



Unit – 01: Introduction to Software Engineering

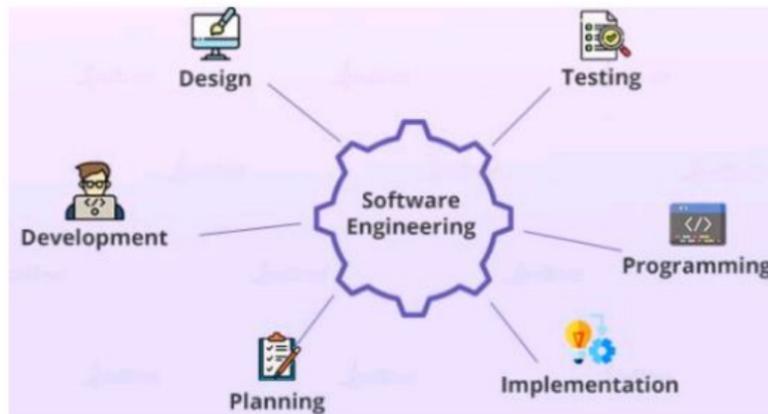
- Introduction to software engineering.
- Lifecycle.
- Process models:
 - Perspective Process Models.
 - Specialized Process Models.
 - The Waterfall model or Linear Sequential model.
 - The Prototyping Model.
 - Spiral Model.
- Software Engineering as Layered Approach and its characteristics.
- Types of software.
 - Traditional v/s Agile Processes and its importance.
 - Selection criteria for software process model.

Questions to be discussed:

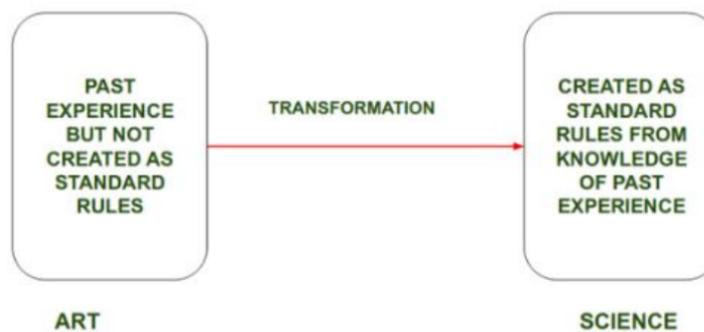
1. What is software engineering? Is it art, craft or a science? Discuss.
2. Why Software Engineering required? What are the goal of software engineering?
3. What are the component of a software? Discuss how a software differs from a program.
4. What is software life cycle? Discuss the generic waterfall model.
5. Sketch a neat diagram of spiral model of software life cycle.
6. Compare the waterfall model and the spiral model of software development.
7. Difference between Traditional and Agile Software Development.

What is software engineering? Is it art, craft or a science? Discuss.

- It is the process of analyzing user needs and designing a software that will satisfy these needs.
- Software engineering is the application of engineering principles to software development.
- In simple words, we can say that software engineering is the process of designing, developing, and maintaining software systems.
- It is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures.
- The outcome of software engineering is an efficient and reliable software product.



- Thus, like other Engineering Discipline, Software Engineering is a Science that is transformed from an Art.



What are the component of a software? Discuss how a software differs from a program?

Software:

- Software is the set of instructions in the form of programs to govern the computer system and to process the hardware components.
- To produce a software product the set of activities is used.
- This set is called a software process.



Components of Software:

- There are three components of the software:
 1. Program,
 2. Documentation, and
 3. Operating Procedures.

Program: A computer program is a list of instructions that tell a computer what to do.

Documentation: Source information about the product contained in design documents, detailed code comments, etc.

Operating Procedures: Set of step-by-step instructions compiled by an organization to help workers carry out complex routine operations.

Difference between Software and Program:

Software	Program
Software's are mainly dependent on operating system.	Programs are mainly dependent on compiler.
Software's can be a program that generally runs on computer.	Programs cannot be a software.
If software's are not present in computers, then computer is useless.	If programs are not present in computer, then also computer can function well because of operating system.
Software's can be downloaded on computer using internet without any need of program.	Program cannot run on computer without any software present in computer.
It requires more time to create software than program.	It requires less time to create program than software.
Examples of software includes Adobe Photoshop, Google Chrome, PowerPoint, Adobe Reader, etc.	Examples of program includes Web browsers, word processors, video games, etc.

Why is Software Engineering required?

