

CSE 411

Software Engineering and System Analysis and Design

Topic 1: Introduction

Introduction

Software Engineering provides a standard procedure to design and develop a software.

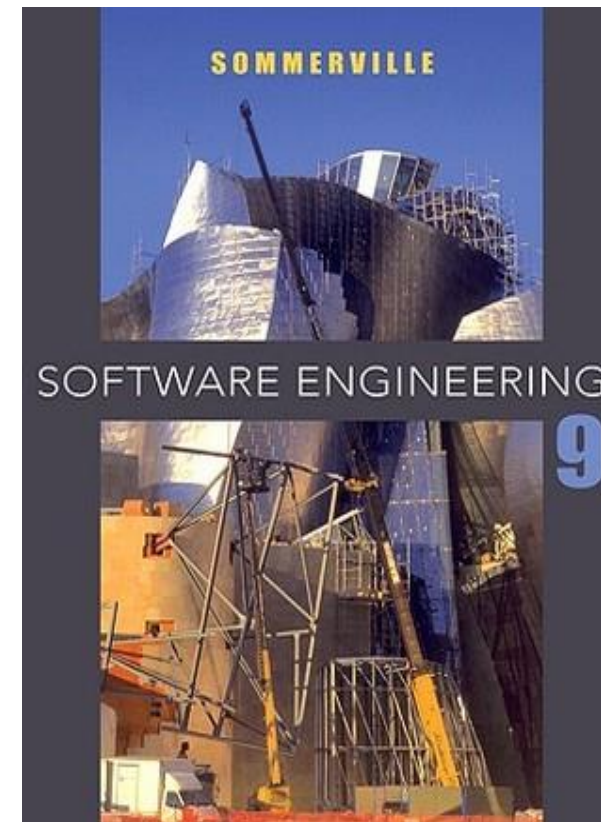
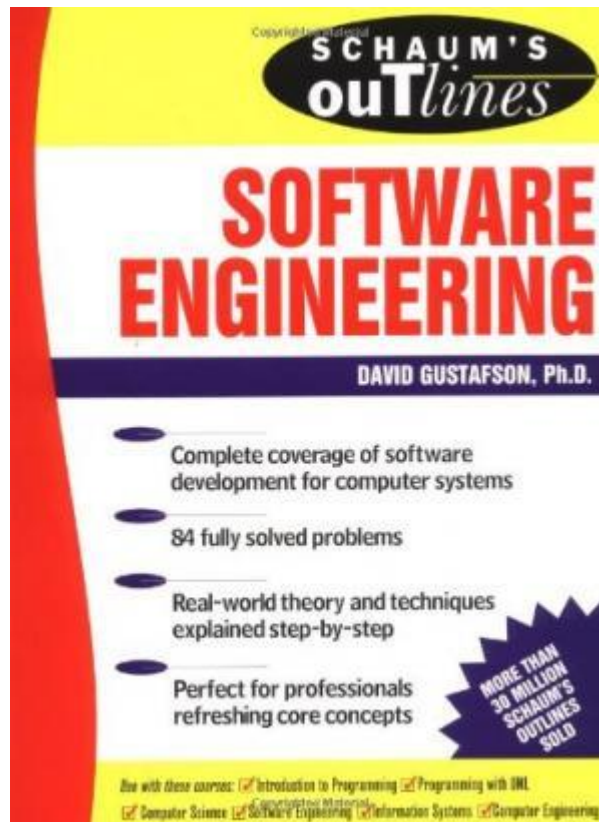
Software Engineering topics:

Software Engineering Models, Software Development Life Cycle, Requirement Engineering, Software Design tools, Software Design Strategies, Software Design levels, Software Project Management, Software Management activities, Software Management Tools, Software Testing approaches, Quality Assurance Vs. Quality control, Manual Testing, Software Maintenance, Software Re-engineering and Software Development Tool such as CASE Tool.

Reference Books

<https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/Software-Engineering-9th-Edition-by-Ian-Sommerville.pdf>

<https://ps.b-ok.asia/dl/738770/09de81>



What is Software Engineering?

- The term software engineering is the product of two words, software, and engineering.
- The software is a collection of integrated programs.
- Software Engineering is an engineering branch related to the evolution of software product using well-defined scientific principles, techniques, and procedures. The result of software engineering is an effective and reliable software product.

