

## Introduction to Software Engineering – Software Engineering

**Software** is a program or set of programs containing instructions that provide the desired functionality. Engineering is the process of designing and building something that serves a particular purpose and finds a cost-effective solution to problems.

### What is Software Engineering?

**Software Engineering** is the process of designing, developing, testing, and maintaining software. It is a systematic and disciplined approach to software development that aims to create high-quality, reliable, and maintainable software.

1. Software engineering includes a variety of techniques, tools, and methodologies, including requirements analysis, design, testing, and maintenance.
2. It is a rapidly evolving field, and new tools and technologies are constantly being developed to improve the software development process.
3. By following the principles of software engineering and using the appropriate tools and methodologies, software developers can create high-quality, reliable, and maintainable software that meets the needs of its users.
4. Software Engineering is mainly used for large projects based on software systems rather than single programs or applications.
5. The main goal of Software Engineering is to develop software applications for improving quality, budget, and time efficiency.
6. Software Engineering ensures that the software that has to be built should be consistent, correct, also on budget, on time, and within the required requirements.

### Key Principles of Software Engineering

1. **Modularity:** Breaking the software into smaller, reusable components that can be developed and tested independently.
2. **Abstraction:** Hiding the implementation details of a component and exposing only the necessary functionality to other parts of the software.
3. **Encapsulation:** Wrapping up the data and functions of an object into a single unit, and protecting the internal state of an object from external modifications.
4. **Reusability:** Creating components that can be used in multiple projects, which can save time and resources.
5. **Maintenance:** Regularly updating and improving the software to fix bugs, add new features, and address security vulnerabilities.
6. **Testing:** Verifying that the software meets its requirements and is free of bugs.
7. **Design Patterns:** Solving recurring problems in software design by providing templates for solving them.
8. **Agile methodologies:** Using iterative and incremental development processes that focus on customer satisfaction, rapid delivery, and flexibility.
9. **Continuous Integration & Deployment:** Continuously integrating the code changes and deploying them into the production environment.

### Main Attributes of Software Engineering

Software Engineering is a systematic, disciplined, quantifiable study and approach to the design, development, operation, and maintenance of a software system. There are four main Attributes of Software Engineering.

1. **Efficiency:** It provides a measure of the resource requirement of a software product efficiently.
2. **Reliability:** It assures that the product will deliver the same results when used in similar working environment.

3. **Reusability:** This attribute makes sure that the module can be used in multiple applications.
4. **Maintainability:** It is the ability of the software to be modified, repaired, or enhanced easily with changing requirements.

### Dual Role of Software

There is a dual role of software in the industry. The first one is as a product and the other one is as a vehicle for delivering the product. We will discuss both of them.

#### 1. As a Product

- It delivers computing potential across networks of Hardware.
- It enables the Hardware to deliver the expected functionality.
- It acts as an information transformer because it produces, manages, acquires, modifies, displays, or transmits information.

#### 2. As a Vehicle for Delivering a Product

- It provides system functionality (e.g., payroll system).
- It controls other software (e.g., an operating system).
- It helps build other software (e.g., software tools).

### Objectives of Software Engineering

1. **Maintainability:** It should be feasible for the software to evolve to meet changing requirements.
2. **Efficiency:** The software should not make wasteful use of computing devices such as memory, processor cycles, etc.
3. **Correctness:** A software product is correct if the different requirements specified in the SRS Document have been correctly implemented.
4. **Reusability:** A software product has good reusability if the different modules of the product can easily be reused to develop new products.
5. **Testability:** Here software facilitates both the establishment of test criteria and the evaluation of the software concerning those criteria.
6. **Reliability:** It is an attribute of software quality. The extent to which a program can be expected to perform its desired function, over an arbitrary time period.
7. **Portability:** In this case, the software can be transferred from one computer system or environment to another.
8. **Adaptability:** In this case, the software allows differing system constraints and the user needs to be satisfied by making changes to the software.
9. **Interoperability:** Capability of 2 or more functional units to process data cooperatively.

### Program vs Software Product

Parameters	Program	Software Product
Definition	A program is a set of instructions that are given to a computer in order to achieve a specific task.	Software is when a program is made available for commercial business and is properly documented along with its licensing.

Parameters	Program	Software Product
		Software Product = Program + Documentation + Licensing.
Stages Involved	Program is one of the stages involved in the development of the software.	Software Development usually follows a life cycle, which involves the feasibility study of the project, requirement gathering, development of a prototype, system design, coding, and testing.

