

Chapter 2

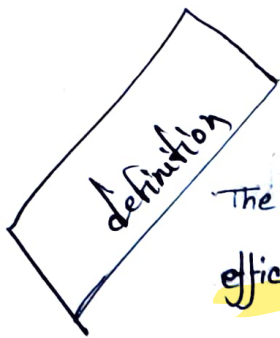
Process Models

- Process flow

- Linear Process flow
- Iterative process flow
- Evaluationary Process flow
- Parallel Process flow

- Prescriptive Process models:

- Water Fall Model
- V - model
- Incremental Process models
- Evaluationary Process Models
 - Prototyping
 - Spiral model
- Concurrent models



Chapter 02

The software process is a step-by-step approach to building software efficiently and with high quality.

- A structured plan that guides software development (What?)
- Software engineers, managers & stakeholders (Who?)
- It keeps the process organized, avoids chaos & ensures better results. (Why?)

- * - software consists of framework activities.
- - activities consists of software engineering actions.
- Actions consists of task set SAAT

* a generic process framework for software engineering defines five framework activities -

- communication ✓
- planning ✓
- modeling ✓
- construction & ✓
- deployment ✓

In addition, { project tracking and control, risk management, quality assurance, configuration management, technical reviews & others.

umbrella activities

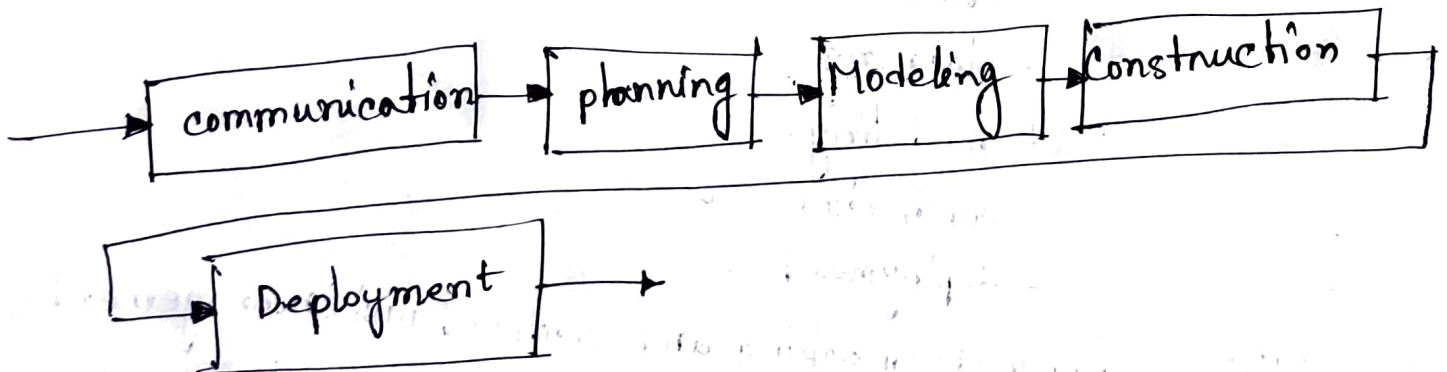
process flow :

Process flow is the way software development activities, tasks and actions are organized in a sequence over time to complete the project efficiently.

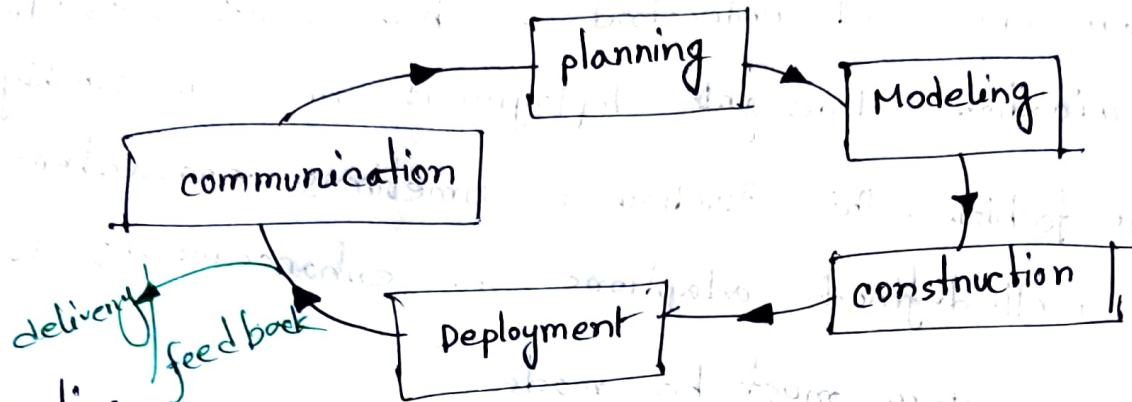
kinds :

