**A logo for a computer

Description automatically generated**

# Software Development Life Cycle Report

# Sprint Review and Retrospective

# CS-250-14870-M01

# Daniel Gorelkin

Table of Contents

[Document Revision History 2](#_Toc102040754)

[1. Applying Roles 3](#_Toc102040758)

[2. Completing User Stories 4](#_Toc102040759)

[3. Handling Interruptions 4](#_Toc102040760)

[4. Communication 5](#_Toc102040761)

[5. Organizational Tools 6](#_Toc102040762)

[6. Evaluating Agile Process 7](#_Toc102040763)

[7. References 8](#_Toc102040763)

Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **Nov 25, 2024** | **Daniel Gorelkin** |  |

# Sprint Review and Retrospective

## Applying Roles

In our recent project, the success of each sprint and the final delivery relied heavily on the distinct contributions of each Scrum-Agile team role. This case study highlights how the efforts of the Product Owner, Scrum Master, and Development Team were coordinated to achieve optimal outcomes. Each Scrum-Agile role was instrumental in transforming stakeholder needs into a polished deliverable. During each Sprint, my project owner refined the backlog by regularly engaging with stakeholders, getting insights, feedback, and constructive suggestions from random users, and ultimately building user stories that helped prevent any shift from the project's primary goals. That was done by capturing the client’s and user’s needs and translating them into concise and clear instructions. For instance, through techniques like the "Five Whys" (Hancock, 2022), the Product Owner drilled down to distinguish real needs from "wants," ensuring that the team focused on high-value features instead of expendable extras. Those meetings and constant communication helped to gain trust within the team and the stakeholders while at the same time, clearly setting priorities and significantly reducing the project uncertainty and diminishing risks.

My role in the project as the Scrum master focused on closing the gaps between the Product Owner and the developers and making sure my team had all the requirements they needed to thrive and deliver the job on time. This included facilitating Sprint Planning, where we defined clear goals and identified any potential roadblocks before work began. During Daily Scrums, I stressed concise, focused updates and supported the team by resolving issues as they emerged, such as providing them with satisfying equipment and removing distortion. Similarly, I managed the Kanban board to provide a visual overview of progress, enhancing transparency and keeping the team aligned on daily targets while conducting the daily ceremony, such as asking key questions such as how likely are we to achieve the Sprint Goal? Or what was accomplished yesterday? Beyond daily interactions, I led Backlog Refinement sessions to assist my Product Owner and reprioritize tasks to ensure the team had a realistic view of the sprint goal. At the end of each sprint, I conducted retrospectives to discuss what worked well and identify improvement areas, setting a learning foundation that prevented issues from repeating in future sprints. In addition, I found myself helping my developers who were stuck with their tasks and assignments and could put the Sprint at risk. This was addressed by suggesting techniques for task prioritization and scrambling workload to minimize the risk of the Sprint running late.

Last but not least, the development team played no less critical role in the Scrum. Their focus was on creating high-quality increments with every Sprint. The team played a pivotal role in our daily Scrum meetings and helped in most Sprint Plannings. Following the Product Owner’s user stories, the team worked on the highest-priority tasks with efficiency and creativity. During each Sprint Review, the Development Team shared their completed features with me and showcased their progress with the Product Owner for validation and client approval to ensure alignment with the client’s and user’s vision. Moreover, during each sprint review, the team actively contributed to discussions and the Sprint backlog evaluation, offering improvements on process adjustments and skill levels. Ultimately, following the Scrum framework model, the collaborative environment allowed each team member to share and gain knowledge or resolve technical challenges by supporting each other and the team as a whole. The synergy of these roles leads to the project’s success, benefiting the strength of the Scrum-Agile framework in tackling complexity while delivering value efficiently due to each member’s contributions being precise and targeted, especially during tasks requiring teamwork to resolve issues quickly.