### Advantages of Software Engineering

There are several advantages to using a systematic and disciplined approach to software development, such as:

1. **Improved Quality:** By following established software engineering principles and techniques, the software can be developed with fewer bugs and higher reliability.
2. **Increased Productivity:** Using modern tools and methodologies can streamline the development process, allowing developers to be more productive and complete projects faster.
3. **Better Maintainability:** Software that is designed and developed using sound software engineering practices is easier to maintain and update over time.
4. **Reduced Costs:** By identifying and addressing potential problems early in the development process, software engineering can help to reduce the cost of fixing bugs and adding new features later on.
5. **Increased Customer Satisfaction:** By involving customers in the development process and developing software that meets their needs, software engineering can help to increase customer satisfaction.
6. **Better Team Collaboration:** By using Agile methodologies and continuous integration, software engineering allows for better collaboration among development teams.
7. **Better Scalability:** By designing software with scalability in mind, software engineering can help to ensure that software can handle an increasing number of users and transactions.
8. **Better Security:** By following the Software Development Life Cycle (SDLC) and performing security testing, software engineering can help to prevent security breaches and protect sensitive data.

In summary, software engineering offers a structured and efficient approach to software development, which can lead to higher-quality software that is easier to maintain and adapt to changing requirements. This can help to improve customer satisfaction and reduce costs, while also promoting better collaboration among development teams.

### Disadvantages of Software Engineering

While Software Engineering offers many advantages, there are also some potential disadvantages to consider:

1. **High upfront costs:** Implementing a systematic and disciplined approach to software development can be resource-intensive and require a significant investment in tools and training.
2. **Limited flexibility:** Following established software engineering principles and methodologies can be rigid and may limit the ability to quickly adapt to changing requirements.
3. **Bureaucratic:** Software Engineering can create an environment that is bureaucratic, with a lot of processes and paperwork, which may slow down the development process.
4. **Complexity:** With the increase in the number of tools and methodologies, software engineering can be complex and difficult to navigate.
5. **Limited creativity:** The focus on structure and process can stifle creativity and innovation among developers.
6. **High learning curve:** The development process can be complex, and it requires a lot of learning and training, which can be challenging for new developers.
7. **High dependence on tools:** Software engineering heavily depends on the tools, and if the tools are not properly configured or are not compatible with the software, it can cause issues.
8. **High maintenance:** The software engineering process requires regular maintenance to ensure that the software is running efficiently, which can be costly and time-consuming.

In summary, software engineering can be expensive and time-consuming, and it may limit flexibility and creativity. However, the benefits of improved quality, increased productivity, and better maintainability can outweigh the costs and complexity. It's important to weigh the pros and cons of using software engineering and determine if it is the right approach for a particular software project.

### Questions For Practice

#### 1. A software configuration management tool helps in [GATE CS 2004]

- (A) keeping track of the schedule based on the milestone reached
- (B) maintaining different versions of the configurable items
- (C) managing manpower distribution by changing the project structure
- (D) all of the above

**Solution:** Correct Answer is (B).

#### 2. Which of the following statements is/are true? [UGC NET CSE 2018]

**P:** Software Reengineering is preferable for software products having high failure rates, poor design, and/or poor code structure.

**Q:** Software Reverse Engineering is the process of analyzing software with the objective of recovering its design and requirement specification.

- (A) P only
- (B) Neither P nor Q
- (C) Q only

(D) Both P and Q

**Solution:** Correct Answer is (D).

**3. The diagram that helps in understanding and representing user requirements for a software project using UML (Unified Modeling Language) is: [GATE CS 2004]**

(A) Entity Relationship Diagram

(B) Deployment Diagram

(C) Data Flow Diagram

(D) Use Case Diagram

**Solution:** Correct Answer is (D).

### **Conclusion**

Software engineering is a key field that involves creating and maintaining software. It combines technical skills, creativity, and problem-solving. As technology advances, the need for software engineers increases, making it a great career choice. Whether you're new to the field or want to learn more, understanding software engineering is crucial. Keep exploring, learning, and enjoying the challenges and opportunities this field offers.

