Object Oriented Software Engineering (OOSE)

22CS017

Software Process

Index

- Software Process
- Software Process Model
- Need of Software Process Model
- SDLC Model
- Types of SDLC Models
- Water Fall Model
- Practice Questions

Software Process

A software process is the set of activities and associated outcome that produce a software product. Software engineers mostly carry out these activities. These are four key process activities, which are common to all software processes. These activities are:

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

Software Process Models

A software process model is a specified definition of a software process, which is presented from a particular perspective. Models, by their nature, are a simplification, so a software process model is an abstraction of the actual process, which is being described. Process models may contain activities, which are part of the software process, software product, and the roles of people involved in software engineering.

1. **A workflow model:** This shows the series of activities in the process along with their inputs, outputs and dependencies. The activities in this model perform human actions.

2. **A dataflow or activity model:** This represents the process as a set of activities, each of which carries out some data transformations. It shows how the input to the process, such as a specification is converted to an output such as a design. The activities here may be at a lower level than activities in a workflow model. They may perform transformations carried out by people or by computers.

3. **A role/action model:** This means the roles of the people involved in the software process and the activities for which they are responsible

Need for Process Model

- The software development team must decide the process model that is to be used for software product development and then the entire team must adhere to it. This is necessary because the software product development can then be done systematically. Each team member will understand what is the next activity and how to do it. Thus process model will bring the definiteness and discipline in overall development process. Every process model consists of definite entry and exit criteria for each phase. Hence the transition of the product through various phases is definite.
- If the process model is not followed for software development then any team member can perform any software development activity, this will ultimately cause a chaos and software project will definitely fail without using process model, it is difficult to monitor the progress of software product. Thus process model plays an important rule in software engineering.

SDLC Models

- Software Development life cycle (SDLC) is a spiritual model used in project management that defines the stages include in an information system development project, from an initial feasibility study to the maintenance of the completed application.
- There are different software development life cycle models specify and design, which are followed during the software development phase. These models are also called "**Software Development Process Models**." Each process model follows a series of phase unique to its type to ensure success in the step of software development.
- **Here, are some important phases of SDLC life cycle:**

