SCSVMV

(Deemed to be University u/s 3 of UGC act 1956)

SOFTWARE ENGINEERING

UNIT-I

Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC)

Objectives

- The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

- It aims to Build solutions using different technologies, architectures and life-cycle approaches in the context of different organizational structures

Software Development Life Cycle (SDLC)

Outcomes

- One can able to develop and conduct appropriate experimentation, analysis and data interpretation, and use engineering judgment to draw conclusions in choosing an apt software development model
- One can able to satisfy the customer expectations, reaches completion within time and cost evaluations, and works effectively and efficiently in the current and planned Information Technology infrastructure by choosing a Suitable software development model.
- One can able to acquire and apply new knowledge as needed, using appropriate learning strategies

Software Development Life Cycle (SDLC)

Pre-requisites

- Basic Knowledge of systematic and operational language
- Need Basic Knowledge of Sound Engineering Principles

Software Development Life Cycle (SDLC)

Terminology used

- Agile
- RAD
- ARA

Software Development Life Cycle (SDLC)

Definition

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software.

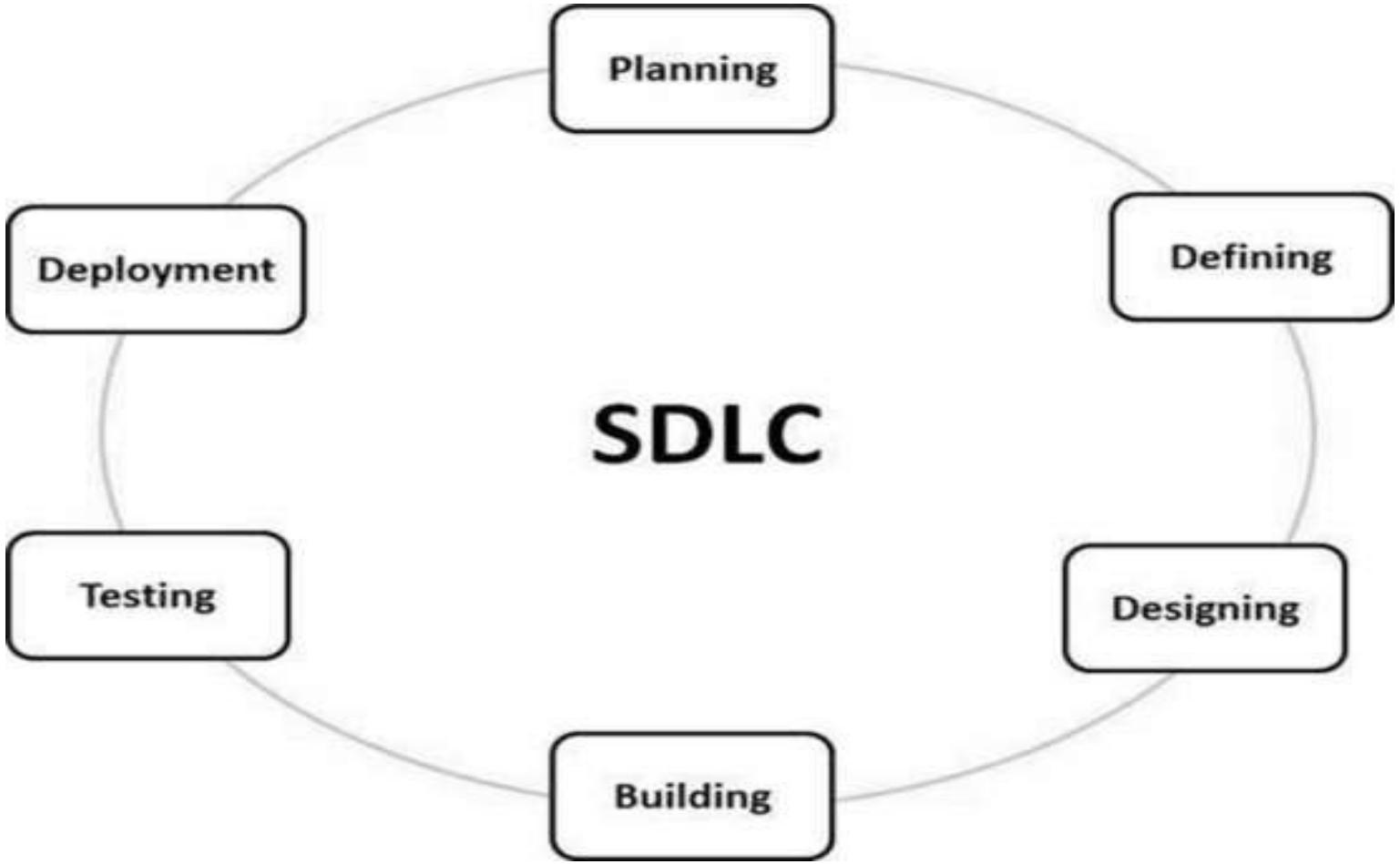
Software Development Life Cycle (SDLC)

- SDLC is a process followed for a software project, within a software organization.

