Seth Sharp

Professor Hirpara

CS:250-Software Development Lifecycle

15 December 2024

***7-1 Final Project***

The Product Owner played an important role by prioritizing user stories in the product backlog based on business value, which ensured the team focused on the most significant features first. For instance, the mock-up email provided in Module 5 or 6, which was written from the perspective of the Product Owner to the Development Team, helped to streamline the process of improving the final product by directing focus to manageable changes that had clearly defined requirements. The Scrum Master role helped to maintain the smooth execution of Scrum ceremonies like daily stand-ups, sprint planning, and sprint retrospectives, facilitating open communication and keeping the team focused on goals. This was directly seen in the clarification of User Stories that made the development of key features possible. The team organized to assign tasks based on individual strengths and quickly resolved issues while minimizing time on product. One way this manifested was in the efficiency of the team in meeting goals without taking a lot of time to do so, such as the creation and extension of the Travel Deals slideshow within a few hours of receiving the specifications. In summary, the success of the SNHU Travel project was achieved through the coordinated efforts of the Product Owner, Scrum Master, and Development Team.

A Scrum-Agile approach to the software development lifecycle helps User Stories by promoting iterative development, fostering collaboration, and ensuring continuous improvement. This happens because during sprint planning the team reviews the product backlog and selects user stories to work on in the next sprint. This keeps the team focused on important and valuable features, such as the simulated meetings that offered insight into how features and changes should be prioritized during the SNHU Travel project. Daily stand-ups promote transparency and enable open communication that allows issues to be addressed promptly. For instance, if a developer is working on a user story and experiences a technical problem, then it can be addressed by the team during the stand-up to find a collaborative solution. Finally, sprint retrospectives offer valuable reflection on what went right, what went wrong, and where improvements can be made. This fosters an environment of continuous improvement by allowing the team to address group shortcomings and work on overcoming them. This was apparent during the group discussions in Module 7 in which the retrospective allowed me to identify areas in need of improvement, which in turn helped to communicate more effectively with group members on the best approach for future projects.

A Scrum-Agile approach supported project completion when the project was interrupted and changed direction because it is designed to handle changes and interruptions effectively. The interative nature of Scrum specifically adds flexibility in the development approach to accommodate changes and interruptions. For instance, when the project was changed from a webpage to a slideshow presentation it required a lot of reworks to accommodate, but by utilizing Scrum the team could quickly adjust to the change in the product backlog and modify sprint plans accordingly. Another important aspect of Scrum-Agile approaches is that they empower the team to make decisions without waiting for higher authority to approve. This empowerment allows the team to adapt swiftly to the new direction without waiting for top-down direction and approval. This ensures that the project continues to progress smoothly. To conclude, these practices ensure that teams can quickly adapt to new requirements, stay aligned with team members, and deliver valuable functionality despite interruptions.

“Since I chose the developer role, we will work closely, as you stated in your post. I agree with you that test-driven development and continuous integration/continuous delivery will benefit the team. However, I have limited experience and capabilities of implementing such an approach depending on project requirements going forward. I will adjust to this approach as best I can without sacrificing team productivity and efficiency. One approach could be a form of pair programming where you provide tests that need solutions, or I provide units that need tests. This approach is not traditional for the concept of pair programming, and it can be changed if the need arises, but it may provide the best method for establishing better team cohesion and communication moving forward.” (6-1 Student Discussion)

This sample of communication was effective in its context and encouraged collaboration among team members by identifying limitations of my abilities, while also providing solutions and requesting input from other members on how best to address issues as they arise. It was effective in its context because it identified my personal limitations in context of the team’s needs and future project requirements. This approach informs the team of development hurdles that might not be caught without a retrospective or daily stand-up, as that is how these personal limitations were identified. It encouraged collaboration by directly requesting input from team members for a resolution to the proposed issues, such as my ignorance regarding TDD possibly being mitigated by an unconventional approach to pair-programming that was stated to be adjustable to the needs and comfortability of other members. This approach allowed for the most efficient and open team communication that instigated cooperation and collaboration among its members.

The organizational tools and Scrum-Agile principles that helped the team be successful can be evaluated as follows. The use of JIRA for issue tracking and management allowed team members to create and track user stories, among other tasks. This visibility helped the team to stay informed and focused regarding progress and barriers to success. Specifically, JIRA helped during sprint planning to select and prioritize backlog items. Additionally, Brightspace facilitated the best communication by providing real-time communication channels. This allowed for the quick resolution of issues and supported the integration of JIRA to streamline workflows. These interacted in tandem with Scrum-Agile principles, such as continuous improvement and transparency, thereby filling a crucial role that led to the success of the team. This was evident in sprint retrospectives in which the team identified areas for improvement and developed plans to address them. This practice helped to optimize processes and enhance productivity. Continuing to sprint planning and daily stand-ups, these were crucial for maintaining open communication and aligning team priorities. These events ensured that team members were aware of the status of stories and functionality, as well as changing priorities and directions.

The goal of the project was to create a travel site for a company named SNHU Travel that allowed users to view and interact with a list of the top five travel destinations. The team consisted of developers, testers, a Product Owner, and a Scrum Master. A few pros of the Scrum-Agile approach were iterative development, flexibility, and collaboration. By delivering incremental updates the team gathered continuous feedback that allowed for better improvements. The daily stand-ups and sprint planning allowed for open communication which helped keep team members focused and aligned on stories and priorities. Finally, the Scrum framework allowed the team to adapt to changes more effectively and efficiently which facilitated better flexibility within the team. The cons include the communication overhead, learning curve, and dependency management. Due to the regularity of stand-ups, retrospectives, and reviews, there is a direct correlation to the time commitment of these events resulting in large communication overhead This can lead to less time for actual work and development which pushes back the product schedule. As for managing dependencies, this can be challenging with multiple teams working on multiple features. For example, delays in back-end development often lead to delayed work on the front end because of software and team dependencies. To end, team members can also experience a sharp learning curve when initially learning Scrum-Agile approaches. Training and adaptation certainly take time, so this plays into the drawbacks of implementing this approach. In conclusion, while the Scrum-Agile approach presented some challenges, it was highly effective for the SNHU Travel site. The iterative development ensured continuous improvements while the improved collaboration and flexibility allowed the team to adapt to changing requirements and improve communication. Despite the communication overhead, learning curve, and dependency management concerns detailed above, the benefits of enhanced flexibility, collaboration, and development outweigh the drawbacks.