**Sprint Review and Retrospective**

**Introduction**The SNHU Travel application project was conducted using a Scrum-Agile approach, marking ChadaTech’s first initiative in transitioning from a Waterfall methodology. This retrospective evaluates how Scrum roles, Agile principles, and iterative development contributed to the project’s success while assessing the benefits and challenges of this approach. Insights are drawn from studies on Agile practices and their real-world implications (Overeem et al., 2021; Spiegler et al., 2021).

**Applying Roles**The Scrum framework provided clearly defined roles, each crucial to project success. As **Scrum Master**, I facilitated sprint planning, daily stand-ups, and retrospectives, ensuring alignment and resolving blockers. For instance, during planning, I guided the team in breaking down complex user stories into manageable tasks (Boren, 2016).

As **Product Owner**, I prioritized the backlog, ensuring the team focused on high-value features such as a user-friendly booking system. Collaboration with stakeholders ensured that deliverables aligned with business needs. Studies emphasize the importance of Product Owners in maintaining project focus and delivering value by bridging team goals with stakeholder priorities (Spiegler et al., 2021).

The **Development Team** implemented tasks collaboratively, ensuring timely delivery of core features like authentication and search functionality. Research highlights that well-structured roles in Agile frameworks significantly enhance team productivity and accountability (Overeem et al., 2021).

**Completing User Stories**  
Agile’s iterative approach ensured the steady completion of user stories by breaking them into manageable tasks during sprint planning. For example, the secure login functionality was split into subtasks, including API development and UI integration, enabling focused progress. Research highlights how iterative development promotes clear task prioritization and continuous progress (Ebert & Paasivaara, 2017).

Frequent sprint reviews allowed stakeholders to provide immediate feedback. For instance, suggestions to improve the search feature were incorporated seamlessly into subsequent sprints. This iterative process ensured that deliverables met user expectations while maintaining high quality (Boren, 2016).

**Handling Interruptions**  
Agile’s flexibility was critical when the client requested multi-factor authentication mid-project. Using backlog refinement, the team reprioritized tasks, incorporating the change into the next sprint without disrupting progress. Daily stand-ups provided a platform to address dependencies, ensuring the project stayed on track despite evolving requirements. Research by Ebert and Paasivaara (2017) underscores backlog refinement as a vital practice for adapting to mid-project changes while maintaining project timelines.

**Communication**  
Clear communication was key to collaboration. Daily stand-ups allowed team members to share updates and resolve blockers quickly. For example, one team member highlighted a dependency issue during a stand-up, which was resolved the same day (Spiegler et al., 2021).

Written updates, such as post-sprint summaries, kept stakeholders informed and engaged. Studies have found that consistent communication practices in Agile teams not only foster collaboration but also improve decision-making processes (Overeem et al., 2021). By fostering open communication, the team stayed aligned on objectives and maintained productivity.

**Organizational Tools**  
Tools like Jira were instrumental in managing tasks and visualizing progress. The Kanban board tracked user stories and identified bottlenecks, while sprint retrospectives used Jira metrics to evaluate team performance. These tools supported core Scrum principles by increasing transparency and accountability (Boren, 2016).

Scrum events, such as retrospectives and planning sessions, enabled the team to reflect on past sprints and adapt to changes, further demonstrating the effectiveness of Agile tools and principles (Verwijs & Russo, 2023).

Evaluating Agile Process  
The Scrum-Agile approach presented several advantages:

* Collaboration: Frequent interactions ensured alignment with client needs.
* Flexibility: The ability to adapt to changes, such as adding MFA, was a significant benefit.
* Incremental Delivery: Progress was continuous and measurable (Ebert & Paasivaara, 2017).

However, the approach also had challenges:

* Learning Curve: Transitioning from Waterfall required significant adjustments (Spiegler et al., 2021).
* Scope Creep: Frequent changes risked expanding project scope (Verwijs & Russo, 2023).

Despite these challenges, Agile was the most effective methodology for this project due to its adaptability and focus on iterative improvements (Boren, 2016).

Conclusion  
This retrospective highlights the strengths of Agile in managing dynamic software projects. Through structured roles, effective communication, and organizational tools, the team delivered a high-quality application aligned with client expectations. Agile’s flexibility and emphasis on collaboration make it a strong candidate for broader adoption at ChadaTech (Ebert & Paasivaara, 2017; Spiegler et al., 2021).

**References**

1. Overeem, M., Spoor, M., Jansen, S., & Brinkkemper, S. (2021). An empirical characterization of event sourced systems and their schema evolution — lessons from industry. *Journal of Systems and Software*, *178*, 110970. <https://doi.org/10.1016/j.jss.2021.110970>
2. Spiegler, S. V., Heinecke, C., & Wagner, S. (2021). An empirical study on changing leadership in agile teams. *Empirical Software Engineering: An International Journal*, *26*(3). <https://doi.org/10.1007/s10664-021-09949-5>
3. Boren, A. (2016). [Review of *Scrum: The Art of Doing Twice the Work in Half the Time*, by J. Sutherland]. *Technical Communication*, *63*(3), 273–273. <http://www.jstor.org/stable/44809507>
4. Verwijs, C., & Russo, D. (2023). *Do Agile Scaling Approaches Make A Difference? An Empirical Comparison of Team Effectiveness Across Popular Scaling Approaches*. <http://arxiv.org/abs/2310.06599>
5. Ebert, C., & Paasivaara, M. (2017). Scaling Agile. *IEEE Software, Software, IEEE, IEEE Softw*, *34*(6), 98–103. <https://doi.org/10.1109/MS.2017.4121226>