## Completing User Stories

## The Scrum-Agile approach supports breaking down a large and complex job into smaller micro-projects called Sprints and completing the project in iterative increments, where new or updated user stories emerge between each increment. The goal of a practical Agile approach is to categorize each task (user story) by its value and complexity and define its Definition of Done (DOD). Hence, during every Scrum planning, the top valuable stories are chosen to be handled during the coming Sprint. Throughout the Sprint, on the daily Scrums and next to its completion, and after all tests are completed and the DoD is marked as done, the team meets to present the increment to the stakeholders. The goal of that checkpoint is to deliver a partial piece of working software to the client and get his approval and feedback that will guide the team on the next Sprint cycle. For example, in our initial meeting with our client, SNHU Travel, we were asked to build a platform that expands the company’s customer base and sells niche travel packages. Next, during our research meeting with some users, we concluded that our users would get the best value from getting the best vacation deals based on their personal user profiles. Therefore, during our Sprint planning meeting, the priorities were set, the DoD was established, and the least valuable user stories were shifted down the queue. In such a manner, we completed one Sprint after another and eventually delivered fully functional software that met our client and user’s satisfaction on time.

## Handling Interruptions

“People change their minds for many reasons, and do so on a regular basis” (The Agile Modeling Method, 2023b). Thus, Agile welcomes changes and allows the team to be more flexible in the approach to development due to an iterative and incremental delivery that could be updated between the Sprints by shifting requirements up and down the scale, even during an advanced development stage by resetting requirements and shifting priorities and complexity of the updated user stories. In the Agile SNHU Travel project in which I took part, my team faced an unpredicted change in direction in the middle of the development process as our stakeholder requested to shift the focus from a niche touring packages platform into a detox/wellness travel direction development. Unlike the Waterfall model, the Agile methodology helped us overcome the unforeseen easily. Instead of scraping all the work that was already marked as done and starting the project from scratch, we managed to modify the project to the new stakeholder requirements by rearranging the Product Backlog, re-evaluating the values of the user stories, and setting new goals for the coming Sprint. The key to our success was the team’s communication and collaboration; during our Sprint planning meeting, we used the Planning Poker Cards technique to evaluate the weight of the new stories and decide which ones were the most valuable and could be completed in the Sprint. Hence, the collaboration of the team following the Agile methodology placed us back on track without leading to financial loss, stakeholder frustration, or delivery delays.

## Communication

As mentioned above, our key to success was great communication within the team, where every single team player knew what to do and contributed to the project. For example, I maintained confident, effective, and engaging communication by clearly setting timelines, requesting clarifications, appointing potential obstacles, and driving my team members to action. The following is a sample of an email sent to my PO and the Scrum master just after our project changed its development direction:

A screenshot of a computer

Description automatically generated

As can be seen in the example above, the request is informative, concise, and engaging. It calls to action and motivates the team to work cooperatively towards the common goal of delivering a completed increment on time and achieving stakeholder satisfaction. Another way of effectively communicating with my team was by using modern technological tools such as JIRA.

A screenshot of a computer program

Description automatically generated

The JIRA tool encouraged collaboration and called for immediate action and response from the whole team. It helped my team stay on track by sharing everyone’s progress, defining timelines, and managing dependencies openly and effectively.

## Organizational Tools

Our team succeeded by using modern tools and following openness and transparency. Combining the two allowed us to stay on track and communicate the information in a valuable way. Our team adopted the Scrum-Agile principles, such as frequently delivering valuable increments after each Sprint session to the stakeholder's satisfaction, by being flexible and adapting to changing requirements, even in advanced stages of the project, for example, when stakeholder requested us to shift the project towards detox/wellness travel development. Our success was achieved by swiftly adopting and changing the project's direction to satisfy the new requirements and collaborating with the client instead of negotiating the contract. Therefore, we delivered functional software that greatly benefited our client, SNHU Travel. However, as suggested in Agile Manifesto, “Individuals and interactions over processes and tools.” (Agile Alliance, 2023), we maintained valuable interactions and communication through regular face-to-face meetings while using radiator boards to present and keep track of the user stories, the running Sprints, and the overall progress of every member of the team.

