Jonathan Kim

Julia Hoffmann

Bryan Edman

Tyler Merritt

Connor Morley

CS-2450-601

Quality Control of UVSim Documentation

It is important to be concerned about the quality of UVSim because the software development needs to meet all the requirements of UVU. This documentation will show how IED controls the quality of the product such as the scrum process throughout the Software Development, how incremental prototype has met the requirements of UVU, and how Class Diagram implements many design patterns and guidelines. These reasons will prove that there shouldn’t be any concern for current and future development of UVSim.

Throughout the development of UVSim, we have been applying Agile method. The key values include where individuals of the teams have interacted over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change from following a plan. These key values helped us as a team to sustain the quality of our work because it allowed us to hold responsibility for their part, communicate the process of development by meeting online twice a week, and assist each other whenever problems arose during the development. It allowed us to deliver value to UVU as quickly as possible, and not bother about extensive plans. This in turn allowed us to involve the users in every step of development of UVSim.

Along with the agile method, implementing scrum is a key aspect of our development process where we focus on communication and collaboration done the right way. The scrum process required us to have product backlogs, sprint backlogs, scrum meetings, and the result of the product. Every member of our team knows how to implement the Scrum methodologies to achieve great results. UVU responded saying it meets their requirement for UVSim. The scrum has been a great way to maintain a constant supervision of daily responsibilities of each person and track changes and issues as they arise.

For milestone II, UVU wanted a justification paper regarding the prototype of UVSim. We approached with Incremental Prototype, which allowed us to accommodate new or changed user’s requirement. Instead of coming up with a completely different design, we allowed the software to grow by increment and enhance UVSim simulator. This prototype allowed us to focus on the essential features such as functionality of the simulator. Incremental prototype requires greater attention to quality factors such as flexibility and modularity, which will enhance the overall quality of design. For example, if IED approached the development of UVSim with a different development method such as the waterfall model, there is a greater chance that the developers won’t fulfill the users’ requirements during the development stages. This happens because there’s no communication between developers and users until the product is finalized. This results in a system that users aren’t satisfied with. However, as IED approached with Incremental prototype, it fulfilled basic needs of the UVU requirements, and it will continue to enhance until the expectations of UVU are met.

During the software development, UVU required IED to come up with a Class Diagram as the baseline for design. The class diagram has Model View Controller (MVC) and Façade software design patterns. MVC design allows UVSim to have three main logical components: the model, the view, and the controller. Each of these components are built to handle certain development aspects of the software program. The MVC design allows user to see the relation with main logical components as well as correlation with specific classes within UVSim simulator. The Façade software design pattern is important because it hides the complexity of the class diagram. UVSim currently has seven interdependent classes which can be difficult to understand for the user. The Façade software design pattern solved a recurring problem and is built to visualize how each class in the UVSim work together.

Other evidence that improvised design is effective can be seen through high cohesion and loose coupling design guidelines. Coupling measures the strength of the inter-module connections. However, loosely coupling design allows classes to be relatively independent and easy to comprehend. For UVSim, IED follow guidelines that includes format of stamp coupling (complete data structures are passed from one module to another) and data coupling (simple data is passed between modules) which meets expectation of UVU’s guideline. We have made sure that our software prototype codes have high cohesion. This means it is keeping parts of a code base that are related to each other in a single place. These guidelines result in UVSim having simpler communication, simpler correctness proofs, and improvement of comprehensibility.

Even though there may be multiple approaches to meet the expectation of UVU, we believe that the approach we have taken has met the criteria and specification of UVU. With the agile method, incremental prototype, and evidence of improved design by applying MVC, Façade software design pattern, high cohesion, and loose coupling design guidelines allowed UVSim to be in a great position for the software development process. These examples should help ease the worry of quality of UVSim and to have confidence in the development process. IED will continually apply agile method by constantly communicating with UVU as well as each other. We will also resolve any current or potential issues until the UVSim product is finalized. We hope that we will have UVU’s utmost trust and confidence with the development of UVSim and guarantee the quality will meet the needs of UVU.