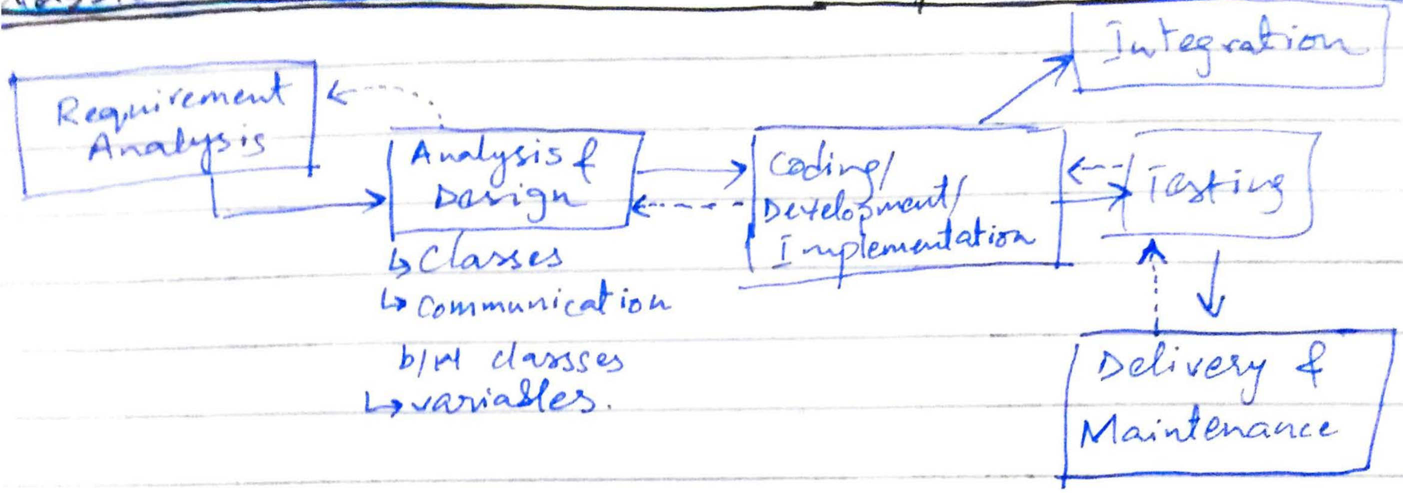
# SOFTWARE ENGINEERING

August 21

## lassic Software Process Model/Waterfall Model/Linear Model



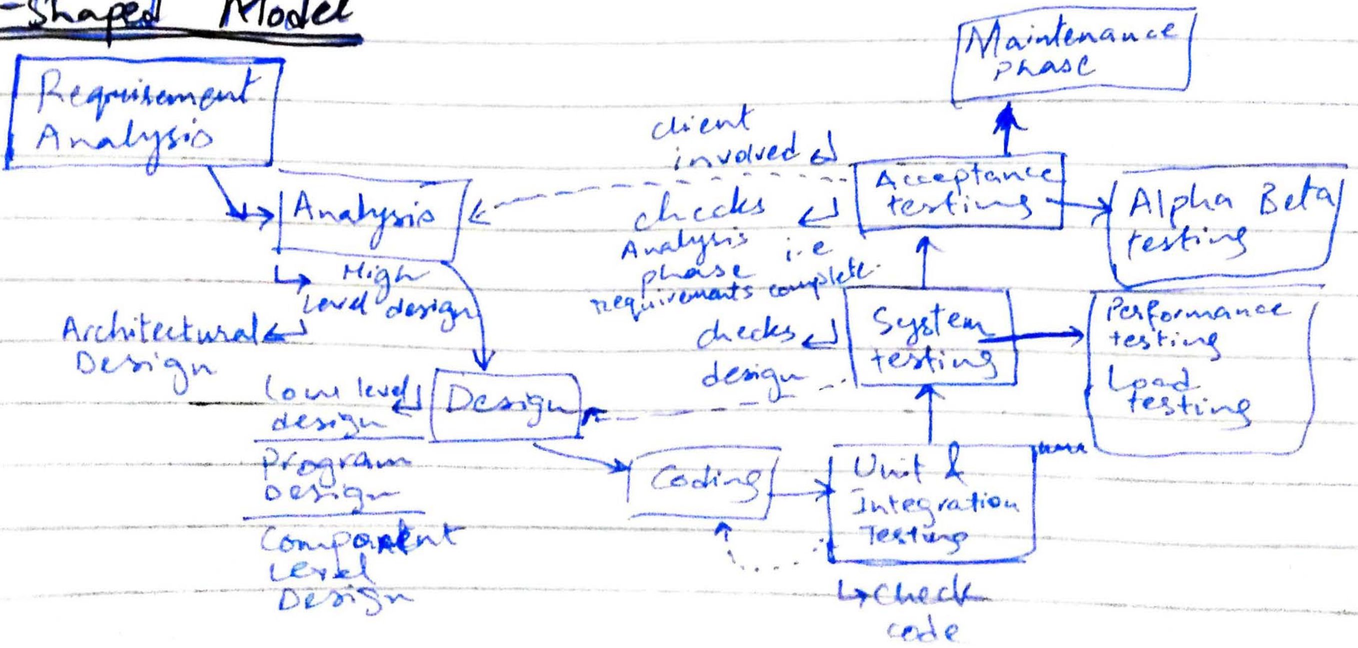
- Requirements are clear & well defined (non-ambiguous).
- Long term projects.

Disadvantages:-

- If requirements are not frozen then not a good model
- Client gets a working product very late in SDLC.

1) **Waterfall Model with feedback** (shown by dotted arrows in above digram) and no integration

