**Nakul Patel**

**RUID – 165008860**

**Software Engineering – Group 13**

**Reflective Essay**

Software Engineering has been a very technically challenging, yet organizationally frustrating course. The course began with the creation of teams and a project proposal and has had quizzes, reports, and demos throughout, having us constantly engaged in working for the class. From the start, the course pushed me into a technically challenging place because a lot of the concepts that would be required for our project, Trade Kings (stock market fantasy game), I had not been exposed to or have had to use for any course. On the other hand, the course design logistically and in terms of grading was troubling because of the volume and uncertainty created.

Coming into the course, I came in with limited software engineering experience. I had developed skills primarily in object-oriented programming in Java and Python over my first two years in college. Also, I had some experience in systems programming in C and data structures in the last year. Since the project we took upon was a stock market fantasy league, the task was to create a website. With no web development experience, I realized that it would entail both front end and back end work. This includes working with the database, the Finance API, data processing and manipulation, user interface, etc. My work focused on two functionalities, the stock and cryptocurrency research pages, as well as the forecasting page. I was challenged to learn how to use JavaScript, HTML, CSS, and PHP because each allowed for the development of different aspects of the webpage. The process began with a lot of reading and testing of smaller blocks of code, and then I went into developing the actual page. The key struggle in the process was designing and creating a decent user interface. After a lot of time, I was able to use a combination of Bootstrap and CSS Grid to solve my problem.

One of the reasons this class was extremely challenging was because the ECE department provides no courses prior to this one on software methodology, web development, or databases. Therefore, as a student the course requires a lot of self-learning which has both positive and negative consequences.

The software engineering techniques were taught in the course helped give us a direction on how to plan out the functionalities and the timeline to complete them as well as how to divide the work amongst the nine members of the group. Moreover, on a more detailed level, the concept we learned help us organize how to solve the smaller problems in each functionality. One of the techniques that was very useful to us was dividing the functionalities amongst subgroups and having each subgroup carry out the full implementation including front and backend, only leaving the integration for others. This ensured that everyone was completely focused on solving and understanding their functionality. Another useful technique was to use my pencils before my keyboard. In other words, I learned various methods to plan out how to solve a problem by hand. For example, before writing the code for the user interface of the research pages, I made sure to have a detailed sketch in place which allowed me to code with a plan in mind.

On the other hand, the UML diagraming techniques that were taught in class were not that helpful because while they were done for the documentation, they did not help in the design. The diagrams pushed one’s more towards the technical requirements of making a proper diagram, rather than actually designing and brainstorming to solve a software engineering problem.

As a group, we decided to work on a stock market fantasy league game which was a predefined project. As a result, we started with a plan of requirements which we had to add and build on. Based on my experience, both working on an already defined project or defining one’s own are very good approaches. As long as the student is being honest in their work and efforts, there is a lot to learn when it comes to developing a larger-scale project from scratch because one is forced to learn all the different aspects of web development. It is easy to assume that defining one’s own project is more beneficial because it forces one to brainstorm and design an idea before even developing. However, when working on a defined project, it is essential to design and implement new features in existing ideas, which is innovation by itself.

Working on the stock market fantasy league was not only a challenge from a technical perspective, but also from a collaborative one. Having mostly worked individually and in partners in previous projects, including programming projects, this was a new experience. By working in a group of nine people, the first and utmost problem faced was communication. Using emails and group chats, an effort was made to make sure messages of design, implementation, and documentation were known by everyone, but not everyone in the group successfully complied, or were regularly on top of their assigned work. This resulted in partial communication and divided messages. In addition, the division of work was a major problem. We divided functionalities by subgroups, but it was evident that some functionalities were much more work than others, and that resulted in conflict at times. Also, the inability of members to do their portion of the work on time to a group set deadline helped in holding back the progression of the project. I often found myself frustrated that I was on top of my work, but some members were not and that it would harm my grade. Overall, the key challenges faced were communication amongst each other and varying levels of motivation to put in the work to succeed.

Although there were plenty of struggles working in a group, there were also many benefits. First, I gained exposure to working in larger teams which is common in the software engineering field when it comes to developing a project. This exposure taught me that not everyone who is your peer, or of equal rank, will listen to everything you have to advise or ask. Additionally, people will not be willing to put in the same effort or time as you are. I also learned how work needs to be divided not only so that everyone has to do work, but that it should cater to the strengths of the various members or else the team is not performing efficiently. Finally, the vital skill that was emphasized was the need for proper communication, so that everyone is on the same page and have the understanding that the project is above any one individual’s feelings.

If I were given another semester, I would think differently about how the work was divided amongst the group. While I took upon a functionality that I certainly felt capable of achieving, as a group, we failed to do the work efficiently and that resulted in preventable roadblocks. Consequently, I would recommend or try to implement more strict group milestones which help the group stay on track and completely the work in a timely fashion allowing for greater success.

In general, this course forced me to learn a lot of different things from the challenges to working in a team to the different technical skills necessary to develop a website. I was put in a position of having no experience, and so I was forced to create that experience for myself and put the extra effort to solve problems. Not only was this greatly beneficial technically, but also professionally because I was forced to understand that succeeding in the software engineering industry requires a lot of time, effort, and skill. I do feel that this course came up a little short in teaching design in comparison to documentation because the focus was more on knowing different forms of documenting rather than problem solving and actual innovation. Regardless, the course overall, was decently effectively in teaching me what is necessary from my perspective to develop a software project from the design to the development to the testing phase by forcing me to learn these things to make a successful project.