



① Software Products

* What is software engineering?

- It is just another branch of engineering.
- associated with development of software product using well defined scientific principles, methods and procedures.
- Software Engineering helps us to produce a reliable software product.

* What is software product?

- merchandise consisting of a computer program that is offered for sale.
- They can be classified as
 1. Generic Products
 2. Customized Products

* Generic Products :-

- They are marketed and sold to any customer who wishes to buy them.
- examples : Graphics software - Photoshop, Office automation - MS Office ; Accounting software - Tally, Design software - Auto CAD

* Customized Products :-

- Software that is commissioned by a specific customer to meet their own needs.
- examples : Banking systems, Air traffic control software, railway ticket reservation system, Hospital management system, etc

① Software Process

* What is Software Process?

- It is the process of splitting software development work into distinct stages/phases containing activities with the intent of better planning and management.

* What are process Models?

- A process model describes the sequence of phases for the entire lifetime of a product. Usually there are three main phases:

1. Concept Phase
2. Implementation Phase
3. Maintenance Phase

1. Concept Phase :-

- a. Software Specification - functionality of the software and its constraints.

2. Implementation Phase :-

- a. Software Development - Production of software product or system
- b. Software Validation - Checking the software to make sure it is what customer wants.

3. Maintenance Phase :-

- a. Software Evolution :- changing the software in response to changing demands.

Software Engineering fundamentals

* Functional Requirements :-

Features or requirements demanded by the customer

* Non-Functional Requirements :-

- Obvious feature that are not necessary to be discussed
- It must be dependable, reliable;
- Integrity - privacy must not be breached.

① Reusableility : e.g COTS - commercial off-the-shelf.

↳ products are packaged or canned (ready-made) hardware or software, which are adapted aftermarket.

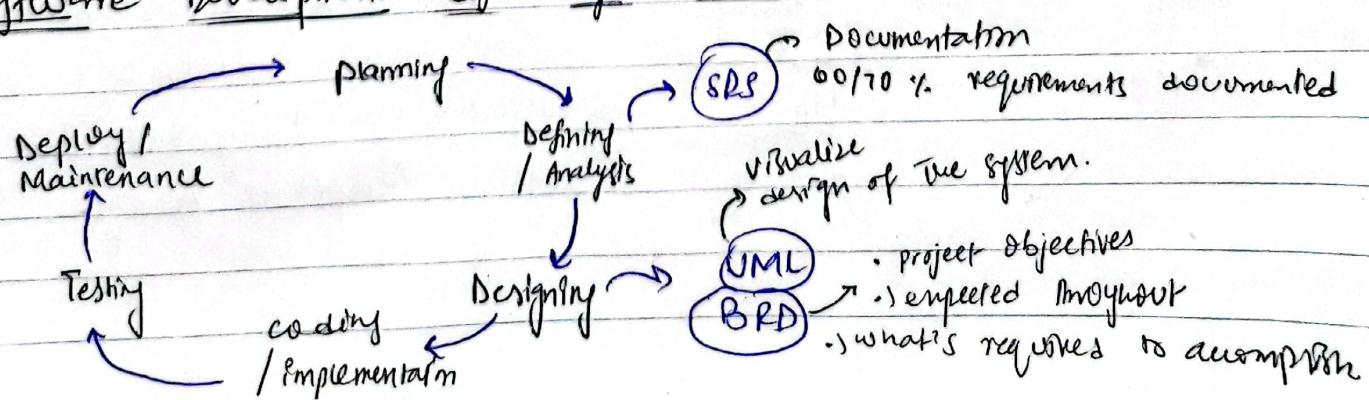
API - Application Program Interface

↳ Type of software interface offering a service to other pieces of software!

