**CS 250 Final Project: Sprint Review and Retrospective**

This retrospective reflects on my learning and participation throughout the CS 250 course, focusing on Agile Scrum principles, roles, ceremonies, and tools explored during the Vision Quest Software case study discussions and related coursework. While the course didn’t involve executing a full software development project, the assignments and activities provided valuable insights into Agile methodologies and their practical applications and has helped me understand many aspects of my own job at Institute for Social Research (ISR) at the University of Michigan.

Throughout the course, I got to complete assignments as the core Scrum roles such as the Scrum Master, Product Owner, Developer, and Tester, understanding their responsibilities and how they contribute to Agile team success. For example, as a Developer, I explored techniques like Planning Poker and Pair Programming to improve estimation accuracy and foster collaboration. The Product Owner role emphasized backlog prioritization to maintain focus and prevent shifting priorities, while the Scrum Master role highlighted the importance of facilitating daily stand-ups and backlog refinement to keep the team aligned and responsive (Mountain Goat Software, 2025; Agile Alliance, 2025). One of the biggest “Ah hah!’ moments for me was the difference between the Scrum Master and the Product Owner really hit home.

The course structure also helped me grasp how Scrum organizes work into manageable sprints, each beginning with sprint planning and concluding with a sprint review and retrospective. Although we didn’t build actual software, we analyzed user stories such as wellness-themed travel filters and destination categories, discussing how iterative refinement and incremental delivery support alignment with customer needs.

A key learning point was Agile’s flexibility, especially how it enables teams to adapt to changing requirements without losing momentum. For instance, when wellness features were introduced mid-sprint in our discussions, we examined how backlog refinement and sprint planning ceremonies facilitate re-prioritization and goal adjustment with minimal disruption. I had been introduced to Waterfall and Agile methodologies before in previous classes, but never in such a full software development scenario which I found helpful in growing some of my current DevOps responsibilities at my job.

I saw how effective communication is a cornerstone of Agile success, as well as seen in my real life work experience how the lack of it can derail a project. A cornerstone of Agile success is effective communication, which I practiced in our team activities. For instance, in the Vision Quest Software discussion, I assumed the Developer role and provided a specific recommendation to my team by posting: **"**To support this transition, I recommend implementing Planning Poker and Pair Programming... Pair Programming... will help break down the silos that previously existed." This sample was effective because it proposed a specific, actionable solution for the "silo" problem our team faced. Furthermore, as I noted in my journal, when the Product Owner introduced the new wellness-themed requirements mid-project, I initiated **"a structured email outlining specific questions about destination categories and testing expectations"**. This proactive communication was critical for fostering collaboration as it ensured my work as a Developer would align with the new requirements *before* implementation began, preventing wasted effort.

I see how these interactions can foster trust and transparency and encouraging collaboration across roles. Reflecting on questions like balancing experience levels in Pair Programming deepened my understanding of mentorship and team dynamics within Agile teams (CodeLucky, 2025; Agile Alliance, 2025).

The course also introduced organizational tools like Jira, which support Scrum teams by managing product backlogs, assigning story points, and tracking progress through Scrum boards and burndown charts. These tools act as information radiators, providing real-time visibility and helping teams maintain focus on delivering value. Combined with Scrum ceremonies, these tools support a disciplined yet flexible workflow (Atlassian, 2025).

Throughout the course, I connected these Agile concepts to my real-world work experience at the ISR. Moving into a hybrid role, I’ve seen firsthand how Agile principles like backlog prioritization and iterative delivery help my team manage shifting priorities and deliver incremental value, especially in the ongoing overhaul of our proprietary MSMS software used for our interviewing processes.

Overall, this course experience underscored the value of Scrum-Agile methodologies in fostering collaboration, adaptability, and customer focus and really opened my eyes and really helped me understand some of the Scrum and Agile practices that were and are already happening at my job that I was rather oblivious to before this course. While I can see challenges such as the initial learning curve and the need for disciplined backlog management, the principles and practices studied provide a strong foundation for effective Agile adoption and have I feel made me more effective at my real world job. Based on this analysis, the Scrum-Agile approach was definitely the best methodology for the SNHU Travel project. The key evidence for this was the Module Five interruption, where the Product Owner pivoted the project focus to "detox/wellness travel". Under a rigid waterfall model, this change would have been disastrous, likely forcing the project back to the design phase and jeopardizing the deadline. Because we used Agile, the Product Owner was able to simply "deprioritize other stories in the Product Backlog", allowing the development team to adapt and deliver the new, higher-value features without derailing the entire project

**References**

Agile Alliance. (2025). Pair Programming. Agile Alliance Glossary. <https://www.agilealliance.org/glossary/pair-programming/>

Atlassian. (2025). Master Agile Project Management with Jira. Atlassian. <https://www.atlassian.com/software/jira/agile>

CodeLucky. (2025). Pair Programming: Benefits and Best Practices for Agile Development Teams. CodeLucky. <https://codelucky.com/pair-programming-benefits-best-practices/>

Mountain Goat Software. (2025). Planning Poker: An Agile Estimating and Planning Technique. Mountain Goat Software. <https://www.mountaingoatsoftware.com/agile/planning-poker>

Scrum Alliance. (2025). What Is Continuous Integration? Scrum Alliance. <https://resources.scrumalliance.org/Article/continuous-integration>

Scrum Alliance. (2025). What Is Test-Driven Development? Scrum Alliance. <https://resources.scrumalliance.org/Article/test-driven-development>