What is Software Engineering?



Software Engineering is required due to the following reasons:

- To manage Large software
- For more Scalability
- Cost Management
- To manage the dynamic nature of software
- For better quality Management

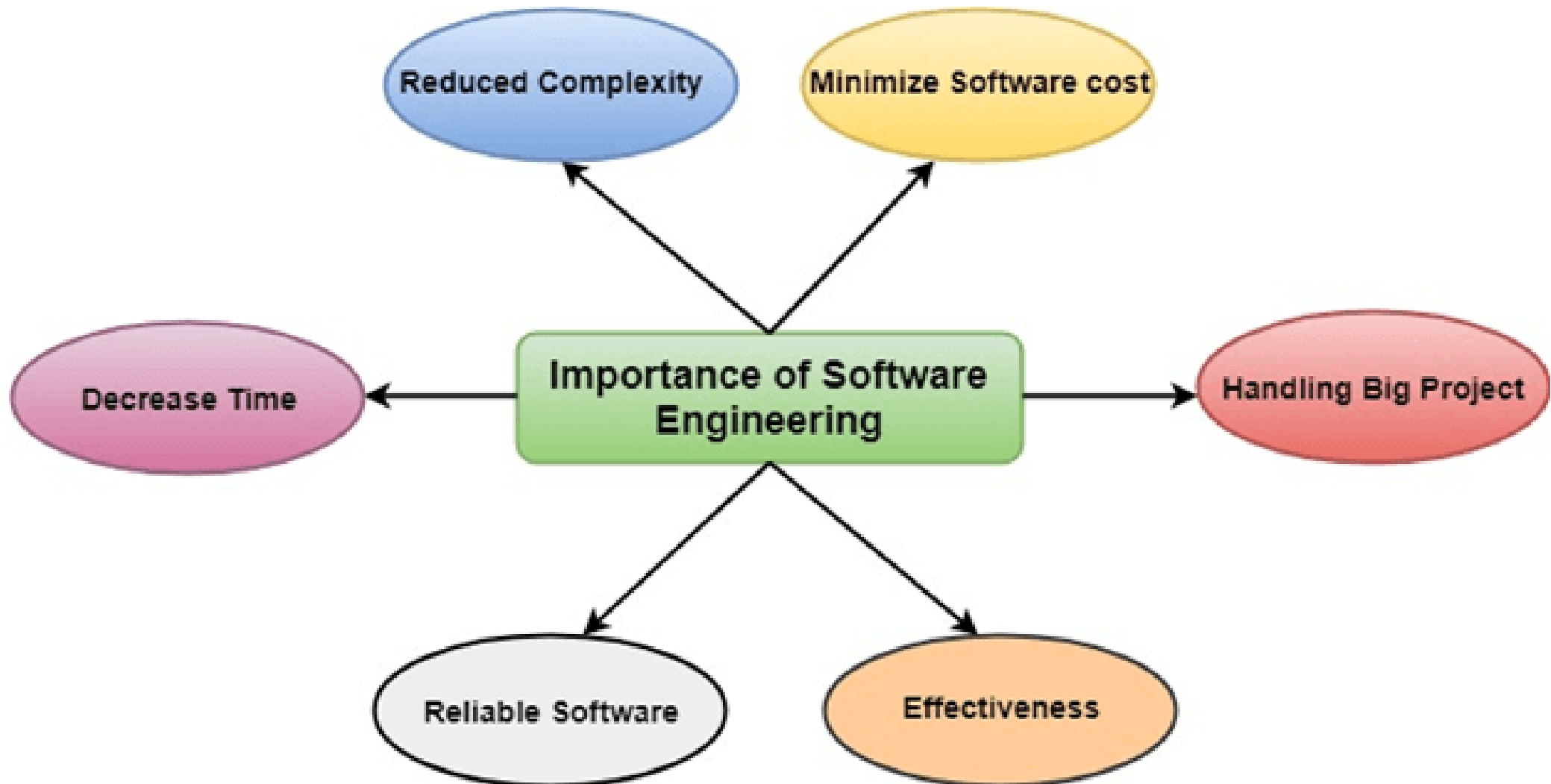
The necessity of **software engineering** appears because of a higher rate of progress in user requirements and the environment on which the program is working.