- Linear process flow ✓
- Iterative process flow ✓
- Evolutionary process flow ✓
- Parallel process flow ✓

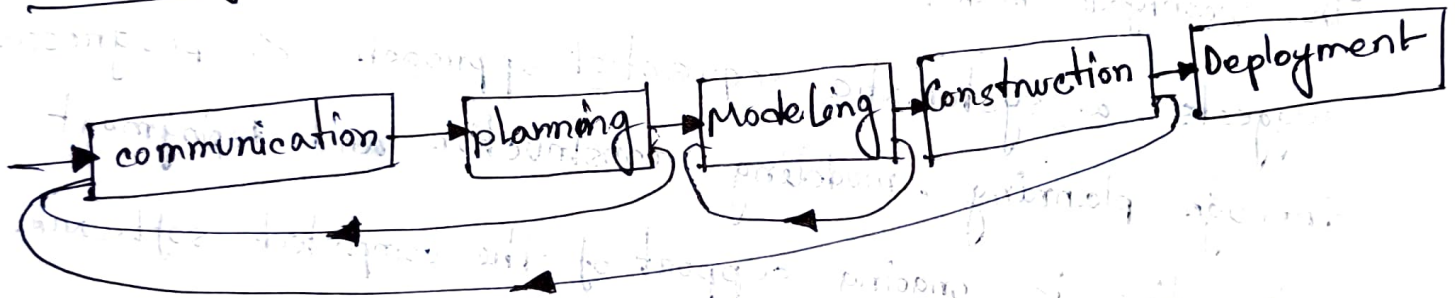
Linear process flow :



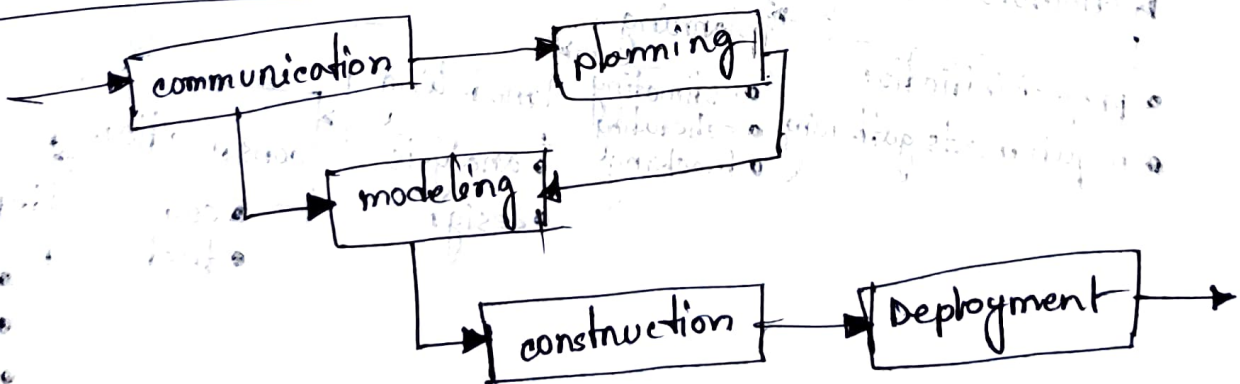
Evolutionary ~~Iterative~~ process flow:



Iterative ~~Evolutionary~~ process flow:



Parallel process flow:

