Software Production process

Software Production process

- The process we follow to build, deliver, deploy and evolve the software product from idea to delivery of product
- Goal of Production process
  - Quality product
  - On time
  - Within budget
  - Reliable and efficient

Software Process Model

- Earlier Software development
  - SINGLE person task
  - Mathematical nature software
  - Programmer and end user same
  - Code and Fix Model
    - Messy
    - New modifications was harder
  - Now a days Software is made on requirements
    - End user is not programmer
    - More reliability needed (failures can be deadly)
    - Now it’s a group activity
  - Code and fix was inadequate
    - Lack of documentation
    - Unstructured way of coding
    - No anticipation of change

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:

- The tasks to be performed
- The input and output of each task
- The pre and post conditions for each task
- The flow and sequence of each task

Software development Life cycle

Software life cycle models describe phases of the software cycle and the order in which those phases are executed

When a process involves building a software (product), the process may be referred to as software (product) lifecycle

Waterfall Model

The waterfall model is the classic model or oldest model and is known as mother of all the model. It is widely used in government projects and many vital projects in company.

The waterfall model is also called as **'Linear sequential model'** or **'Classic life cycle model'.**

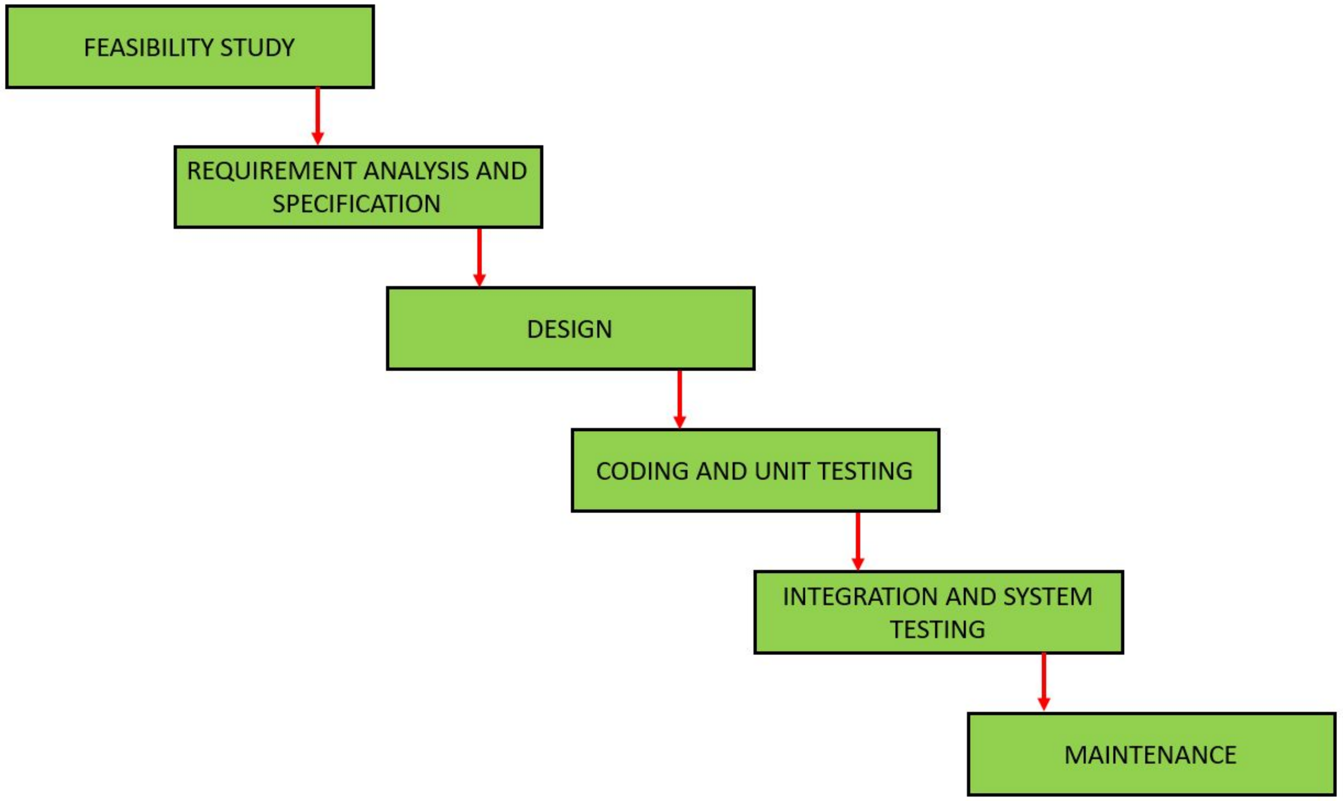
In this model, each phase is executed completely before the beginning of the next phase. Hence the phases do not overlap in waterfall model.

