

13.4.3 Reverse Engineering

Reverse engineering is the process of analyzing the program of the existing software system, its documentation, and behavior in order to develop a new program at a higher abstraction level. Note that reverse engineering does not involve changes to the system or creation of a new system; instead it is a process of examining the system without changing its overall functionality.

The key objectives of reverse engineering are to generate alternative views, recover lost information, detect side effects, synthesize higher abstractions, and facilitate reuse. Other objectives of reverse engineering are listed below.

- Removing access restrictions from the software system
- Customizing the existing system to recreate it into new form
- Analyzing both good and bad features of the software system
- Improving the performance and features of the existing software system
- Duplicating an existing component or software system to understand the security mechanism
- Updating obsolete software system and their old development processes with better, effective, and less expensive technologies.

Reverse engineering is not only used in software engineering but also in various other fields such as entertainment, consumer products, microchips, electronics, and mechanical designs. Before starting reverse engineering, it is essential that a well-planned life cycle analysis and cost-benefit analysis of the software system is carried out.

Reverse engineering is a time consuming process. However, nowadays this process has become easy due to present usage of Information Technology. The reasons for using Information Technology for reverse engineering are listed below.

- Engineering techniques have become computerized. Due to this, the task of preparing designs is done by computer, which makes it easier for the reverse engineering team to recognize and interpret the future design of the software system.

- Artificial intelligence techniques are being used for pattern recognition, parsing, and interpretation in order to determine the structures along with their location within the software system. Thus, it becomes easy and less time consuming to determine the structure of software during reverse engineering.

To perform reverse engineering, a reverse engineering team is formed, which analyzes the software system and generates information about it. Sometimes, it is observed that the team, which was involved in developing the existing system, is also involved in developing the new system. In such a case, it is difficult for the team to avoid the assumptions or considerations of the existing system. This in turn leads to generation of a system which is similar to the existing system. To avoid this, the team is divided after specification such as when the design and code specification has been produced. However, before being divided, it is important for the team to make the specification in an abstract manner.

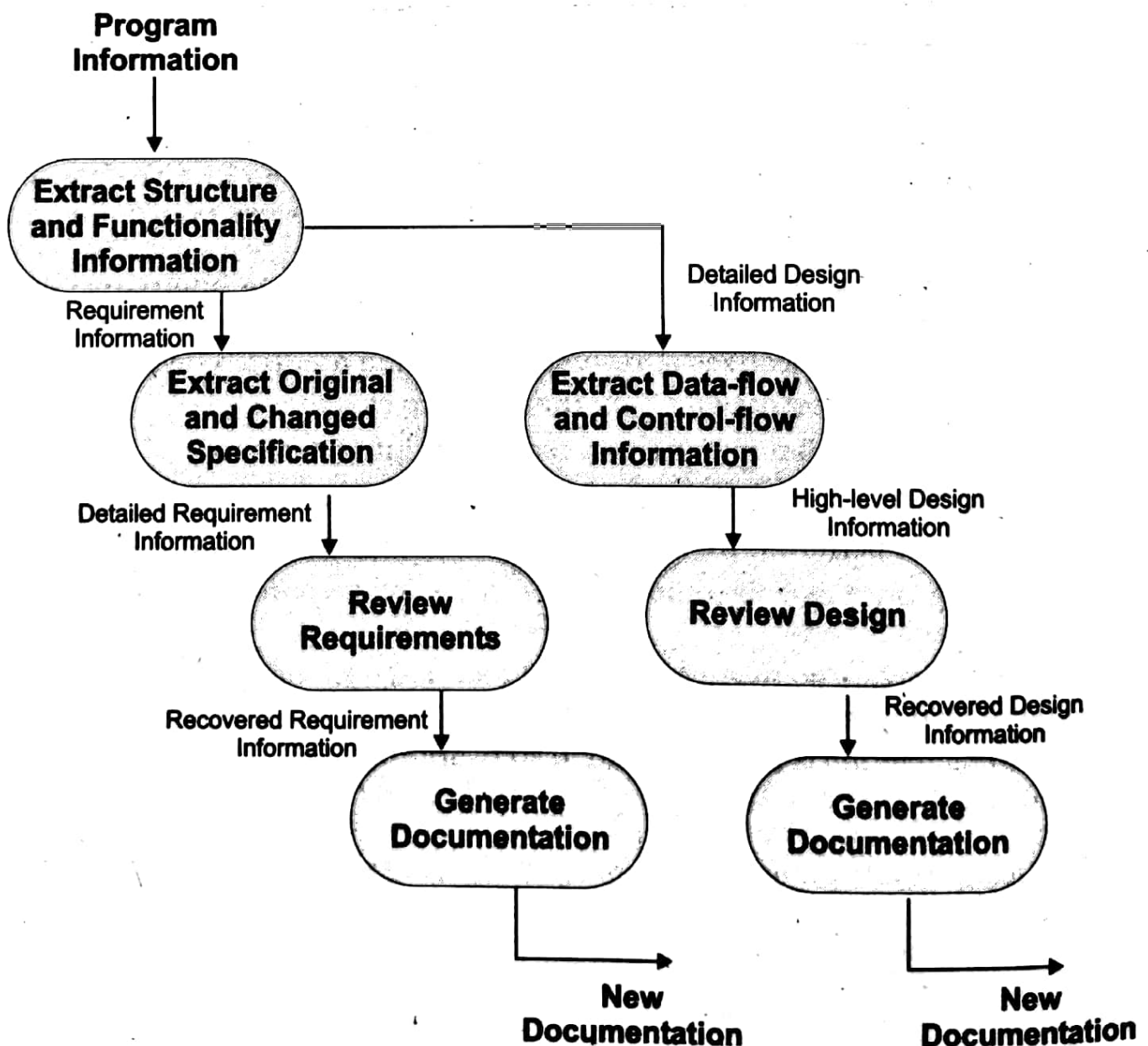


Figure 13.6 Reverse Engineering Process

As shown in Figure 13.6, the reverse engineering process begins by extracting the requirements and detailed design information from the source code and the documentation of legacy system. A requirements document is created and a high-level design abstraction is extracted and expressed using data-flow and control-flow diagrams. Next, the recovered design is reviewed for consistency and correctness. Once the requirements and design information is achieved, the reverse engineering documentation is generated.

Reverse engineering occurs at various levels of abstraction such as program level and system level. At the program level, internal program data structures are reverse engineered. At the system level, global data structures such as files and databases are re-engineered to incorporate new database management paradigms. For example, moving a relational database to a new platform.