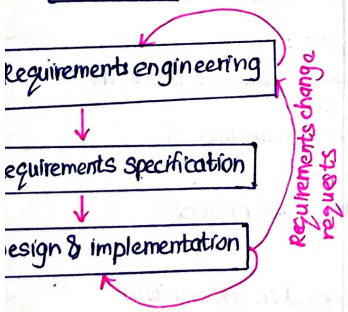
SOFTWARE PROCESS MODELS

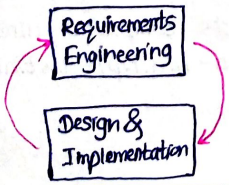
Generic Activities of s/w process

- Specification :-
  - define what system should do
- Design & Implementation :-
- Validation :-
  - checking that it does what the customer wants
- Evolution :-
  - changing the system in response to changing customer needs.

Plan-driven processes



Agile Processes



• Waterfall Model

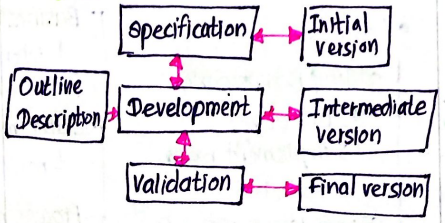
- plan driven
- linear sequential

- Requirements analysis & definition
- system & s/w design
- Implementation & unit testing
- Intergration & system testing
- Operation & maintenance

- Can be used,
  - requirements not changing - clear
  - not complicated & big
  - project - short
  - environment clear
  - technology & tools - stable
  - resources available trained
- Advantages,
  - simple - easy to understand & use
  - rigid - easy to manage
  - phases - not overlap
  - small projects - stable requ.
- Disadvantages
  - can't go back to previous stages
  - working s/w - at the end
  - risk & uncertainty
  - not good for complex and object oriented projects
  - poor model for long & ongoing projects
  - not good for projects - requirements highly changing.

• Incremental Development

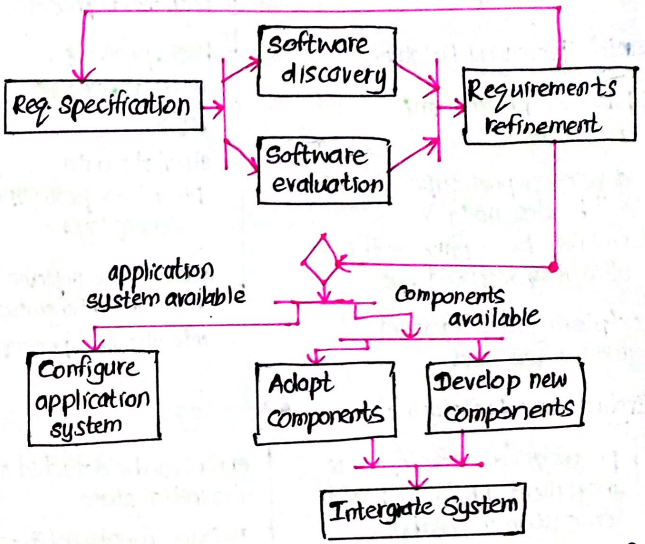
- many mini development projects
- highest priority requirements at first



- Benifits
  - quick
  - flexible & less expensive → change requirements & scope
  - changes are accepted at any stage
  - less costly
  - frequent customer ~~inte~~ feedback
  - errors - identify - easily
- Problems
  - process is not visible
  - system structure - tends to degrade as new increments are added

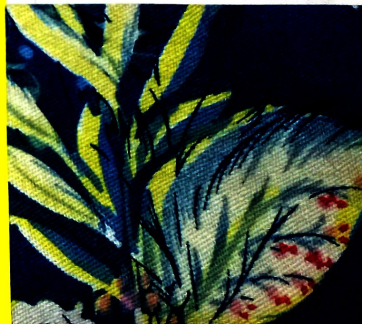
• Reuse Oriented (Intergration & Configuration)

= Reuse oriented development



- Requirements specification
  - brief descriptions of essential requirements
- s/w discovery & evaluation
  - a search made for components & systems ~~tat~~ that provide functionality required and evaluate whether them are suitable
- Requirement Refinement
  - refine using the info ~~ab~~ about reusable components & applications & change specifications
- Application System Configur
  - if off-the-shelf application available - configure them
- Component adaptation & intergration
  - modify reusable compone -ts and develop new components.

- Advantages :-
  - reduce cost
  - low risk
  - save time & effort
  - very efficient in nature
- Disadvantages :-
  - not always work
  - compromise in requirements → not fullfill requirements of the user
  - old system components may not compatible with other components



• Prototyping

- when not all the requirements are known
- iterative
- trial-and-error process

Rapid Throwaway Prototype

- based on preliminary requirements
- developed prototype will be discarded & will not be a part of the ultimately accepted one
- exploring ideas getting instant feedback

Evolutionary Prototyping

- prototype is developed is incrementlly refined based on customer feedback
- save time & effort
- a project which uses new technology
- complex projects

Incremental Prototyping

- final product is decimated into different small prototypes and develop individually
- different prototypes are merged into a single product

Extream Prototyping

- in web development
- 3 sequential phases
  - basic prototype with all existing pages
  - simulate data process → prototype service layer
  - services are implemented and intergrated into the final prototype

• Advantages

- errors can be detected in the initial state
- missing functionalities can be identified
  - ↳ reduce the risk of farlure
- effective communication
- customer satisfactation

• Disadvantages

- slow & time consuming
- prototype developing cost is wasted
- excessive change requests
- poor documentation

• Iterative Model

- begin with smaller set of requirements
- analyze them & gradually implement features
- this cycle is "iteration"

• Advantages

- coding begins early
- cost effective
  - ↳ implement requir. changes.
- streamlined (process) management → iterations
- bugs & deffects are spotted at earlier

• Disadvantages

- difficult - risk analyze
- design & architecture issues in later phases
- too reliant on baseline plan
- resource heavy model

• RAD

- iterative
- short cycle
- if requirements are well understood & project scope is constrained

Business modelling

- identify business functionalities

Data modelling

- refine data objects

Process modelling

- transform data objects into the information flow

Application generation

- generate system using reusable

Testing and turnover

- test new components
- check all interfaces

• Issues :-

- sufficient human resources
- customer must interact frequently
- system should be modularized
- technical risks are high

• Spiral Model

- iterative + waterfall
- delivers projects in loops
- phase of risk assesment

• Advantages

- well documented
- on time - on budget
- risk assesment
- can apply changes to new iterations

• Disadvantages

- success depends on skilled risk managers
- large resource pool
- time consumine

---

* Waterfall :-
  small / mid sized projects - clearly defined requirements

* Incremental :-
  - loosely-coupled parts and projects
  - clear requirements

* Iterative :-
  - large scale projects

* RAD :-
  - developed in short time
  - known requirements

* Spiral
  - large projects - complex requireme
  - new product - multiple testing phase