**William Wootton**

**SNHU**

**CS250**

**6.29.2025**

Different roles work in different ways so that all must come together cohesively to contribute to a well-developed and implemented project. These roles inside an agile style approach to managing the project include the product owner, who is commissioning the product and may have different requirements for the product and may have different ideas on how the product should look and operate. Next, we have the scrum master, this member works directly with the product owner to communicate accurately with the development team to make sure any requirements are added to the product, any changes that may be added in the future are implemented, and to coordinate the team so that all tasks are done in a timely fashion and completed as such. The development team, is a group of individuals who directly work on the product they include roles such as the tester, which work with the dev team to decide which parts of the product are working, or not working and help a team understand where problems are occurring and what needs to be fixed or what is completed. Next, we have the developers which directly work on the code of the product. This group will be asigned tasks by the scrum master and work to complete them to the best of their ability within a sprint. This group also may need to return to a previous part of the product if a bug or issue is found by the tester.

All these roles together, along with any other roles that may be added to help the team, must work together to complete a product that is sound and up to the product owner's wants and needs. When working these past 7 weeks I have noticed the testers' role is more important than I may have originally believed, this is because when coding many issues may occur, or bugs may appear which may not be discovered directly by the developers. This means that the testers must be able to accurately find, and label bugs found while also communicating this with the team and offering solutions to larger issues that may be because of an issue with the product as a whole. Such as things like UI design, a tester must be able to realize if a UI is hard to navigate or use and must provide an alternative or alternate idea for the development team to approach. The development team must also have good communication with the scrum master so they can convey if a task is going slower or quicker than originally planned, and they must also be able to address issues that are found during sprints.

User stories are probably the most useful tools when creating a product, especially with the agile style approach to development. User stories provide ideas for implementations in the product; a product owner may only have a vague outline or idea for a product which is where user stories come in to help form a complete and comprehensive product. For example, SNHU Travel had a basic idea of implementing a website into an app, however when looking at the user stories ideas such as likes and dislikes, and activity reviews can greatly elevate the usefulness of the app as people can easily plan and curate their vacations to their exact wants.

For any and all interruptions agile allows for a flow like development cycle, certains tasks are going to be completed faster or slower, and certain tasks may run into issues. This may cause a change in the way a product may need to be coded and will change the final product, the agile style of development can allow for this as new ideas are easily brought to the scrum masters plate then he can bring this to the product owner and if cleared can then assign and change tasks around to accurately fit the needs of the team currently. This also works if a team member seems to be falling behind; another member can be added to assist with that task if their own is ahead of schedule. This flexibility not only with the team, but also the product allows for something not available in a traditional waterfall style system that is rigid and does not offer much change during the process of implementation. Agile all around is easier to work with and allows a team to be much more involved in the product and design as a whole allowing for improvements in the original design and development continues.

A development team member may realize that a task will be difficult to implement but also may have experience with a similair process that works just as well as the tasks they were originally assigned. The team member then may go to the scrum master and explain his or her case and with approval from the scrum master may implement this change which can cut down cost and development time, freeing up useful resources that can then be pushed into other parts of the product. This also works for new ideas that may come to mind for a team member that, if the product has the available resources, can implement which may improve the overall product itself. A tester may realize that the way a menu works is confusing and hard to use if the user is using the product for the first time, they can bring this to the attention of the scrum master who can then brainstorm with the team to develop a strategy to tackle the issue and implement the change. The tester may then evaluate it again and provide any more feedback they may have. This communication is key and will be the most important part of the process as without communication between the team then the product itself will suffer greatly.

When doing research for the organizational tools I found some of them to be incredibly useful and easy to use and understand. Jira for example, provides a well detailed look at the project you are a part of. When I first looked at Jira, I decided to try to use it for my own personal projects and found it to be easy to use and work with. I can add comments and add features I wish to implement, I can see how the code itself is going and can assign team members to given tasks which allows the scrum master to accurately tell where all their resources are currently being used. Another main feature is the summary, this is found on most other sites as well but provides a very important overview of the product, it shows the distribution of tasks which are in progress, completed, or need to be worked on. It also shows any changes made by anyone apart of the team and can accurately show how the product and the current sprint are progressing.

Scrum-Agile worked well in mine own experience with the SNHU Travel app, it allowed me to listen to and gather new ideas and information from the user stories which was vital in what I felt like not only translating the website to an app, but improving the product as a whole. Diving into the different roles allowed me to understand how the roles may look differently at tasks. A tester will look to find issues and problems, while also trying to figure out ways to improve the product. While a developer may work to implement a task and find a better way to do so or come up with an entirely new idea that helps push the product to the next level. The scrum master must communicate with not only their team, but also convey the team's current mood, how the work is progressing, and discuss new ideas with the product owner accurately. This can be difficult depending on the product owner who may be stubborn in their approach, which may require the scrum master to debate and come to a mutually beneficial agreement for both parties.

For this project I found Scrum-Agile to be the best approach and would say that most of the time this is the case. Many products which refuse to listen to feedback, and criticism fail miserably and can even kill the reputation of a company. I firsthand have seen this happen and it makes me angry to see any company from a software development, to game design, to tv show producers continuously let ego, or stubbornness cause their products to flounder and eventually fail because of their own unwillingness to adapt and listen to their consumers.