**Victor Udeh**

**CS410  
Software Requirements Engineering Journal**

**Date:**

1. **Define: What is Software Requirements Engineering?**

* Software requirements engineering is the process of identifying, analyzing, documenting, and validating the functions and features that a software system must have to meet stakeholders’ needs. This process ensures that project goals are clear, feasible, and aligned with user expectations, serving as a foundational phase in the development cycle.

1. **Purpose: Why is Software Requirements Engineering Important?**

* Software requirements engineering is crucial in the software development life cycle because it minimizes misunderstandings, reduces the need for costly modifications, and ensures the final product meets user expectations and project goals. By defining precise requirements early on, teams can avoid many common pitfalls, thereby saving time and resources.

1. **Comparison: How Does Software Reverse Engineering Differ from Software Requirements Engineering?**

* Software requirements engineering is a proactive process focused on defining and documenting what a system should do before development begins. In contrast, software reverse engineering is retrospective; it aims to understand, document, and reconstruct the functionality of an existing system, especially when original documentation is insufficient or missing.

1. **Impact: Thoughts on Round-Trip Engineering and its Impact on Computer Science**

* Round-trip engineering, which integrates forward and reverse engineering, allows for continuous updates and synchronization between a system's design and its implementation. This approach can greatly benefit legacy system maintenance and streamline modern development practices, ultimately enhancing efficiency, adaptability, and real-time documentation. Such advancements could positively impact the computer science field by fostering greater agility in handling evolving software systems.