### **FAQs on Software Engineering**

#### **What is Software Re-Engineering?**

***Ans:** Software Re-Engineering is basically a process of software development that helps in maintaining the quality of the system.*

#### **2. State some Software Development Life Cycle Models?**

***Ans:** Some of the Software Development Life Cycle Models are mentioned below.*

- Waterfall Model
- Big-Bang Model
- Spiral Model
- Iterative Model
- V-Model

#### **3. What is Verification and Validation in Software Engineering?**

***Ans:** Verification refers to the set of activities or functions that checks whether software has implemented the correct function or not.*

*Validation refers to set of activities that ensures that the software is built as per the requirement of the client.*

*For more, refer to Verification and Validation in Software Engineering.*

## **Reverse Engineering - Software Engineering**

Software Reverse Engineering is a process of recovering the design, requirement specifications, and functions of a product from an analysis of its code. It builds a program database and generates information from this. This article focuses on discussing reverse engineering in detail. What is Reverse Engineering?Reverse engineering can extract design

## **Difference between Software Engineering process and Conventional Engineering Process**

Software Engineering Process and Conventional Engineering Process, both are processes related to computers and development. In this article, we will see the similarities as well as differences between both, that is Software Engineering Process and the Conventional Engineering Process. Table of Content Software Engineering ProcessConventional Engine

## **Re-engineering - Software Engineering**

Software Re-engineering is a process of software development that is done to improve the maintainability of a software system. Re-engineering is the examination and alteration of a system to reconstitute it in a new form. This process encompasses a combination of sub-processes like reverse engineering, forward engineering, reconstructing, etc. Tab

## **Difference between Software Engineering and Computer Engineering**

Software engineering and Computer engineering are two distinct disciplines that focus on different aspects of computer systems. While both fields require a strong foundation in computer science and mathematics, software engineering is focused on software development processes, while computer engineering is focused on the physical components and sys

## **Evolution of Software Engineering: From an Art To Engineering Discipline**

Software Engineering is a systematic and cost-effective technique for software development. It is an engineering approach to developing software. For example: If someone wants to travel from Punjab to Delhi. There are two approaches one can follow to achieve the same result: The normal approach is to go out and catch the bus/train that is available

## **Requirements Engineering Process in Software Engineering**

Requirements Engineering is the process of identifying, eliciting, analyzing, specifying, validating, and managing the needs and expectations of stakeholders for a software system. Table of Content What is Requirements Engineering?Requirements Engineering ProcessTools Involved in Requirement EngineeringAdvantages of Requirements Engineering Process

## **Introduction Of People Metrics in Software Engineering**

The concept of people metrics originated in the area of HR management. It provides a quantifiable measure of people's activity. These metrics are helpful to provide evidence of performance against things like objectives and goals. People's metrics have become important for Balanced Scorecards and other performance measurement systems. The reason is

## **MOCK (Introduction) - Software Engineering**

A mock, in software engineering, is a simulated object or module that acts as a stand-in for a real object or module. Mocks are often used in testing to isolate the behavior of a particular module or component and to verify that it behaves as expected. Mocks are typically created in code, and they simulate the behavior of real objects or modules by

## **Introduction to Faults in Software Engineering**

In software engineering, a fault is an error or defect in a program that causes it to produce incorrect or unexpected results. Faults can occur at various stages of the software development process, from the initial design to the final deployment. This article focuses on discussing faults in software engineering in detail. Table of Content  
What are

## **Difference between Forward Engineering and Reverse Engineering**

Forward engineering and reverse engineering are two approaches to software development, with different goals and processes. Forward engineering involves creating new software systems from scratch using given requirements and design specifications. It focuses on building new applications through a structured process of analysis, design, implementati

## **Introduction to Business Process Re-engineering**

Business Process Re-engineering (BPR) is a management strategy aimed at improving organizational performance by re-designing and optimizing business processes. BPR is a systematic and radical approach to change, focused on transforming and streamlining core business processes to achieve dramatic improvements in quality, efficiency, and customer sat

## **Software Engineering | Schick-Wolverton software reliability model**

Prerequisite - Jelinski Moranda software reliability model The Schick-Wolverton (S-W) model is a modification to the J-M model. It is similar to the J-M model except that it further assumes that the failure rate at the  $i$ th time interval increases with time  $t_i$  since the last debugging. In the model, the program failure rate function between the  $(i-1$

## **Software Engineering | Responsibilities of Software Project Manager**

Software Project Management (SPM) is a sub-field of Project Management in which software projects are planned, implemented, monitored and controlled. It consists of three terms: Software, Project and Management. So, let us understand each term separately. Software includes a set of programs, documentation and user manual for a particular software p

## **Software Engineering - Hardware Reliability vs Software Reliability**

Reliability in software is software that has no failure and works in a special time period with a special environment. Hardware reliability is the probability of the absence of any hardware-related system malfunction for a given mission on the other hand software reliability is the probability that the software will provide a failure-free operation

