

## **TUTORIAL - 1**

Q1.What do you mean by Reverse engineering?

Q2 Explain the following:”Software engineering as a layered technology”.

Q3 Write down the objective of software engineering.

Q4. Explain software development life cycle. Explain various activities during SDLC

Q5.What are software process models? Distinguish Iterative Enhancement Model and Spiral Model.

## **ANSWER**

### **Ans1:-**

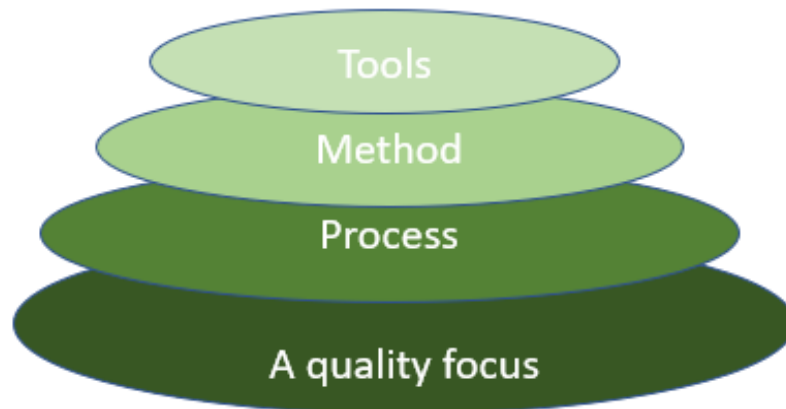
Reverse engineering is the process of analyzing a system or a product to understand how it works and how it was designed. Reverse engineering can be done for various purposes, such as learning from existing solutions, improving existing products, finding vulnerabilities or flaws, recovering lost information or documentation, or creating compatible products or services. Reverse engineering can involve different techniques and tools depending on the type and complexity of the system or product being analyzed. Some common examples of reverse engineering are:

- Disassembling a software program to examine its source code and functionality.
- Decompiling a binary file to recover its original source code or structure.
- Debugging a software program to observe its behavior and identify errors or bugs.
- Extracting data from a database or a file format to understand its structure and content.
- Measuring and testing a physical device or a component to determine its dimensions, materials, properties, and performance.
- Studying a patent or a design document to reproduce its features or functions.

Reverse engineering can have legal and ethical implications depending on the context and the intention of the reverse engineer. Some cases of reverse engineering may be protected by fair use laws or exceptions, while others may infringe on intellectual property rights or contractual obligations. Therefore, reverse engineers should always be aware of the potential risks and consequences of their actions and respect the rights and interests of the original creators or owners of the systems or products they analyze.

### **Ans2:-**

Software Engineering is a layered technology. It is the application of principles used in the field of engineering, which usually deals with physical systems, to the design, development, testing, deployment and management of systems. The main objective of software engineering layers is to help software developers obtain high-quality software.



**The layers of software engineering are:**

- Process
- Methods
- Tools
- Quality

The process layer defines a framework for a set of key process areas that must be established for effective delivery of software engineering technology. The methods layer provides a set of systematic procedures, techniques, and tools that can be used to support the process layer. The tools layer provides automated or semi-automated support for the process and the methods layers. The quality layer ensures that the software products and process adhere to the predefined standards and objectives.

**Ans3:-**

The objective of software engineering is to develop and maintain high-quality software that meets the needs of its users, is delivered on time and within budget, and is easy to maintain and modify as needed. It involves the application of engineering principles to software development, including the use of systematic, disciplined, and quantifiable approaches to software development, operation, and maintenance. Would you like to know more about software engineering?

The objective of software engineering is to:

