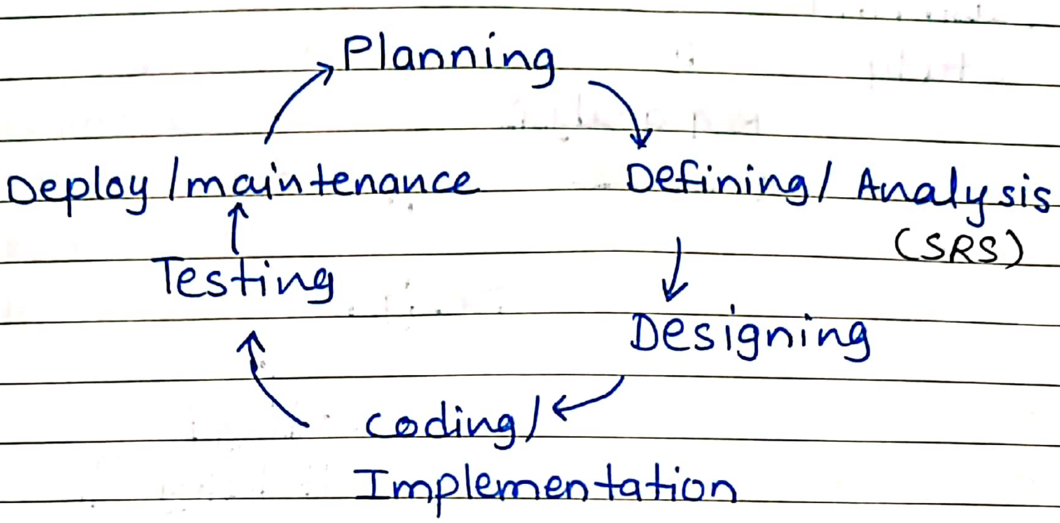
- NOTES:

- SDLC (s/w development life cycle)



- Customers & service provider (2 entities)
- Also called process model.
- Descriptive & diagramatic representation of s/w life cycle.
- Represents all the activities required to make a s/w product transist through its life cycle phas
- Defines entry & exit criteria for every phase.
- Types: classical waterfall model.
  Iterative ——— || ———.
  evolutionary model.
  prototyping model.
  spiral model.
- Maps diff. activities performed on a s/w product from its inception to retirement.

- Classical Waterfall Model:

- 1970, used for small projects.

Feasibility Study ↓

Req. analysis & specs. ↓

Design ↓

coding & unit testing ↓

system testing & integration ↓

mainte nce

(Model is less flexible)

Disadv:
1) No feedback
2) No experiment
3) No parallelism
4) High risk
5) 60% efforts maintenance

Adv:
1) Base Model
2) simple & easy
3) small projects

- Feasibility study: ~~KK~~ ~~wh~~weather your project is technically or financially feasible or not.

- Req. analysis & specifications : team, time, estimation is finalized & SRS doc. is created for customer in written form.

