Aaron Shipley

CS 410: Journal 3-2

Southern New Hampshire University

Brian West

1/20/2023

Software requirements engineering is a fundamental part of developing, producing, and maintaining software. Requirement engineering, according to Software Engineer Insider (2011), is the first phase of the Software Development Lifecycle. Software requirements engineering goes by many names, such a requirements collections requirements analysis, requirements definition, requirements development. These names all describe the same facet of the SDLC. This portion of the SDLC happens before any design, production, implementation, or release of actual software occurs.

The purpose of Software Requirements Engineering is quite basic in its nature. Requirements engineering is comprised of four sections. The first section is requirements elicitation. Elicitation is the tasks to acquire the requirements of the software. It is considered the stage where the requirements engineers try to find out what the stakeholders really want (Fahmi, Choi, 2007). The second section of Requirements engineering is analysis. This section entails understanding, organizing, interpreting, and classifying the requirements that were elicited in the previous step. Within this section, it is where the boundaries of the software will be established and any anomalies that may occur between requirements. This section is the area where one will focus on the problems that may be discovered, but the solution to it is not necessarily focused on.

The third step in requirements engineering is the specifications section. Within this segment an SRS document will be formed. An SRS (Software Requirements Specification) document is used, according to Rosencrance ( 2019), to formally specify the requirements and their the classification of functional and non-functional requirements within the software system.

The SRS acts as an agreement between customer / client and the development team in terms of understanding and agreeing upon what the software will do and accomplish.

The fourth section of requirements engineering is requirements validation. This section ensure that the answers to the problems previously identified are acknowledged and addressed appropriately. This step validates the documents and requirements to ensure all requirements are addressed and all parties are in agreement before design and implementation phases are initiated.

Software reverse engineering is the process of Gathering requirements from the system itself. It essentially is the process of working in reverse to break down a functioning system to analyze, organize and interpret requirements based on the developed software (GeeksForGeeks, 2022). The uses for reverse engineering software can simply be for research into an application and the requirements it is comprised of or to update / upgrade software for more efficient execution

The main differences between reverse engineering and requirements engineering is all in the order in which tasks, goals, or desired data is achieved. With requirements engineering, the basis of the system is developed based on the requirements engineering and is more customer / developer driven and takes place before any code or tests are written, or software is produced. Reverse engineering works in opposite of this and works with the completed software and breaks the developed system down to gather the requirements from what the developed system executes.

The concept of round trip approach to software engineering is the concept of using requirements engineering and reverse engineering in tandem throughout the SDLC. This approach is changing the field of computer science in many ways. As projects and systems become more complex and detailed, the idea of a singular approach to the creation of software is rudimentary and cannot be looked at as a viable approach to successfully create top tier software. By utilizing reverse engineering throughout the SDLC, one can learn more about the requirements in more detail based on the source code of the system. This will only help in matters of ensuring requirements are properly met and the goals of the system are achieved fully. With this approach engineers can be well rounded computer scientists and better understand the what, how, and why of the system, instead of just understanding a portion of it.

References

GeeksforGeeks. (2022, July 28). Difference between forward engineering and reverse engineering. GeeksforGeeks. Retrieved January 22, 2023, from https://www.geeksforgeeks.org/difference-between-forward-engineering-and-reverse-engineering/

GeeksForGeeks. (2023, January 9). How to write a good SRS for your project. GeeksforGeeks. Retrieved January 22, 2023, from <https://www.geeksforgeeks.org/how-to-write-a-good-srs-for-your-project/>

Rosencrance, L. (2019, September 10). Software requirements specification (SRS). Software Quality. Retrieved January 22, 2023, from https://www.techtarget.com/searchsoftwarequality/definition/software-requirements-specification

S. A. Fahmi and H. -J. Choi, "Software Reverse Engineering to Requirements," 2007 International Conference on Convergence Information Technology (ICCIT 2007), Gwangju, Korea (South), 2007, pp. 2199-2204, doi: 10.1109/ICCIT.2007.228.

Software Engineer Insider. (2011). Software requirements engineering: One of the many niches. What is Software Requirements Engineering? | Software Engineer Insider. Retrieved January 22, 2023, from https://www.softwareengineerinsider.com/careers/software-requirements-engineering.html