- Huge programming become extensive engineering; thus need to provide **simple scientific process**.
- **Cost of programming** remains high if the proper process is not adapted.
- Due to the **dynamic nature of programming** the quality of the software is continually changing, new upgrades need to be done in the existing one.
- Better procedure of software development provides a better and **quality software product**.

Characteristics of a good software engineer:

- Exposure to systematic methods, i.e., familiarity with software engineering principles.
- Good technical knowledge of the project range (Domain knowledge).
- Good programming abilities.
- Good communication skills. These skills comprise of oral, written, and interpersonal skills.
- High motivation.
- Sound knowledge of fundamentals of computer science.
- Intelligence.
- Ability to work in a team

Importance of Software Engineering



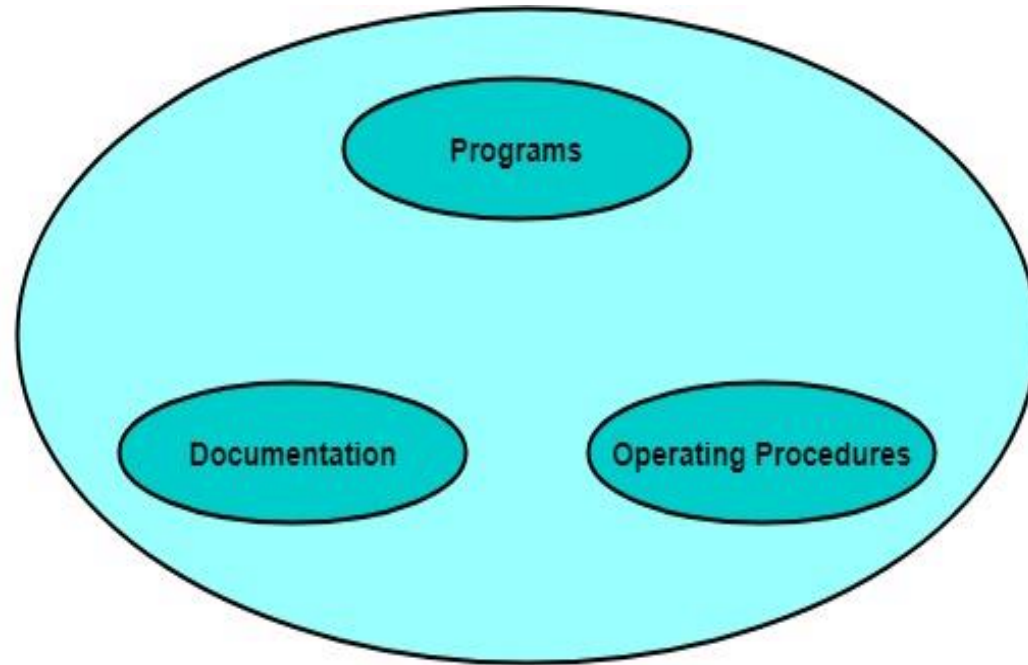
Software Process

A **software process** is the set of activities and associated outcome that produce a software product.

There are four key process activities, which are common to all software processes. These activities are:

- **Software specifications:** The functionality of the software and constraints on its operation must be defined.
- **Software development:** The software to meet the requirement must be produced.
- **Software validation:** The software must be validated to ensure that it does what the customer wants.
- **Software evolution:** The software must evolve to meet changing client needs.

