Alex Akers

CS250

8/14/2025

**Sprint Review & Retrospective — SNHU Travel**

Throughout the development of the SNHU Travel program, Scrum and the agile mindset allowed us to create the SNHU travel app effectively. We started with a simple team agreement about roles, success criteria, and how we’d work day‑to‑day. This sprint review and retrospective sums up what each role brought to the table, how we finished user stories, how we handled change, and which tools and events helped us move fast without getting sloppy.

Applying Roles

As a team, we stuck to clear, lightweight roles. The Product Owner kept the backlog pointed at real value and was available for quick clarifications. When I took the role of Scrum Master, my job was to protect the timeboxes, keep the conversations focused, and remove the little frictions that slow a sprint down. Development owned vertical slices instead of half‑features, and testing translated acceptance criteria into concrete checks we could all see.

Completing User Stories

User stories finished because we kept them small, and because our Definition of Done wasn’t just a checkbox. A typical flow looked like this: we broke an epic down, wrote acceptance criteria that spelled out behavior and edge cases, and made sure dev and test agreed before anything started. Testing planned for empty results and weird inputs up front, not at the end. We only called it done after it was demoed working, tests were green, and notes were updated. It wasn’t fancy, but it kept stories moving all the way to “Done” instead of camping in “Almost.”

Handling Interruptions

Midway through development, we had a pivot toward wellness/detox travel. In a waterfall world, that kind of change hurts. With Scrum, the Product Owner re‑ordered the backlog, we estimated the impact, and we traded a lower‑value item out to hold the sprint goal. Because the acceptance criteria and test cases lived with the story, test could adjust right away and development wasn’t guessing.

Communication

The communication that actually helped was short, specific, and attached to the work. I like posting updates that call out what moved and what’s in the way, not long essays. Here’s the kind of thing I think I’d put on a story: “Yesterday: wired budget/date filters. Today: hook filters to search query”. That’s enough detail for teammates to jump in without a meeting.

When priorities change, I try to be direct and polite with what I need. For the wellness pivot, we drafted a clear note to the Product Owner and Tester laying out the specifics I needed to move forward (destinations/types to showcase, branding direction, and matching test cases) in our journal. Keeping it short and sweet sped up decisions and avoided back‑and‑forth. I also found that asking about the behavior of “no results” (like when filters are too strict) would have saved time later. The pattern is simple in my opinion: put the question where the story lives, give an example, and ask for a small, specific decision.

Email to Product Owner and Tester

Subject: Info Needed to Continue Development

Hi Christy and Brian,

With the new plan to focus on detox and wellness travel, I want to make sure I have everything I need to move forward on development. Christy, could you send a short list of the destinations or types of wellness trips you’d like highlighted in the slideshow?

Brian, once Christy provides those details, can you share the updated acceptance criteria or test cases? I just want to be sure the new slides and descriptions line up with what you’ll be testing for.

Thanks! – Alex

Why I think this worked: it lived with the work item, it made the decision small, and it was written so either role could respond quickly. I feel it encouraged collaboration without pulling the whole team into a meeting they didn’t need. The same idea applied to test comments on pull requests, calling out which acceptance criteria were covered and which edge cases were exercised, so devs had a clear target and testers could extend coverage.

Organizational Tools & Scrum Events

The tools mattered because they supported, not replaced, the conversations. We used a simple board to make the flow visible, and kept WIP reasonable. Backlog refinement mid‑sprint paid off during planning—stories were smaller, better named, and already estimated. Sprint reviews were for demoing real increments and gathering feedback, not slides. Retros were where we tried tiny process experiments: pair test with dev earlier, tighten the Definition of Done, and capture empty‑state behaviors as acceptance criteria. A lightweight test case template plus Definition of Done acted like a quality gate without slowing us down.

Evaluating the Agile Process

Pros: Fast feedback from stakeholders helped us tune the user experience (clearer date filters, better empty states) before we built too much of the wrong thing. Vertical slices exposed integration risks early. Regular retros made continuous improvement normal instead of a once‑a‑semester activity. Most importantly, we could absorb the wellness pivot without derailing the sprint goal.

Cons: Early on, with fuzzy requirements and a brand‑new domain, I feel that we spent more time in backlog refinement than we planned. Estimations moved around until we built a shared feel for scope. And with a small team, ceremony can feel heavy if you don’t timebox hard—so we kept standups to fifteen minutes and parked deep dives for after.

Was Scrum the right fit for SNHU Travel? Yes. The work needed quick validation and had changing priorities. A phase‑gated model would likely have delayed feedback and increased rework. Scrum gave us a steady cadence to ship, learn, and adjust. If we were doing a very fixed‑scope, compliance‑heavy build with few changes expected, I might pick a different approach. But for this project, the iterative loop—plan a little, build a little, show it, adjust—was exactly what we needed.

In short, roles were clear, stories stayed small, change was handled through the backlog, and our communication lived where the work lived. The tools and events gave us just enough structure to move quickly without losing quality. That’s what made this sprint cycle feel calm even when things changed in my opinion: everyone knew the goal, the next step was visible, and the feedback loop was tight.