

→ Different life cycle models are:

- (1) Classical waterfall model.
- (2) Iterative waterfall model.
- (3) prototyping model.
- (4) evolutionary model.
- (5) spiral model.
- (6) RAD model.
- (7) Agile model.

23-07-2016

Difference between process and methodology

① A software development process ~~has~~ ^{has} as much ~~process~~ ^{process} as methodology.

① A software development process has a much ~~broader~~ ^{broader} scope as compared to a software development methodology.

② A process usually describe all the activities starting from the inception of the software to its maintenance.

→ It also recommends specific methodology for carrying out each activity.

→ In contrast a methodology describes the steps to carry out only a single ~~or~~ or atleast a few individual activity.

Why to use a development process:

- The primary advantage of ~~giving~~ ^{following} a process is that it encourages development of a software in systematic and discipline manner.
- Adhering to a process is especially important to the development of professional software which requires team effort.
- When a software is developed by a team rather than by a individual programmer use of a life cycle model is inevitable.

Phase entry and exit criteria. (Imp)

Besides clearly identifying different phases in the life cycle of a software product, a life cycle model normally defines the phase entry and exit criteria for ~~every~~ every phase.

→ The phase entry and exit criteria is usually expressed as a set of conditions that needs to be satisfied for the phase to start or to complete.

→ As there is no clear specification of entry and exit criteria of every phase and if there is no life cycle model is followed then it is very difficult to chart the progress of the project.

There are different software life cycle models

① Classical waterfall model

The classical waterfall model is the most obvious way to develop software.

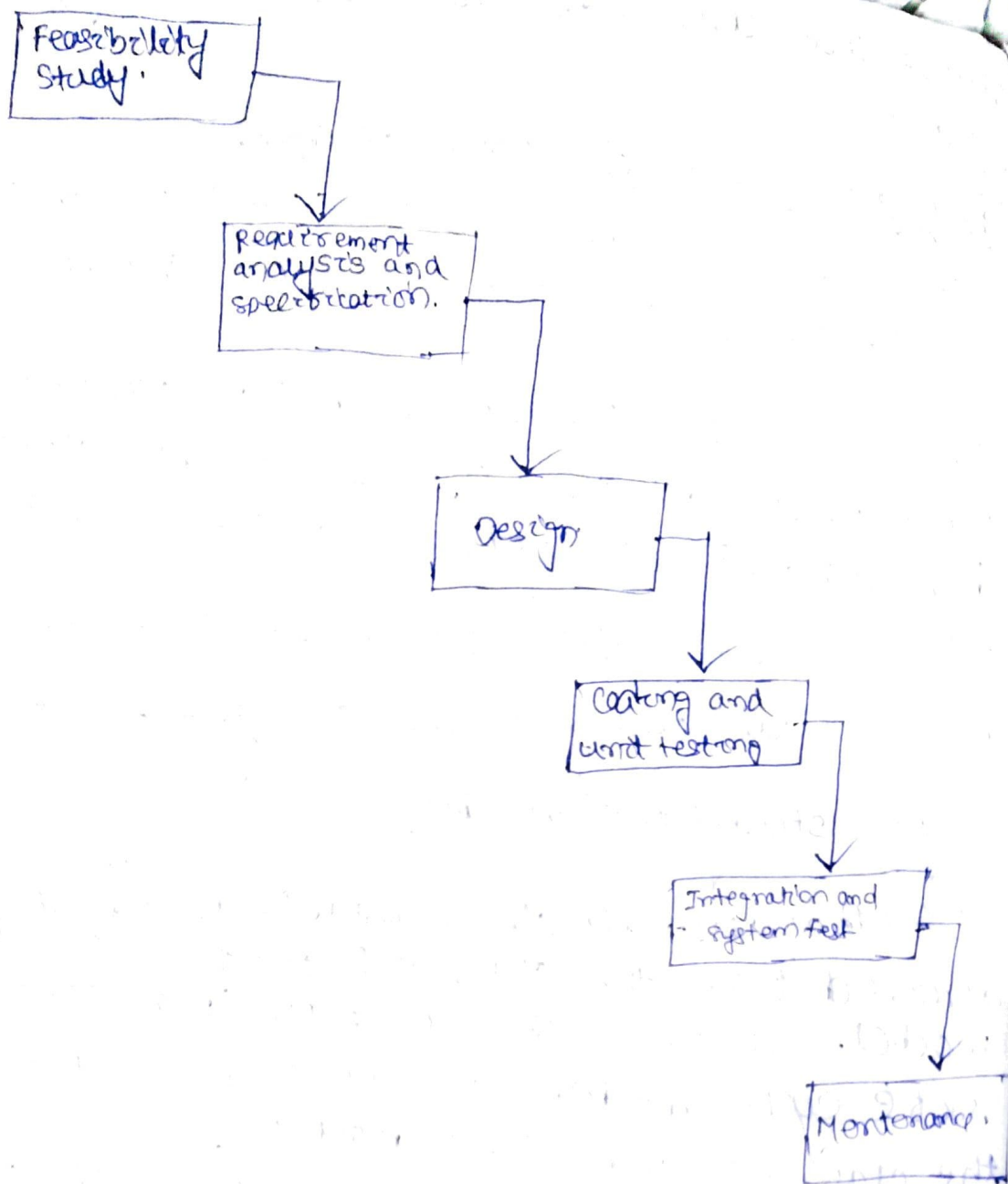
→ It is not a practical model in the sense that it can not be used in actual development of software projects thus we can consider this model to be a theoretical way of developing software.