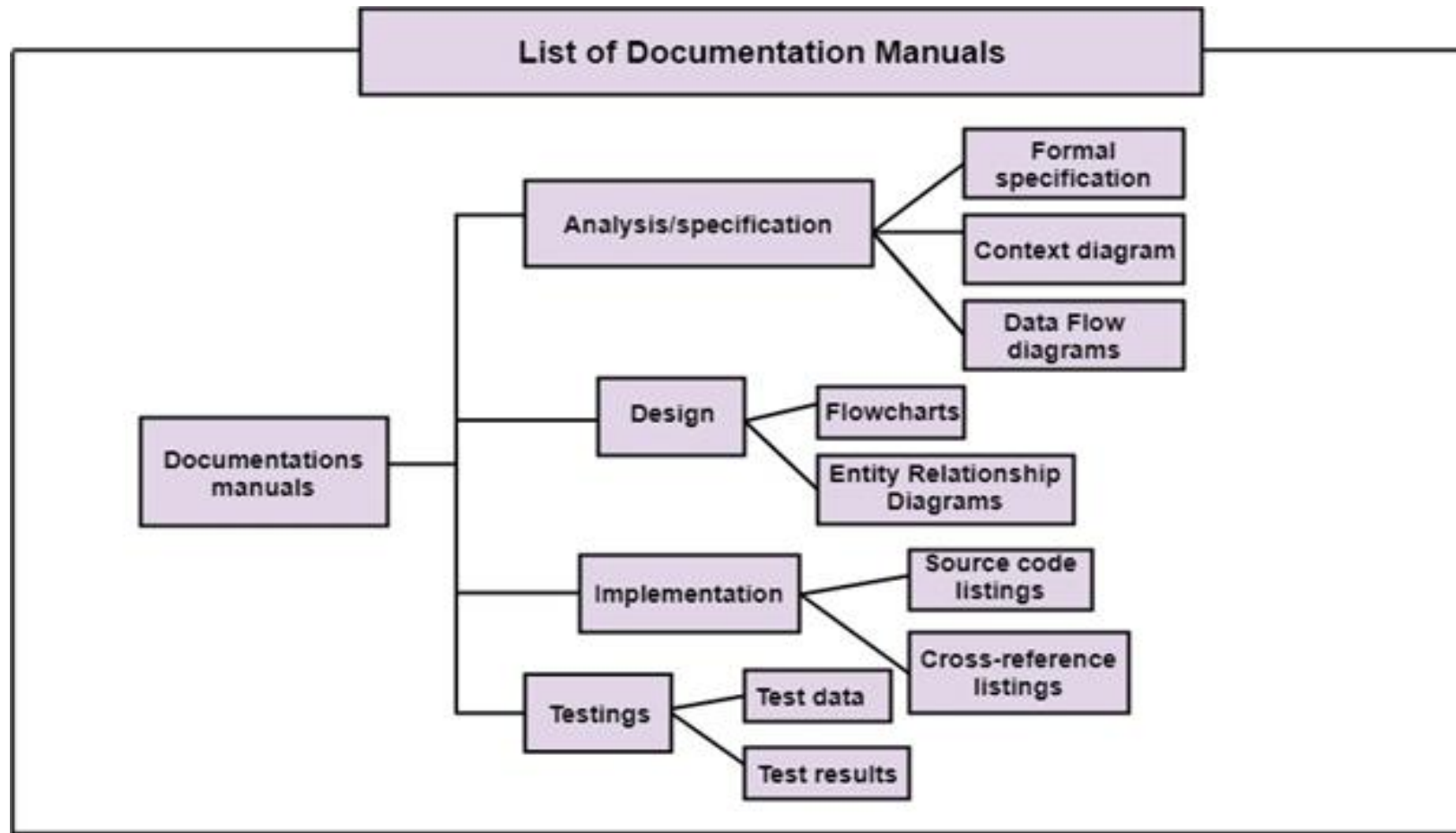
Software is more than programs. Any program is a subset of software, and it becomes software only if documentation & operating procedures manuals are prepared.



