**Module Three Journal**

Aaron Ciminelli

SNHU

CS-410-R1913 Software Reverse Engineering 23EW1

Professor Joe Parker

September 24th, 2023

According to the article, the purpose of software requirements engineering is to discover, analyze, document, validate, and manage the requirements for a software system. It involves understanding what customers, users, and other stakeholders expect from the system. Software requirements engineering also involves creating and maintaining the requirements specifications, which are used to describe the system's functions, behavior, and performance. These specifications are used to develop the software system and ensure that it meets all the requirements.

One crucial aspect of requirements engineering is to make sure that the development team creates a system that fulfills the actual needs of the users. If the requirements are ill-defined, the system may not solve the intended problem. By reducing rework, time and money can be saved. This requires clear communication between the user and the development team, and a clear understanding of the user's needs. Additionally, requirements engineering should include testing and maintenance to ensure the system is functioning as expected.

Reverse-engineering involves recovering the architecture, design, and code of an existing system, while requirements engineering focuses on defining the needs of a new system. Unlike reverse engineering, requirements engineering begins with the user, not implementation. Requirements engineering involves understanding the user's needs, creating specifications, and prototyping the system. It also involves testing the system to ensure it meets the user's needs. Finally, requirements engineering involves designing the system to conform to regulations and standards.

By using the proposed integrated round-trip approach, it is possible to enhance the process of requirements gathering by taking advantage of the knowledge available from the legacy system. This can lead to more accurate validation of the requirements, increased chances for reuse, and easier tracking of changes. If implemented correctly, this approach can result in improved quality and reduced risks associated with re-engineering. Additionally, this approach can also help to identify any potential issues with the legacy system, making it easier to address them before the new system is launched. Furthermore, this approach can also help reduce development costs, as well as improve the overall efficiency of the project.

Integrating reverse engineering and requirements engineering can provide a more comprehensive understanding of existing and necessary systems. If further research is conducted to improve these methods, they may prove beneficial in reengineering legacy systems. This could lead to more cost-effective, efficient, and resilient systems, as well as better system adaptability and scalability. Additionally, reverse engineering and requirements engineering can provide valuable insights into system design, maintenance, and improvement.

**Resources:**

Jeganathan, S. (2019). DevSecOps: A Systemic Approach for Secure Software Development. ISSA Journal, 17(11), 20–27.