**ChadaTech Sprint Review and Retrospective**

**SNHU Travel Project**

Matthew Biletnikoff

CS-250: Software Development Lifecycle

Prof. Haruka Konishi

06/19/2025

**Applying Roles:**

As the Scrum Master for the SNHU Travel project, I played a key role in guiding our team through the Scrum-Agile development process. Throughout the project, each Scrum role; Scrum Master, Product Owner, and Development Team, was critical to our success. As the Scrum Master, I facilitated Scrum events such as Sprint Planning, Daily Scrums, and Sprint Retrospectives. I was responsible for maintaining focus and ensuring we followed Agile principles.  
  
 When I served as the Product Owner in a previous sprint, I gathered feedback from the SNHU Travel focus group and used it to refine the Product Backlog. For example, one user emphasized the need for easy access to wellness packages, which I translated into a prioritized user story. I also defined acceptance criteria to make sure the development team knew what a successful feature looked like.  
  
 The Developers were responsible for implementing these stories. They collaborated effectively, splitting larger tasks such as the top destinations carousel into sub-tasks, which included designing UI components, implementing clickable links, and ensuring image paths worked in the JAR file. Each team member brought different strengths, and our cross-functional approach enabled faster and more reliable feature delivery.

**Completing User Stories:**

Using the Scrum-Agile framework allowed our team to complete user stories efficiently and transparently. Each story was broken down into achievable tasks during Sprint Planning. For instance, one of our primary user stories stated: “As a traveler, I want to view the top five destinations so that I can choose my next vacation spot easily.” This story was split into components such as loading destination names from a data file, displaying formatted descriptions, showing relevant images, and linking each to...  
  
 During the sprints, we used the Definition of Done (DoD) to verify completion. We constantly checked that every task met our quality standards, passed integration tests, and was reviewed by the team. Because of this structure, we finished our stories on time, and stakeholders could validate features immediately during Sprint Reviews. This process helped ensure we were delivering value with each iteration.

**Handling Interruptions:**

Agile’s adaptability was especially helpful when we encountered a major issue related to image paths in the packaged Java JAR file. Initially, our images loaded correctly in the development environment but failed once the program was exported. Rather than halting progress or waiting until a future release cycle, as might happen with Waterfall, we addressed the problem immediately in the next Sprint.  
  
 We discussed the issue in our Retrospective and identified the root cause: the image files were using absolute paths rather than relative ones. To solve this, we refactored the file structure, moved all assets into a dedicated resource folder, and updated our code to load images using relative paths with `getClass().getResource()`. After retesting and confirming the solution, we documented this fix in our shared knowledge base for future team reference.  
  
 This ability to pivot mid-sprint without significantly impacting our momentum was a testament to the resilience and flexibility of Agile development.

**Communication:**

Clear, consistent communication played a central role in keeping our team aligned and efficient. I facilitated Daily Scrum meetings where each member shared their progress, plans, and blockers. This structure kept our momentum going and ensured that no one was left struggling in silence. When a team member reported difficulty integrating the image-loading function into the UI, we allocated time to pair-program and quickly resolved the issue.  
  
 In addition to verbal communication, we used online tools to stay organized. Our task board allowed team members to assign themselves stories, mark progress, and attach relevant code or screenshots. I also posted regular updates such as: “The JAR export issue is now resolved. Please test the slideshow component before tomorrow’s demo.” These messages were brief, actionable, and encouraged immediate collaboration. This communication style was essential for staying focused, especially while balancing remote wor...

**Organizational Tools:**

Scrum’s structured approach, paired with supportive tools, helped our team stay focused and deliver value each sprint. We used the Product Backlog to capture user requirements and conducted Backlog Refinement to keep it updated and relevant. During Sprint Planning, we used story points and team velocity to estimate effort, which helped us balance scope and capacity.  
  
 The Burndown Chart offered a real-time visual of our progress, and when it began trending upward unexpectedly due to unresolved bugs, we prioritized debugging and technical debt in the next sprint. These metrics gave us a clear view of performance and areas needing improvement. Retrospectives gave us space to reflect on what went well and what didn’t. For example, after struggling with test failures due to late merging, we established a policy to freeze feature branches 24 hours before the demo.  
  
Overall, these organizational tools and events allowed us to self-manage effectively and remain accountable without micromanagement.

**Evaluating Agile Process:**

The Scrum-Agile process was especially effective for the SNHU Travel project. Its iterative and incremental approach allowed us to gather feedback and implement changes quickly. This was a major contrast to the Waterfall model, where design, development, and testing are performed in separate, often disconnected stages. For this project, where we were responding to real-time feedback from a mock client, Agile gave us the adaptability and speed we needed.  
  
 The main benefits we experienced included early delivery of working features, better visibility into team progress, and improved alignment with user needs. We also encountered some challenges. For instance, task estimation using story points was difficult in our early sprints. However, over time, our confidence and consistency improved. Additionally, Agile requires continuous involvement from stakeholders and a high degree of discipline among team members.  
  
 Compared to Waterfall, where changes late in the process can be costly and slow, Agile enabled us to integrate new insights without resetting the entire project. Based on our success, I strongly believe Scrum-Agile should be adopted more broadly at ChadaTech. The approach promoted cross-functional collaboration, continuous improvement, and frequent delivery of customer value, making it a better fit for modern software projects.

# References

Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. Scrum.org. https://scrumguides.org/scrum-guide.html