

Software Engineering

Software engineering is the systematic approach to the development, operation, maintenance and retirement of software. Software Engineering is the application of science and mathematics by which the capabilities of computer equipment are made useful to man via computer programs, procedures, and associated documentations.

Basic Objective

The basic objective of software engineering is to develop methods and procedures for software development that can scale up for large systems and that can be used consistently to produce high-quality software at low cost and with a small cycle of time.

Need for Software Engineering

- As Software development is expensive so proper measures are required so that the resources are used efficiently and effectively.
- Cost and time considerations are another factor, which arises the need for Software Engineering.
- Reliability factors

Software Development Phases

A development process consists of various phases, each phase ending with a defined output. The phases are performed in an order specified by the process model being followed. The main reason for having a phased process is that it breaks the problem of developing software into successfully performing a set of phase, each handling a different concern of software development.

This ensures that the cost of development is lower than what it would have been if the whole problem were tackled together. A phased development process is central to the software engineering approach for solving the software crisis. In general, we can say that any problem solving in software must consist of these activities:

Requirement Analysis

The requirement analyses and identifies what is needed from the system. Requirements analysis is done in order to understand the problem the software system is to solve. The problem could be automating an existing manual process, developing a new automated system, or a combination of the two. The two parties involved in software development are the client and the developer. The developer has to develop the system to satisfy the client's needs. The developer usually does not understand the client's problem domain, and the client often does not understand the issues involved in software systems. This causes a communication gap. This phase bridges the gap between the two parties. The outcome of this

phase is the software requirement specification document (SRS). The person responsible for the requirements analysis is often called the analyst.

Software Design

It is the first step in moving from problem domain to solution domain. The purpose of the design phase is to plan a solution of the problem specified by the requirement documents. Starting with what is needed, design takes us towards how to satisfy the needs. The design of a system is perhaps the most critical factor affecting the quality of the software. It has a major impact on the later phases, particularly testing and maintenance. The output of this phase is the design document. This document is similar to a blueprint or a plan for the solution and is used later during implementation, testing and maintenance. It is further of two types:

System Design or Top level Design: It identifies the various modules that should be in the system, the specifications of the modules and interconnections between the various modules. At the end of system design all the major data structures, file formats, output formats, and the major modules in the system and their specifications are decided.

Detailed Design: It identifies the internal logic of the various modules. During this phase further details of the data structures and algorithmic design of each of the modules is specified. Once the design is complete, most of the major decisions about the system

have been made. However, many of the details about coding the designs, which often depend on the programming language chosen, are not specified during design.

Coding

This phase translates the design of the system into code in a given programming language. This phase effects the both testing and maintenance. Well-written code can reduce the testing and maintenance efforts. The focus is on developing programs that are easy to read and understand. Simplicity and clarity should be strived for during the coding phase.

Testing

It is a major quality control measure used during software development. Its basic function is to detect errors in the software. Its basic function is to detect errors in the software. After the coding phase, computer programs are available that can be executed for testing purpose. This implies that testing not only has to uncover errors introduced during coding, but also errors introduced during the previous phases. Thus, the goal of testing is to uncover requirement, design, and coding errors in the programs. Consequently, different levels of testing are used.

The different types of testing are:

1. **Unit Testing:** It tests each module separately and detects ceding errors.
2. **Integration Testing:** When the various modules are integrated, this testing is performed to detect errors from the overall system.
3. **System Testing:** The system is tested against the system requirements to see if all the requirements are met and if the system performs as specified in SRS.
4. **User Acceptance Testing:** The system is tested against the user requirements to see if the user is satisfied.