This model is used for small projects.

In this model, feedback is taken after each phase to ensure that the project is on the right path.

Testing part starts only after the development is completed.

Waterfall Model


FEASIBILITY STUDY
REQUIREMENT ANALYSIS AND SPECIFICATION
DESIGN
CODING AND UNIT TESTING
INTEGRATION AND SYSTEM TESTING
MAINTENANCE

Another view of Waterfall Model

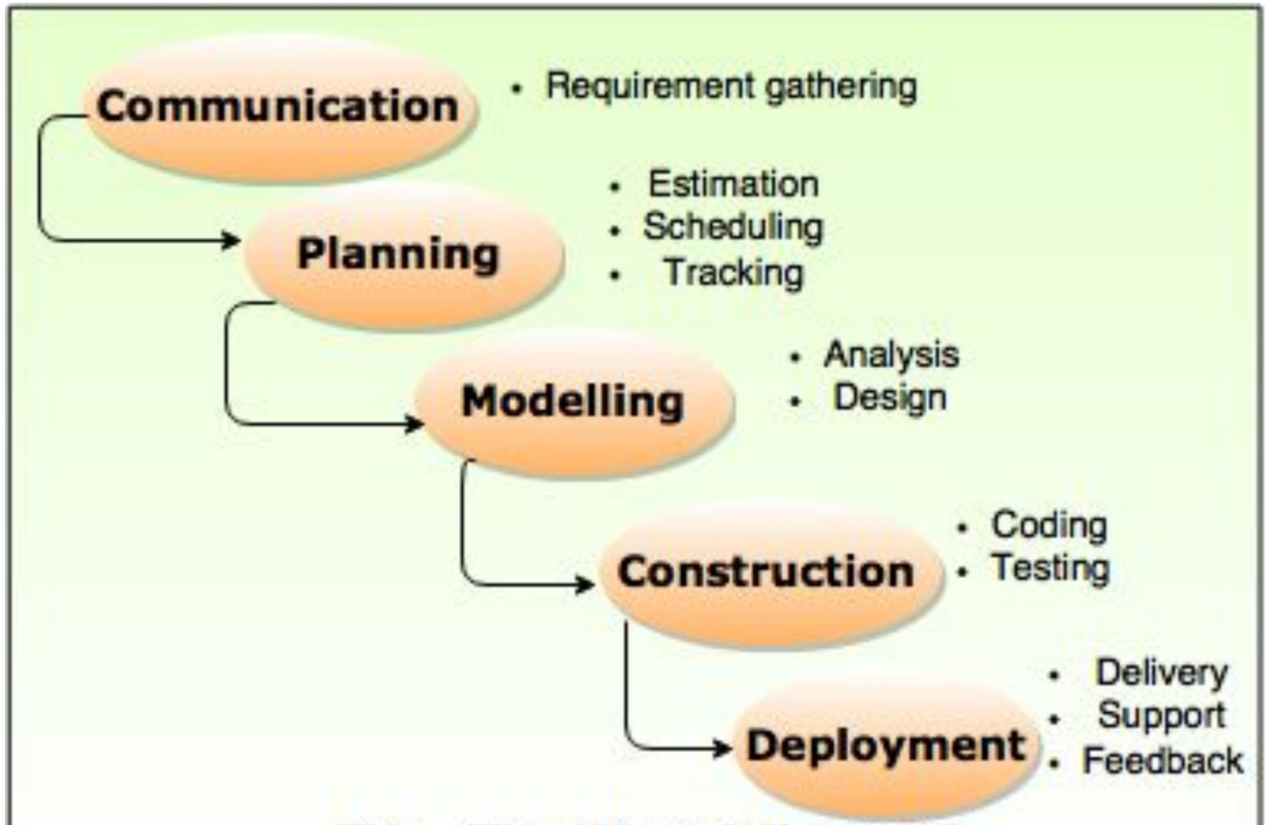


Fig. - The Waterfall model

Another view of Waterfall Model

i) Communication
The software development starts with the communication between customer and developer.

ii) Planning
It consists of complete estimation, scheduling for project development.

iii) Modeling

Modeling consists of complete requirement analysis and the design of the project i.e algorithm, flowchart etc.

iv) Construction

Construction consists of code generation and the testing part.

Coding part implements the design details using an appropriate programming language.

Testing is to check whether the flow of coding is correct or not.

Testing also checks that the program provides desired output.

v) Deployment

Deployment step consists of delivering the product to the customer and taking feedback from them.

If the customer wants some corrections or demands for the additional capabilities, then the change is required for improvement in the quality of the software

Advantages of Waterfall

This model is simple and easy to understand and use.

It is easy to manage due to the rigidity of the model

In this model phases are processed and completed one at a time

Waterfall model works well for smaller projects where requirements are very well understood.

Disadvantages of the Waterfall model

This model is not good for complex and object oriented projects.

In this model, the changes are not permitted so it is not fit for moderate to high risk changes in project.

It is a poor model for long duration projects.

The problems with this model are uncovered, until the software testing.

The amount of risk is high.

V Model

V model is known as Verification and Validation model.

This model is an extension of the waterfall model.

In the life cycle of V-shaped model, processes are executed sequentially.

Every phase completes its execution before the execution of next phase begins.

based on association of a testing phase for each corresponding development stage.

Verification phases on one side of the ‘V’

Validation phases on the other side.

Coding phase joins the two sides of the V-Model.

V Model

Verification: It involves static analysis technique (review) done without executing code. It is the process of evaluation of the product development phase to find whether specified requirements meet.
Validation: It involves dynamic analysis technique (functional, non-functional), testing done by executing code. Validation is the process to evaluate the software after the completion of the development phase to determine whether software meets the customer expectations and requirements.

