The experience of using a Waterfall Development Process had many shortcomings, but also many benefits. The initial stage of creating a Software Requirement Specifications (SRS) document was greatly expedited by the use of ChatGPT. However, the entire process and the document itself felt unnecessary for this particular project. While it still took multiple iterations of generations to correct it, the process still would've taken far longer by hand. The next step of creating UML diagrams and pseudo code ended up being quite helpful in the development process; the pseudo code, especially, assisted in implementing the general flow of the program in the main function.

The testing stage of the Waterfall Development Process was definitely where expectations and reality were not in harmony, but also where ChatGPT had the most influence. After the creation of the initial prototype, ChatGPT was consulted about the creation of the JUnit test cases. Since the bulk of the program flow and logic is centralized in the main function, it was asked about testing the main function directly, which is heavily discouraged. Therefore, the main flow of the program had to be tested directly in the command line interface, by hand. This definitely deviates from the expected course of action for the Waterfall Development Process, but it was necessary to ensure the creation of a working program that followed all of the requirements and necessary functionality. ChatGPT was never used to directly create or advise the team regarding contents of unit test cases, but it was helpful in teaching the proper use of JUnit tests and their syntax.