Portfolio: Reflection Lessons Learned (CSC470)

Alexander Ricciardi

Colorado State University Global

CSC470: Software Engineering

Dr. Vanessa Cooper

February 9, 2025

Portfolio: Reflection Lessons Learned (CSC470)

The winter semester B Software Engineering (CS470) course at CSU Global has introduced me to software engineering concepts and given me skills that I can apply software to model and develop software. The course emphasized the evaluation of system requirements and the analysis of software through the use of Unified Modeling Language (UML) diagrams. The course has also provided me with a good understanding of the entire software development lifecycle, from requirements gathering to deployment and maintenance. In this reflection, I will explore the key lessons that I learned and how they can apply and be applied toward more effective software engineering practices and modeling.

Before taking this course I had a layperson understanding of software development, I was aware that diagrams were used to model systems. However, I did not how they were used within the software development system. Additionaly, the only diagram types that I had some knowledge of how to use were flow charts and UML class diagrams. Furthermore, I was underestimating the crucial role that modeling plays in software development. The course taught me that modeling is not just about creating diagrams to understand a process; it's about creating shareable material that models, analyzes, and designs entire systems using various types of diagrams and methodologies such as Agile. Moreover, UML, provides diverse diagram types that have distinct purposes that can be used to capture various aspects of a system such as user interactions (use case diagrams), object structure and relationships (class diagrams), dynamic behavior and object interactions (sequence and state machine diagrams), and system workflows (activity diagrams). All these diagrams work together to analyze, model, and design systems providing a blueprint of the entire software lifecycle of those systems.

In my future software engineering endeavors, I will prioritize modeling, using UML diagrams that include:

- Use case diagrams which are the starting point for understanding user needs.
- Class diagrams that provide the foundation for designing object structures, which I will apply before coding.
- Sequence diagrams visualize object interactions, which I will use to identify design flaws early.
- Activity diagrams model describe the workflow behavior and dynamic aspect of systems, which I will use to model workflows and business processes.
- State machine diagrams visualize how objects change state over time, which I will use to model the dynamic behavior of individual objects within the system.

UML incorporates additional diagram types that can be used through the software lifecycle to analyze and design systems. Combining UML diagrams with a strong understanding of Object-Oriented principles, database modeling, and methodologies such as Agile, all concepts taught in the Software Engineering (CS470) course, will enable me to implement more effective software engineering practices and modeling in my professional life.

In conclusion, the CSC470 Software Engineering course gave me a new and more detailed understanding of software development. It has allowed me to transition from a novice programmer with limited UML diagram knowledge and understanding to a better-equipped junior software engineer with a good understanding of the entire software development lifecycle. In my future career as a software engineer, the concept of systems analysis, modeling, and design using UML, combined with an understanding of object-oriented principles, database systems,

and agile development practices, will allow me to contribute effectively to project teams and deliver high-quality software.