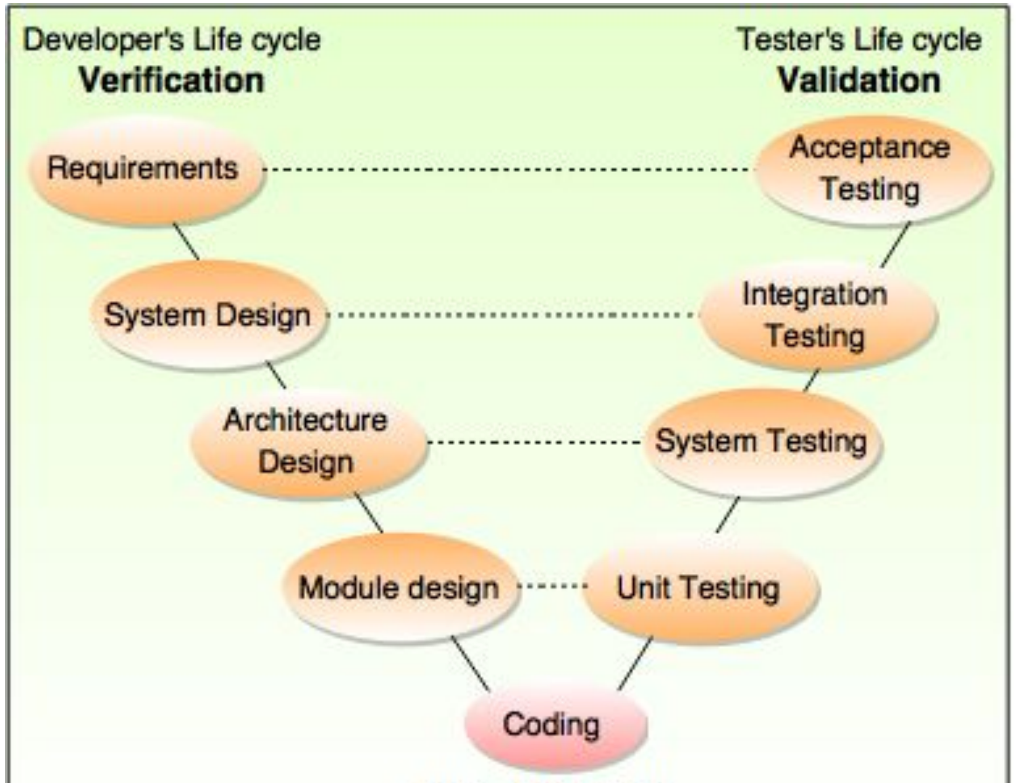


Fig.- V-Model

Verification Phase of V-model:

Requirement analysis: This is the first step where product requirements understood from the customer's side. This phase contains detailed communication to understand customer's expectations and exact requirements.

System Design: In this stage system engineers analyze and interpret the business of the proposed system by studying the user requirements document.

Architecture Design: The baseline in selecting the architecture is that it should understand all which typically consists of the list of modules, brief functionality of each module, their interface relationships, dependencies, database tables, architecture diagrams, technology detail, etc. The integration testing model is carried out in a particular phase.

Module Design: In the module design phase, the system breaks down into small modules. The detailed design of the modules is specified, which is known as Low-Level Design

Coding Phase: After designing, the coding phase is started. Based on the requirements, a suitable programming language is decided. There are some guidelines and standards for coding. Before checking in the repository, the final build is optimized for better performance, and the code goes through many code reviews to check the performance.

Validation Phase of V-model

Unit Testing: In the V-Model, Unit Test Plans (UTPs) are developed during the module design phase. These UTPs are executed to eliminate errors at code level or unit level. A unit is the smallest entity which can independently exist, e.g., a program module. Unit testing verifies that the smallest entity can function correctly when isolated from the rest of the codes/ units.

Integration Testing: Integration Test Plans are developed during the Architectural Design Phase. These tests verify that groups created and tested independently can coexist and communicate among themselves.

System Testing: System Tests Plans are developed during System Design Phase. Unlike Unit and Integration Test Plans, System Tests Plans are composed by the client?s business team. System Test ensures that expectations from an application developer are met.

Acceptance Testing: Acceptance testing is related to the business requirement analysis part. It includes testing the software product in user atmosphere. Acceptance tests reveal the compatibility problems with the different systems, which is available within the user atmosphere. It conjointly discovers the non-functional problems like load and performance defects within the real user atmosphere.

When to use V-Model?

When the requirement is well defined and not ambiguous.

The V-shaped model should be used for small to medium-sized projects where requirements are clearly defined and fixed.

The V-shaped model should be chosen when sample technical resources are available with essential technical expertise.

Advantage (Pros) of V-Model:

Easy to Understand.

Testing Methods like planning, test designing happens well before coding.

This saves a lot of time. Hence a higher chance of success over the waterfall model.

Avoids the downward flow of the defects.

Works well for small plans where requirements are easily understood.

Disadvantage (Cons) of V-Model:

Very rigid and least flexible.

Not a good for a complex project.

Software is developed during the implementation stage, so no early prototypes of the software are produced.

If any changes happen in the midway, then the test documents along with the required documents, has to be updated.