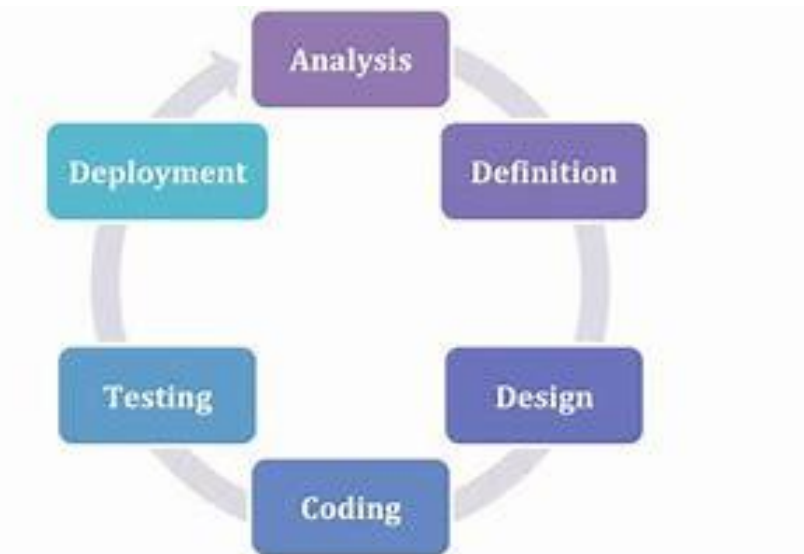
- **Apply systematic**, disciplined and quantifiable approaches to the development, operation and maintenance of software systems that meet the needs and expectations of users and stakeholders.
- **Ensure the quality**, reliability, security and usability of software products and

services through the application of engineering principles and practices.

- **Manage the complexity**, uncertainty and evolution of software systems through the use of appropriate methods, tools and processes.
- **Enhance the productivity**, efficiency and creativity of software developers and managers through the use of best practices and standards.
- **Contribute to the advancement** of knowledge and innovation in the field of software engineering through research, education and collaboration.

#### Ans4:-

The Software Development Life Cycle (SDLC) is the process of planning, designing, building, testing, deploying, and maintaining software.



#### The SDLC has six stages:

1. **Planning & Analysis:** This stage includes two parts: the planning stage where you are gathering requirements from your client or stakeholders and the requirement analysis stage where you're looking into the feasibility of creating the product, revenue potential, the cost of production, the needs of the users, etc.
2. **Design & Definition:** In this stage, the software architecture is designed, and the technical specifications are defined.
3. **Development & Coding:** This stage involves the actual coding of the software.
4. **Testing:** In this stage, the software is tested to ensure that it meets the requirements and specifications.
5. **Deployment:** In this stage, the software is deployed to the production environment.

6. **Deployment & Maintenance:** This stage involves the ongoing maintenance and support of the software.

Each stage in the SDLC has its own set of activities that need to be performed by the team members involved in the development project. For example, in the planning and analysis stage, the activities include gathering requirements, analyzing the feasibility of the project, and creating a project plan<sup>3</sup>. Similarly, in the design stage, the activities include creating a software architecture, defining technical specifications, and creating a design document<sup>3</sup>. In the development stage, the activities include coding, code reviews, and unit testing<sup>3</sup>. In the testing stage, the activities include creating test cases, executing tests, and reporting defects<sup>3</sup>. In the deployment stage, the activities include preparing the software for deployment, deploying the software, and verifying the deployment<sup>3</sup>. Finally, in the maintenance stage, the activities include fixing defects, updating the software, and providing support

#### **Ans5:-**

Software process models are a set of activities that are used to develop software. They provide a framework for software development that is used to guide the development process. There are many different software process models, including the **Iterative Enhancement Model** and the **Spiral Model**.

#### **Difference between Spiral model and Incremental model:-**

S.No.	Spiral Model	Incremental Model
1.	Spiral model is a software development model and is made with features of incremental, waterfall or evolutionary prototyping models.	Incremental Model is a software development model where the product is, analyzed, designed, implemented and tested incrementally until the product is finished.
2.	In spiral model requirements and early stage planning is also necessary.	In incremental model requirements and early stage planning is necessary.
3.	Flexibility to change in spiral model is not that difficult.	Flexibility to change in incremental model is Easy.

4.	There is low amount risk in spiral model.	There is also low amount risk in incremental model.
5.	While cost of spiral model is very expensive.	Cost of incremental model is also Low.
6.	Spiral model can handle large project.	Incremental model can't handle large project.
7.	In spiral model overlapping of phases is not possible.	In incremental model overlapping of phases is possible.
8.	Testing is done in spiral model at the end of the engineering phase.	Testing is done in incremental model after every iteration of phase.
9.	Returning to previous stage/phase in spiral model is possible.	Returning to previous stage/phase in incremental model is possible.
10.	In spiral model large team is required.	In incremental model large team is not required.