**Sprint Review and Retrospective**

Over the course of the SNHU Travel project, I had the opportunity to step into every major role on the Scrum team—Product Owner, Scrum Master, Developer, Tester, and Communicator. In each role, I learned different lessons about how Agile teams function. As the Product Owner, I learned the importance of being the voice of the customer. Our user interviews helped prioritize what features to develop, such as a customizable user profile, which aligned with multiple requests. This reflects Cobb’s statement that one of Agile’s greatest strengths is its focus on business value and the ability to adapt through direct user engagement (Cobb, 2015).

Acting as the Tester highlighted the consequences of unclear user stories. There were certain missing details in the user stories that had an impact on what to test, the default values for a slider for example. It required sending an email asking for clarification, and it also presented an opportunity to propose an enhancement for usability. This aligns with Agile's principle that continuous feedback and adaptation are essential for quality, as well as the value of cross-functional collaboration where testing is not just a hand-off at the end. Quality isn’t something you inspect, it’s something you build in, as emphasized by Deming’s Total Quality Management roots embedded in Agile (Cobb, 2015).

In my role as Scrum Master, the focus was on structuring events. Daily scrums, backlog grooming mid-sprint, and retrospectives at the end of each iteration helped ensure we stayed Agile in practice. This structure supported an adaptable framework that evolved through regular reflection, just as Principle 12 of the Agile Manifesto recommends: “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly” (Beck et al., 2001). Cobb (2015) also supports this structure, noting that Scrum’s strength lies in its simplicity and flexibility, encouraging common sense over rigid instructions.

One of the most interesting experiences came when a stakeholder wished to shift focus onto wellness trips, based on market research. What would have been a major roadblock in a traditional project was relatively easy to accommodate thanks to Agile’s rolling-wave planning (Cobb, 2015). Because we designed with flexibility in mind, all that was needed was to update tags, metadata, and filters. Communication practices like timely emails, standups, and visual task tracking helped keep everyone aligned during the adjustment. The emphasis on working software and responding to change over sticking to a rigid plan was critical here (Beck et al., 2001).

Scrum’s time-boxed sprints and cadence were also crucial to keeping momentum. This is part of Agile’s flow-based thinking, which focuses on maximizing throughput and reducing bottlenecks (Cobb, 2015). By breaking work into manageable chunks and allowing for regular checkpoints, we were able to stay productive without overloading any team member.

Despite Agile’s successes, there were still complications. For example, when a team member showed a pattern of tardiness, it challenged Scrum’s emphasis on self-organization, showing that sometimes stronger leadership is necessary. Cobb (2015) mentions this as well, stating that the Scrum Master may need to shift between servant leadership and coaching depending on the team’s maturity.

Looking at the overall process, I believe Scrum was an excellent fit for this project. Agile’s iterative nature helped the team adapt quickly, while it’s emphasis on communication and user feedback ensured that we stayed aligned with what was needed. The case study of Valpak shows how Agile can scale at the enterprise level, supporting both innovation and structure when tailored to a company’s needs (Cobb, 2015). While this project was obviously not at that same scale, the principles remain consistent.

Were this a real team project going forward, I’d continue using Agile as a way to refine the process further. I’d advocate for better story clarity from the start, and greater accountability from all the team members in the daily scrums. That could inspire them to suggest their own improvements as well, fostering an environment that is always improving.

**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*. Wiley.