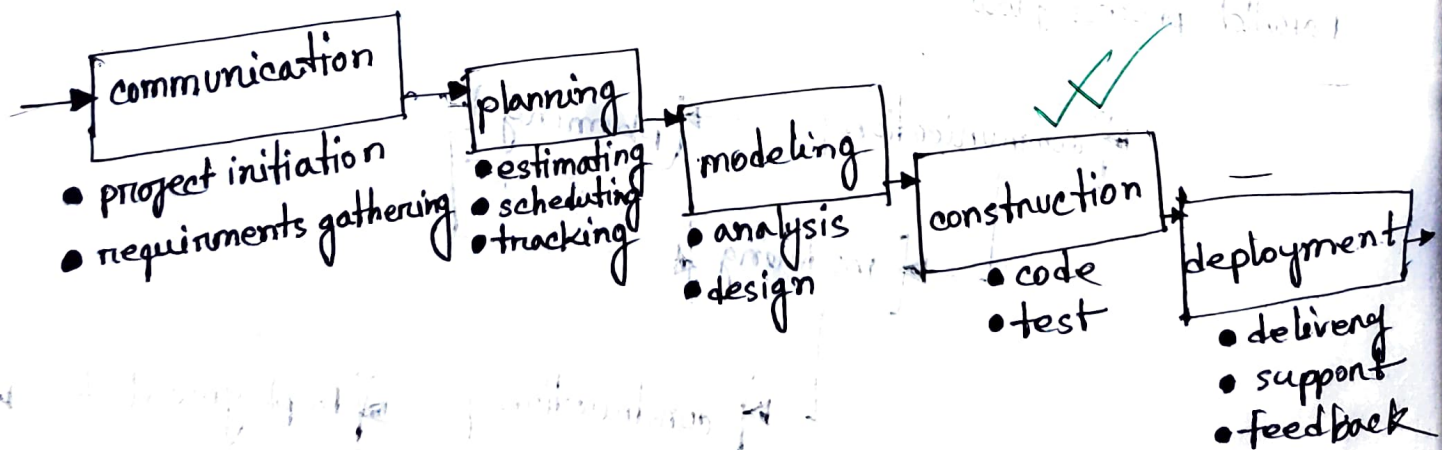


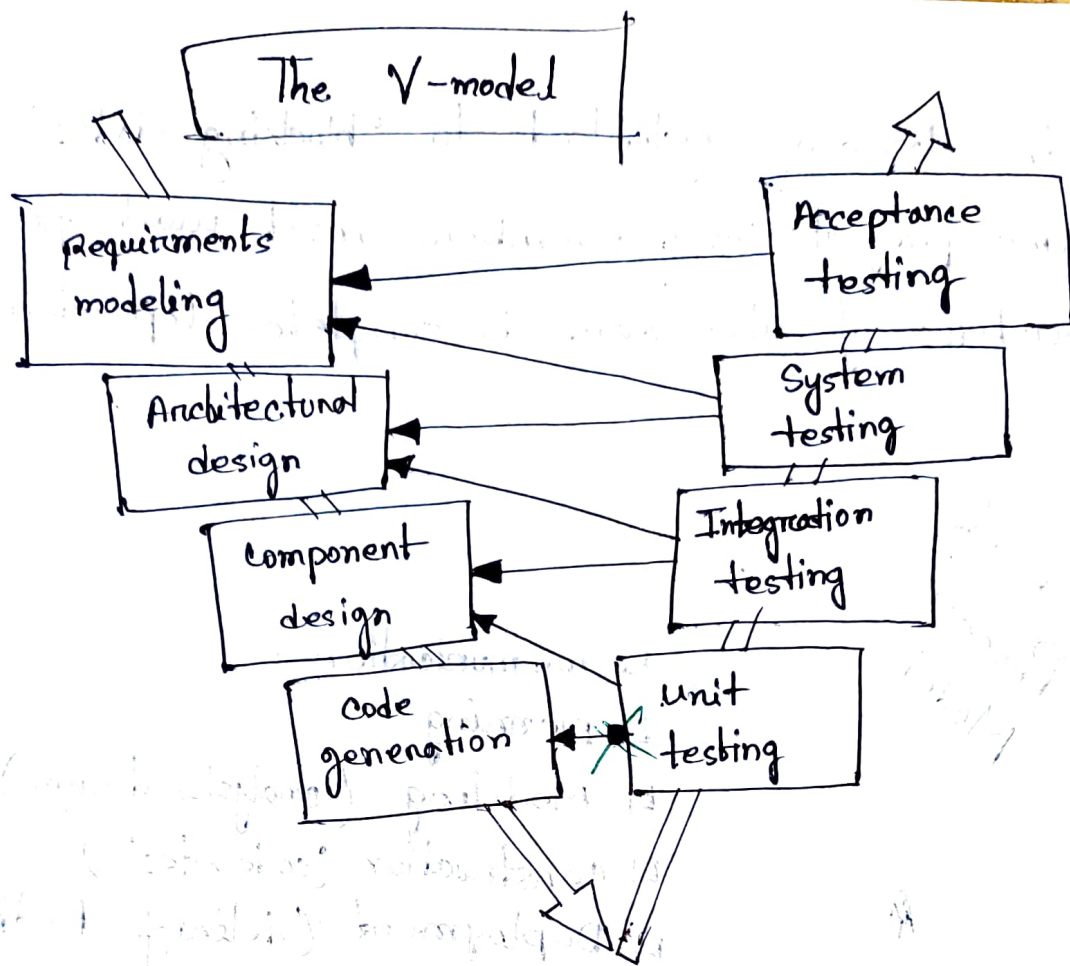
prescriptive
process
model

The waterfall model

There are times when the requirements for a problem are well understood - when work flows from communication through deployment in a reasonably linear fashion. This situation is sometimes encountered when well-defined adaptations or enhancements to an existing system must be made.