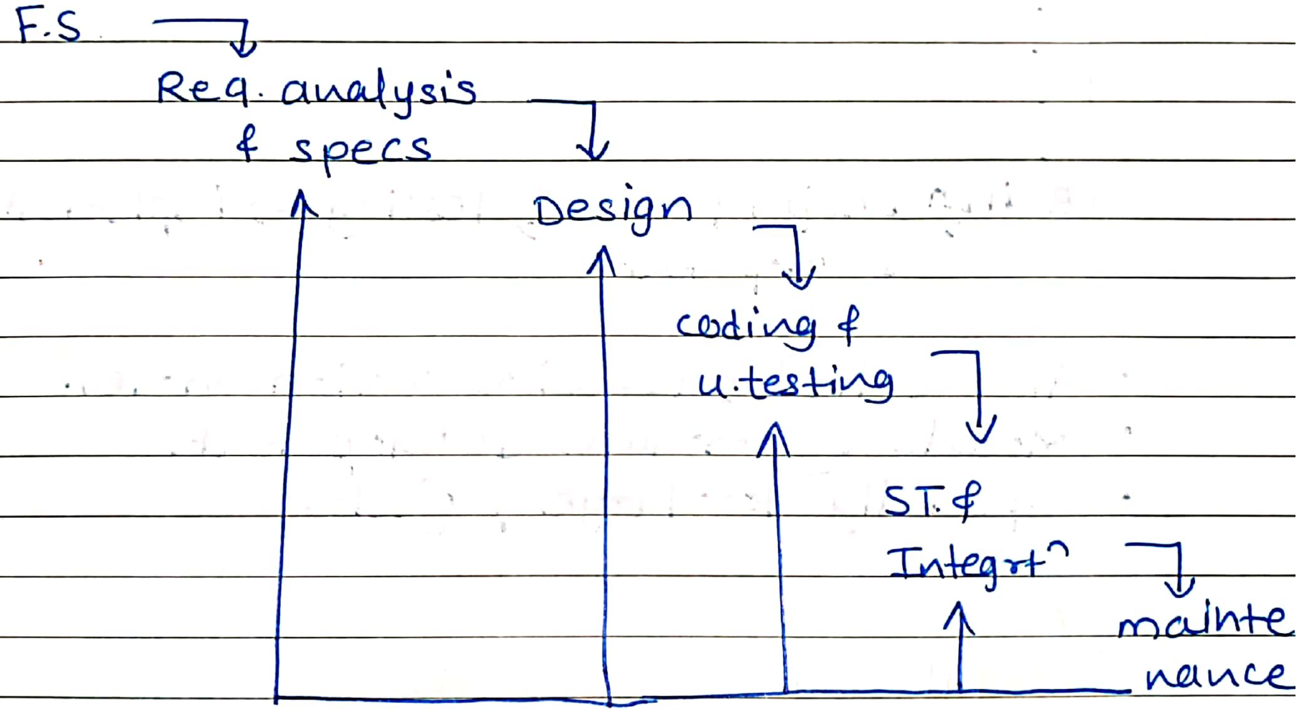
- Design: Designing of front-end, DFD, creates blueprint for coding phase.

- Coding & unit testing: Implementation is done & testing is done for each & every small unit.

- System testing : Integrating small units & integration & then tested.

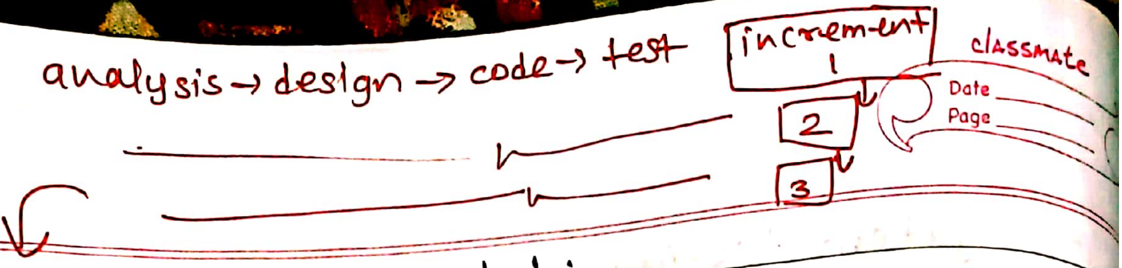
- Maintenance : very imp phase. Max. (60%) efforts. Fix bugs.

- Iterative waterfall model :



Adv :
1) Base model
2) simple & easy
3) small projects
4) feedbacks.
5) well organised.

Disadv :
1) No phase overlapping
2) No intermediate delivery
3) Rigid (No change)
4) Less customer interaction



- Incremental model:

Build1 → Design & Development → testing → implementat?

Requirements (SRS) — Build2 → Design & Development → testing → implementat?

Build3 → Design & Development → testing → implementat?

- Also called successive version model.
- Module by module development.
- specially for large projects.

