Jacob Rethmeier

Professor Krupa

CS 250

10/20/2024

7-1 Chadatech

Throughout the CS250 course, I took on multiple roles within the Scrum-Agile team, and each role contributed significantly to the success of the project. The Scrum framework encourages team members to fulfill distinct roles to ensure transparency and collaboration across the team (Schwaber & Sutherland, 2017). As the Product Owner, I acted as the primary coordinator between the client (SNHU Travel) and the Development Team. My responsibility was to gather requirements, create user stories, prioritize the backlog, and ensure that the most valuable features were delivered first. As a Developer, my role required implementation of key features like the top destination’s carousel, ensuring the code met project requirements while collaborating closely with the Tester to maintain quality. As a Scrum Master, I facilitated daily standups, sprint planning, and sprint reviews, helping to remove obstacles and keep the team aligned with project goals. In the Tester role, I created and executed test cases to catch bugs early, ensuring the application’s functionality and reliability throughout development. This approach ensures that the team delivers value continuously rather than waiting for an entire product to be built, which aligns with Agile principles of incremental development (Beck et al., 2001). For example, I worked closely with the client to prioritize features such as a top destinations carousel and user profile management, helping the team focus on the most important aspects of the product early on.

When I assumed the role of Scrum Master, my role shifted to facilitating communication and maintaining transparency. I organized Sprint Planning sessions, daily standups, and Sprint Reviews to keep the team aligned and ensure that any issues or roadblocks were addressed quickly. According to Schwaber & Sutherland (2017), “the Scrum Master is responsible for promoting and supporting Scrum as defined in the Scrum Guide,” which means ensuring the team adheres to Agile principles and practices. The daily standups were particularly useful, as they allowed each team member to share their progress and any blockers they were facing. This helped create an environment of continuous feedback, allowing the team to make necessary adjustments in real-time.

The Scrum-Agile approach to completing user stories allowed us to break down complex tasks into manageable pieces, ensuring that the team could focus on specific features for each sprint. User stories are central to Agile as they define functionality from the perspective of the user, emphasizing the delivery of customer value (Cohn, 2004). By creating user stories for features such as the top destinations carousel and user profile creation, we were able to incrementally build out the SNHU Travel application while ensuring that each piece added measurable value. The iterative nature of Scrum also enabled us to gather feedback early and often, allowing us to refine features as needed throughout the development process. This approach helped the team remain flexible and focused on the end user’s needs. Breaking down the project into sprints allowed us to complete the user stories for each feature independently, which helped reduce complexity and allowed for continuous improvement. As Cohn (2004) notes, “user stories drive planning, estimating, and acceptance testing,” meaning they are vital in ensuring the product meets both the client’s expectations and technical requirements.

The flexibility of the Scrum-Agile approach was evident when we encountered interruptions and changes in project direction. According to a study by Serrador and Pinto (2015), Agile is effective in dynamic environments due to its ability to adapt to changes quickly and effectively. This was crucial when the SNHU Travel project shifted focus from general travel to wellness and detox travel packages. Instead of overhauling the entire project, we adjusted our backlog, revised user stories, and reprioritized our sprints to accommodate the new requirements. The Scrum framework’s adaptability allowed the team to maintain momentum while incorporating these changes without significant disruptions. This flexibility would have been difficult to manage in a traditional Waterfall approach, where changes late in the development process could cause significant delays and require reworking large portions of the project. In contrast, Agile’s iterative process allows for continuous refinement and adaptation, which was essential in ensuring that we met the client’s evolving needs.

Effective communication was a key component of our success. Scrum emphasizes transparency, inspection, and adaptation, which are supported by daily communication practices (Schwaber & Sutherland, 2017). For instance, I used daily standups to foster open communication and ensure that the team was aligned on progress and potential roadblocks. Additionally, regular email chains between team members, such as communication between the Product Owner and Tester, allowed us to clarify requirements and plan for upcoming test cases. According to a study by Drury, Conboy, and Power (2012), Agile teams benefit from increased collaboration and communication due to frequent interactions, which improves decision-making and problem-solving. Our team’s communication fostered an environment of collaboration, where each member felt comfortable raising concerns and discussing ideas. For example, when the wellness travel feature was introduced, the development and testing teams were able to collaborate effectively to incorporate the new requirements seamlessly. This constant feedback loop helped ensure that any issues were addressed quickly, improving the overall quality of the final product.

We leveraged organizational tools such as Jira and Azure DevOps to manage our sprints, backlog, and tasks. These tools helped the team visualize our progress and keep track of priorities, ensuring that nothing was overlooked. Jira allowed us to break down user stories into tasks and manage them across sprints, while Azure DevOps supported code integration and version control. Using these tools, we maintained transparency and stayed aligned on project goals, even when working remotely. According to Bass, Haxby, and Moe (2018), “tools such as Jira facilitate collaboration by providing a centralized platform where teams can organize work and communicate progress.” These tools were essential in keeping our team organized, especially when balancing multiple roles and responsibilities across sprints. They also allowed us to visualize project timelines and adjust priorities based on the evolving needs of the client.

The Scrum-Agile approach proved to be highly effective in the SNHU Travel project, though it came with both advantages and challenges. One of the key benefits was its flexibility, which allowed us to respond quickly to changes in requirements and maintain frequent communication with the client. This adaptability is one of the core strengths of Agile methodologies, enabling teams to deliver high-quality products that meet customer expectations (Beck et al., 2001). However, one potential downside of Agile is the risk of scope creep, as frequent changes to the project can lead to an expanding scope and delayed timelines. This was mitigated by maintaining a well-organized backlog and holding regular sprint reviews to ensure that the team stayed on track.

I personally believe the Scrum-Agile approach was the best choice for the SNHU Travel project because it provided the flexibility needed to adapt to evolving client requirements while ensuring the team stayed focused on delivering a high-quality product. The iterative nature of Agile allowed for continuous improvement and regular feedback, which helped us refine the product with each sprint. This engagement with the client throughout the development process ensured that the final product closely aligned with their changing expectations. Agile’s emphasis on collaboration fostered open communication among team members, which made it easier to identify and address potential issues early, keeping the project on track. Although Agile’s flexibility carries some risks, such as scope creep, we were able to mitigate these risks by maintaining a well-organized backlog and prioritizing key tasks. In the end, the adaptability, increased collaboration, and regular feedback loops inherent in Agile allowed us to create a more refined and valuable product, far outweighing any challenges that arose during development. This demonstrates the effectiveness of Agile in driving successful outcomes in dynamic and evolving project environments.

Works Cited

Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., … Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved from https://agilemanifesto.org/

Bass, J. M., Haxby, A., & Moe, N. B. (2018). Agile project management and its role in knowledge transfer. Journal of Software: Evolution and Process, 30(4), e1915.

Cohn, M. (2004). User Stories Applied: For Agile Software Development. Addison-Wesley Professional.

Drury, M., Conboy, K., & Power, K. (2012). Obstacles to decision making in Agile software development teams. Journal of Systems and Software, 85(6), 1239-1254.

Schwaber, K., & Sutherland, J. (2017). The Scrum Guide. Retrieved from https://scrumguides.org

Serrador, P., & Pinto, J. K. (2015). Does Agile work? A quantitative analysis of Agile project success. International Journal of Project Management, 33(5), 1040-1051.