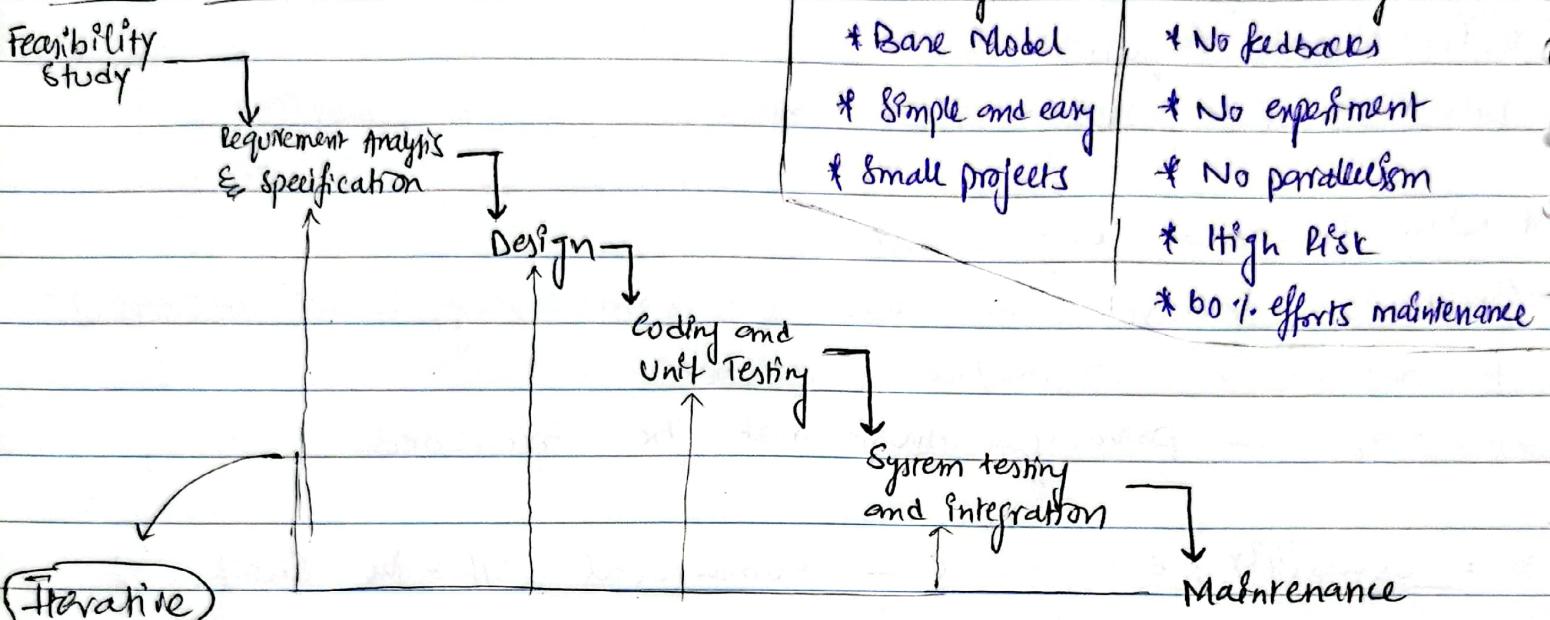
SaaS - Software as a Service

↳ Way of delivering applications over the Internet - as a service, instead of installing and maintaining.

② Software Development Life Cycle (SDLC) :-



○ Classic Waterfall Model :-



Advantages

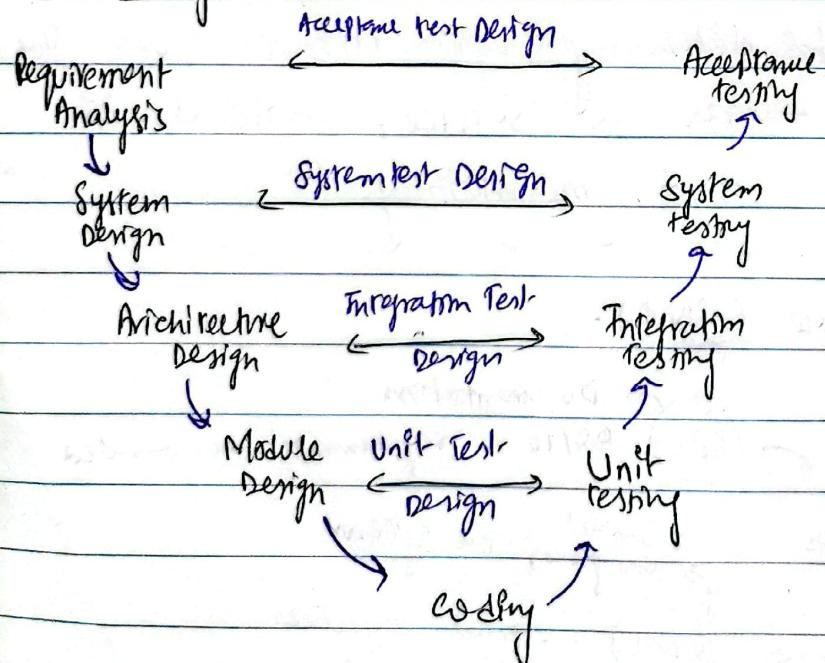
- * Baseline Model
- * Simple and easy
- * Small projects