The waterfall model sometimes called "classic life cycle". suggests a systematic sequential approach & progresses through planning, modeling, construction and deployment culminating in ongoing support of the completed software.





The V-model provides a way of visualizing how verification and validation actions are applied to earlier engineering work.

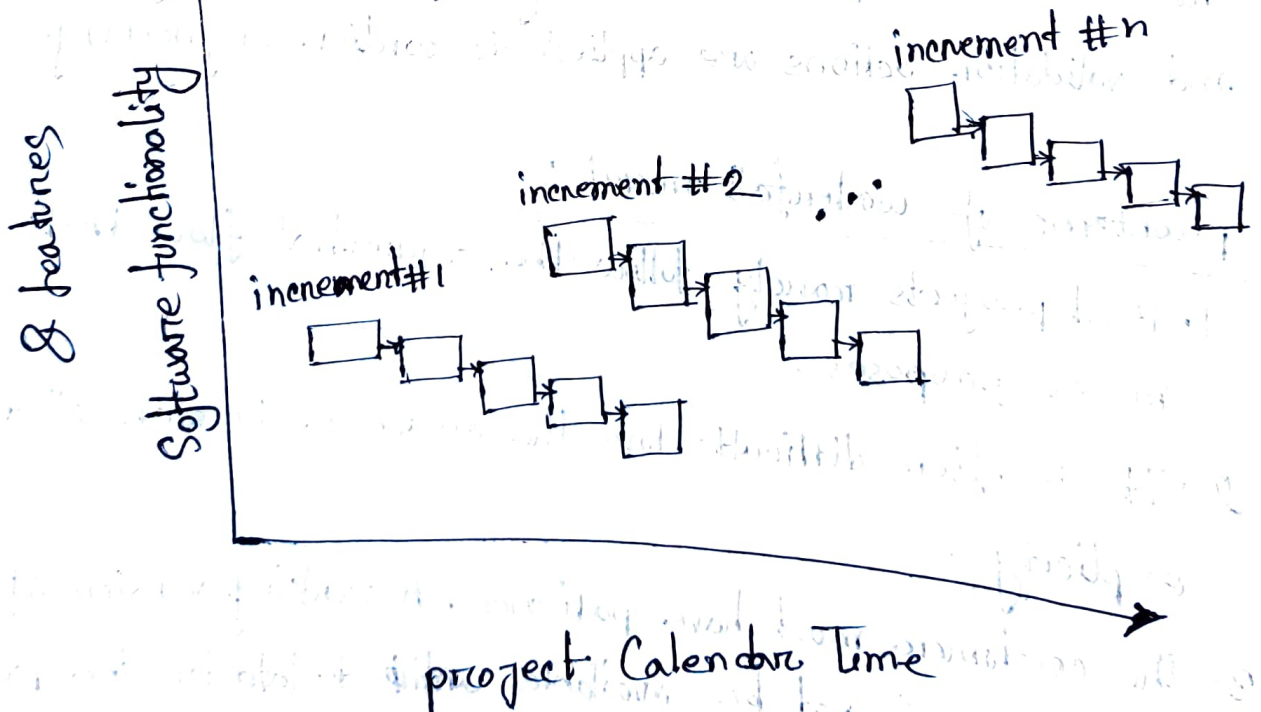
problems of waterfall model:

1. Real projects rarely follow the sequential flow that the model proposes.
2. It is often difficult for the customer to state all requirements explicitly.
3. The customer must have patience. A working version of the program will not be available until late in the project time span.

* the classic life cycle leads to "blocking states" is which some project team members must wait for other members of the team to complete dependent tasks.

Incremental Process Model

- communication
- planning
- Modeling (analysis, design)
- construction (code, test)
- Deployment (delivery, feedback)



— combination of linear & parallel process model

example:

*** Word Processing Software

1st increment: File management, general editing, documentation

2nd increment: Higher level editing, adding document production

features

3rd increment: Spelling & grammar checking

4th increment: advanced page layout capability.

* any increment can incorporate the prototyping paradigm.

Incremental process is particularly useful when staffing is unavailable for a complete implementation

— Early increments can be implemented with fewer people.

If the core product is well received, then additional staff can be added.

Prototyping :

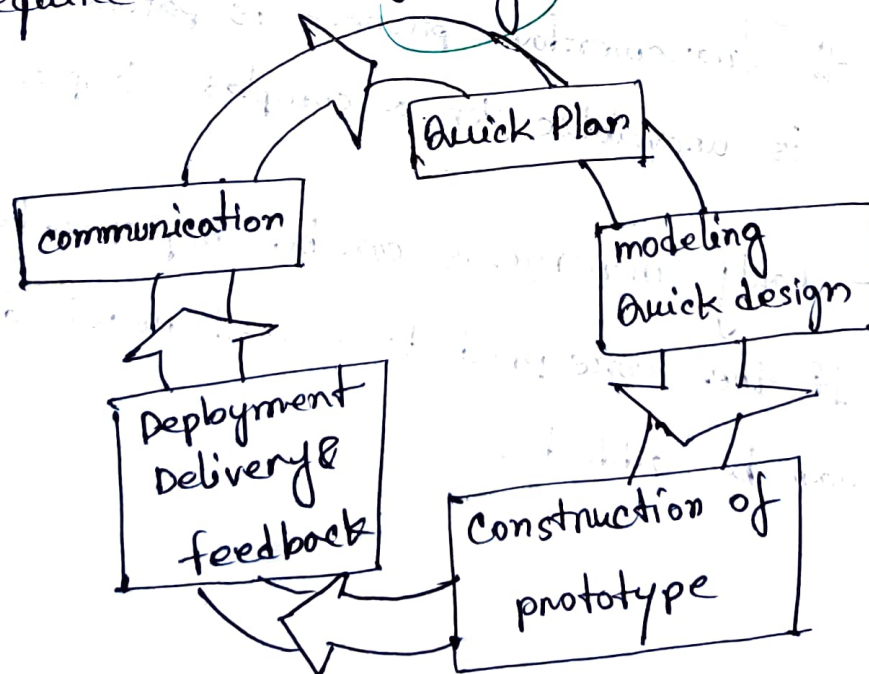
- aim fixed ✓
- but features, functions not identified ✓

In this case, a prototyping paradigm may offer the best ✓

Although prototyping can be used stand-alone ✓
but that can be implemented within the context of any one

the prototyping paradigm assists ^{someone} to better understand.
- what ~~to do~~ is to be built when requirements are fuzzy ✓

The Prototyping Paradigm



The Spiral Model:

- An evolutionary software process model that couples
- the iterative nature of prototyping with the controlled & systematic aspects of the waterfall model.
 - Risk-Driven process model

system's degree of

Characteristics:

- Cyclic Approach for incrementally growing
- definition ✓
- implementation ✓
- Decreasing its degree of risk ✓

- Divided into a set of framework

↳ into segment

- implied by a circuit around the spiral in a clockwise direction beginning at the center.

development: 1st circuit around: Product Specification
next circuit around: Develop a prototype

then progressively more sophisticated versions

Each pass-through the planning region results in adjustments to the project plan.

