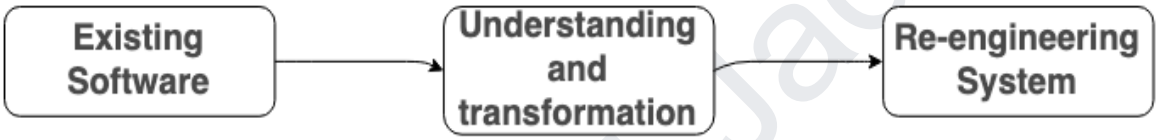
# Reverse Engineering

It is the procedure to get system specification by analyzing and understanding the existing system.

Reverse Engineering is also known as backward engineering, is the process of forward engineering in reverse. In this, the information is collected from the given or existing application. It takes less time than forward engineering to develop an application. In reverse engineering, the application is broken to extract knowledge or its architecture.




**Reverse Engineering**

**Steps of Software Reverse Engineering:**

Collection Information:

This step focuses on collecting all possible information (i.e., source design documents, etc.) about the software.

Examining the information:

The information collected in step-1 is studied so as to get familiar with the system.

Extracting the structure:

This step concerns identifying program structure in the form of a structure chart where each node corresponds to some routine.

Recording the functionality:

During this step processing details of each module of the structure, charts are recorded using structured language like decision tables, etc.

Recording data flow:

From the information extracted in step-3 and step-4, a set of data flow diagrams is derived to show the flow of data among the processes.

Recording control flow:

The high-level control structure of the software is recorded.

Review extracted design:

The design document extracted is reviewed several times to ensure consistency and correctness. It also ensures that the design represents the program.

Generate documentation:

Finally, in this step, the complete documentation including SRS, design document, history, overview, etc. is recorded for future use.

- # Restructuring

It is a modification of source code and/or data.

In general, no modification of all program architecture. It focuses on design details of individual modules and local data structure. If it involves program architecture, it becomes forward engineering. Restructuring occurs when the basic architecture is solid.

## Code Restructuring

The most common type of restructuring. It is a better design to perform the same function. If modules were coded in a way that makes them difficult to understand, test and maintain then code restructuring can be used.

To accomplish this activity:

- The source code is analyzed using a restructuring tool.
- Violations of structured programming constructs are noted.
- Code then restructured manual or automatically.
- The resultant restructured code is reviewed and tested.

Techniques of code restructuring:

- Warnier's logical simplification techniques
    - Boolean algebra

- Conversion of "spaghetti-bowl" code into structured program

- Reengineering tools
  - Resource exchange diagram maps program module and resources
  - Program architecture is restructured to minimize coupling.

### Data Restructuring

Firstly "Data Analysis " must be done.

- Current data architecture is examined in detail and data models are defined.
- Data objects and attributes identified, and existing data structures are reviewed for quality.

Then "Data Redesign" commences.

- Clarifies data definitions to achieve consistency among data item names and physical record formats.

At the end "Data Name Rationalization"

- Ensures all data naming conventions conform to local standards.

## Forward Engineering

Forward Engineering is a method of creating or making an application with the help of the given requirements. Forward engineering is also known as Renovation and Reclamation. Forward engineering requires high proficiency skills. It takes more time to construct or develop an application. Forward engineering is a technique of creating high-level models or designs to make in complexities and low-level information. Therefore this kind of engineering has completely different principles in numerous package and information processes. Forward Engineering applies to all the software engineering processes which contain SDLC to recreate associate existing applications. It is near fulfilling new needs of the users into re-engineering.

Forward engineering and reverse engineering are two approaches to software development, with different goals and processes. Here are some key differences between the two:

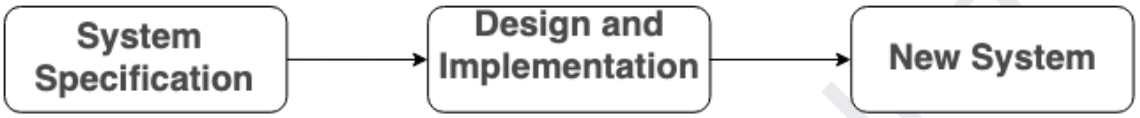
- Goal: The goal of forward engineering is to develop new software from scratch, while the goal of reverse engineering is to analyze and understand an existing software system.
- Process: Forward engineering involves designing and implementing a new software system based on requirements and specifications. Reverse engineering involves analyzing an existing software system to understand its design, structure, and behavior.
- Tools and Techniques: Forward engineering often involves the use of software development tools, such as IDEs, code generators, and testing frameworks. Reverse engineering often involves the use of reverse engineering tools, such as decompilers, disassemblers, and code analyzers.
- Focus: Forward engineering focuses on the creation of new code and functionality, while reverse engineering focuses on understanding and documenting existing code and functionality.
- Output: The output of forward engineering is a new software system, while the output of reverse engineering is documentation of an existing software system, such as a UML diagram, flowchart, or software specification.

Characteristics of forward engineering:

1. Forward engineering is a variety of engineering that has different principles in numerous package and information processes.
2. Forward engineering is vital in IT as a result of it represents the 'normal' development process.
3. Forward engineering deals with the conversion of business processes, services, and functions into applications.
4. In this method business model is developed first. Then, a top-to-down approach is followed to urge the package from the model developed.
5. Forward engineering tools are accustomed to move from implementation styles and logic to the event of supply code.
6. It essentially permits the user to develop a business model which may then be

translated into data system components.

7. These tools basically follow the top-to-down approach. System creator and visual Analyst is a forward engineering CASE tool.




**Forward Engineering**

Difference between Forward Engineering and Reverse Engineering

| Forward Engineering | Reverse Engineering |
|---|---|
| In forward engineering, the applications are developed with the given requirements. | In reverse engineering or backward engineering, the information is collected from the given application. |
| Forward Engineering is a high proficiency skill. | Reverse Engineering or backward engineering is a low proficiency skill. |
| Forward Engineering takes more time to develop an application. | While Reverse Engineering or backward engineering takes less time to develop an application. |
| The nature of forward engineering is Prescriptive. | The nature of reverse engineering or backward engineering is Adaptive. |

| | |
|---|---|
| In forward engineering, production is started with given requirements. | In reverse engineering, production is started by taking the existing products. |
| The example of forward engineering is the construction of electronic kit, construction of DC MOTOR , etc. | An example of backward engineering is research on Instruments etc. |
| Forward engineering Starts with requirements analysis and design, then proceeds to implementation and testing. | Reverse engineering Starts with an existing software system and works backward to understand its structure, design, and requirements. |
| Forward engineering is used to create new software applications from scratch. | Reverse engineering is Used to modify and improve an existing software application. |
| Forward egineering is process of moving from a high-level abstraction to a detailed implementation. | Reverse engineering is a process of moving from a low-level implementation to a higher-level abstraction. |
| Requires a clear set of requirements and design specifications. | Requirements and design specifications may not be available, making it necessary to reconstruct them from the code itself. |
| Forward engineering is generally more time-consuming and expensive. | Reverse engineering is generally less time-consuming and less expensive. |

| | |
|---|---|
| The final product is completely new and independent of any existing software system. | The final product is typically a modified or improved version of an existing software system. |
| Involves a series of steps such as requirements gathering, design, implementation, testing, and deployment. | Involves steps such as code analysis, code understanding, design recovery, and documentation. |
| Forward engineering is commonly used in the initial stages of software development. | Reverse engineering is commonly used in the maintenance stage of the software development life cycle. |