Thakur Educational Trust's (Regd.)

THAKUR RAMNARAYAN COLLEGE OF ARTS & COMMERCE

ISO 21001:2018 Certified

PROGRAMME: B.Sc (I.T)

CLASS: S.Y.B.Sc (I.T)

SUBJECT NAME: SOFTWARE ENGINEERING

SEMESTER: IV

FACULTY NAME: Ms. SMRITI DUBEY

UNIT I

Chapter 1 – Introduction to Software Engineering

Concepts:

Software and its types

Characteristics of software

Software Engineering

Software Development Lifecycle

Software

In a computer system, **the software is basically a set of instructions or commands that tells a computer what to do. Or in other words, the software is a computer program that provides a set of instructions to execute a user's commands and tell the computer what to do**. For example, like MS-Word, MS-Excel, PowerPoint, etc. There are two fundamental types of software:

1) **Generic Software Products:** These are the stand - alone system that is produced by the development organization and sold on open market to any customer who is able to buy them. **Example** – databases, word processors.
2) **Customized Software Products:** These are the systems which are commissioned by particular customer**. Example –** Systems support a particular business process, Air traffic control systems.

They can be differed in features:

	Generic software products	Custom software products
1	The generic software development is done for developing a general-purpose software.	Customer software development is done to develop a software product as per the needs of particular customer.
2	In this development process, the software developers have to depict the end-user's specifications.	In this development process, the end-user requirements can be aggregated by communicating by them.
3	From designing and marketing perspective, this type of development is very difficult.	This development does not require marketing, because it is developed for appropriate group of users.
4	Large number of users may be using this kind of software.	This type of software is used by limited number of users.
5	Quality of the product is not a preference for generic software.	Quality is the main criterion in customer software product. Best quality of the product is focused for customer or company.
6	Development team controls the process of generic software development.	Customer determines the process of software development in this type of product.

7	Generally, the software developed is economical. There may be some hidden costs such as installation and implementation cost.	Software product is of high cost as the particular product for customer is developed.
8	Example of generic software product development is Word-editing software.	Inventory control and management system are examples of customer software

Software Engineering

Software engineering is **defined as a process of analyzing user requirements and then designing, building, and testing software application which will satisfy those requirements. IEEE, in its standard 610.12-1990, defines software engineering as the application of a systematic, disciplined, which is a computable approach for the development, operation, and maintenance of software.**

Software Crisis & its Solution

What was the Software Crisis?

- It was in the late 1960s when many software projects failed.
- Many software became over budget. Output was an unreliable software which is expensive to maintain.
- Larger software was difficult and quite expensive to maintain.
- Lots of software not able to satisfy the growing requirements of the customer.
- Complexities of software projects increased whenever its hardware capability increased.

- Demand for new software increased faster compared with the ability to generate new software.

All the above issues lead to 'Software Crisis.'

The Solution

Solution was to the problem was transforming unorganized coding effort into a software engineering discipline. These engineering models helped companies to streamline operations and deliver software meeting customer requirements.

- The late 1970s saw the widespread uses of software engineering principles.
- In the 1980s saw the automation of software engineering process and growth of (CASE) Computer-Aided Software Engineering.
- The 1990s have seen an increased emphasis on the 'management' aspects of projects standard of quality and processes just like ISO 9001

Attributes for Software Products

The characteristics of any software product include features which are displayed by the product when it is installed and put in use.

They are not the services which are provided by the product. Instead, they have related to the products dynamic behavior and the use made of the product.

Examples of these attributes are:

