Ryan Murphy and Brady Barnes

Professor Brinkley

CPSC 3720

31 August 2025

Reflection Report

Regarding the strengths of the Waterfall process we used during the creation of the Study Buddy app, there were some clear advantages. The waterfall design process allowed our group to begin work on the web app quickly and finish its development in a timely manner. As a result of not needing to create and then test components in steps (the Agile process), we could see that the waterfall process could work great for smaller projects and short deadlines. However, there are also many drawbacks to using this method that our group discovered. One example is that we could not change our Software Requirements Specification (SRS) after we had agreed to it. This led to us not being able to deviate from the SRS when one of us had an idea they believed to be better once we started development. There will nearly always be instances of a change in the original idea, and the waterfall process does not accommodate well for these instances. Another problematic issue is that the testing phase of the web app was at the end. Thankfully, our finished project did not have that many bugs-- but if there were some complications, saving testing until the end is more complicated than testing portions and components of the project.

Initially, we expected the Waterfall process to be straightforward and uncomplicated for this project. We thought that we would have stuck relatively close to our desired timeline and that we would not have to revisit older work. However, through the process we discovered that the data storage process was going to be an issue. Neither of our group members had experience with JSON storage files, which is what AI suggested we used. As a result, we had to tweak out

execution to instead use a simple JavaScript file to store our data. Furthermore, while our pseudocode was cohesive it took a while to understand how each of the files would communicate with each other.

Our team used a mix of Claude (Anthropic LLM) and ChatGPT to structure and assist our web app. Out of our team, only one of us had prior ReactJS, JavaScript, and CSS experience although our ReactJS knowledge was quite limited. As a result, we used Claude Opus 4.1 to draft our pseudocode and some portions of the ReactJS pages. Specifically, Opus 4.1 was used to create the formdata requests/statements to save user provided information to their profiles. We also used AI to help structure the matches algorithm and give guidance on the onclick calls in our React code. AI use was also prominent in fixing errors as it was very time efficient. AI was also used to generate tests for our team to use to see if there were any errors in our system, such as seeing if you could join a session for a class you were not a part of.