→ why to study this model?

Ans → Because all other models are eventually derived from the classical waterfall model. So in order to appreciate other life cycle models we must first learn the classical waterfall model.

The classical waterfall model derives its name from the different phases namely feasibility study, requirement analysis and specification, design, coding and unit testing, integration and system testing and maintenance.

→ The name of this model is justified by its diagrammatic representation which resembles a cascade of waterfall.



As shown in the diagram the different phases starting from the feasibility study to the integration and system testing phase are known as development phase.

And the phase which starts after the completion of development phase is known as maintenance phase.

Feasibility study: The main aim of the feasibility study is to determine whether it would be financially and technically feasible to develop a product.

→ The following activities are done in the feasibility study phase.

- ① An abstract problem definition.
- ② Formulation of different solution strategies.
- ③ Analysis of alternative solution strategies.

Requirement analysis and specification:

The main aim of requirement analysis and specification stage is to understand the exact requirement of the customer and to document them properly.

→ This phase consists of two distinct activities namely (1) Requirement gathering and analysis (2) Requirement specification.

(1) Requirement gathering analysis: The goal of the requirement gathering analysis is to collect all relevant information regarding the product to be developed.
→ It must be consistent and concrete, because inconsistent requirements may lead to confusions.

→ This activities ^{is} began by collecting all the relevant data regarding the product from the users and customers through interviews and discussion.

→ It may also contain ^{some} ambiguous data. After all ambiguities, ~~at~~ inconsistencies and incompleteness have been resolved, the user requirement ^{are} systematically organised into a software requirement specification (SRS) document.

Requirement specification:

✓ The customer requirement identified during the requirement gathering and requirement analysis activity are organised into a SRS document.

→ The important component of these document are:
1. Functional requirement and non functional requirement and the goals of implementation.

Note

1 → Documenting the functional requirements involve the identification of the functions to be supported by the system.

→ Each function can be characterized by the input data, the processing required on the input data and the output data to be produced.

(2) → The non-functional requirement identify the performance requirement, the required standards to be followed etc.

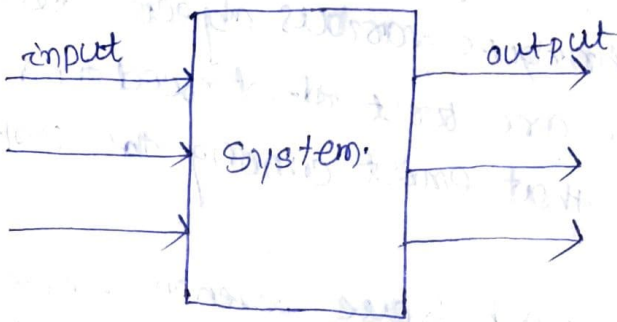
→ After the SRS document received and approved by the customers it can be treated as a contract between the development team and the customers.

