**Final Project**

Brandon Cook

Southern New Hampshire University

CS 250: Software Development Lifecycle

Professor Tarik Iles

June 21, 2024

**Final Project**

The various roles of the Scrum team create a cohesive unit that all work together to achieve project goals. The Product Owner takes on the task of acting as a medium between the business needs of the organization and the technical needs of the Scrum team. They have a thorough understanding of Agile practices and develop Product Backlog items that the development team can refer to in order to understand what the user needs of the final product. Organizing these items in order of importance lets the development team prioritize the required functionality without being distracted by direct interface with the clients and stakeholders. The Scrum Master plays an important role in ensuring collaboration and clarity. They do this by organizing and scheduling the various events that take place throughout a sprint: Sprint Planning, Backlog Refinement, Sprint Review and Retrospective. They act as an intermediary for each event to help coach the development team to be more self-organized and cross-functional. Testers take information from user stories to create test cases that the development team, who are creating the product and implementing functionality, can use to ensure they are meeting all of the required acceptance criteria that is needed. All of these roles come together and lead to an efficient process of product development that leads to success.

User Stories are an effective way to manage the needs of the clients and stakeholders, presenting that information in a way that is useful to the testers and development team. Interviews with clients about their needs lead to a list of items that will need to be implemented into the product, and those needs are organized into a simple list with a common format: “As this role, I want something so that I can receive this benefit.” This gives a concise overview of the functionality needed, and these user stories can then be organized by how important they are. Acceptance criteria, which is a list of requirements that must be met before the story is considered complete and ready for deployment, can then be added to the stories for further detail. When I was working on the SNHU Travel project, creating User Stories alongside the acceptance criteria to accompany the stories helped with creating a visual idea of what the product needed to do. The user wanted to see “hot deals” based on their profile settings and viewing history, and creating acceptance criteria allowed me to think through how that would actually work in a technical sense.

In Agile development, interruptions are all but guaranteed. The environment needs to remain flexible in order to continuously meet the needs of the user, regardless of how often they are changing. When needs change, backlog items need to be reprioritized around those changes and new acceptance criteria needs to quickly be implemented. When working on the SNHU Travel project, the client decided that for the top 5 destinations, they wanted to shift focus to detox and wellness vacations as they were the current, ongoing trend. This forced the team to shift focus and quickly rework existing code; rather than restarting from scratch, test cases just needed to be updated and segments of code edited. The schedule did not shift, as is common with Agile, so other user stories that existed in the Product Backlog were deprioritized to make this shift the main focus of the project. Remaining flexible and adapting to the change allowed the project sprint to remain on course.

When it comes to communication, the concept of “openness and transparency” is especially important in an Agile environment. This is supported by the use of Scrum events: the planning, review, and retrospective. Daily stand-up meetings also achieve this. By remaining open to colleagues and peers, they can feel more comfortable communicating concerns and offer assistance when working on more difficult problems. Asking questions for clarity can keep the team on task without doubts.

In this age, there are many automated and visual tools that will help with organizing a sprint in an Agile environment. One of those tools that I particularly enjoy is an information radiator. A large screen is displayed in front of the development team, showing items from the backlog that need to be completed, who is working on certain tasks, and which items have already been implemented. This provides a high level of clarity for the team and helps with self-organization, as there will be no grounds for confusion on current progress of backlog items. Another popular tool is Jira, which is similar to a radiator. It allows tracking for workflows and tasks in a similar way, but offers many other features that are beneficial in an Agile environment, such as the development of tickets to assign deadlines to particular items.

An Agile approach has many benefits. It allows for a flexible, adaptive approach to product development. Customer satisfaction is the priority, and the fast-paced nature of the environment allows teams to maintain knowledge about current trends and lead the charge for innovation in their field. It helps to develop cross-functional skills and challenges developers to become proficient in more than just a single area of expertise. Despite the benefits, there are some drawbacks. It is difficult to plan things out in Agile; the nature of the environment is to begin projects without a clear direction towards an end result. Therefore, there is no way to determine what the end result of the product should be. This is effective for certain projects, but there are others which need an extra layer of planning to be successful. I believe that the Agile approach was the best approach for the SNHU Travel project, as determined by the efficiency of changing requirements during the fifth week. Remaining adaptable made the shift to a different focus very easy and straightforward, regardless of how backlog items were deprioritized. The website needed to be up and running in a short amount of time, which is something that wouldn’t be possible in a waterfall-based approach.