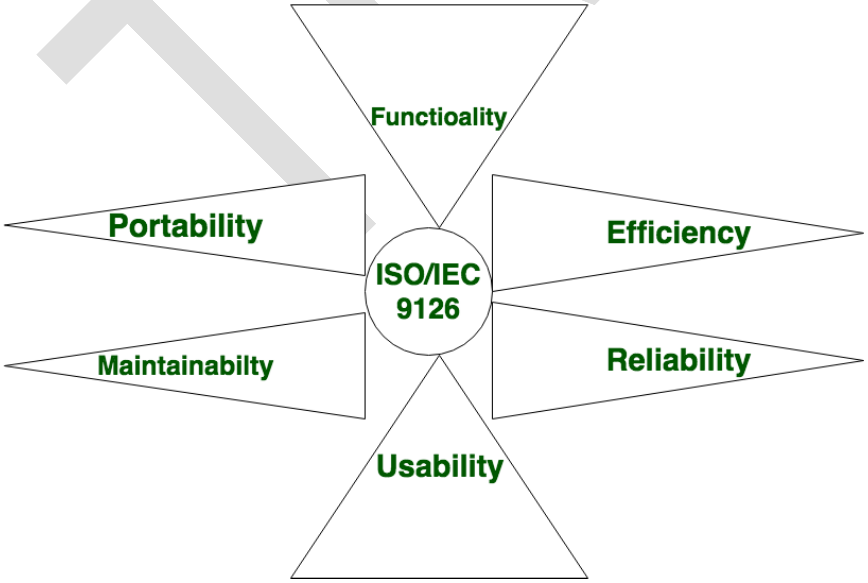
Efficiency, reliability, robustness, maintainability, etc.

However, the relative importance of these characteristics varies from one software system to another.

Product Characteristics	Description
Maintainability	The software should evolve to meet the changing demands of the clients.
Dependability	Dependability includes various characteristics. Dependable software should never cause any physical or economic damage at the time of system failure.
Efficiency	The software application should overuse system resources like memory and processor cycle.
Usability	The software application should have specific UI and documentation.

Characteristics of good Software

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



These components are described below:

Functionality:

It refers to the degree of performance of the software against its intended purpose.

Reliability:

A set of attributes that bears on the capability of software to maintain its level of performance under the given condition for a stated period of time.

Efficiency:

It refers to the ability of the software to use system resources in the most effective and efficient manner. The software should make effective use of storage space and executive command as per desired timing requirements.

Usability:

It refers to the extent to which the software can be used with ease. the amount of effort or time required to learn how to use the software.

Maintainability:

It refers to the ease with which the modifications can be made in a software system to extend its functionality, improve its performance, or correct errors.

Portability:

A set of attributes that bears on the ability of software to be transferred from one environment to another, without or minimum changes.

Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) is a framework that defines the steps involved in the development of software at each phase. It covers the detailed plan for building, deploying and maintaining the software.

SDLC defines the complete cycle of development i.e., all the tasks involved in planning, creating, testing, and deploying a Software Product. Every phase of the SDLC life Cycle has its own process and deliverables that feed into the next phase.

Software Development Life Cycle Process

SDLC is a process that defines the various stages involved in the development of software for delivering a high-quality product. SDLC stages cover the complete life cycle of a software i.e. from inception to retirement of the product.

Adhering to the SDLC process leads to the development of the software in a systematic and disciplined manner.

Purpose:

Purpose of SDLC is to deliver a high-quality product which is as per the customer's requirement. SDLC has defined its phases as, Requirement gathering, Designing, Coding, Testing, and Maintenance. It is important to adhere to the phases to provide the Product in a systematic manner.

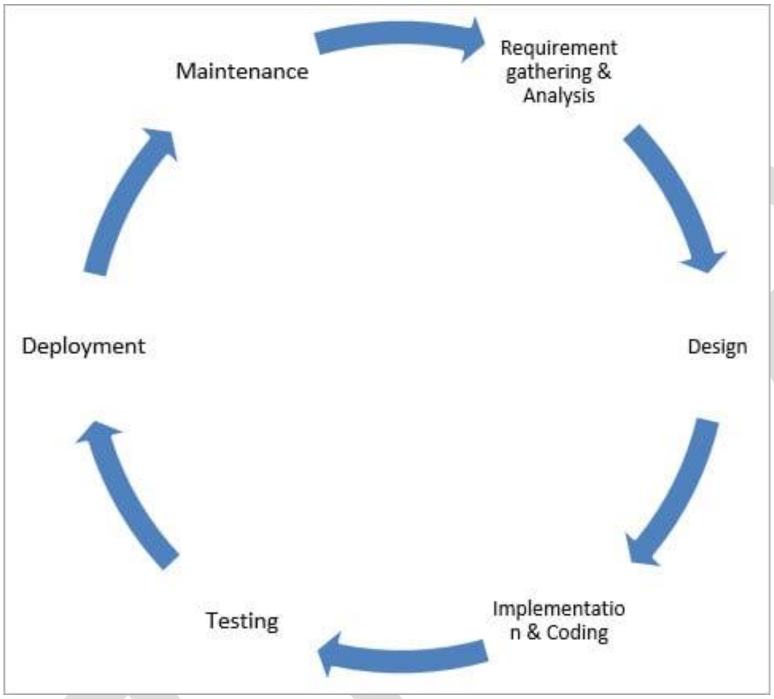
For Example, A software has to be developed and a team is divided to work on a feature of the product and is allowed to work as they want. One of the developers decides to design first whereas the other decides to code first and the other on the documentation part.