- It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software.

- The life cycle defines a methodology for improving the quality of software and the overall development process.

Stages of a typical SDLC



Stage 1: Planning and Requirement Analysis

- Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

- Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Defining Requirements

- Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts.

- This is done through an **SRS (Software Requirement Specification)** document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture

- SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.
- This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.
- A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

- In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

- Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product

- This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC.

- However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance

- Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

- Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

Software Development Life Cycle Models

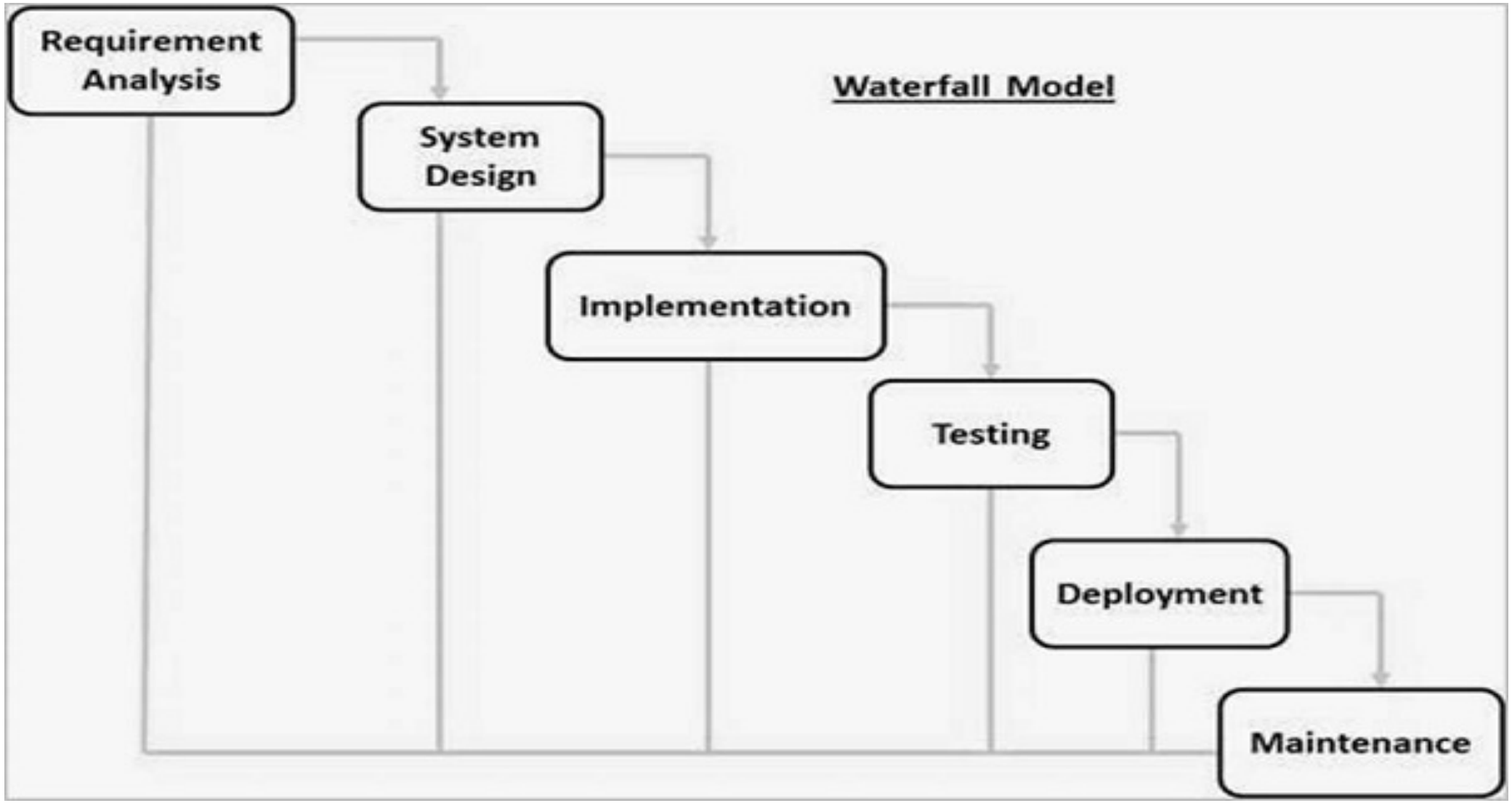
- Waterfall Model (**For sample, Only this model is explained in detail**)
- Iterative Model
- Evolutionary Model
- Prototype Model
- Spiral Model
- RAD Model
- Agile Model
- Incremental Model