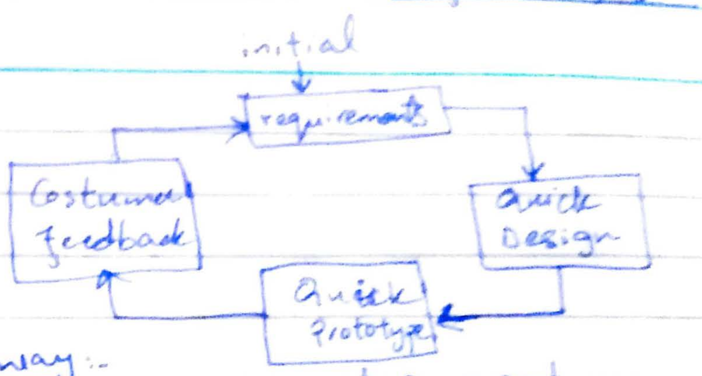
2) **V-Shaped Model**



August 26, 2019

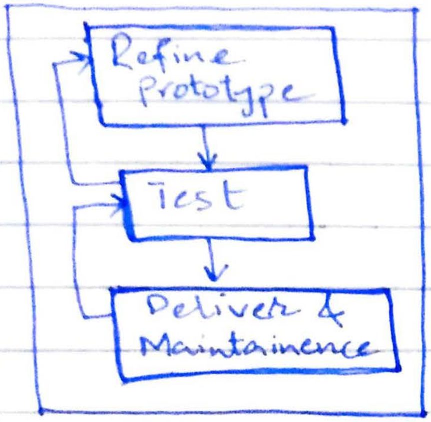
## Prototyping:-

- → Evolutionary
- → throw away



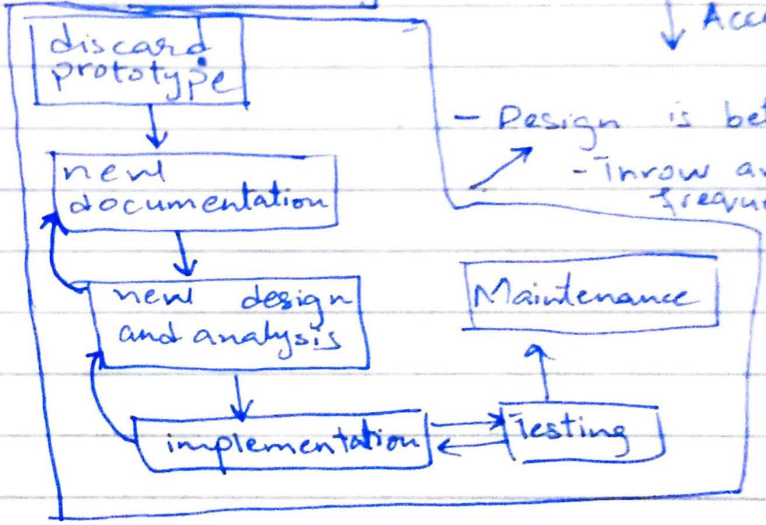
↓ Accepted.

### Evolutionary:-



- Time/cost effective

### Throw away:-



- Design is better