→ The requirement analysis and specification phase concentrate on what ~~nee~~ needs to be done and carefully avoid the solution (how to do) aspect.

Note →

The SRS document produced at the end of this phase is also called the black box specification of the problem.

→ Since in this document the system is considered as a black box whose external behavior is specified (Input/output) and the internal details are not known.



Design.

The goal of design phase is to transform the requirement specified in the SRS document into a structure i.e. suitable for implementation in some programming language.

→ In technical term during the design phase the software architecture is designed from the SRS document.

→ Two different design approaches are available.

- ① Traditional design approach
- ② Object oriented design approach

Traditional design approach:

→ It consists of two different activities.

① A structured analysis of the requirement is carried out where the detail structure of the problem is examined.

② Structured design activity where the results of structured analysis are transformed into the software design.

→ Here we are using datablow diagram (data) to perform structured analysis.

→ Structured design is undertaken once the structured analysis is completed. This is some time referred to as the software architecture.

Object oriented design approach.

In this technique various objects occur in the problem domain are first identified and various relationships that exist among the objects are identified.

The OOD approach has several benefits such as lower development time and effort and better maintainability of the product.

Coding and unit testing.

During this stage the software designed is realised as a set of programs or program units.

→ It is also known as implementation phase.

→ Unit testing involves verifying that each unit meets its specification.

→ ~~reason~~ The reason behind the unit testing is that the other modules with which this module has to be integrated. interrelated may not be ~~ready~~.

Integration and system testing

The individual program units or programs are integrated and tested as a complete ~~se~~ system to ensure that the software requirements have been ~~met~~ met.

→ Generally the integration is carried out incrementally over a number of steps and finally when all the modules have been successfully integrated and tested. The system testing is carried out.

→ It consists of three different kinds of testing activities.

(i) Alpha testing (α -testing): It is the system testing performed by the development team.

(ii) β -testing: It is the system testing performed by a friendly set of customers.

(iii) Acceptance testing: It is the system testing performed by the customer itself after the product is delivered.

Based on the result, the customer may reject or accept the delivered product.

4591

Maintenance :- Maintenance of a typical software product requires much more effort than the efforts necessary to develop the product itself which is roughly on 40:60 ratio.

→ Maintenance involves performing any or of the following 3 kinds of activities.

(1) Corrective Maintenance: ~~these~~ ^{these} ~~are~~ ~~the~~.

errors are corrected that were not discovered during the product development phase.

perfective maintenance:

→ ~~These~~ Here the functionalities of the system according to the customer requirement are enhanced.

(2) Adaptive maintenance:

Here the software is ported to work in a new environment.

for ex: porting may be required to get the software to work on a new computer platform or with a new operating system.

Shortcomings / Drawbacks of classical waterfall

→ Though it is a very simple and useful model it suffers from several shortcomings.

1) No feedback path: The classical waterfall model is idealistic in the sense that it assumes that no error is ever committed by the developers during any of the life cycle phases. Therefore it does not incorporate any mechanism for error correction.

→ Contrary to the fundamental assumption made by classical waterfall model, the developers do commit as large no. of errors in almost every activity they carry out during various phases of life cycle.

These defects usually get detected much latter later in the life cycle.

2) Difficult to accommodate change request:

This model assumes that all customer requirements can be completely and correctly defined at beginning of the project, but it is hard to achieve this even in ideal project scenarios. The customer requirements usually keep on changing with time, but this is unable to accommodate any changing request.

3) Inefficient error correction:

→ No overlapping of phases.

→ This model recommends that the phases be

carried out sequentially and a new phase can only start after the completion of the previous phase. but hardly it is possible to obey such recommendations because it leads to have a large numbers of team members to carry out a particular phase.