- Can be adopted to apply throughout the life of the computer software.
- iteration until concept development is complete.
- Can commence "new product development project".
- Enhance product.

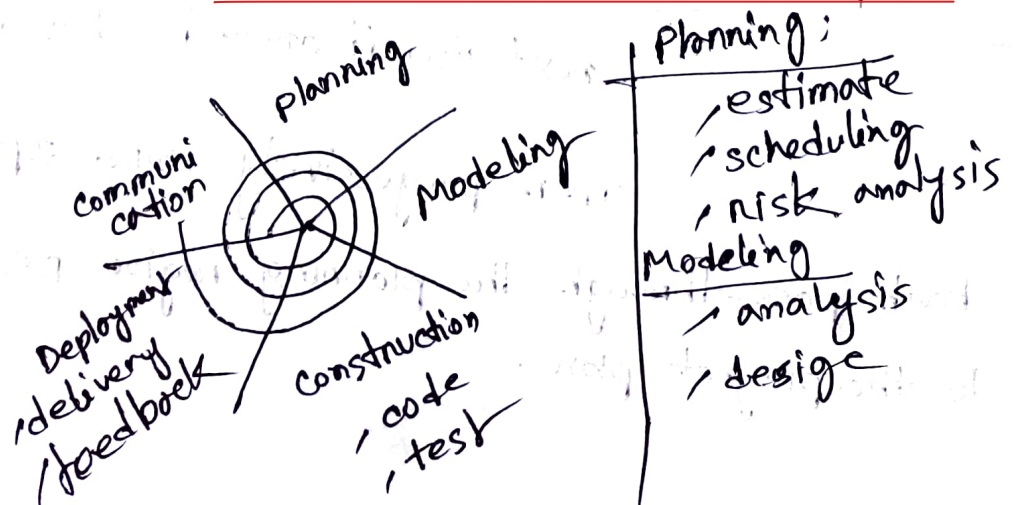
*** The spiral model is a realistic approach to the development of large-scale systems & software.