Software Engineering is required due to the following reasons:

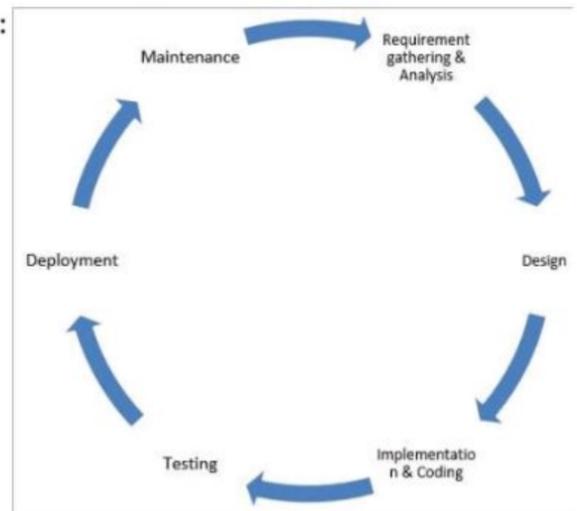
- To manage Large software
- For more Scalability
- Cost Management
- To manage the dynamic nature of software
- For better quality Management

Goals of Software Engineering:

- The goals of software engineering are to produce high-quality software that is reliable, efficient, and easy to maintain.
- Also being developed in a way that is cost-effective and meets the needs of the users.
- Some specific goals of software engineering include:
 - Producing software that is reliable and free from defects
 - Developing software that is efficient and performs well
 - Creating software that is easy to maintain and modify
 - Developing software in a cost-effective manner
 - Meeting the needs of the users.

Software Development Life Cycle (SDLC):

- SDLC stands for Software Development Life Cycle.
- It is a process used by the software industry to design, develop and test high quality software.
- The aims of SDLC is to produce a high-quality software that meets customer expectations.
- It consists of a detailed plan describing how to develop, maintain, replace and alter a specific software.
- There are various phases in SDLC which are given below:
 1. Requirement gathering and analysis
 2. Design
 3. Implementation or coding
 4. Testing
 5. Deployment
 6. Maintenance



Requirement Gathering and Analysis

- During this phase, all the relevant information is collected from the customer to develop a product as per their expectation.

Design:

- In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

Implementation or Coding

- Coding starts once the developer gets the Design document.
- The Software design is translated into source code.



- All the components of the software are implemented in this phase.

Testing

- Testing starts once the coding is complete and the modules are released for testing.
- In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

Deployment

- Once the product is tested, it is deployed in the production environment.

Maintenance

- After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

What is a software process model?

- A software process model is an abstraction of the software development process.
- The models specify the stages and order of a process.
- So, think of this as a representation of the order of activities of the process and the sequence in which they are performed.
- A model will define the following:
 1. The tasks to be performed
 2. The input and output of each task
 3. The pre and post-conditions for each task
 4. The flow and sequence of each task

Prescriptive process models:

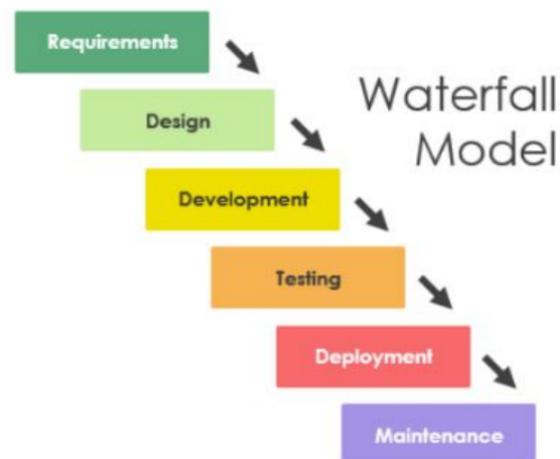
- As we know that the descriptive model describes the history of how a particular software system was developed.
- While a prescriptive model prescribes how a new software system should be developed.
- The name 'prescriptive' is given because the model prescribes a set of activities, actions, tasks, quality assurance and change the mechanism for every project.
- There are three types of prescriptive process models:
 1. The Waterfall Model
 2. Incremental Process model
 3. RAD model

Types of software process models

- There are many kinds of process models for meeting different requirements.
- The most popular and important SDLC models are as follows:
 1. Waterfall model
 2. Prototype model
 3. Spiral model

Waterfall Model(Linear Sequential Model):

- Waterfall model is the very first model that is used in SDLC.
- The Waterfall model was introduced by Winston w. Royce in 1970.
- It is also known as the linear sequential model or 'Classic life cycle model'.
- In this model, the outcome of one phase is the input for the next phase.
- Development of the next phase starts only when the previous phase is complete.
- This model is used for the small projects.
- It has the following phases:
 1. Requirements
 2. Design
 3. Implementation
 4. Testing
 5. Deployment
 6. Maintenance



Advantages of the Waterfall Model:

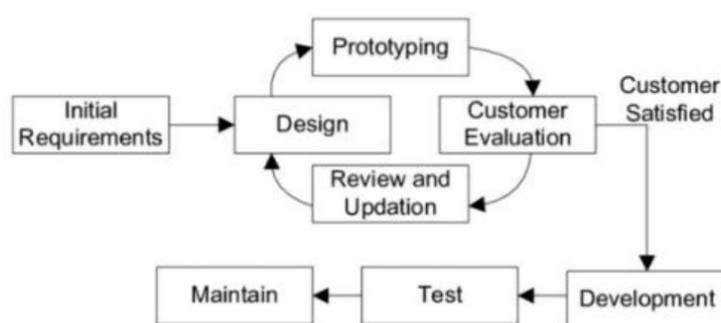
- It is the simple model which can be easily understood.
- In this model all the phases are done step by step.
- Here, each phases are well defined, this leads to no complexity and the project easily manageable.

