Zachary Nicholas

September 19, 2024

CS 410 Software Reverse Engineering

Module Three Journal

* **Define**: What is software requirements engineering?

Software requirements engineering is defined as the process of defining, documenting, and maintaining requirements in the engineering process, This process is broken into 5 parts: The feasibility study, the requirement elicitation, the requirements specifications, the requirements verification and validation, and the requirement management.

* **Purpose**: Why is software requirements engineering an important part of the software development life cycle?

As I stated above there are considered to be 5 different parts of the software requirements engineering, chiefly among them is the feasibility section which looks to see if the program or app is Technically, operationally, economically, legally, and schedule. To say in other words if the program or app can be made with the current hardware the company has, if it can be well supported after its launch, make sure that it follows all laws, regulations, and standards, and finally if the timeline for the project or app is feasible. Another reason why this is so important ins the SDLC (Software Development Life Cycle) is if we don’t properly understand the requirements of the project especially the hardware or certain design considerations that need to be made it could lead to a badly optimized or even a program that isn’t able to run.

* **Comparison**: How does the approach of software reverse engineering differ from the approach of software requirements engineering?

Software reverse engineering typically deals with analyzing already existing software system in order to comprehend its structure, behavior, and functionality. This is often done to when documentation for the program is either not existent or very outdated and this process allows to view how the program would function and allow us to make documentation that would or could help with future maintenance or upgrades to the program. This is often thought to work completely opposite from how forward engineering happens such as with software requirement engineering where it focuses on obtaining the needs and expectations of the owners of the program or app to be made. This is often done by interviews to figure out what is most important for the owners when we design the app or program and prioritizing what can be done.

Software requirements engineering

* **Impact**: What are your thoughts on the proposed new integrated approach of round-trip engineering and its impact on the computer science field?

I find that the round-trip approach as it allows for the use of both of these engineering methods which can help to make “legacy” system easier to manage and potentially modifying and upgrading them to newer systems in the future, as well potentially optimizing and enhancing the software development life cycle by using both of these methods together.

References:

GeeksforGeeks. (2024, July 24). *Requirements Engineering Process in Software Engineering*. https://www.geeksforgeeks.org/software-engineering-requirements-engineering-process/

mo\_innovation. (2023, April 21). *Difference between forward engineering and reverse engineering*. Monarch Innovation Private Limited. https://www.monarch-innovation.com/forward-engineering-vs-reverse-engineering

Software reverse engineering to requirements | IEEE conference publication | IEEE xplore. (n.d.). https://ieeexplore.ieee.org/document/4420580