- Throw away is used more frequently

### Use When:-

- Requirements are not clear
- Domain expertise not available
- Proof of concepts needed

### Advantages:-

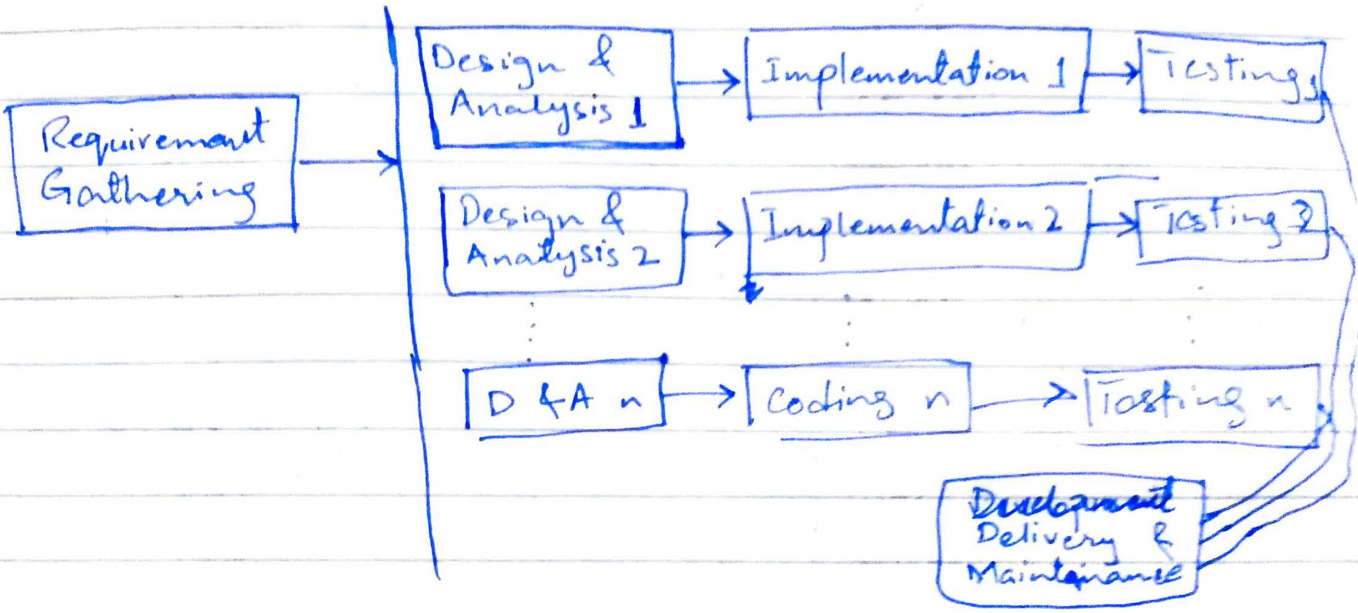
- Requirements are clarified
- Missing functions/bugs identified early
- Working software available early.