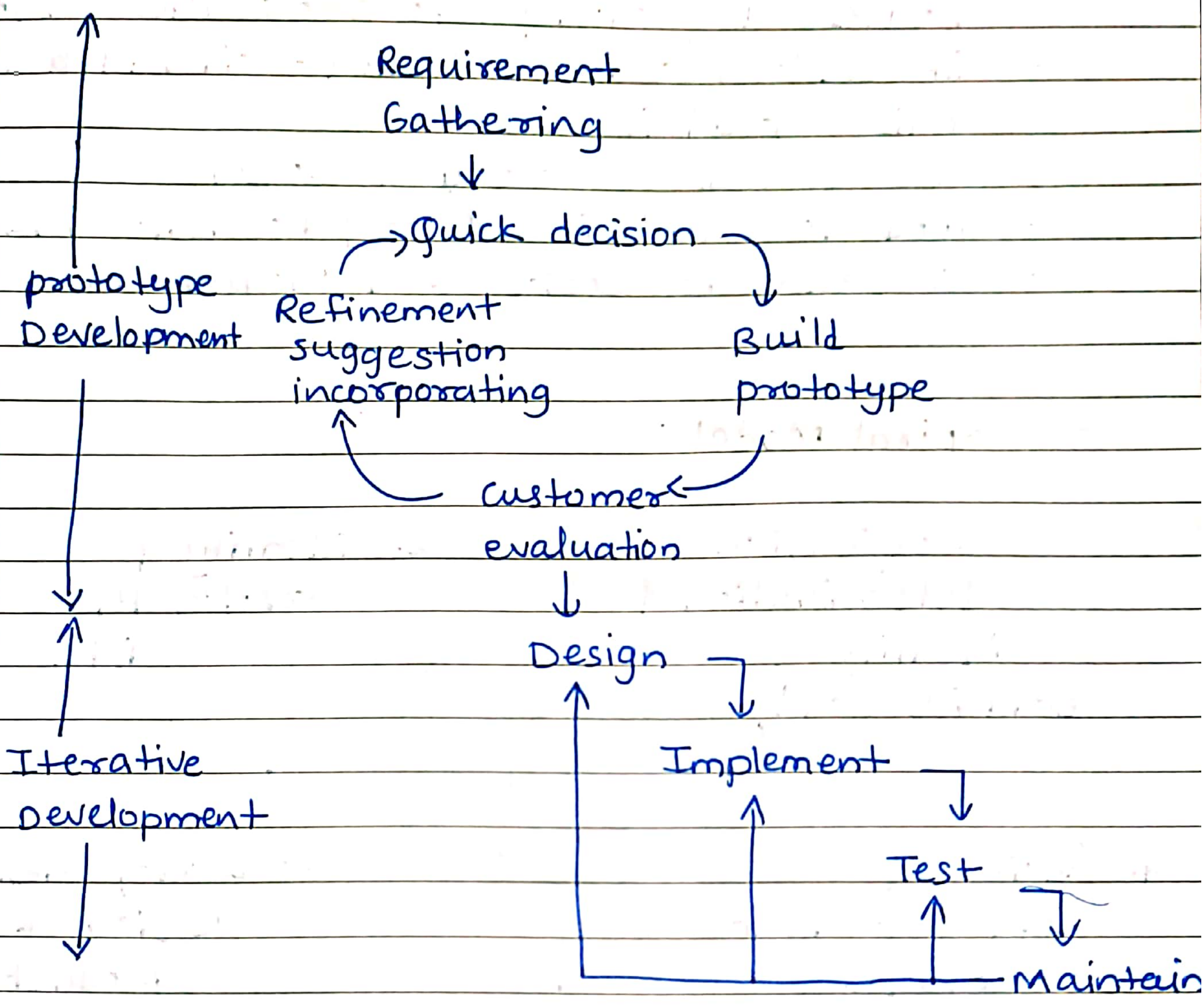
- Adv:

1) Module by module working.
2) Customer interaction maximum.
3) low chances of errors.
4) Early release product demand.
5) Flexible to changes.
6) cost efficient.

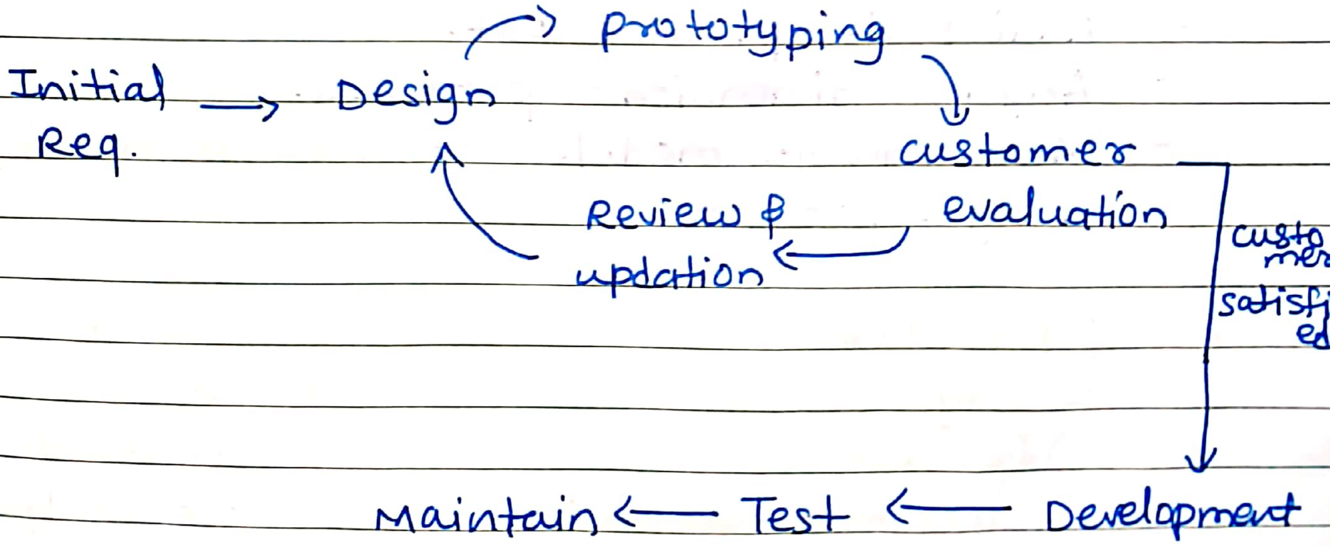
- when to use:

1) project has lengthy development schedule.
2) req. are superior.
3) customer demands quick release of produc
4) S/W team is not very well skilled or trained.

- Prototyping Model:

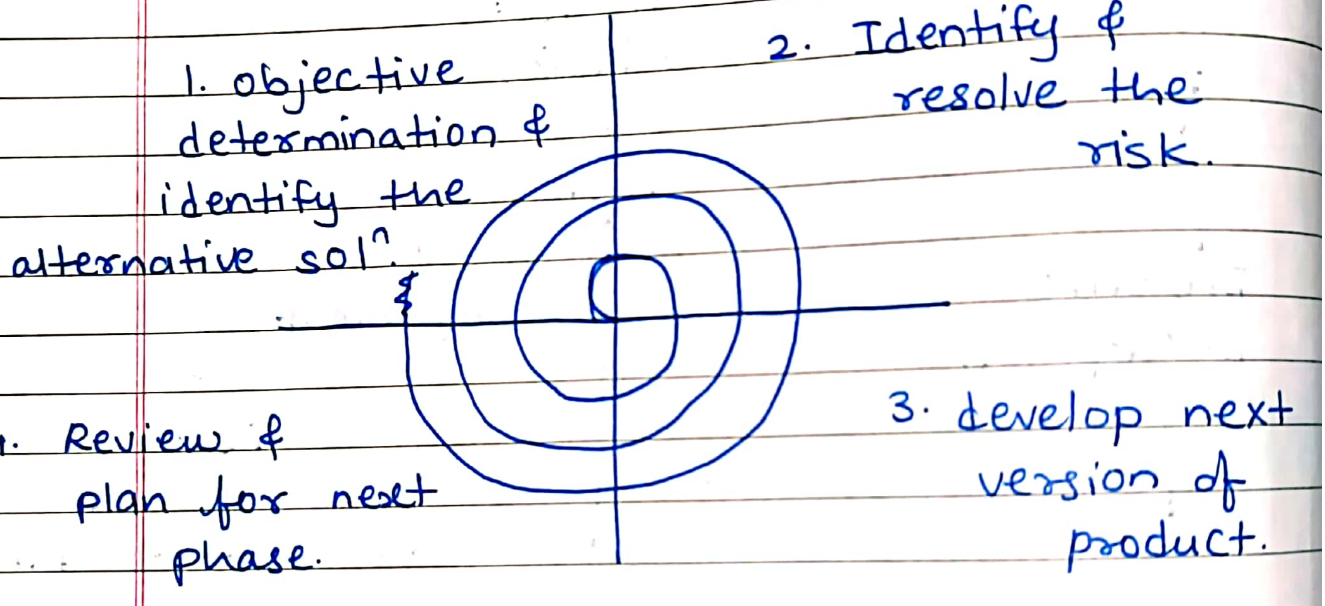


OR



- It is a kind of dummy / toy model.
- Creates a prototype to show the custome
- If customer is not clear with idea the
  this model is used
- also called throwayay model
- good for technical & requirement risks.
- Increase in cost of development.

- Spiral Model :


1. objective determination & identify the alternative soln.
2. Identify & resolve the risk.
3. develop next version of product.
4. Review & plan for next phase.