SDLC Models

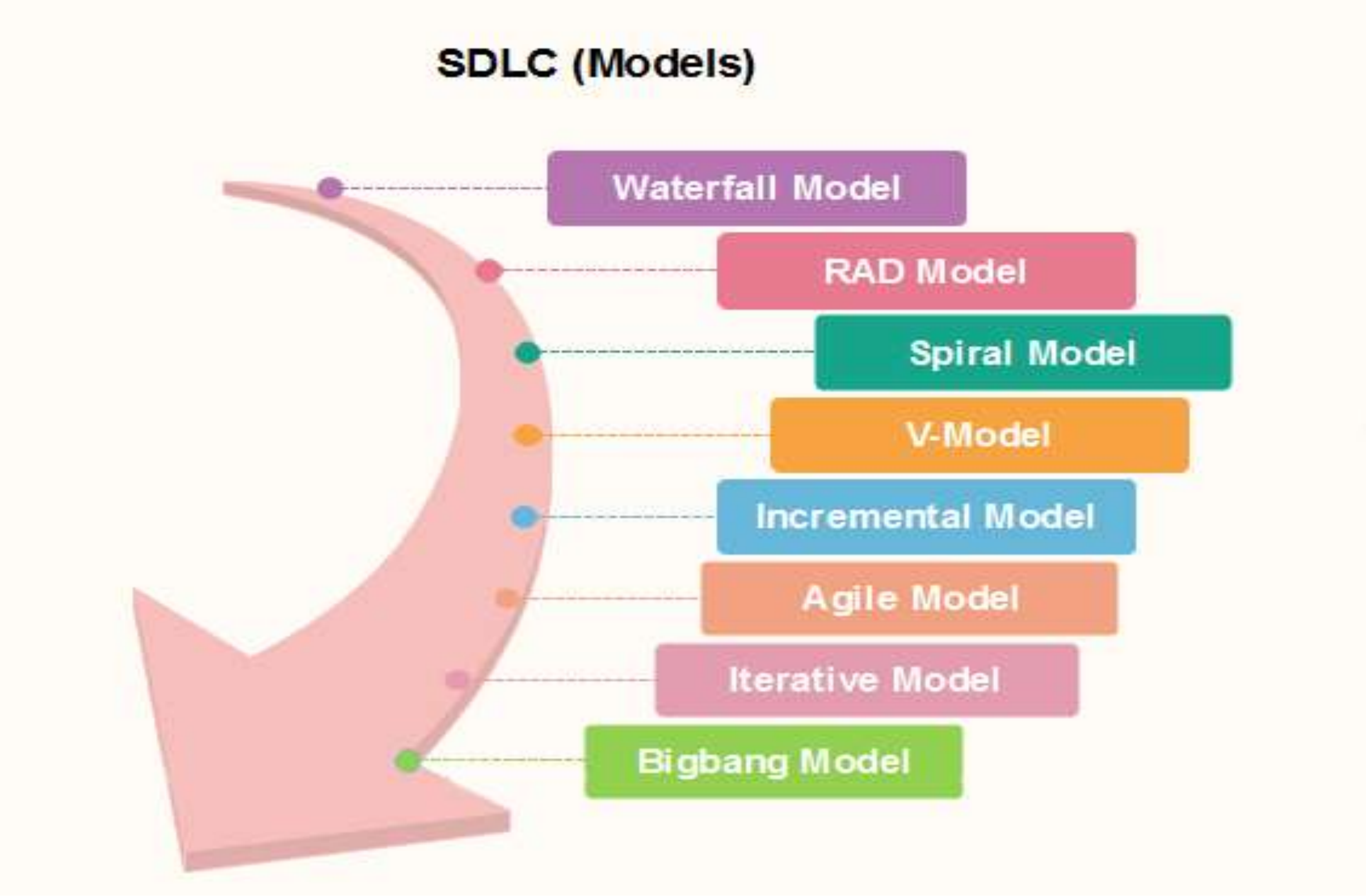

Figure 1: -SDLC Models

Waterfall Model

Winston Royce introduced the Waterfall Model in 1970.This model has five phases:

  - Requirements analysis and specification,
  - Design
  - Implementation, and
  - Unit testing,
  - Integration and system testing, and
  - Operation and maintenance.

- The steps always follow in this order and do not overlap. The developer must complete every phase before the next phase begins. This model is named "**Waterfall Model**", because its diagrammatic representation resembles a cascade of waterfalls.