Waterfall Model

- The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.
- The Waterfall model is the earliest SDLC approach that was used for software development.
- The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

Waterfall Model

- Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project.
- In "The Waterfall" approach, the whole process of software development is divided into separate phases.
- In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.


Waterfall Model
Requirement Analysis
System Design
Implementation
Testing
Deployment
Maintenance

Sequential phases in Waterfall model

- **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

Sequential phases in Waterfall model

- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors.

Some situations where the use of Waterfall model is most appropriate are −

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model - Advantages

- The advantages of waterfall development are that it allows for departmentalization and control.
- A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.
- Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance.
- Each phase of development proceeds in strict order.

Waterfall Model - Advantages

Some of the major advantages of the Waterfall Model are as follows −

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Waterfall Model - Disadvantages

- The disadvantage of waterfall development is that it does not allow much reflection or revision.

- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

Waterfall Model - Disadvantages

The major disadvantages of the Waterfall Model are as follows −

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

Comparison of Various Process Models

Parameter / Process Model→	Waterfall Model	Incremental Model	Prototype Model	Rad Model	Spiral Model	Agile Model
Clear Requirement Specifications	Initial level	Initial level	At medium level	Initial level	Initial level	Change incrementally
Feedback from user	No	No	Yes	No	No	No
Speed to change	Low	High	Medium	No	High	High
Predictability	Low	Low	High	Low	Medium	High
Risk identification	At initial level	No	No	No	Yes	Yes
Practically implementation	No	Low	Medium	No	Medium	High
Loom	Systematic sequence	Iterative sequence	Priority on customer feedback	Use readymade component	Identification of risk at each stage	Highly customer satisfaction and incremental development[09]
Any variation done	Yes-v model	No	No	No	Yes-win win spiral[6]	No
Understandability	Simple	Intermediate	Intermediate	Intermediate	Hard	Much complex
Precondition	Requirement clearly defined	Core product should clearly define	Clear idea of Quick Design	Clean idea of Reuse component	No	No
Usability	Basic	Medium	High	Medium	Medium	Most use now a days
Customer priority	Nil	Nil	Intermediate	Nil	Intermediate	High
Industry approach	Basic	Basic	Medium	Medium	Medium	High
Cost	Low	Low	High	very high	Expensive	Much Expensive
Resource organization	Yes	Yes	Yes	Yes	No	No
Elasticity	No	No	Yes	Yes	No	Very high

Summary

- There are many existing models for developing systems for different sizes of projects and requirements
- Waterfall model and spiral model are used commonly in developing systems
- Each model has advantages and disadvantages for the development of systems , so each model tries to eliminate the disadvantages of the previous model

Questions and cross questions in the topic

- If requirements are easily understandable and defined then which model is best suited?
- Project risk factor is considered in which model.
- What is the meaning of requirement elicitation in software engineering?
- How many numbers of maturity levels in CMM are available?
- Design phase is followed by.
- Which software is used to control products and systems for the consumer and industrial markets?
- Abbreviate the term CMMI.
- First level prototype is evaluated by?
- Which is the Bedrock that supports software Engineering in layered technology.
- The physical connections between elements of the OO design represent?
- In which level goal, objective, work tasks, work products and other activities of software process are carried out.

Final Conclusion

- The structure imposed by the **SDLC** is specifically designed to maximize the probability of a successful software development effort.
- It is critical for the project manager to establish and monitor control objectives during each **SDLC** phase while executing projects.
- On the basis of the features for a particular software project one can decide which of these software development life cycle models should be chosen for that particular project.
- Selecting the correct life cycle model is extremely important in a software industry as the software has to be delivered within the time deadline and should also have the desired quality