### Disadvantages:-

- Cost, target & deadline maybe missed

## RAD model:-

-RAPID Application ~~model~~ Development.



Use when:-

- Product delivery is in 60-90 days
- Low risk project
- Requirements must be clear.
- For large projects.

Disadvantages:-

- Cannot be used for high risk projects.
- Cannot be used if enough resources are not available.

Advantages:

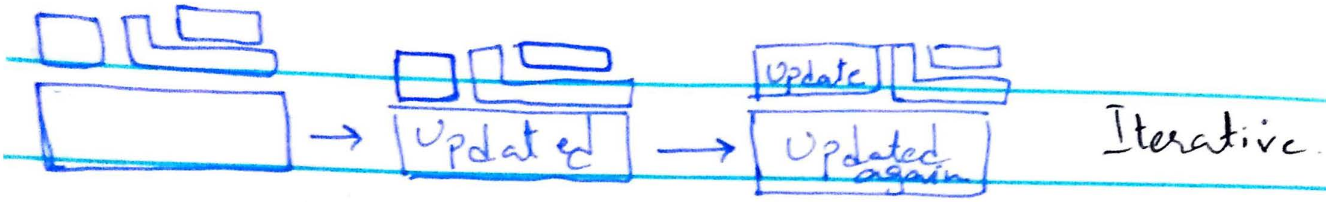
- Early product delivery.

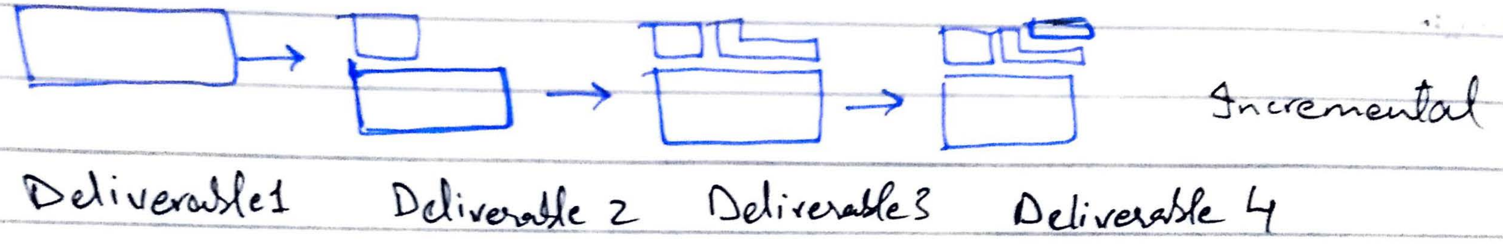
**Phased Development**:- - Follows water-fall model

↳ Iterative

↳ Incremental

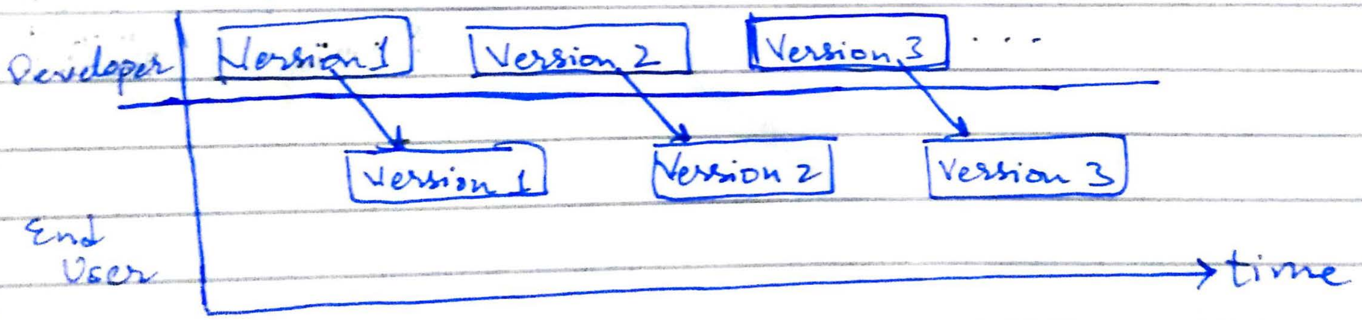
- Usually both of them are performed better.
- Use when requirements are clear.
- Timely delivery is important.