Waterfall Model

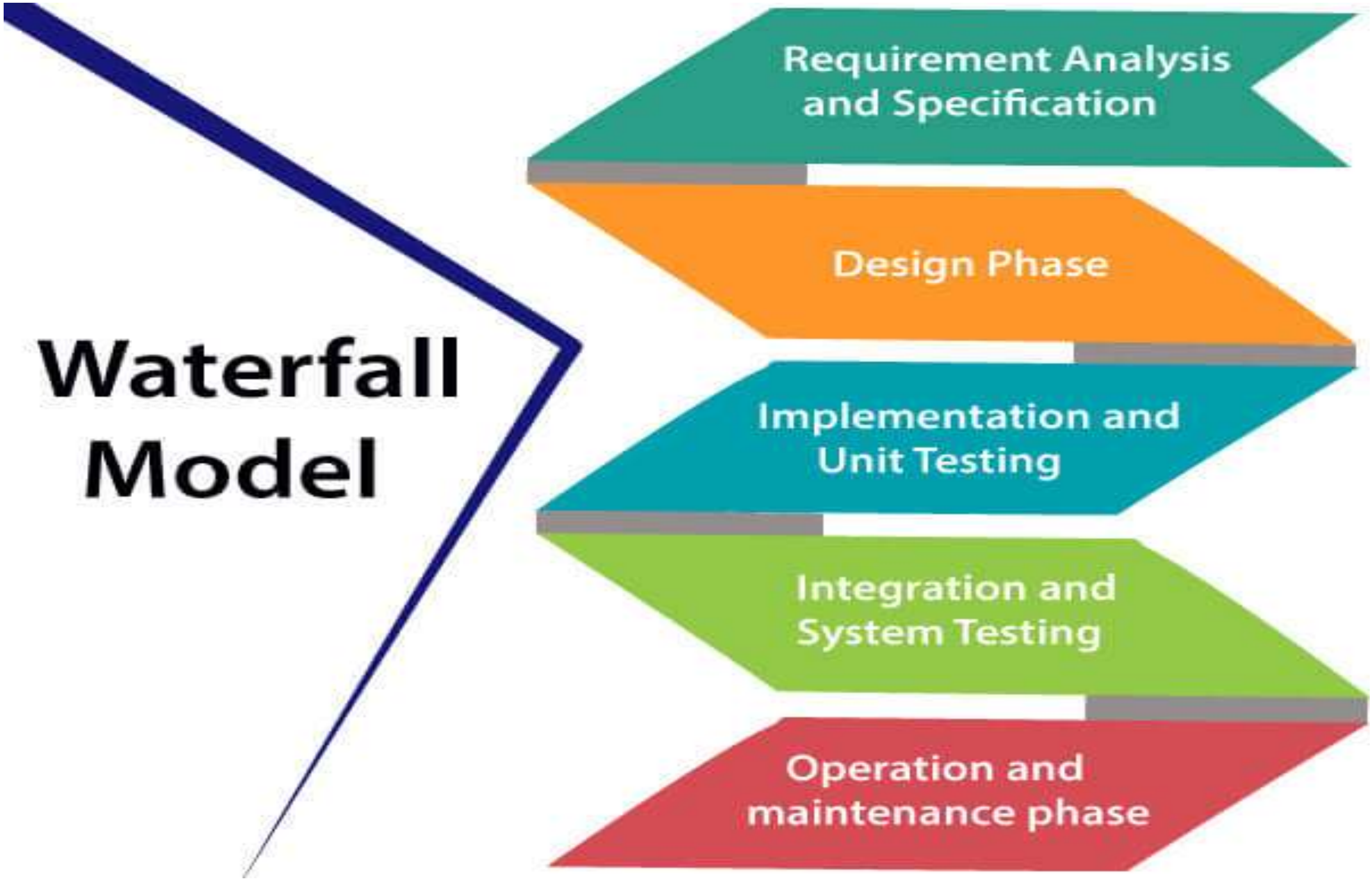
Waterfall Model
Requirement Analysis and Specification
Design Phase
Implementation and Unit Testing
Integration and System Testing
Operation and maintenance phase

Figure 2: -Water Fall Model

Waterfall Model

- **Requirements analysis and specification phase:** The aim of this phase is to understand the exact requirements of the customer and to document them properly. Both the customer and the software developer work together so as to document all the functions, performance, and interfacing requirement of the software. It describes the "what" of the system to be produced and not "how."In this phase, a large document called **Software Requirement Specification (SRS)** document is created which contained a detailed description of what the system will do in the common language.
- **Design Phase:** This phase aims to transform the requirements gathered in the SRS into a suitable form which permits further coding in a programming language. It defines the overall software architecture together with high level and detailed design. All this work is documented as a Software Design Document (SDD).

Waterfall Model

- **Implementation and unit testing:** During this phase, design is implemented. If the SDD is complete, the implementation or coding phase proceeds smoothly, because all the information needed by software developers is contained in the SDD. During testing, the code is thoroughly examined and modified. Small modules are tested in isolation initially. After that these modules are tested by writing some overhead code to check the interaction between these modules and the flow of intermediate output.
- **Integration and System Testing:** This phase is highly crucial as the quality of the end product is determined by the effectiveness of the testing carried out. The better output will lead to satisfied customers, lower maintenance costs, and accurate results. Unit testing determines the efficiency of individual modules. However, in this phase, the modules are tested for their interactions with each other and with the system.
- **Operation and maintenance phase:** Maintenance is the task performed by every user once the software has been delivered to the customer, installed, and operational.

