CS250

Professor Vianelis Martinez

Jermaic Fuentes

August 17, 2025  
 Sprint Review and Retrospective  
 As part of Chada Tech's pilot transition to Scrum-Agile, our team developed the SNHU Travel application. Throughout this process, each team member contributed to different Scrum roles, including Scrum Master, Product Owner, Developer, and Tester. This retrospective reflects on how our roles, user stories, communication practices, organizational tools, and adaptability contributed to the project’s success, and evaluates the effectiveness of Scrum-Agile for this type of work. There were several roles within the scrum team, and the strength of our Scrum-Agile approach came from each role’s unique contributions.  
 As Scrum Master my focus was on facilitating Scrum events and keeping the team aligned. Daily Scrums promoted accountability and quick problem resolution, while backlog refinement avoided confusion at sprint planning. My leadership role was about serving the team ensuring transparency and guiding us toward continuous improvement.

Our Product Owner served as the bridge between users and the development team. By gathering insights from focus groups and translating them into actionable user stories they ensured that development aligned with actual customer needs, such as the development of filters for travelers looking for a trip within their budget or interest. The product owner also worked closely with the testers and scrum master to bring clarity on key functions of the product and how they should behave.

Our developer translated stories into working software increments. They also emphasized the importance of retrospective and communication, they reinforced that retrospectives gave the team opportunities to adapt and improve practices, while tools like Jira helped visualize progress and reduce miscommunication. Our developers also teamed up with testers during the development to create accurate and improved functionality in the app.

Our tester ensured that acceptance criteria were translated into test cases that reflected real user expectations. By collaborating closely with the Product Owner, they clarified ambiguous requirements and validated functionality through early involvement demonstrating the principle of “shift-left testing”. Our tester was also able to work closely with developers to validate functions the developers created. Each role reinforced the others, creating a cycle of feedback, adaptation, and improved delivery.

Scrum enabled us to break down user needs into small, manageable increments. User stories became the key to our project success. For example, the Product Owner ensured clarity by framing needs in customer terms, such as “As a traveler, I want to set a price limit so that I only see packages I can afford.” This gave developers and testers a clear target for design and validation. The Tester pointed out that acceptance criteria guided the design of test cases, this guaranteed that each user story met both functional requirements and user expectations. The Scrum Master supported backlog refinement so that stories were always sprint-ready, avoiding delays during planning. This iterative approach ensured user stories were not just completed but delivered with measurable value.  
 Our primary interruption came from a change in display requirements for the “Top 10 Destinations” feature. Initially, this functionality was envisioned as a list, but later feedback suggested it should be presented as a slideshow to create a more engaging user experience. This ambiguity surfaced during testing when acceptance criteria did not specify the format. The Product Tester identified the gap and reached out to the Product Owner via email for clarification, by requesting wireframes and additional details.

In a traditional waterfall model, this change would have caused significant delays because requirements are locked early in the process. However, Scrum allowed us to adjust quickly. The Product Owner updated the backlog, the Scrum Master facilitated discussion to re-align sprint priorities, and the development team adapted their tasks accordingly. The open communication between tester and Product Owner ensured that the final solution matched stakeholder expectations. This interruption ultimately became an example of how Agile supports responsiveness and collaboration, turning potential setbacks into opportunities for improvement.  
It also demonstrated how communication is the source of Agile Daily Scrums, allowed developers to quickly surface blockers, and keep the team aligned. Tester-PO communication (via email) resolved ambiguities such as whether price filters should allow ranges, or the format of the list, this ensured features met user and stake holder's expectations. Sprint Reviews provided opportunities to showcase increments and gather client feedback, such as adding various filters for the top 10 locations and budget limits. The frequent exchange promoted trust, reduced misunderstandings, and improved delivery speed.  
 Along with our efforts to communicate we had organizational tools like the Jira board which further strengthened communication by making progress transparent to everyone and it centralized the backlog and sprint tracker. Another tool utilized was agile framework tools such as sprint planning, daily scrums, sprint reviews and retrospectives. All enabled adaptation, this helped us set goals, keep us aligned and on track, it even took in input from owners or stake holders. So, these tools enhanced transparency, accountability, and adaptability.  
 For a project like SNHU Travel where requirements evolved and client feedback was essential, Scrum-Agile was clearly the best fit. The pros allowed us to adapt to changes and keep the whole team informed. We were able to receive continuous feedback which improved the development process. And it promoted communication and transparency across roles which built trust and accountability. All of this allowed us to deliver working software continuously, engage stakeholders throughout, and respond effectively to change. On the Con side no one is perfect and sometimes we can leave things out such as when user stories sometimes lacked detail, or we required additional clarification from stakeholders. If there are new team members they also must learn and adapt to the agile framework and that could pose a challenge at first as well.