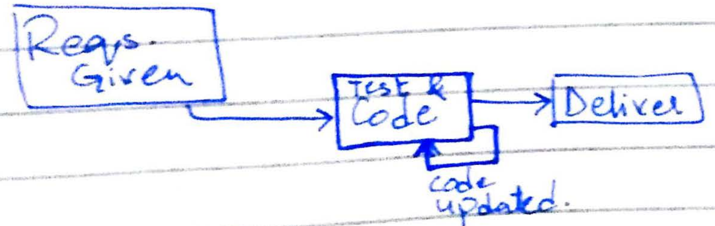


Disadvantages:-

- Integration issues possible
- Removing bug. in one module may result in bugs in other mod
- Multiple design & architectural changes may cause problems



## Code and Fix Model:-



- Can only be used for small projects.

August 28, 2019

-Spiral -Unified Process -Agile Method → Scrum
→ XP

Spiral:-

↳ Start with the small set of requirements.

↳ New functionalities added in each spiral (interation)

↳ Ask Management

↳ Four phases in each iteration.

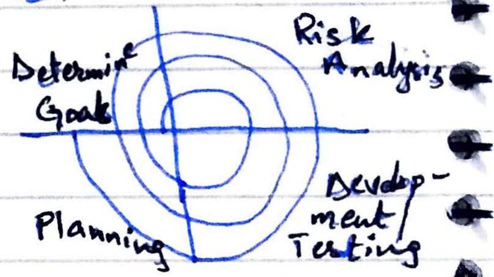
① ↳ Planning

② ↳ Determine goals, constraints, & alternative.

③ ↳ Risk Analysis

④ ↳ Development & testing.

(check book for proper diagram).



Used for:-

- Large projects
- Medium/High risk projects.

Disadvantage:-

- Budget may exceed.