- Risk handeling.
- Radius of spiral = cost.
- Angular dimension= progress.
- Meta ~~Large~~ model. (uses multiple models)
- large projects.

Adv: 1) Risk handling
2) Large projects
3) flexible
4) customer satisfact^n.

Disadv: 1) complex
2) expensive
3) Too much risk analysis
4) Time

- Concurren Model (COMET model):

- UML Based model. (unified modeling lang.)

- Phases :

1) Req. phase:
- functional Req.
- Usecases & actors are also considered.
- Description of use case.
- Developing of req. model.
- There must be a clarity in req.

2) Analysis phase:
- Development of static & dynamic model.

  ↓ ↓

  structural intraction

  relationship
- static : structural relationship bet^n problem domain classes. (class diagram)
- Dynamic: Interaction of objects with use cases. (communication diagram)

3) Design phase:
- S/W architecture of system is designed.
- Mapping of analysis model to design model.
- Subsystem integration & communication through messages.

- Imp components of SRS:

1) functional requirements.
2) non- —||—
3) goals of implementation.

- Types of maintenance:

~~A~~

1) corrective maintence.
2) perfective —||—.
3) adaptive —||—.

- Types of prototype of model:

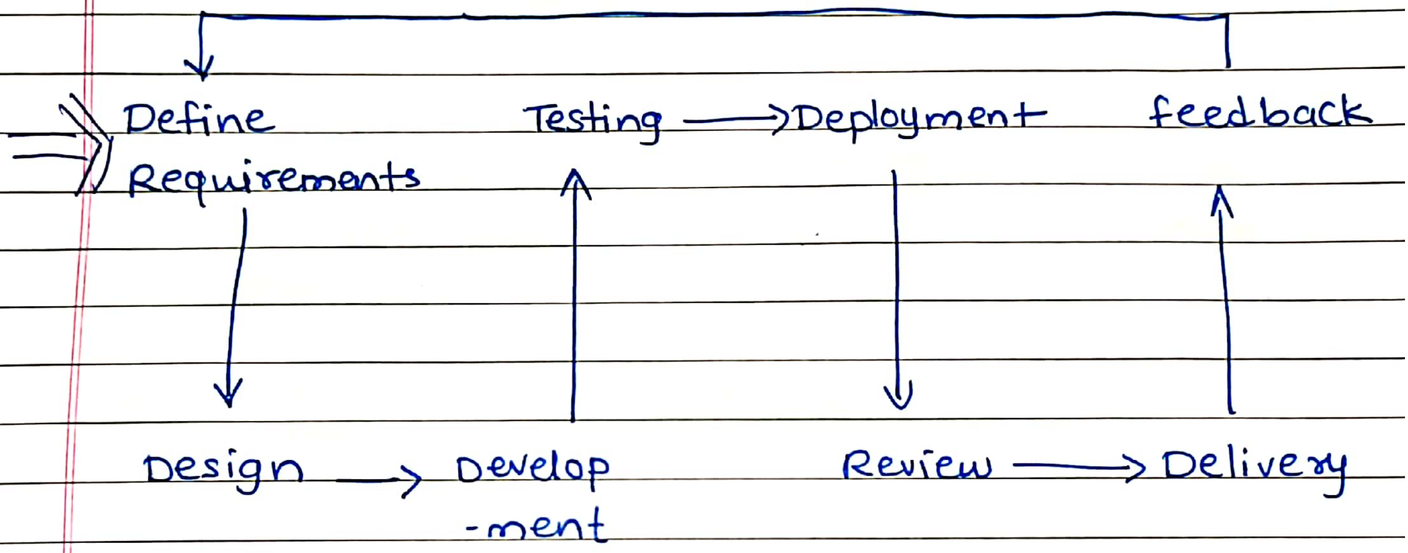
1) Rapid throwaway prototyping.
2) Evolutionary —||—.
3) Incremental —||—.

- Adv. of prototype mode:

- Aspect Oriented development:

- Agile s/w development process:

- Move quickly.
-



- Breakdown large project into small chunks called Iteration.
- Develop, test it.
- Release in market.
- feedback.
- based on feedback enhancement is done.
- Re-release the s/w.

Adv:
1) Frequent delivery.
2) face to face communicatn with client.
3) changes can be done.
4) Time saving.

Disadv:
1) less documentatio
2) Maintanance problem.