Disadvantages

- * No feedbacks
- * No experiment
- * No parallelism
- * High risk
- * 60% efforts maintenance

- ↳ Same as classic waterfall model, but changes could be made (only bugs and errors), not the documented requirements.
- ↳ No intermediate delivery.
- ↳ less customer interaction.

○ V-Shaped Model :-



- Also known as Verification & Validation Model.
- ↳ check after whole system designed.
- Extension of waterfall model.
- Testing is associated with every phase of life cycle.
- Verification Phase (Requirement Analysis, System Design, Architecture Design, Design)
- Validation (Unit testing, Integration, System, Acceptance Testing).

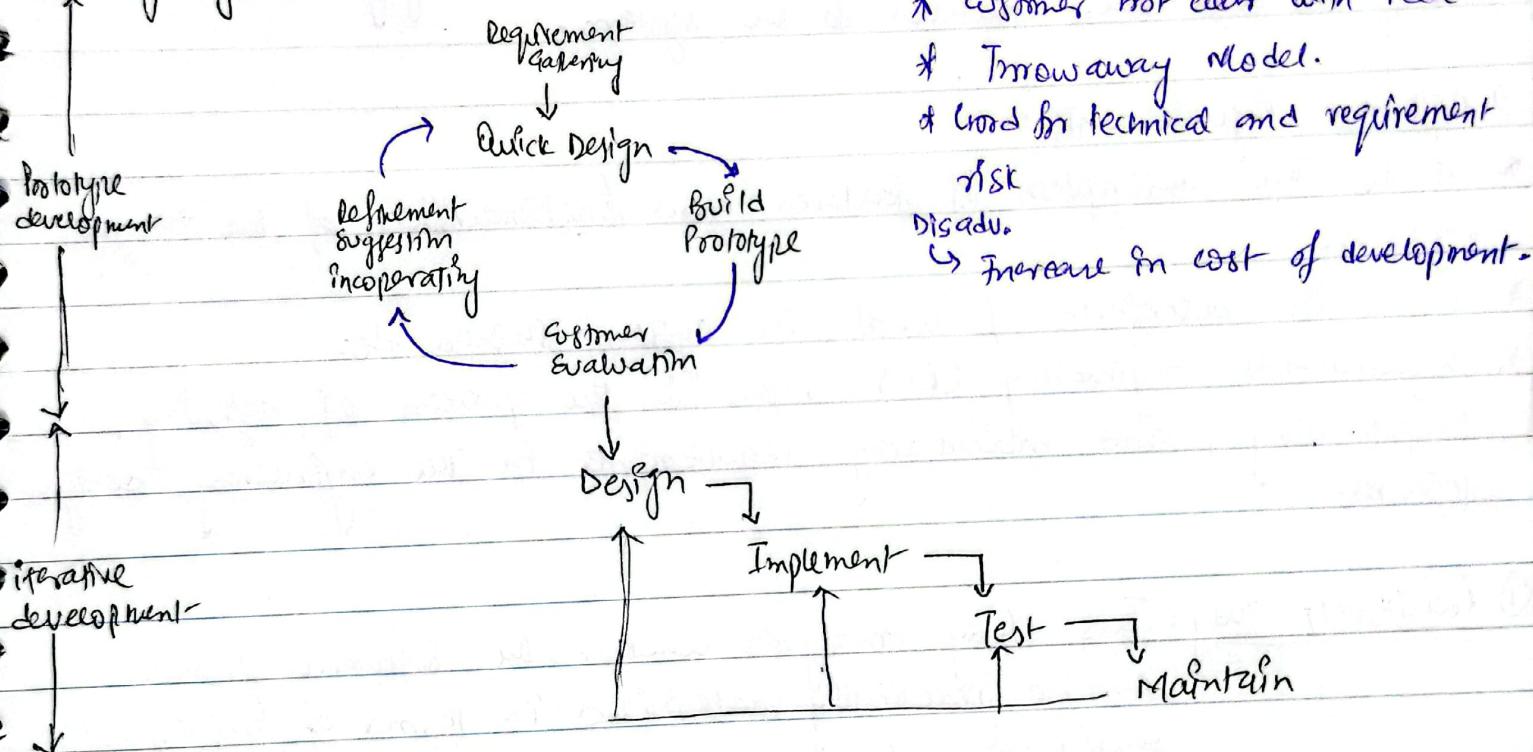
- Advantages

- * Time saving
- * Good understanding of project in the beginning
- * Every component must be testable.
- * Projects can be tracked easily (proactive)
- * Proactive defect tracking.

- Disadvantages

- * No feedback so can scope of changes
- * Risk analysis not done
- * Not good for big or object-oriented programming.

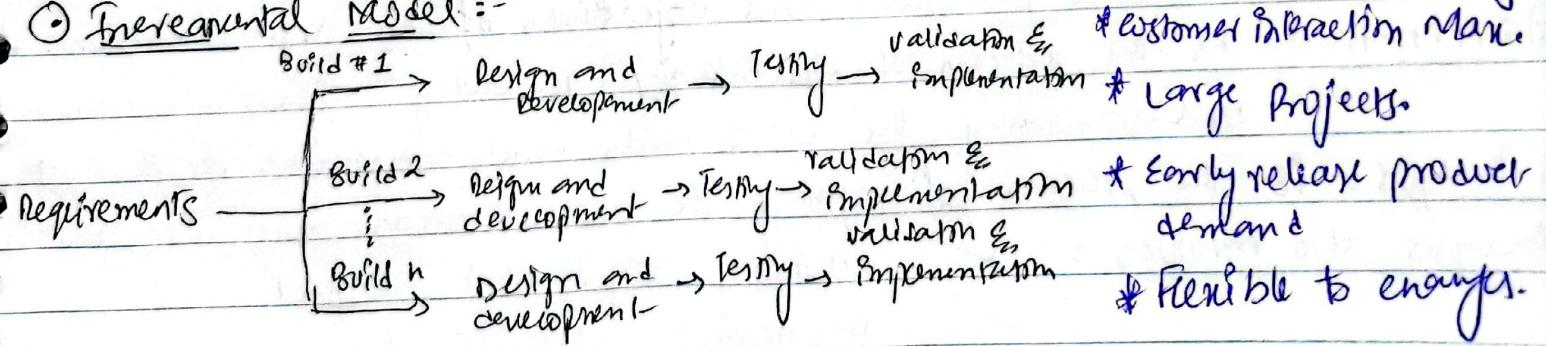
① Prototyping Model :-



good for

- * Customer not clear with Idea
- * Throwaway Model.
- * Good for technical and requirement risk discus.
- ↳ Increase in cost of development.

② Incremental Model :-



Problems from Management perspective :-

- 1) The process is not visible. Managers need regular deliverables to measure progress. If systems are developed quickly, it is not cost effective to produce documents that reflect every version of the system.
- 2) System structure tends to degrade as new increments are added. Regular change leads to messy code as new functionality is added. In whatever way is possible. It becomes increasingly difficult and costly to add new features to the system.

① Software Requirements :-

- * It is the description of features and functionalities of the target system.
- * It is the description of what the system should do.
- * Requirements Engineering (RE) refers to the process of defining, documenting, and maintaining requirements in the engineering design process.

① Feasibility Study: This study analyzes whether the software product can be practically materialized in terms of implementation contribution of project to organization, cost constraints and as per values and objectives of the organization.

② Elicitation: is the process of discovering, reviewing, documenting and understanding the user's needs and constraints for the system.

③ Analysis: is the process of refining the user's needs and constraints. Analysis also involves : - ✓ Avoiding requirement conflict ✓ Requirement prioritisation.

- ④ Specification: is the process of documenting the user's needs and constraints clearly and precisely.
- ⑤ Verification/Validation: is the process of ensuring that the system requirements are complete, correct, consistent and clear.