This will lead to project failure because of which it is necessary to have a good knowledge and understanding among the team members to deliver an expected product.

SDLC Cycle

SDLC Cycle represents the process of developing software.

Below is the diagrammatic representation of the SDLC cycle:



SDLC Phases

The entire SDLC process divided into the following SDLC steps:

Phase 1: Requirement collection and analysis

Phase 2: Feasibility study

Phase 3: Design

Phase 4: Coding

Phase 5: Testing

Phase 6: Installation/Deployment

Phase 7: Maintenance

Phase 1: Requirement collection and analysis

The requirement is the first stage in the SDLC process. It is conducted by the senior team members with inputs from all the stakeholders and domain experts in the industry. Planning for the quality assurance requirements and recognition of the risks involved is also done at this stage. This stage gives a clearer picture of the scope of the entire project and the anticipated issues, opportunities, and directives which triggered the project.

Requirements Gathering stage need teams to get detailed and precise requirements. This helps companies to finalize the necessary timeline to finish the work of that system.

Phase 2: Feasibility study

Once the requirement analysis phase is completed the next sdlc step is to define and document software needs. This process conducted with the help of 'Software Requirement Specification' document also known as 'SRS' document. It includes everything which should be designed and developed during the project life cycle.

There are mainly five types of feasibilities checks:

Economic: Can we complete the project within the budget or not?

Legal: Can we handle this project as cyber law and other regulatory framework/compliances.

Operation feasibility: Can we create operations which is expected by the client?

Technical: Need to check whether the current computer system can support the software

Schedule: Decide that the project can be completed within the given schedule or not.

Phase 3: Design

In this third phase, the system and software design documents are prepared as per the requirement specification document. This helps define overall system architecture.

This design phase serves as input for the next phase of the model.

There are two kinds of design documents developed in this phase:

High-Level Design (HLD)

Brief description and name of each module

An outline about the functionality of every module

Interface relationship and dependencies between modules

Database tables identified along with their key elements

Complete architecture diagrams along with technology details

Low-Level Design (LLD)

Functional logic of the modules

Database tables, which include type and size

Complete detail of the interface

Addresses all types of dependency issues

Listing of error messages

Complete input and outputs for every module

Phase 4: Coding

Once the system design phase is over, the next phase is coding. In this phase, developers start build the entire system by writing code using the chosen programming language. In the coding phase, tasks are divided into units or modules and assigned to the various developers. It is the longest phase of the Software Development Life Cycle process.

In this phase, Developer needs to follow certain predefined coding guidelines. They also need to use programming tools like compiler, interpreters, debugger to generate and implement the code.

Phase 5: Testing

Once the software is complete, and it is deployed in the testing environment. The testing team starts testing the functionality of the entire system. This is done to verify that the entire application works according to the customer requirement.

During this phase, QA and testing team may find some bugs/defects which they communicate to developers. The development team fixes the bug and send back to QA for a re-test. This process continues until the software is bug-free, stable, and working according to the business needs of that system.

Phase 6: Installation/Deployment

Once the software testing phase is over and no bugs or errors left in the system then the final deployment process starts. Based on the feedback given by the project manager, the final software is released and checked for deployment issues if any.

Phase 7: Maintenance

Once the system is deployed, and customers start using the developed system, following 3 activities occur

Bug fixing – bugs are reported because of some scenarios which are not tested at all

Upgrade – Upgrading the application to the newer versions of the Software

Enhancement – Adding some new features into the existing software

The main focus of this SDLC phase is to ensure that needs continue to be met and that the system continues to perform as per the specification mentioned in the first phase.