CS 250 Final Project: Sprint Review and Retrospective  
Luis Tomassini

Throughout the SNHU Travel project at ChadaTech, I had the unique opportunity to step into each role on the Scrum-Agile team—Scrum Master, Product Owner, Developer, and Tester. Experiencing the development process through all perspectives gave me a deeper appreciation for how interconnected and essential each function is to successful Agile delivery. This retrospective reflects on the challenges, lessons, and strengths of applying the Scrum-Agile approach throughout the sprint, especially as we transitioned away from a traditional waterfall mindset.

Serving as Scrum Master, I was focused on promoting transparency, facilitating daily stand-ups, and removing impediments that could slow progress. I quickly realized how vital consistent, open communication is to a team’s success. Organizing and participating in stand-up meetings allowed the team to stay aligned, surface blockers early, and promote collaboration between roles that had previously operated in silos. According to Cobb (2015), one of the core responsibilities of the Scrum Master is to foster an environment where the team can function without distractions or misalignment. One of the most impactful practices I implemented in that role was encouraging the team to develop a shared **Definition of Done**. This set a clear, unified standard for what completion meant across roles—particularly between developers and testers—which reduced rework and ambiguity.

In my time as Product Owner, I focused on backlog refinement and the evolving prioritization of user stories. I ensured our work reflected stakeholder needs while remaining feasible for the team. One example that stood out was when new feedback mid-sprint led to a suggestion to simplify the user interface for the trip planning tool. Thanks to Agile's flexibility, we were able to acknowledge the value of this change, adjust our backlog, and plan for its integration in the next sprint—something that would have been disruptive under waterfall. According to Mahnič (2021), the role of the Product Owner in Agile involves continuous prioritization and negotiation with stakeholders, especially in response to changing requirements. I also worked closely with testers to refine acceptance criteria, ensuring clarity and testability in every user story.

As a Developer, I experienced the hands-on application of iterative development. Sprint planning and Planning Poker exercises helped our team estimate effort collaboratively and avoid overcommitting. When priorities shifted or new ideas emerged, we responded by re-evaluating story points and adjusting scope rather than forcing unfinished or low-value work forward. Agile gave us permission to deliver in smaller, functional increments without being bogged down by an all-or-nothing mentality (Beck et al., 2001).

Stepping into the Tester role, I reinforced the importance of **continuous integration and feedback**. I prioritized test case creation based on user stories and acceptance criteria while collaborating with developers to test features early and often. This reduced the risk of bugs piling up late in the sprint. Additionally, I worked to ensure that every test was aligned with user needs by validating the stories directly with the Product Owner—myself in a previous sprint—which highlighted the real-world importance of role communication and user-centered thinking. As noted by Hossain and Babar (2020), incorporating testers earlier into Agile workflows helps improve overall code quality and reduce last-minute risk.

Across all roles, communication remained the cornerstone of our success. Tools like burn-down charts and the evolving product backlog helped visualize our progress, while daily stand-ups, retrospectives, and constant cross-role feedback loops ensured that no team member was working in isolation. We all contributed to a continuous learning cycle, where reflection at the end of each sprint helped us adapt and improve for the next one.

To support our workflow, we used several organizational tools grounded in Agile principles. The evolving product backlog gave us the flexibility to respond to change, while story points provided a more effective way to estimate effort than fixed time predictions. Burn-down charts offered a visual snapshot of our remaining work, helping us monitor progress and maintain sprint velocity. These tools reinforced Agile principles such as responding to change over following a rigid plan and delivering working software over exhaustive documentation (Beck et al., 2001). These values helped us stay focused on continuous delivery and adaptation.

Overall, the Scrum-Agile process proved to be highly effective during the SNHU Travel project. One of its strongest advantages was its adaptability. The team was able to shift focus quickly, react to feedback, and prioritize tasks that delivered the most value. Collaboration was stronger, team morale was high, and the pace of delivery was consistent. One challenge we did encounter was the early learning curve—crafting clear user stories and assigning accurate story points took time and practice. However, as the team settled into the Agile rhythm, our workflow became much more fluid and productive.

Based on our results, I would highly recommend that ChadaTech continue expanding its Agile transformation. The pilot project demonstrated that Agile not only improved how we worked but also what we delivered. We were able to respond to change, build stronger team dynamics, and maintain high product quality in a way that would have been much harder under the traditional waterfall model. As ChadaTech looks to modernize its development approach across teams, the Scrum-Agile methodology presents a clear path toward faster delivery, higher adaptability, and greater overall value.

**References**

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., … & Thomas, D. (2001). *Manifesto for Agile Software Development*. <https://agilemanifesto.org/>

Cobb, C. G. (2015). *The Project Manager's Guide to Mastering Agile: Principles and Practices for an Adaptive Approach*. John Wiley & Sons.

Hossain, E., & Babar, M. A. (2020). *Agile quality assurance strategies: A survey-based empirical study*. Journal of Systems and Software, 163, 110519. <https://doi.org/10.1016/j.jss.2020.110519>

Mahnič, V. (2021). *Agile requirements engineering practices: An empirical study*. Computer Standards & Interfaces, 74, 103521. <https://doi.org/10.1016/j.csi.2020.103521>