Antonio Bethune

CS 350 : Final Project

April 17, 2021

A scrum-agile team is a cohesive unit of individuals who take on specific roles when working on a project to complete a task. What makes this team different from other teams, specifically teams that use the waterfall method, is the fact that they embody the ideas of agile development. This means that each team member is flexible and willing to make changes in their respective disciplines as needed throughout a projects life cycle to achieve the best product. The main roles of a scrum-agile team, in no order, is the product owner, scrum master, developer, tester and some also consider the customer. Now even though some of the roles have multiple people filling the role or sub teams, the full size of the team is usually no bigger than the amount of people that can share 2 pizzas.

Starting with the product owner, this is the team member who is an active liaison between the developers for the product and the customer or commissioner of the product. One of their duties is working on the product backlog; making the initial draft, keeping it up to date and making sure it is readily accessible to the team. This role is usually held by one person and they hold this position for the entire life cycle of the product until it is released. Whenever there is new information or something that needs to be changed, the product owner sees to making sure each person on the team is aware of it and then records this information in the product backlog and on any other platform that the team maybe using for record keeping, such as on a Kanban board. During the development of the SNHU Travel project, the product owner made sure to have the product backlog for the user stories completed and disseminated to the developers. By creating this backlog for the stories, the product owner was able to assess the customers wants and needs for the application and place them in a well formatted document that was recorded for the life cycle of the product and easily accessible for the team. Whenever a change might have arose for the application to be edited, the product owner was able to go back to a document and edit the changes as needed and then give the new product backlog out to the team. This helped to keep the team organized but also gave a nice record of the stages and changes made to the product mid cycle which is an advantage of the agile method of developing.

Another role is that of the scrum master. This position is also often held by one person, even though many people on the team may have had similar training or have done the job of a scrum master before. The main objective the scrum master must achieve is to ensure active communication and allotted times for the team to work out any problems they might have. A scrum master would do this by setting up daily meetings, standups, or communication chains such as emails to achieve this objective. While working on the SNHU Travel project, it was observed that the scrum master would make daily standups in which topics would be created and assessed so that the team would be able to effectively communicate on their progress. The scrum master also made sure to time for topics that were not related to the current standup to be assessed, as seen in the video we were given for the scrum standup in a previous module and made rules for what happens when someone misses the scrum meeting with or without a valid excuse. The scrum master is also aware of any prior obligations by the team members and relays that information as needed during scrum meetings. For my portion of the project in particularly when I had to work in the setting of a team with my classmates, I found it nice to have a scrum master that was flexible with everyone’s schedules and made the use of chain emailing a priority communication tool so that everyone knew where others were at in the project discussion stage without everyone having to be available at one time.

The role of developer in a scrum-agile team is one that cannot be overlooked, and it is one of the roles that usually needs a sub team. This is because the developer’s role is to make/code the product that will be released for the company/customer. A developer does exactly what their name implies, they take the ideas and then documentation that the customer and product owner have and create the actual product itself. Depending on the level and complexity of the project, more developers might be suited to handle different sections of the project. Developers usually use some sort of board to create a sprint system in which they work within a sprint timeframe (usually a few days to a week) in which they try to develop a certain portion of the product and then test and integrate their new code/product components into the main product and push it up to be apart of the final product. For the SNHU Travel project, the developers were constantly working on their application to meet certain standards and story points requested by the product owner. This was the main development for the product that would have been released by the due date for the product.

A tester is another in house contributor to the scrum-agile team. They usually are a group of people who have expertise in finding bugs and breaking components of the product created by the developers and work hand in hand with hem to fix them. Their role is very important as one of the biggest downfalls of any released product is if it is defective and usable by the actual customers. For our project, we had several iterations where our testers had to make sure that there were no bugs in the code. We also played the role of tester and developer during assignments to make sure that we found bugs and corrected them and then tested them ourselves to make sure we were meeting the product requirements.

Using a scrum-agile approach with developing our user stories was a big help in defining what needed to be built. The process for the development of the user stories was simple in all we had to do was listen to the customers and figure out how to incorporate their needs into an application. The only thing that I believed would have made this more difficult for a product owner creating this backlog would have had is if they had no idea on how an application would function. The agile approach made it so that the user stories would have a broad overview of what the product should look like and do, that way the developers could decipher information and build from a sort of skeleton like model of the application. The product owner’s intelligence on how they application should flow from the customers feedback was important to this step. Writing in acceptance criteria and labeling the levels of the user stories by relevance and importance was a great way for the product owner to relay good information that can be worked on by the developer. The developer in turn made sure to evaluate the acceptance criteria and use those as building blocks for the code needed in the application. The main acceptance criteria were to show the top 5 destinations for the customer/user, and this was incorporated into the application for the 5-page slide show. Having the acceptance criteria of allowing the user to scroll through them, was a backbone for making it a slide show and have the developers something to go from to make it either a rotating carousel or a clickable one, depending on the UI/UX interest of the customer. Even if the developer used the “wrong” method to rotate through the images, an agile team can go back and fix these minute changes without destroying the project completely.

Changing the project flow midway is also not a huge deal for a professional scrum team either. During this project it was decided to switch the focus on the types of vacation packages shown to the user on the top 5 carousels. With other methods, such as the waterfall method, going back and changing this feature would require a lot of back logging and there might be more hoops to jump through to approve this kind of change. Also depending on the time frame of the project’s completion time, it would automatically be pushed back further to implement a late game change due to the nature of non-agile methods. With an agile team though, this change was easily made due to the way the product was being developed. Agile teams usually develop components of a project that are put together, so the changing of a component might take some time, but the continuous integration process allows for changes to be made simultaneously while the rest of the developers can continue with the other parts of the product.

During this project there were times where I had to communicate with my classmates as a team. This started out at a difficult stage since we all live far from each other and do not have a place to meet up. Also due to the global pandemic it would be better and safer to find alternatives than to congregate in person. Due to this we used chain emailing and the discussion boards as our main form of communicating. This was an agile principle to remain flexible especially with all the team members different schedules. We used the emails to take on designated roles for the project and keep a backlog of records for everyone to refer to. Also, the discussion boards allowed us to showcase our communication through a live board and was a better form of communication for other team members who checked the class discussion more often than their emails. This helped us become a successful unit by allowing us to communicate with a way that accommodated mostly everyone’s time and other commitments. Gave a sense of responsibility to all team members by allowing them 24 hours to respond to emails and make their post on time after a certain email was sent. While also allowing for a good backlog of information that could be referred to, using chain emailing was a good scrum-agile practice that we implemented with our situation.

SNHU Travel defiantly needed a scrum-agile team for the development of their application. They needed a team that was able to change on the fly with new ideas that they had down the line, and they knew this because their initial idea for their application was vague. This was good because if their idea was fully fleshed out and had no gaps in data at the beginning (they went solely based on their web application), then the changes down the line would have been difficult to incorporate, and their mobile application would be the same as their website which would lead to a sort of redundancy in making a mobile application since moist smarty phones can access the internet anyway. The only real con that the agile process might have had is the movement of release dates depending on the progress of the team, but I believe that would be a problem with any method, such as the waterfall method, if the same problems arose using an alternative developing approach. Therefore, the best thing that the company could have and did do was use an agile based team to get the job done.