Unified Process:-

↳ Two dimensional approach

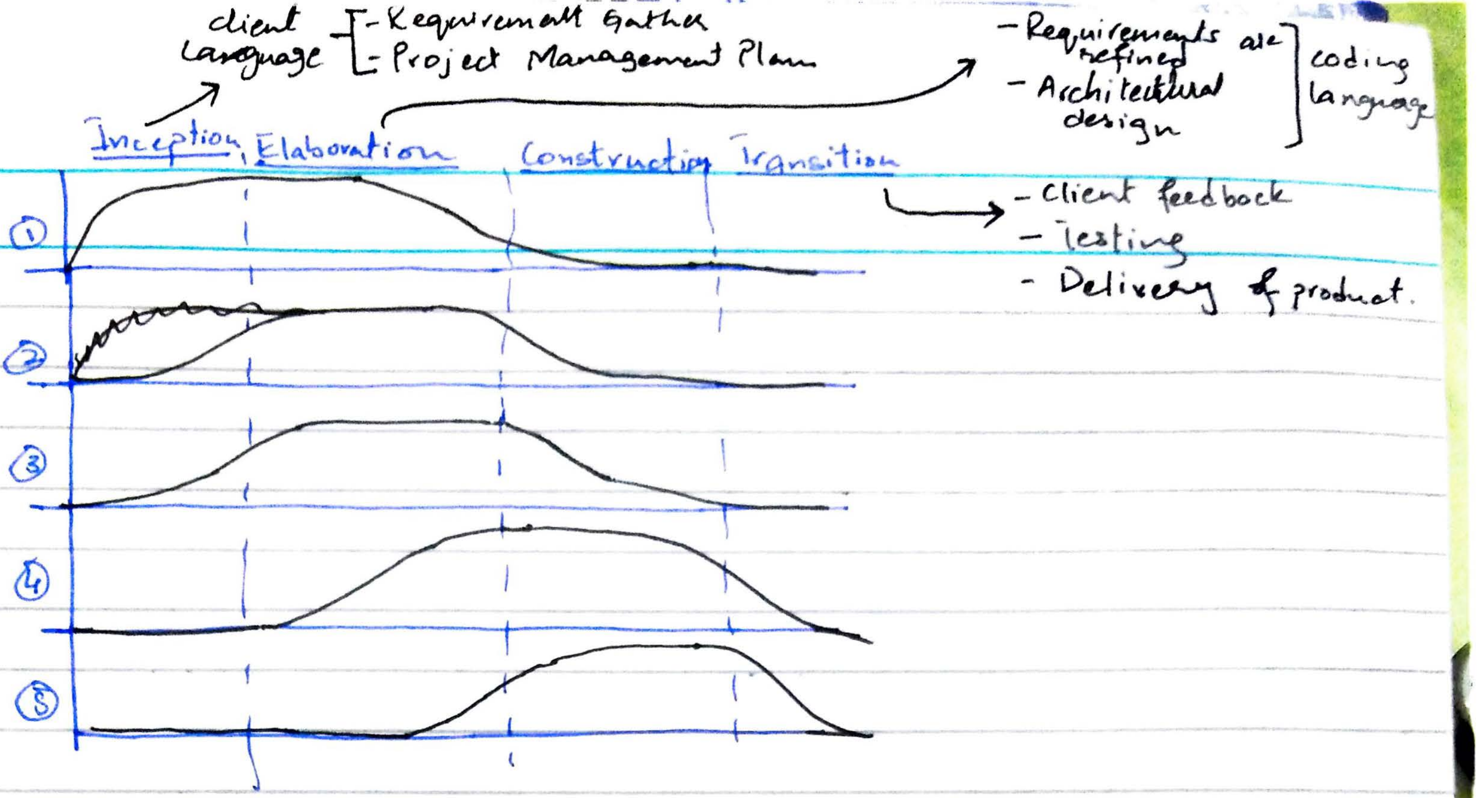
① Requirement workflow

② ~~Design~~ Analysis WF

③ Design WF

④ Development WF

⑤ Testing WF.



Agile Manifesto:-

- Individuals & interactions over processes & tools.
  (Design / Analysis phase is not focused).
- Working software over documentation.
- Client involvement over following a plan.
- Adapt to change.

*Scrum:-

- Product owner
  ↳ Client interaction
  ↳ Decides what tasks are to be done in spiral.
- Scrum Master
- Development Team
- Product Backlog
  ↳ Fuctions added that needs work to be done.
- Product Backlog grooming
  ↳ Fuctions/tasks are assigned priority with client.
  " " size with dev. team + client.

Sprint
↳ 2-3 weeks period
↳ Tasks assigned need to be completed.
stances

| | | |
|---|---|---|
| XS | ½ day | 1 points |
| S | 1 day | 2 points |
| M | 2-3 days | 5 points |
| L | 5 days | 8 points |
| XL | 5+ days | 13 points |

↳ Try to breakdown

## Sprint Planning

↳ Tasks are assigned to team members.

↳ Tasks are divided into sub tasks.

## Sprint Backlog

↳ Tasks to be completed in current sprint.

## Daily Meeting:-

↳ 15 minute meeting

↳ Tasks done yesterday

↳ Tasks to do today

↳ Issues if any

## 16th minute

→ Big issues are discussed

## Sprint Review

- tasks not done reasons
- Product owner + Scrum Master + Dev team

## Sprint Retrospective

↳ What to start

↳ What to stop

↳ What to continue.

## eXtreme Programming

↳ Client is always present on workplace.

↳ Pair programming

↳ work only on features requested.