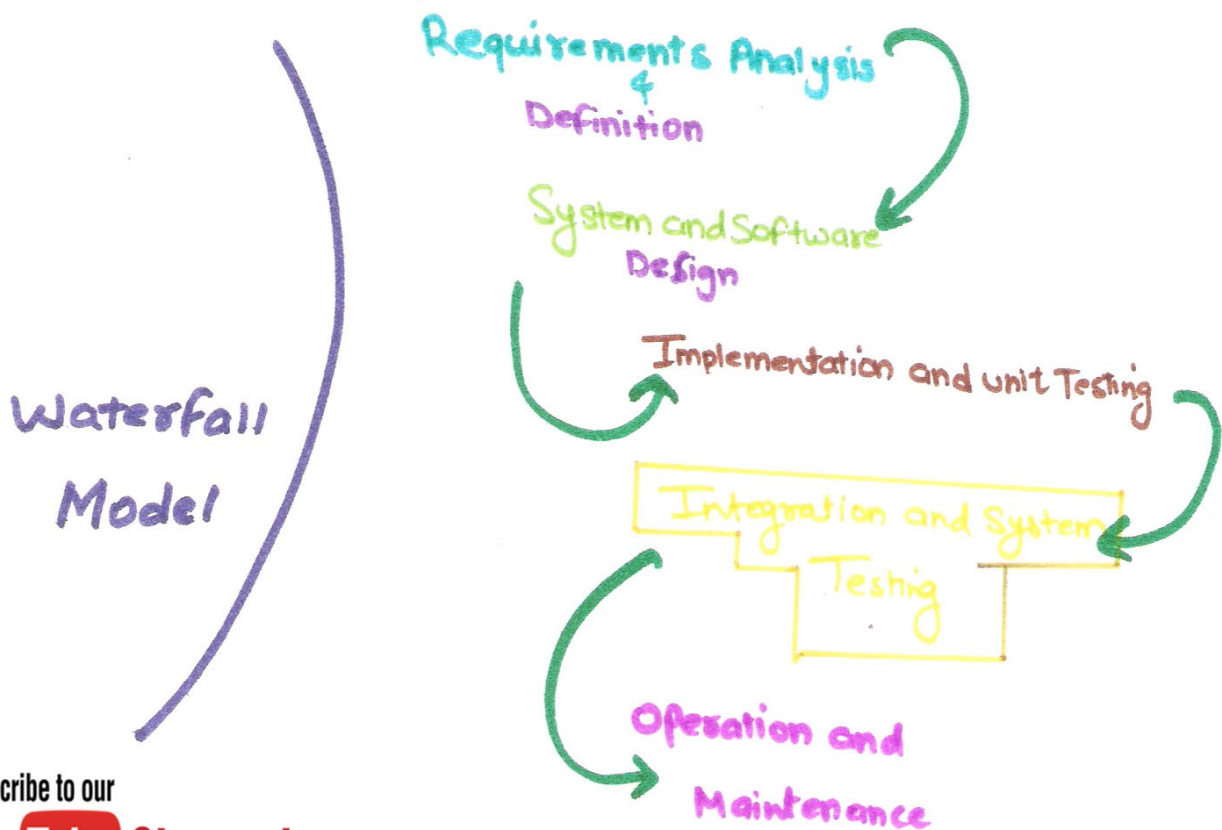# WATERFALL MODEL

Waterfall Model or Classical Waterfall Model:

→ Waterfall Model is a Simplest Model of Software Development Paradigm.

→ All the phases of SDLC will function one after another in linear manner. That is, when the first phase is finished then only the second phase will Start and so on.

The Waterfall model was first process model to be introduced and also called Linear - Sequential life Cycle model.

Waterfall Model

Requirements Analysis & Definition

System and Software Design

Implementation and unit Testing

Integration and System Testing

Operation and Maintenance

Computer Science Lectures By ER. Deepak Garg

This type of model is basically used for the project which is small and there are no uncertain requirements.

At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project.

In this Testing starts only after the Development is completed and Phases do not overlap.

Advantages

1) Discipline :- The staged development cycle enforces Discipline Every phase has a defined start and end point, and progress Can be conclusively identified (through the use of milestones) by both vendor and client.

2) Minimal Wastage : The emphasis on Requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of Schedule slippage,

3) Improves Quality :- It's much easier to Catch and Correct possible flaws at the design stage than at the testing stage; after all the Components have been integrated and tracking down specific errors is more Complex.

Computer Science Lectures By ER·Deepak Garg

Disadvantages

- **Uncertain Nature of Customer needs :-** Due to this estimating time and costs with any degree of accuracy (as the model suggests) is often extremely difficult.

- **Implicit Assumption :- to Road block** Implicit Assumption that designs can be feasibly Translated into Real products, this sometimes runs into Road blocks when developers begin implementation.

- **Not good Model :-** Not a good model for Complex and object-Oriented projects as world changing fast as technologies changes.

  - High amount of risk and uncertainty
  - Poor Model for long and ongoing projects.
  - Not Suitable for the projects where requirements are at a moderate to high Risk OF CHANGING.

Computer Science Lectures By ER·Deepak Garg