner.** Before I go in depth as to what the Product Owner’s job is, I’m going to sum up their job in a simple sentence, and then I’m going to go in depth at what they do. The Product Owner’s job is to maximize the value of a product, to both stakeholders and potential customers. Now with that idea in your head, let’s get into how they do that. These won’t be all the things that they do, but just a few of the major things that they do to get a better generalization of their impact on the product/software. Firstly, Product Owner’s prioritize items in a product’s backlog. What this means is that depending on the needs of either customers or stakeholders, one feature will be implemented before another. Ideally, Product Owners will want to implement the feature with potential the most impact that’ll make consumers go, “Whoa! I must tell others about this!” and spread good word of mouth. Next is an obvious one, that being clear communication. A Product Owner needs to be crystal clear in telling the dev team the vision of a product. This will help the dev team since by being clear, the dev team can get a better understanding of the product and then will be able to think of updates or features that can elevate that product vision. Lastly, and usually the make or break, stakeholders. You’re probably looking at this like, “Huh? What about stakeholders?” Well, Product Owners need to have meetings with stakeholders so that they can get their feedback on what they currently like about the product and what they’d like to see in the future, AKA, see their vision. Sometimes, the vision of the stakeholder and the Product Owner can clash, and in those situations, it would be ideal to find a middle ground. Next up with have the backbone of the team… **the dev team** (haha)! Since without these guys, there wouldn’t be a product to begin with! Similarly to how I explained Product Owners, I will only be doing a few major things that they do, rather than everything that they do. One of the things that helped us during the project was all the different uses of each section of the team: the software designers (UI, Front Page, etc.), the product testers, and the entire dev team! All their talent comes together to make the product the best it can be. Next up is task ownership. In other words, the dev team decides who is working on what during an Agile Sprint. During this process, the team also looks at how the task can be done as fast as possible as well as efficiently as possible. Lastly, and another make or break, making sure the product works/making it as high quality as possible. The last thing you want to do is give a client a product and then they report back that the product doesn’t work, or that the product doesn’t exactly meet their vision. A team wants to meet the client’s vision and more! Emphasis on “**and more**”. You want to deliver a product where the client will love it so much, they won’t want it to be changed, at all. That is the ideal that I want my dev team to follow. Next up is… **the Scrum Master**, AKA me! As mentioned at the very beginning, my job is to make sure my team stays on task while in an Agile Sprint. As such some things that I must do are as follows. Facilitate, specifically facilitate for team. I lead the planning of sprints, I lead the sprint reviews, and daily standups (short daily meetings with the team to discuss how things are moving along in the project). While facilitating, I can also see what type of hinderances my team is facing during development and see what I can do to remove it or ease it. Lastly, as mentioned before, I must make sure that my team follows Agile principles, throughout the project development process. Since if it falters at any point, the path to completion will become mucky. In some cases, the Agile Process may not work, and in those cases, it’s my job to make sure I find an alternative plan for our development process, so that my team can get back on track. By using a **Scrum-Agile approach**, we as a dev team were able to utilize the feedback from user stories and apply the feedback into the product. However, sometimes we can’t do a completion 100% follow through on the user stories, but we’ll try to implement according to their feedback the best we can. For instance, during the SNHU travel app development, we had user stories about what users, specifically, users who used the product the most, wanted to see implemented. One of the stories requested that they wanted to have a section that gives them good deals on travel packages. Depending on the user’s usage of the app, we would make sure the app is as customizable as possible to the user. Next up are **interruptions**. Unfortunately, these things can happen during development and can happen for a multitude of reasons. Maybe the client changed their vision entirely or maybe the client doesn’t want the product anymore, just a few examples. In this situation, it’s best to adapt as best you can and make sure you can still deliver a good product in the end (if the client still wants the product.) For example, I’ll use the SNHU Travel project. There can be a case where deals that’re meant to give clients the best offers, aren’t working as intended; and in this case we would remove the feature and rework to work properly. Next is **communication**. Here is an example of one of my explanations I’d have about a session with my team: “At one point while we were ending development, our client requested to do a full reset of the project. As unfortunate as this was, our Product Owner guided us to an alternative way to complete the project.” Now comes the **organizational tools**. Some tools that helped us out are but aren’t limited to: Product Backlog (A list of things that need to be completed for the project), Sprint Board (Tells the dev team their current progress through a project), and lastly a burndown chart (which similarly to a Sprint Board, shows the remaining amount of work in a sprint or project.) Now it’s time for the **Agile Process**! Specifically, the Scrum-Agile effectiveness for the SNHU Travel project. Pros: Get a better understanding of what consumers want in the application. Cons: Depending on the complexity of requested features, it might take too long to implement and overshadow other potential add-ons. Overall, the Scrum-Agile approach was good since we’re able to take back feedback from users who well… used our application the most and take their feedback on what would make our product better. This in turn can make the customers loyal and spread good word of mouth to get new potential customers who in turn can also become loyal!