Project 7 Sprint Review and Retrospective

Areesha Tariq

6/22/2025

As Scrum Master for the SNHU Travel application development team, I facilitated my team through the Scrum-Agile processes mapping various software development lifecycle (SDLC) project phases into an iterations format. This retrospective outlines how each of us contributed to the successful completion of the project in for each role, how we closed user stories, how we adapted to interruptions, how we maintained effective communication, and how we employed Agile tools. Most importantly, it will analyze Scrum-Agile's overall effectiveness, for a software project such as this.

Every Scrum role added their own perspective and effort to contributing towards the success of the project at hand. As product owner, I was kept responsible for reprioritizing how the product backlog was organized based on the stakeholder needs I gathered. An example of this is when the client came to us asking to integrate a search function for vacation packages. I added it to the product backlog and moved it to the top for the next sprint for the development team to work on.

The Scrum Master (my last role) kept the development team engaged and removed any blockers as they arose. During Sprint 2 the team struggled to get a clear idea of how the booking API was to actually be implemented. I organized a meeting with the developers to lay out and clarify the requirements again, finally resuming development.

At an individual level as a developer, I worked alongside the development team to deliver software increments that demonstrated working features of the product at hand. As an example, in Sprint 1, I helped with designing the UI for selecting a destination for the vacation package, while making sure to focus on accessibility and responsiveness across different devices. As a team, we practiced pair programming and had code reviews to keep quality up to standard.

The Scrum-Agile framework promoted our ongoing work on user stories through the use of iterations, feedback, and working as a team. At the beginning of our sprints, we held planning meetings where we would select user stories from the backlog. An example of a user story we planned was to implement a function where users could filter travel packages by price range. We would break the user story into several tasks: Creating the UI, writing filtering logic on the backend (which might include additional filtering), and testing.

Daily standups were utilized to help us be aware of our progress and quickly resolve blockers when they arose. For instance, when it was discovered that a team member was delayed as the client had not provided complete data, we adjusted the sprint plan and changed the focus to the stories that could be completed as required. The flexibility of Agile facilitated ongoing value being delivered, without risk to quality or timelines.

Agile-Scrum's greatest advantage became clear when the client changed direction while we were developing the app. Initially, SNHU Travel asked us to create a desktop-first web app. However, during Sprint 3, we received feedback from the client about the need to prioritize mobile responsiveness because they were already getting an increasing number of users on mobile.

With Agile, we were able to easily pivot. For our next Sprint Planning, we updated our Definition of Done, amended tasks, and then communicated how we were going to give mobile compatibility priority over desktop compatibility. These kinds of changes would be exceptionally difficult in Waterfall as late changes are expensive. The best part about Agile is that it is increment-based. We can reprioritize our work without throwing the project into disarray.

The effective communication of the team was a major factor to being successful in the project. We were able to adopt funcational and simulated tools such as a shared project board, and team chat to guide the coordination of our work. As Scrum Master, I made it a goal of mine to frame participants and foster open conversation during our retrospectives through guided discussion questions; "What went well, and what can we improve next sprint?".

When developers had a misunderstanding regarding testing, it resulted in duplicate work and a lack of clear communication. I coordinated a meeting that allowed each developer to explain their process and contributed to the team’s testing checklist, which facilitated collaboration and ensured that duplicate work of miscommunications would not exist. These communication habits allowed effective routines to form, and establish a productive, group-centered environment.

Organizational tools and Scrum ceremonies played a significant role in our success. We used a digital Kanban board to track sprint tasks which visually conveyed our progress and easily allowed us to re-prioritize tasks. Backlog grooming kept our work organized, and sprint reviews allowed for us to demo features and receive simulated feedback from stakeholders.

Retrospectives allowed us time to reflect. After Sprint 1, we realized our tasks were larger than we expected and broke our work into better tasks in future sprints. Breaking down our work improved velocity and reduced developmental blockers mid-sprint. The rhythm of the Scrum events made our gap feel manageable, transparent and adaptive.

Overall, the Scrum-Agile approach was a good fit for the SNHU Travel project. Its benefits included adaptability, continuous feedback, team morale, and faster delivery of functional features. It fostered a focus on transparency and collaboration and rapid iteration.

Yet, there were drawbacks. Agile is challenging in that it required solid time management and frequent check-ins from team members, which were demanding. There were feedback loops that we simulated because we did not have a real client, which affected the realism.

Though there were limitations in this project, Agile still was the right approach. If the project had followed a Waterfall path, this project would not have benefited from feedback until late in developmental cycle. Collectively responding to changes like mobile-app first requirements would have been challenging without Agile. Scrum was well suited for the dynamic environment to build a user-centered application with minimized bureaucracy and more agility to promote iterative development.

The SNHU Travel project shows that using Agile roles, principles, and effective ceremonies helped to develop functional software. The teams Amelia would stay adaptable, communicate often, and use the Agile ceremonies with the intention to provide value each sprint. This retrospective provided some insights into how Agile can support continuous improvement that can be leveraged to help with a wider adoption of the Scrum framework at ChadaTech.