Software Development Life Cycle: Sprint Retrospective

Matthew Dunfee

Department of Computer Science, Southern New Hampshire University

CS-250-12948-M01: SDLC

Tammy Morrison

December 6, 2024

**Sprint Review and Retrospective**

**Over the course of this project, I gained valuable insights into the Scrum-Agile framework by actively assuming the roles of Scrum Master, Product Owner, and Developer. This experience allowed me to explore various tools, communication methods, and common scenarios like workflow interruptions. I will now assess the effectiveness of the Scrum-Agile approach for the SNHU Travel development project.**

**Applying Roles**

**Each role in a Scrum-Agile team plays a critical part in the success of a project:**

* **Product Owner: Responsibilities such as requirements analysis and organizing the product backlog are foundational to the project. These steps guide the team toward deliverables aligned with stakeholder expectations. For example, during the requirements analysis phase, I opted for a modern web application design, which, while practical, conflicted with the rigid confines of the assignment. This misstep demonstrated the importance of aligning broader ideas with the project’s specific scope.**
* **Scrum Master: Contrary to its title, this role focuses on servant leadership—ensuring team cohesion, planning events, and removing blockers. The Scrum Master fosters team values like bravery, openness, and respect, which are crucial to Agile success (Schwaber & Sutherland, 2020). For example, after the backlog was established, a client requested modifications. The Scrum Master’s intervention helped the team adjust while maintaining focus and morale.**
* **Developer: This role is the most straightforward, involving tasks such as meeting the Definition of Done, participating in standups, and ensuring deliverables are transparent and scalable. For example, I realized that failing to structure data dynamically limited scalability and maintainability in the final application, which could have been avoided with better planning.**
* **Tester: The Tester role is critical for ensuring that the deliverables meet quality standards and align with user expectations. This involves validating the application against user stories and identifying gaps or discrepancies in functionality. For instance, during the project, discrepancies between the wireframe and user stories were discovered. The Tester paused their current tasks to notify the team, preventing unnecessary work and ensuring alignment with project goals. This role emphasizes precision and collaboration, as testing provides crucial feedback to guide development and maintain quality.**

**Completing User Stories**

**User stories are one of the most powerful tools in the Agile SDLC, as they translate user expectations into objective deliverables. They focus on end-to-end behavior rather than specific functionalities, empowering developers to build logic that supports scalability and dynamic content. For instance, rather than hardcoding display cards on the website, I could have created objects and iterated over them dynamically. This approach would have ensured scalability and ease of maintenance.**

**Handling Interruptions**

**Agile thrives in handling interruptions, as its iterative nature is built for unexpected changes. Two key mantras of Agile—“fail early, fail often” and “just barely good enough”—underscore its lean philosophy (Beck et al., 2001). These principles minimize waste and allow teams to adapt quickly. In this project, Agile facilitated rapid adjustments when user stories and wireframes revealed discrepancies. This responsiveness prevented unnecessary work and redirected focus efficiently.**

**Communication**

**Effective communication is crucial for precise teamwork. Agile emphasizes this through tools like user stories, test cases, and wireframes. However, as tasks grow complex, tracking systems like Jira become indispensable. For example, while preparing test cases, comparing user stories with the wireframe revealed major discrepancies. The tester paused their task to alert the team, preventing wasted effort. This scenario highlights the importance of open communication and the ability to adapt workflows based on new findings.**

**Evaluating Agile Process**

**The Scrum-Agile approach presented clear advantages and disadvantages during the SNHU Travel project:**

* **Pros: The methodology’s adaptability, responsiveness, and customer-first focus were assets. Agile allowed the team to iteratively refine requirements and adapt to changes.**
* **Cons: For a project of this size and simplicity, Agile introduced unnecessary overhead. Agile’s "just barely good enough" principle was compromised, as the static nature of the application could have been addressed with a less resource-intensive approach. In this context, the methodology's heavy structure felt misaligned with the project's straightforward requirements.**

**While Agile is a powerful framework, applying it to this specific project felt akin to using a flathead screwdriver on a Phillips screw—possible, but not optimal. For future projects of this scale, simpler methodologies may be more efficient.**

**References**

* **Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). Manifesto for Agile software development. Agile Alliance. Retrieved from** [**http://agilemanifesto.org**](http://agilemanifesto.org)
* **Schwaber, K., & Sutherland, J. (2020). The Scrum guide. Scrum.org. Retrieved from** [**https://scrumguides.org**](https://scrumguides.org)