In addition, without contradicting the Agile Manifesto, we used modern technological tools such as JIRA, which helped the team effectively manage the project's progress and increase efficiency. JIRA also helped us create a well-defined sprint planning by choosing and time framing the selected stories while considering their story points to fit the sprint timeframe. Not less, it helped the team to stay updated while allowing each team member to process the stories individually, mark them as done, or share potential issues that emerged and could affect the Sprint increment. JIRA boards were also an effective tool in our sprint retrospectives. They provided feedback on the team's efficiency and progress by providing visual metrics that helped us discuss the obstacles and remove them in the next Sprint, which led us to improve and evolve as a team.

## Evaluating Agile Process

In conclusion, the Scrum-Agile approach was appropriate for the SNHU Travel development project. The Scrum-Agile approach is the best fit for projects that require flexibility, frequent customer feedback, requirement change, and rapid adaptation, while the initial requirements contain high uncertainty. Some of the strengths of the Scrum-Agile approach are faster delivery of valuable increments to the stakeholder, high capability of flexibility and adaptability, and handling situations where requirements are unclear or likely to change. As has been seen in the SNHU project, which started as developing a platform to tailor customized travel packages based on popularity and user profile and later shifted its focus to detox/wellness travel packages following a waterfall development model, could lead the team to a failure due to the disability to modify the project in an advanced development stage after investing significant efforts and time in developing and documenting the requirements. However, the Scrum-Agile approach is not the optimum approach to all development requests and has weaknesses. For example, despite the existence of technological tools like JIRA, communication, and collaboration can become more complex with distributed teams, making it challenging to maintain Scrum principles, and the methodology highly relies on the team skills and proficiency of the team members. In addition, in projects with high predictability and very well-defined and fixed requirements, conducting the Agile ceremonies and investing time in meetings instead of rich documentation and development could be a burden and decrease the team’s performance. Another drawback of Agile compared to a traditional software development life cycle (SDLC) is the project delivery due date and the development potential costs, making it hard to predict and commit upfront for stakeholders who want concrete deadlines and costs.

Therefore, the Scrum-Agile methodology was the best fit because our project had only a high vision overview and unclear requirements with a high risk that those could change throughout the SDLC.

**References:**

Agile Alliance. (2023, October 18). *Agile Manifesto for Software Development | Agile Alliance*. Agile Alliance |. <https://www.agilealliance.org/agile101/the-agile-manifesto/>

Atlassian. (n.d.). Jira Software - Features | Atlassian. <https://www.atlassian.com/software/jira/features>

Charles G. Cobb. (2015). The Project Manager’s Guide to Mastering Agile : Principles and Practices for an Adaptive Approach. Wiley. <https://eds.p.ebscohost.com/eds/ebookviewer/ebook/bmxlYmtfXzkzNzAwOV9fQU41?sid=6ab2f8ad-3b31-43da-9da1-97edef7d8de3@redis&vid=3&format=EB&rid=1&runquerystringmethod=eBookDownloadundefined_click_bookcheckouts%3acheckBookAvailability%3aready&runquerystringmethod=eBookDownloadundefined_click_bookcheckouts%3acheckBookAvailability%3aready>

Hancock, J. (2022, July 1). MindTools | Home. <https://www.mindtools.com/a3mi00v/5-whys>

Sliger, M. (2012). Agile estimation techniques. Paper presented at PMI® Global Congress 2012—North America, Vancouver, British Columbia, Canada. Newtown Square, PA: Project Management Institute.

The Agile Modeling (AM) Method. (2023b, November 23). *Agile Requirements Change Management*. The Agile Modeling (AM) Method - Effective Strategies for Modeling and Documentation. <https://agilemodeling.com/essays/changeManagement.htm#PrioritizingRequirements>