⑧ Iterative & Risk-driven Software development process

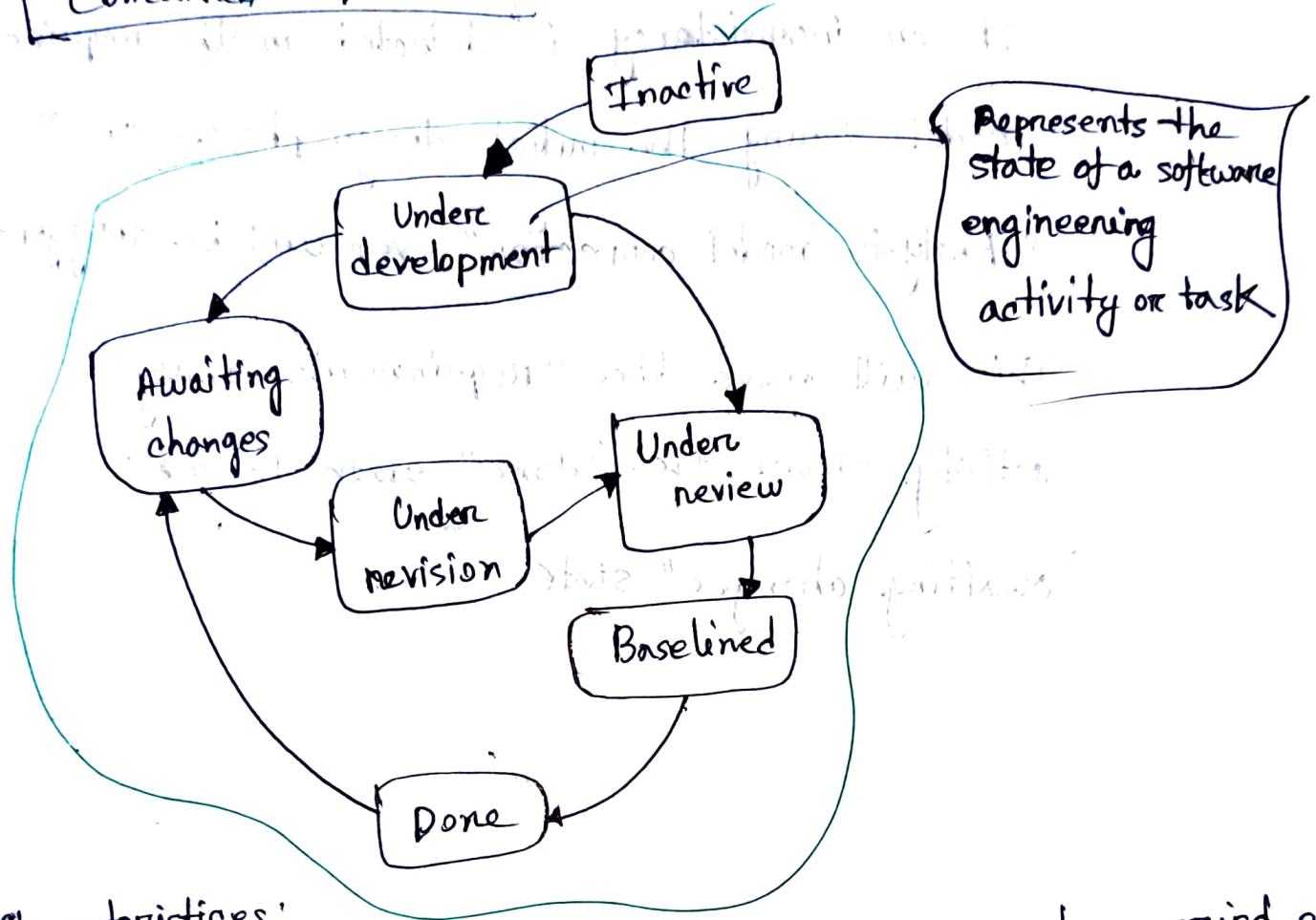
Disadvantage:

- Costly
- need more time
- not efficient for small one

Diagram:



Concurrent Model :



Characteristics:

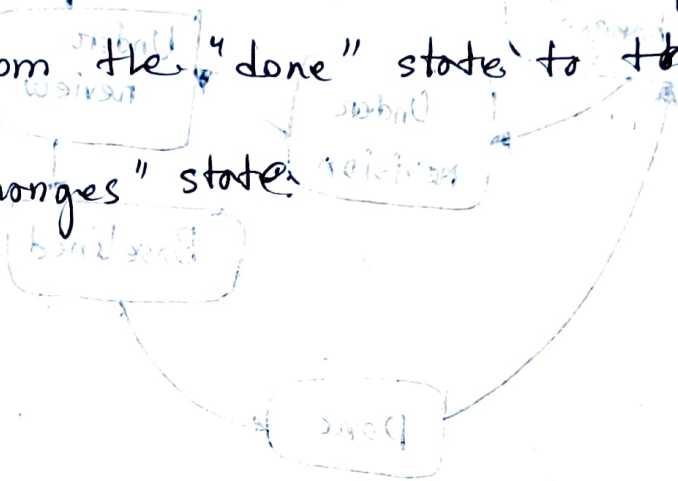
- Each phase of software development can be carried out simultaneously or iteratively.
- Quick adaption to changes is possible
- Effective for large and complex software projects

*** All software engineering activities can proceed concurrently, but they exist in different states.

For Example :

If an inconsistency is detected in the requirements model during the initial design phase, an "Analysis model connection" event will be triggered.

This will move the "requirements analysis" activity from the "done" state to the "awaiting changes" state.



④ PSP (Personal Software Process):

PSP is a structured method that helps individual software developers improve their work quality & efficiency.

focuses on personal performance.

PSP Activities:

1. Planning:

- Estimate size, time & defects
- Identify tasks and create schedule

2. High Level Design:

- Design components & external specs.
- Build prototypes if needed.

3. High Level Design Review:

- use formal reviews to find error

4. Development:

- Refine design, write & review code, test & record
matrices

5. Postmortem:

- Analyze all collected data to see
 - what worked well
 - what needs improvement.