

Reflection Report (1-2 pages, team-authored):

Strengths & weaknesses of the process used

Comparison of actual vs. expected outcomes

How ChatGPT influenced requirements, design, coding, and testing

**Kellen Grossenbacher:**

The biggest fall back I noticed during the waterfall process was the necessity of clear and exact requirements up front in order to ensure the outcomes and implementation later on would fit the expectations of our final product. When prompting ChatGPT to generate a SRS, we had to go back and add some more specific aspects of the requirements so that all desired functionality was going to be properly achieved. We had to be very nit-picky when it came to the description of the features that would be used to implement our requirements. However, what became very clear as we went was that the more specific and exhaustive our SRS was, the easier it was to prompt ChatGPT for our other artifacts. By using our SRS as context, we were easily able to generate in-depth pseudocode that laid the foundation for complete implementation. With both the SRS and pseudocode as context, generating a usable and functioning code base in a single prompt with minor tweaking was made possible.

When comparing our expected/actual outcomes, it became very apparent to me that the waterfall process, although top heavy, allows for precise and high-functioning code to be generated with relative ease at the back end of the process. All functionality was achieved with a single prompt and even passed the test cases we generated after the code. Using the waterfall process for this project gave me a lot of insight into where different development processes can be stronger than others, and the specific shortcomings of the waterfall process. It also taught me the value in curating exhaustive prompts in order to get the most out of an AI tool like ChatGPT.

**Jalil Harris**

Going into this project the waterfall method was one that was used in most of our coding experiences so I've had some experience working with it. The distinct phases of requirements, design, implementation, and testing provide a structured format I tend to appreciate. A major strength of the approach was the SRS being locked down early so the requirements were able to be followed. The separation of design and coding allows for very high-level and disciplined planning so that the corresponding phases work as a success. I appreciate that side of the waterfall method.

That being said, the weaknesses of the waterfall method become apparent very fast. Once the requirements are locked in, there is little flexibility to adapt to new issues and changes. Not being able to go and iteratively improve and adapt to changes places a very large limit on the method. If a design oversight or missing requirement is designed during implementation and

testing it is too hard to change it without disrupting the workflow of the code. This major setback of the waterfall method led me to believe that an agile methodology is a more efficient approach. When comparing actual vs. expected outcomes the system met the core functionality requirements but some edge cases and inputs weren't validated and threw a lot of errors, a major drawback of using a AI system like Chat GPT for generation. I found Chat GPT to be very helpful with the SRS generation, Pseudocode generation and with the coding, of our project as it really helped give a clear roadmap on how the project could be completed. I found that when actually helping generate code Chat GPT would have some mistakes that would need to be cleaned up but this project really enhanced for me the power of these systems if prompted correctly!

### **Hayden Lorenz**

The biggest strength of this process was that in my previous experiences building projects, we utilized the waterfall model, thus making this process a little more familiar. Utilizing ChatGPT to assist in the SRS process was beneficial because it was my first exposure to writing a full SRS document. Additionally, with ChatGPT it became apparent how specific we needed to be with our prompts in order to get our desired outcome. Additionally, I believed that the waterfall process was beneficial to our group because I often like outlining the project beforehand in order to have a clear direction going forward. Furthermore, I believe that utilizing a process we were familiar with and that focused on the requirements heavily to start was beneficial.

Even though the waterfall process was something we utilized before, there were still some weaknesses that became apparent. For instance, the time taken to write and finalize our requirements took a good portion of our time and limited us with the ability to begin coding earlier. This slowed down the time taken to complete our project by a significant amount. Although this was a weakness with the time restraint of the in-class project, it was also beneficial to ensure that our requirements were solid from the beginning. Moreover, the time taken to complete our project due to the waterfall process was a weakness, however, the overall quality of our project was more preferable.