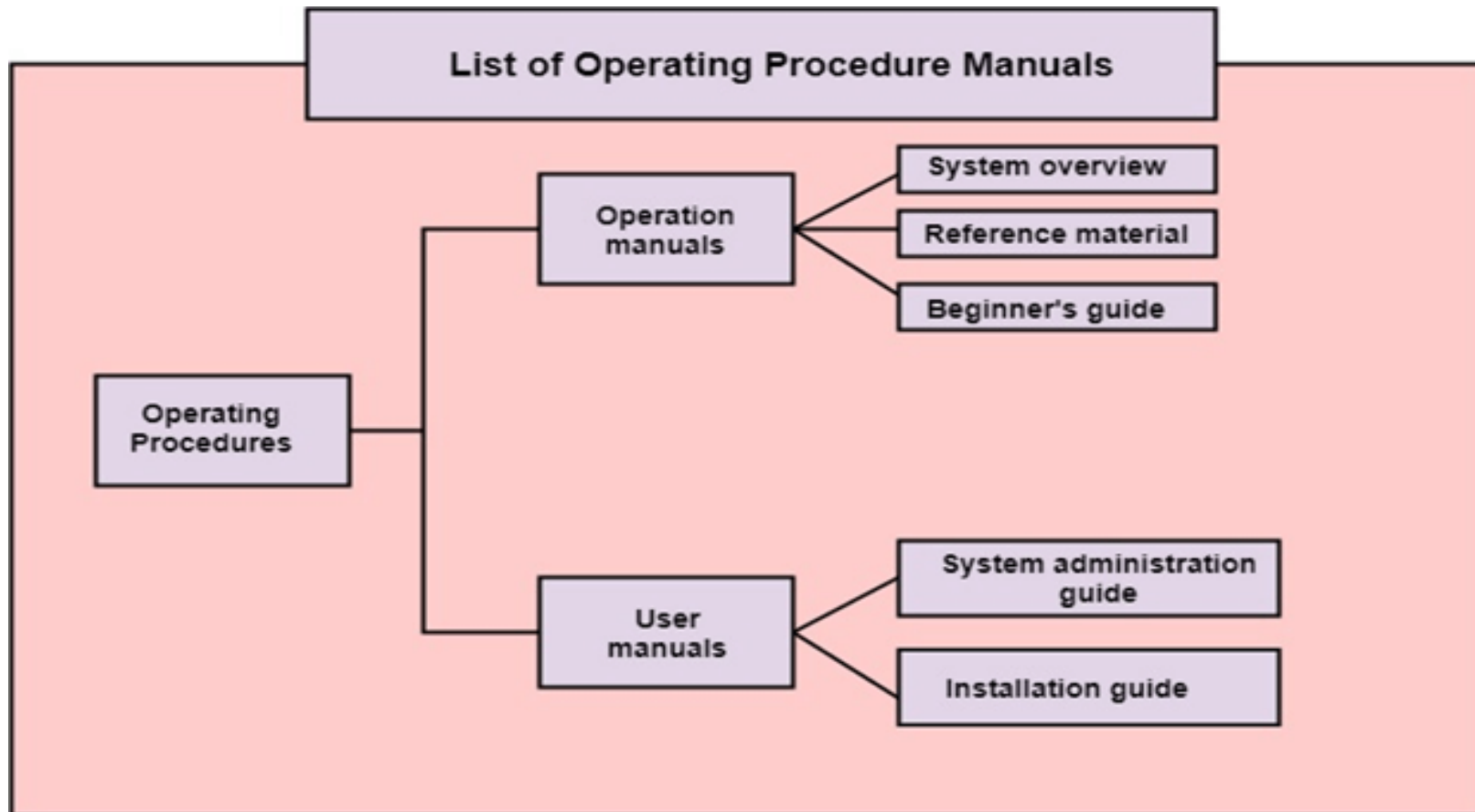
✓ Software = Program + Documentation + Operating Procedures

Fig:Components of Software

Examples of documentation manuals are: Data Flow Diagram, Flow Charts, ER diagrams, etc.



Example of operating system procedures manuals is: installation guide, Beginner's guide, reference guide, system administration guide..



