AGILE VS. WATERFALL

AGILE ROLES

- SCRUM Master: Holds the entire SCRUM team together by fostering collaboration and communication amongst team members.
- Product Owner: Communicates with stakeholders and presents their wants to the rest of the SCRUM Team through and organized product backlog.
- Developer: Develops the product according to the user requirements in the product backlog.
- Tester: Tests the product and communicates with the developer about any needed adjustments or concerns.

SDLC PHASES AND AGILE

- Planning: Defining the scope of the project and the software's goals. Fosters collaboration and communication in an agile team.
- Requirements Analysis: Identifying what end users are looking to gain from the software. Keeps the team focused on delivering the best possible product for the end user.
- Design: Designing functional software according to the end user's requirements.
- Coding: Converting the work done in the design phase into tangible and functional code.
- Testing: Testing the code for any bugs or errors and communicating these issues back to the developers.
- Deployment: Delivering the final product to the end users.
- Maintenance: Constant testing and improvements to the product after release.

WATERFALL METHOD

If the SNHU Travel Project had used Waterfall instead of Agile the development process would have been more rigid and not as flexible. This would have presented a problem when the goal of the project changed to offering detox/wellness vacations as I would have had to go back and redo every previous step which would have thrown off the momentum of the project. With agile I was able to make the adjustments in one single sprint, and I did not have to redo previous steps.

AGILE VS. WATERFALL

When choosing between Agile or Waterfall, one must consider the size of the project, the deadline, and the flexibility required for the project. With the SNHU Travel Project, Agile was the better fit due to the flexibility needed to adapt to changes fast while still meeting the original deadline. Because the project was developed in each module over the course of the term, I was able to adjust the project based on what I learned each week, this would have been more difficult in a Waterfall model as I would have had to redo previous work. If the project and clear and concrete requirements that I knew would not change throughout the development cycle, Waterfall would have been the better fit.

REFERENCES

Atlassian. (n.d.). *Agile scrum roles and responsibilities*. https://www.atlassian.com/agile/scrum/roles

Zhezherau, A. (2022, November 16). *The Agile Software Development Life Cycle | Wrike Agile Guide.* Wrike. https://www.wrike.com/agile-guide/agile-development-life-cycle/#the-six-phases-of-the-agile-life-cycle

The seven phases of the software development life cycle. (2025, April 16). *Harness.io*. https://www.harness.io/blog/software-development-life-cycle-phases#phase-1-planning

GeeksforGeeks. (2025, April 11). *Waterfall Model Software Engineering*. GeeksforGeeks. https://www.geeksforgeeks.org/waterfall-model/