**CS-250**  
**Retrospective – Week 7**  
**Daniel Loranger**

Our team has now completed five weeks of sprints. Each team member has taken to the new the Agile-Scrum process we set out to pilot. It took some getting used to at first, but over time we found our rhythm and established a predictable velocity towards the end which will enable us to better maintain sustainable development standards. One event in this project that showcased for me was how well this development process helped the team was in handling change and still delivering progress each sprint. I’m convinced that continuing this process will help the SNHU-Travel team continue to adapt quickly and produce successful products going forward.

As we moved through development, I watched the coding team take user stories such as “Top 5 destinations” and turn it into a working feature. The team continued to make progress, but also stayed flexible whenever the product manager updated the requirements based on new market feedback. One clear example was when we built the first prototype without knowing exactly what kind of trip type or presentation format the final product would need. Even with those unknowns, we still managed to get a working version out quickly that showed our direction and helped everyone visualize what was envisioned for the final product.

Once that first prototype was ready, the testers started building their test scenarios and automated regression tests. The early tests demonstrated how valuable it was to see test results right away, developers could instantly tell if a change broke something or if everything still worked as intended. It made it much easier to track which stories were truly “done” and ready to show at the end of each sprint.

Early in the project, our product owner interviewed potential users and turned their input into short, focused user stories. I liked how this kept everyone on the same page. Instead of long requirement documents, we worked from concise goals written in plain language. It made the project feel more connected to what users actually wanted instead of just following technical checklists. This allowed the developers to think creatively and provide code optimized to the user need, not the written requirement (which might be wrongly translated).

When we shared our first working prototype, the product owner came back with new market feedback that changed the focus. We switched from general travel planning to wellness vacations and changed the presentation from a roller frame to a slideshow format. Normally, that kind of mid-project shift could cause chaos as detailed work would be underway already, but using a minimum functioning code format, the impact was minimal. Because we were using Agile-Scrum, we could adjust our backlog, plan for the change in the next sprint, and keep going without major delays. Within just two weeks, we had a new prototype that matched the new direction. This new working model would have taken much longer in a waterfall process.

If this had been a traditional waterfall project, those changes could have meant a huge setback; throwing out work, redoing plans, and losing weeks or even months. Instead, Agile helped us pivot quickly with minimal impact. That flexibility really demonstrated the value of the Scrum approach and the principle of embracing change.

As Scrum Master, I made sure we held daily stand-ups where everyone gave a quick update, talked about challenges, and offered help when needed. These short meetings made a big difference in the dynamic of the team. They built trust within the team as everyone knew where everyone was in their tasks and helped me stay aware of any obstacles before they grew into bigger issues. Our team used JIRA to track our sprint tasks, burndown charts, and team velocity. Having those tools visible helped keep everyone on task and accountable without feeling called-out.

At the end of each sprint, we held a sprint review and retrospective event. These events allowed an opportunity for reflection because they helped the team see where we succeeded and where we could individually and collectively improve. Over time, we got better at estimating our workloads and selecting stories that fit comfortably within the sprint and our individual capacities. This ultimately will make our team more efficient and more predictable.

Looking back, I think the biggest takeaway from this experience is how effective Agile-Scrum can be for short projects. The sprint planning and retrospectives gave us a clear achievable goal framework to work in and kept everyone focused on delivering usable results every week. The clear impact demonstrated in this project is when a project has a short timeline and evolving requirements, these short, time-boxed cycles make it much easier to stay on track and respond to change before it becomes a problem. Each sprint allowed us to test ideas, get feedback, and make adjustments based on fast feedback. From these micro-development sprints, we created a steady flow of progress. Even when guidance was changed, the feedback loops were quick allowing us to adapt quickly.

From this pilot effort, it has demonstrated to me that Agile-Scrum was the right approach for this project and an effective process once well understood by the entire team. It can take a while to adapt to, but with incomplete requirements at the start, and a timeline that was laughable, Agile was demonstrably a perfect fit. If we had used a waterfall model, I am confident the team would have run into delays and rework. Instead, we finished on time with a working product that met the customer’s changing needs.