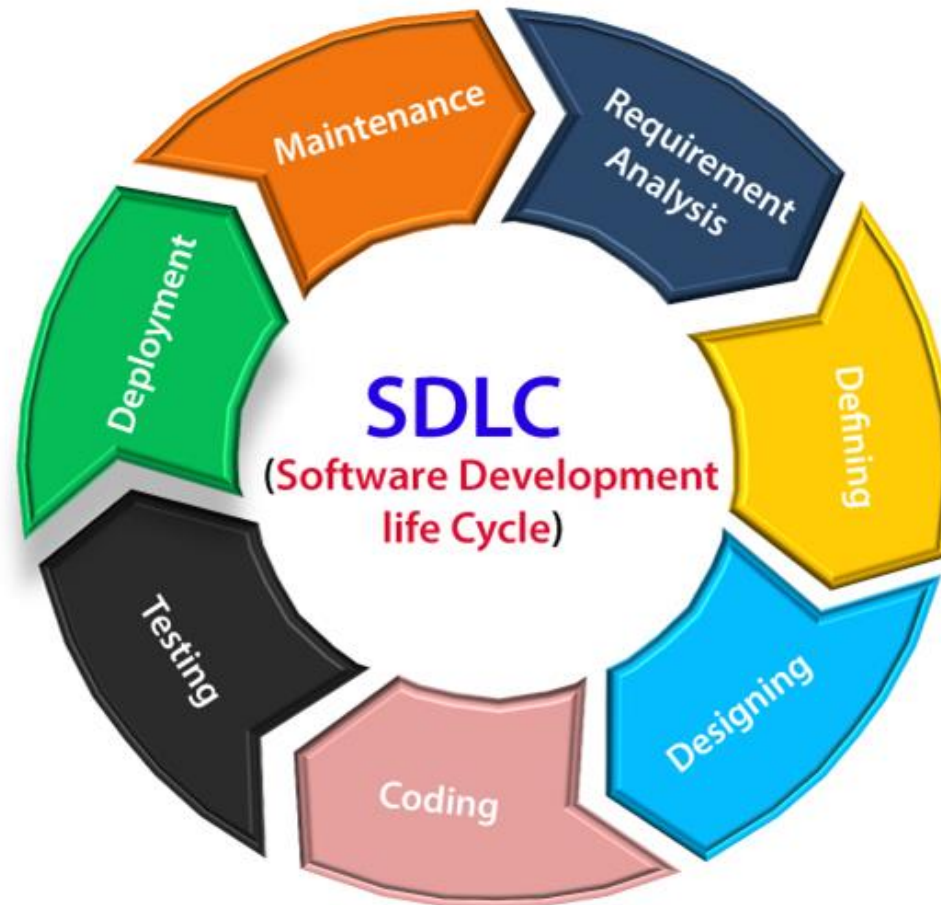
Software Development Life Cycle (SDLC)

A software life cycle model (also termed process model) is a pictorial and diagrammatic representation of the software life cycle.

- A life cycle model represents all the methods required to make a software product transit through its life cycle stages.
- Without using an exact life cycle model, the development of a software product would not be in a systematic and disciplined manner. When a team is developing a software product, there must be a clear understanding among team representative about when and what to do. Otherwise, it would point to chaos and project failure.

SDLC Cycle represents the process of developing software; includes the following steps:

* 6 Stage of SDLC



Stage 1: Planning and requirement analysis

- The senior members of the team perform it with inputs from all the stakeholders and domain experts or SMEs in the industry.
- Planning for the quality assurance requirements and identifications of the risks associated with the projects is also done at this stage.
- Business analyst and Project organizer set up a meeting with the client to gather all the data like what the customer wants to build, who will be the end user, what is the objective of the product. Before creating a product, a core understanding or knowledge of the product is very necessary.

Stage 2: **Defining** Requirements

- Once the requirement analysis is done, the next stage is to certainly represent and document the software requirements and get them accepted from the project stakeholders.
- This is accomplished through "SRS"- Software Requirement Specification document which contains all the product requirements to be constructed and developed during the project life cycle.

Stage 3: Designing the Software

- The next phase is about to bring down all the knowledge of requirements, analysis, and design of the software project. This phase is the product of the last two, like inputs from the customer and requirement gathering.

Stage 4: Developing the project

- In this phase of SDLC, the actual development begins, and the programming is built. The implementation of design begins concerning writing code. Developers have to follow the coding guidelines described by their management and programming tools like compilers, interpreters, debuggers, etc. are used to develop and implement the code.

Stage 5: Testing

- After the code is generated, it is tested against the requirements to make sure that the products are solving the needs addressed and gathered during the requirements stage.
- During this stage, unit testing, integration testing, system testing, acceptance testing are done.

Stage 6: Deployment

- Once the software is certified, and no bugs or errors are stated, then it is deployed.
- Then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment.
- After the software is deployed, then its maintenance begins.

Stage 7: Maintenance

- Once when the client starts using the developed systems, then the real issues come up and requirements to be solved from time to time.
- This procedure where the care is taken for the developed product is known as maintenance.

End of Topic 1