Reflection

When working on coding projects, there are two ways to go about it. One is the waterfall method, and one is the agile method. Both are designed to help teams reach goals, but each has its own pros and cons that make it better suited for certain situations.

The waterfall method is a method where you follow things step-by-step, and you’re planning things all the time. A major pro for this method is that it’s very structured. Everyone knows what’s coming up next, and it has a clear plan, and it's easier to track processes. Another good strength is that it produces a lot of documentation, which can be used later for training or fixing problems. One of these documents is the UML diagrams, which clearly show what is going on and how everything is going to work. However, there are cons that follow the water ball method, and that is it’s not very flexible. It follows a very strict process when it comes to coding the projects if something changes midway through the project it’s very expensive and you have to go all the way back and start all over again at the same time customers do not see the finish project until the project is nearly done, which can lead to disappointment if the product does not work as intended.

The agile method takes a different approach and sets up and step-by-step plan. The work is divided into cycles. These are often called sprints; each team delivers a small part of the project that already works, which can be tested. One of the big pros of this process is the flexibility that comes with it. If products need to be changed, you can do it relatively quickly. Another advantage is that the processes of visible much earlier since the products are delivered regularly. A big proponent of agile is that it delivers a lot of teamwork. Some of the cons are that because the partridge is always changing is harder to predict how long it will take and how much it will cost. Also, what car is constantly involved with customers and users, which is not always realistic, and agile tends to focus more on producing results rather than writing detailed documentation, which can make it harder to maintain or improve the project later.

In our study buddy application, we used the waterfall method. Since this is a smaller application, the waterfall method worked really well; however, in terms of a grand scale, if we wanted to have this application to be flexible and make changes to it anytime, anywhere, the waterfall method would not work. The agile method would because of the flexibility factor. We drew several UMLs that showed us what we needed to implement into our application. Apart from that, we implemented test cases that made us ensured that our application worked as we intended it to. Overall waterfall worked well on this scale, but in a grander form, I believe agile would have worked better in terms of updating the app and adding more features to it in the long run.

ChatGPT also influenced our project in a huge way. Our project served as a tool to code this project up. AI helped us eliminate the grunt work where whereas without AI, it would have taken us a lot longer. Troubleshooting was a lot easier with the help of AI. If we had any bugs in our code, we could have AI look in its database, and it would help fix our issues. We also had AI debug our test cases, but on top of that, it would give us pointers on how to improve our existing code, as well. In terms of design, ChatGPT was a big help. It also helped us create the UML, which would then be the blueprint for the project. AI was a big help in fleshing out our dead and going in depth then with. My group would come up with the ideas, and then we would ask ChatGPT to give us pointers or go in-depth with our ideas. ChatGPT was a major help with the efficiency aspect of our project.