CS-410-T2855 Software Reverse Engineering

3-1 Journal

Eric Wallace

Nov. 13, 2022

* **Define**: What is software requirements engineering?  
  Software requirements engineering takes place during the first phase. The goal is to document, define and the maintenance of the software development process. Known as the SRS, the software requirements specification, is used to determine what software is developed and for gathering information from the customer to extrapolate the requirements of the system or application to be developed. All of this takes place before any design or development starts. The SRS isolates any potentials problems before they happen so not to disrupt the process.
* **Purpose**: Why is software requirements engineering an important part of the software development life cycle?  
  Software requirements engineering is important because it is the requirements themselves that dictates how software is designed and what features is in software. They help to find software errors during the requirements gathering stage which is a lot less expensive than find errors during the development stage. The requirements are determined by the customer in the form of wants and needs. Not only do requirements lower the expenses of software development but it could also prevent harm in some cases. Take for instance, a weapons system on a military ship. If there was an error that made it through to the live software that could be a matter of live and death. So, software requirements are very important to the development process.
* **Comparison**: How does the approach of software reverse engineering differ from the approach of software requirements engineering?  
  Software requirements engineering is all about how software is to be developed by gathering information from the customer and to find errors before they occur. Reverse software engineering is the breaking down of software to find out how software works. The difference between the two approaches is that one takes place before software development begins and the other after the software has been developed and deployed. Another difference is that in the process of reverse engineering is breaks the application down into small chunks to determine how it was written and reverse engineering requirements is the documenting how software is to be developed.
* **Impact**: What are your thoughts on the proposed new integrated approach of round-trip engineering and its impact on the computer science field?  
  My thoughts are the document was written in 2007 and in 2022 there is probably better techniques than that of round-trip. But, back in 2007, being that it leads to a better understanding of the requirements of legacy software to determine what can be reused, what is retained and what has to be thrown away could potentially cut down on development time of new software.