Disadvantages of Waterfall model:

- Waterfall model is time-consuming & cannot be used in the short duration projects.
- In this model a new phase cannot be started until the ongoing phase is completed.
- It is a poor model for long projects.

Prototype Model:

- Prototype model is a systems development method in which a prototype is built.
- It requires that before development of actual software, a working prototype of the system should be built.
- Dummy functions are used to create prototypes.
- This is a valuable mechanism for understanding the customers' needs.
- Feedbacks are implemented and the prototype is again reviewed by the customer for any change.
- This process goes on until the model is accepted by the customer.
- Once the customer approves the prototype, it is used as a requirement for building the actual software.
- The actual software is build using the Waterfall model approach.



PROTO TYPE MODEL

Advantages of Prototype Model:

- Prototype model reduces the cost and time of development as the defects are found much earlier.
- Missing features can be identified in the evaluation phase.
- Involvement of a customer from the initial stage reduces any confusion in the requirement.

Disadvantages of Prototype Model:

- Since the customer is involved in every phase, the customer can change the requirement of the end product which increases the complexity of the scope and may increase the delivery time of the product.

Spiral Model:

- It is a SDLC model that provides a systematic and iterative approach to software development.
- Spiral Model is a risk-driven model, that means it focus is on managing risk through multiple iterations.
- The Spiral Model includes iterative and prototype approach.
- The spiral model, proposed by Barry Boehm in 1986.
- Spiral model phases are followed in the iterations.
- Spiral Model has four phases:
 1. Planning
 2. Risk Analysis
 3. Engineering
 4. Evaluation





Advantages of Spiral Model:

- Risk Analysis is done extensively using the prototype models.
- Any enhancement or change in the functionality can be done in the next iteration.

Disadvantages of Spiral Model:

- The spiral model is best suited for large projects only.
- The cost can be high as it might take a large number of iterations which can lead to high time to reach the final product.

Compare the waterfall model and the spiral model of software development:

Waterfall Model	Spiral Model
The Waterfall model is simple and easy.	The spiral model is more complex.
The waterfall model works in a sequential method.	While the spiral model works in the evolutionary method.
The waterfall model is adopted by customers.	While the spiral model is adopted by developers.
The waterfall model is applicable for small projects.	While the Spiral model is used for large projects.
There is high amount risk in waterfall model.	There is low amount risk in spiral model.
Waterfall model is comparatively inexpensive.	While cost of spiral model is very expensive.
It requires least maintenance.	It requires typical maintenance.

Traditional Software Development:

- Traditional software development is the software development process used to design and develop simple software.
- It is used when the security and many other factors of the software are not much important.
- It is used by freshers to develop the software.
- It consists of five phases:
 1. Requirements analysis
 2. Design
 3. Implementation
 4. Coding and Testing
 5. Maintenance



Agile Software Development:

- Agile software development is the software development process used to design complicated software.
- It is used when the software is quite sensitive and complicated.
- It is used when security is much more important.
- It is used by professionals to develop the software.
- It consists of three phases:
 1. Project initiation
 2. Sprint planning
 3. Demos

Difference between Traditional and Agile Software Development:

Traditional Software Development	Agile Software Development
It is used to develop simple software.	It is used to develop complicated software.
In this methodology, testing is done once the development phase is completed.	In this methodology, testing and development processes are performed concurrently.
It follows a linear organization structure.	It follows an iterative organizational structure.
It provides less security.	It provides high security.
Client involvement is less.	Client involvement is high.
It supports a fixed development model.	It supports a changeable development model.
It is used by freshers.	It is used by professionals.
Development cost is less using this methodology.	Development cost is high using this methodology.
It majorly consists of five phases.	It consists of only three phases.



Selection criteria for software process model:

- Choosing the right software process model for your project can be difficult.
- If you know your requirements well, it will be easier to select a model that best matches your needs.
- You need to keep the following factors in mind when selecting your software process model:
 1. Project requirements
 2. Project size
 3. Project complexity
 4. Cost of delay
 5. Customer involvement
 6. Familiarity with technology
 7. Project resources