## **Failure Curve for Software in Software Engineering**

Pre-requisites: Software Engineering In Software Engineering, the concept of a software failure curve is part of the hardware bathtub curve. According to the diagram, the failure rate is a function of time for Hardware. This curve shows that hardware exhibits a relatively high failure rate early in its life (these failures are often design or manuf

### **Identifying Software Development Metrics - Software Engineering**

Companies invest a fair proportion of their resources in the expensive staff that drive the development of technology on which an organization runs. It is therefore imperative to explore and quantitatively measure the performance of such investments by empowering teams with the ability to track their efficiency. This task is delegated to a designat

### **Software Business and Development - Software Engineering**

Software Business means selling software for the business or sometimes it is referred to as selling information about various software made for business purposes that are so-called business software. This article focuses on discussing Software Business and Development in detail. What is a Software Business? Software business is one of the commercia

### **Software Quality Framework - Software Engineering**

Software Quality Framework is a model for software quality that ensures quality by connecting and integrating the different views of software quality. This article focuses on discussing the Software Quality Framework. What is a Software Quality Framework?Software Quality Framework connects the customer view with the developer's view of software qua

### **Software Evolution - Software Engineering**

Software Evolution is a term that refers to the process of developing software initially, and then timely updating it for various reasons, i.e., to add new features or to remove obsolete functionalities, etc. This article focuses on discussing Software Evolution in detail. What is Software Evolution?The software evolution process includes fundament

### **Characteristics of Good Software - Software Engineering**

Software is treated as good software using different factors. A software product is concluded as good software by what it offers and how well it can be used. The factors that decide the software properties are divided into three categories: Operational, Transitional, and Maintenance. What is a Good Software?Software engineering is the process of de

### **Software Quality Assurance (SQA) Set 2 - Software Engineering**

Software Quality Assurance (SQA) consists of a set of activities that monitor the software engineering processes and methods used to ensure quality. Software Quality Assurance (SQA) EncompassesA quality management approach. Effective software engineering technology (methods and tools). Some formal technical reviews are applied throughout the softwa

### **Software Project Management Plan (SPMP) - Software Engineering**



Once project designing is complete, project managers document their plans during a software package Project Management setup (SPMP) document. The SPMP document ought to discuss an inventory of various things that are mentioned below. This list will be used as a doable organization of the SPMP document. Organization of the software package Project M

### **Changing Nature of Software - Software Engineering**

The software is an instruction or computer program that when executed provides desired features, function, and performance. A data structure that enables the program to adequately manipulate information and documents that describe the operation and use of the program. Characteristics of software: There is some characteristic of software which is gi

### **Software Review - Software Engineering**

Software Review is a systematic inspection of software by one or more individuals who work together to find and resolve errors and defects in the software during the early stages of the Software Development Life Cycle (SDLC). A software review is an essential part of the Software Development Life Cycle (SDLC) that helps software engineers in valida

### **Agile Software Development - Software Engineering**

Agile Software Development is a software development methodology that values flexibility, collaboration, and customer satisfaction. It is based on the Agile Manifesto, a set of principles for software development that prioritize individuals and interactions, working software, customer collaboration, and responding to change. Agile Software Developm

### **Role and Responsibilities of a software Project Manager - Software Engineering**

A software project manager is the most important person inside a team who takes the overall responsibilities to manage the software projects and plays an important role in the successful completion of the projects. This article focuses on discussing the role and responsibilities of a software project manager. Table of Content Who is a Project Manag

### **Classification of Software - Software Engineering**

Software Engineering is the process of developing a software product in a well-defined systematic approach software engineering is the process of analyzing user needs and then designing, constructing, and testing end-user applications that will satisfy these needs through the use of software programming languages. Table of Content Parameters Defini

### **Software Design Process - Software Engineering**

The design phase of software development deals with transforming the customer requirements as described in the SRS documents into a form implementable using a programming language. The software design process can be divided into the following three levels or phases of design: Interface DesignArchitectural DesignDetailed DesignElements of a SystemAr

### **Software Product - Software Engineering**

Software Products are nothing but software systems delivered to the customer with the documentation that describes how to install and use the system. In certain cases, software products may be part of system products where hardware, as well as software, is delivered to a customer. Software products are produced with the help of the software process

### **Software Characteristics - Software Engineering**

Software is defined as a collection of computer programs, procedures, rules, and data. Software Characteristics are classified into six major components. Software