Waterfall Model

- **ADVANTAGES:**
  - Waterfall model works well for smaller projects where requirements are very well understood. This model is simple to implement also the number of resources that are required for it is minimal.
  - The requirements are simple and explicitly declared; they remain unchanged during the entire project development.
  - The start and end points for each phase is fixed, which makes it easy to cover progress.
  - The release date for the complete product, as well as its final cost, can be determined before development.
  - It gives easy to control and clarity for the customer due to a strict reporting system.
  - It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
  - In this model phases are processed and completed one at a time. Phases do not overlap.

Waterfall Model

- **DISADVANTAGES:**
  - Very less customer interaction is involved during the development of the product. In this model, the risk factor is higher, so this model is not suitable for more significant and complex projects.
  - This model cannot accept the changes in requirements during development.
  - It becomes tough to go back to the phase. For example, if the application has now shifted to the coding phase, and there is a change in requirement, It becomes tough to go back and change it.
  - Since the testing done at a later stage, it does not allow identifying the challenges and risks in the earlier phase, so the risk reduction strategy is difficult to prepare. Once an application is in the **testing** stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
  - No working software is produced until late during the life cycle.
  - High amounts of risk and uncertainty.
  - Poor model for big and complex projects.
  - Not suitable for the projects where requirements are at a moderate to high risk of changing.

Waterfall Model

- **When to use the waterfall model:**
  - This model is used only when the requirements are very well known, clear and fixed.
  - Technology is understood.
  - There are no ambiguous requirements
  - Ample resources with required expertise are available freely
  - The project is short.

FAQ

- Q.1 Changes made to an information system to add the desired but not necessarily the required features is called
  - (A) Preventative maintenance.
  - (B) Adaptive maintenance.
  - (C) Corrective maintenance.
  - (D) Perfective maintenance

- Q.2 The most important feature of spiral model is
  - (A) requirement analysis.
  - (B) risk management.
  - (C) quality management.
  - (D) configuration management.

FAQ

- Q.3 For a well understood data processing application it is best to use
  - (A) The waterfall model (B) prototyping model
  - (C) the incremental model (D) the spiral model

- Q.4 If every requirement stated in the Software Requirement Specification (SRS) has only one interpretation, SRS is said to be
  - (A) correct. (B) unambiguous.
  - (C) consistent. (D) verifiable.

- Q.5 In the spiral model ‘risk analysis’ is performed :
  - (A) In the first loop (B) in the first and second loop
  - (C) In every loop (D) before using spiral model

FAQ

- Q.6 Modifying the software to match changes in the ever changing environment is called
  - (A) adaptive maintenance (B) corrective maintenance
  - (C) perfective maintenance (D) preventive maintenance

- Q.7 Changes made to the system to reduce the future system failure chances is called
  - (A) Preventive Maintenance (B) Adaptive Maintenance
  - (C) Corrective Maintenance (D) Perfective Maintenance

Review

- Explain the use of different software model.
- Differentiate between increment and iterative model.
- What is V model.
- Explain the applications of prototype model.

Task of the day

- Design a SRS document for Restaurant Management System.

- **Scenario:** The system helps to manage the restaurant more effectively and efficiently by computerizing meal ordering, billing and inventory control. The system processes transaction and stores the resulting data. Reports will be generated from these data which help the manager to make appropriate business decisions for the restaurant.

Bibliography

- https://www.tutorialspoint.com/software_engineering/
- https://courses.cs.vt.edu/csonline/SE/Lessons/Qualities/index.html
- https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm
- https://www.geeksforgeeks.org/software-engineering-spiral-model/
- https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/

happy

LEARNING