Brian Engel

Module 3 Journal

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

Software requirements engineering is used to determine user needs, constraints, and performance required from software. It is typically an iterative process that involves continuous user feedback and interaction.

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

Software requirements engineering is possibly the most important part of the software development process. Without clear requirements to use to build a system, you wouldn’t know what is needed and what isn’t, and probably waste a lot of time and money and possibly come up with a system that is not needed or unusable.

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

Software reverse engineering is an approach of analyzing an existing or legacy system and figuring out its components and how they work. While they both are used to find out how a system works / should work, the key difference is that reverse software engineering deals with existing systems and software requirements engineering deals with new systems.

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

I think that by using both software reverse engineering and software requirements engineering in an iterative process to design a new system based on an older system is the only way to do it right. As a matter of fact, I assumed this is how it has always been done and wasn’t a new approach. I think that software requirements engineering is the most straightforward approach and works best when you have knowledgeable stakeholders, but there are situations where software reverse engineering will really shine. Let's say we are building a new system from an existing outdated system. There is a stakeholder that needs to pull information out of a database. The requirements can be collected straight from that person in software requirements engineering. In the old system there was a person inputting customer information into the database and would know the requirements for that aspect of the system, but to make things more efficient that position is no longer there and the customer inputs information directly through a form on the webpage. Now there is no one that knows the ins and outs of the input aspect and might get underrepresented in the requirements. In this case reverse engineering can help to discover what is needed to make this operation flow smoothly. By using both methods of finding requirements and using the round-trip approach we can make sure that all aspects of a system are represented.