The product owner is essential for communicating with the customer, pulling information from customer input, and translating it to small but detailed increments of work known as user stories that are then organized by priority into a product backlog. In our case, the product owner conducted a focus group meeting where they gathered opinions of customers on the site and what features they would like to see. From the information they have gathered priority seems to lie in personalized vacation lists, profile settings, vacation types, etc. They then communicated these features when they created user stories. The stories were to be clear, concise, while still being detailed enough for the team to understand what features the customer wants.

The developers do what they can to meet the customer’s needs by actually developing the product and addressing any bugs brought up. Developers should be quick to react to any changes and work together to overcome any challenges. When the product owner informed the team of the features the customer wanted, it was up to the developers to translate that into code for the site. The team was able to address each user story starting with the Top 5 list. Similarly, the testers help with quality assurance and let the team know of any shortcomings or bugs that need to be addressed. Testers came up with test cases for each of the user stories to ensure that the product did what was expected. A few examples of this include whether or not any changes made to the profile actually saved to the site’s database and if each slide of the hot deals user story showed all of the information needed.

The scrum master ensures that the rest of the team follows scrum practices. They act as a coach by helping the team adhere to scrum implementation and addressing any problems preventing the team from going forward in production. Scrum masters oversee meetings, help answer questions members may have on scrum or agile, are able to help the product owner if there are any questions related to backlog management, and lead by example in scrum adoption. When the company switched from waterfall to agile, I took on the role of scrum master and did my best to offer guru advice and walk our team through each of the meetings. I helped open lines of communication and was able to answer any questions as to how we should go about these changes.

With waterfall, the development process is somewhat easy to understand because it is a step-by-step process where each task is completed in a more linear format. This, however, comes with drawbacks. With a strictly waterfall model, quality checks can have a major impact on development as this is often only done towards the end of production. It is also difficult to allow the customer to have any input on any changes or additions during production because they usually will not see the product until the end of production. Scrum-agile attempts to solve the problems waterfall has by prioritizing adaptiveness and communication. A sprint will have regular communication through sprint planning, daily meetings, sprint review and retrospective. Opportunities to test the product with both testers and the customer are also frequent during this development cycle.

Scrum-agile was integral to helping us reach the sprint goal it encouraged communication and teamwork, allowed our team to be flexible, and helped ensure customer satisfaction through frequent testing and customer feedback. Sprint planning allowed us to take a look ahead at the product backlog. Planning also served to allow each member of the team to negotiate with the product owner on what tasks were to be put into the backlog. An example of this came when the product owner discussed with the team which tasks would be user stories to be placed in the backlog. The daily standup meetings were great for communication because it allowed each member of the team to update the rest of the scrum team on what they’ve done, what they were working on and any challenges they were facing on a daily basis during the sprint. The daily standups allowed us to discuss as a team who would work on the Top 5 list while who would be setting up the profile settings for the site.

Scrum-agile also allowed for our team to be prepared for a change in direction. Our team was well into development when the product owner informed us that the customer wanted the lists to instead be in a ‘slideshow’ format. In a waterfall approach this kind of change may push the release date back to account for the change. Instead of scrapping what we have worked on, we placed more focus on the changes the customer wanted and shifted development to cater to that vision. Clear and frequent communication with the customer allowed us to react to what they wanted earlier rather than waiting until the end of development.

When our company initially adopted Scrum-agile as an SDLC, it was important as a team to be supportive of each other in facilitating change in the work environment. As a developer at the time, I made sure to direct any questions to the scrum master and the product owner and communicate as often as possible. When we weren’t available to talk in person, email was the next option. For daily stand ups where we are discussing ownership and our progress, a whiteboard might be an inexpensive but effective visual aid to help the team keep track. I also suggested the idea of utilizing Azure Boards to ensure everyone involved has quick access to any planning that might otherwise only be seen in person. This in conjunction to daily standups would help everyone be on the same page as far as planning and progress.

Scrum agile encourages communication, flexibility, efficiency, and visibility. The frequent meetings allow for teams to really know what needs to be done and who owns which tasks. The communication allows not only teams to keep each other informed but also allows the customer to give valued feedback during development. Scrum-agile allows for congruent development, meaning several user stories can be tackled at a time leading to good time management. However, there are some downsides a software development company might need to be aware of. Scrum only works if each member of the team is committed and cooperative. Scrum also relies on experience, making adoption difficult. Having a flexible end date can lead to scope creep where a manageable project becomes bloated from features being added in over time. (Chandana, 2021) Overall, I believe Scrum-agile worked out well for the SNHU Travel Project. The boon in communication and flexibility especially helped us to keep up with the customer’s needs while empowering each team member to work towards the team’s goals.

Chandana. (2021, February 25). *Advantages and Disadvantages of Scrum*. Simplilearn.com. https://www.simplilearn.com/scrum-project-management-article.