

UNIT II : Prescriptive Process Models

* A Generic Process Model

- A process model was defined as a collection of work activities, actions and tasks that are performed when some work product is to be created
- Each of these activities, actions and tasks reside within a framework or model that designs & defines their relationship with the process & with one another
- From the Fig. below, each framework activity is populated by a set of software engineering activities
- A generic process framework for software engineering define five framework activities
 - Communication
 - Planning
 - Modeling
 - Construction
 - Deployment

Communication: The intent is to understand stakeholders' objectives for the project and to gather requirements that help define software feature & functions.

Planning: It act as an map (software project plan) which defines the software engineering work by describing the technical tasks to be conducted, the risks resources, work product & work schedule.

Modeling: Modeling is used for better understanding of software requirements & the design that will achieve those requirements.

Construction: It combines code generation & the testing that is required to uncover errors in the code.

Deployment: The software is delivered to the customers who evaluates the delivered product & provides feedback based on evaluation.

Software Engineering process framework activities are complemented by a number of umbrella activities. Typical umbrella activities includes:

- a) Software Project Tracking & Control : Allows the software team to assess progress against the project plan & take away necessary action to maintain schedule.
- b) Risk Management : Assesses risk that may affect the outcome of the project or quality
- c) Software Quality Assurance : Defines and conducts the activities required to ensure software quality
- d) Technical Reviews : Accesses software engineering work product in an effort to uncover & remove errors before they are propagated to next quality
- e) Software Configuration Management : Manages the efforts of change throughout the software process
- f.) Reusability Management : Defines criteria for work product reuse
- g.) Work Product Preparation & Production : In compasses the activities required to create work products such as models, documents, logs, forms and lists.

Software Process

Common Process Framework

Umbrella Activities

Framework Activities

[Tasks]

Milestone, Deliverable

SQA Points

Fig. A software Process Work

* Software Engineering a Layered Technology

- The foundation for software engineering is the process layer. It is the glue that holds the technology layers together and enables rational and timely development of computer software
- Process defines a framework that must be established for effective delivery for engineering technology
- The software process forms the basis for management control of software project and establishes the context in which technical methods are applied, work product is produced, milestones, quality is ensured
- Software engineering method provides the technical how-to's building software. Methods include communication requirement, analysis, designs modeling, program construction, testing and support.
- Software engineering tools provides automated or semi automated support for the process or the methods. Eg :- computer - aided software engineering

- Any engineering approach must rest on an organizational commitment to quality. Total quality management, Six Sigma provide a continuous process improvement culture. The bedrock that support software engineering is a quality focus.

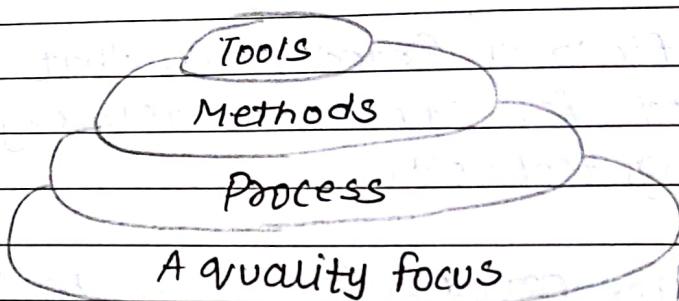
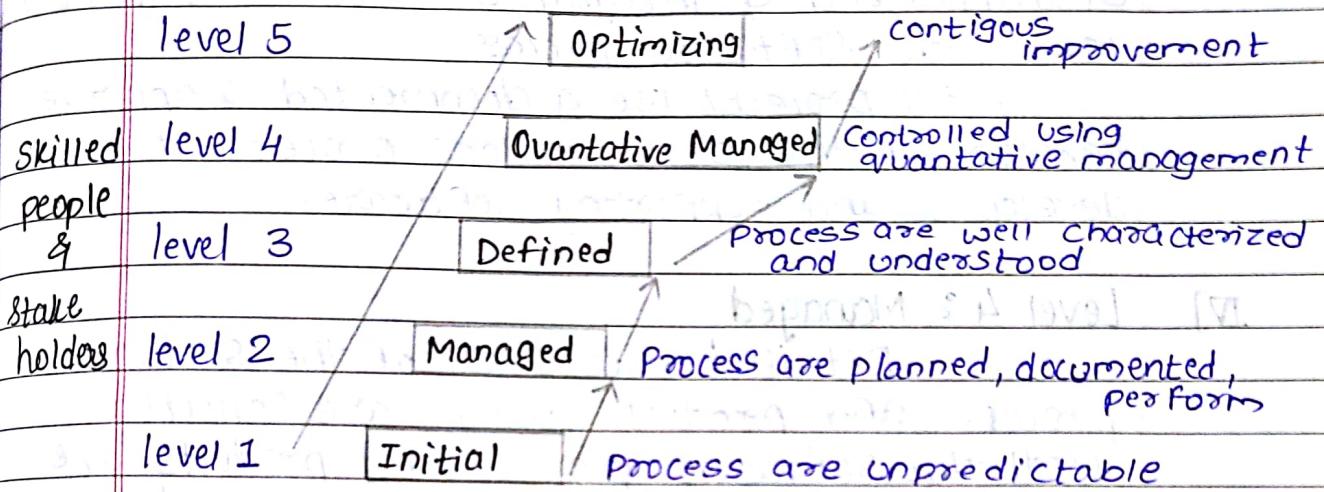


Fig:- Software engineering Layers

* CMMI [Capability Maturity Model Integration]



- The SEI approach provides a measure of the global effectiveness of a company's software engineering practises and establishes five process maturity levels that are defined in the following manner.

I] Level 1 : Initial

- The software process is characterized as adhoc & occasionally even chaotic. Few processes are defined & success depends on individual efforts

II] Level 2 : Repeatable

- Basic project management processes are established to track cost, schedule and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar application

- III] Level 3 : Defined
- The software process for both management and engineering activities is documented, standardized & integrated into an organization wide software process
 - All projects use a documented & approved version of the organizations process for developing and supporting software.

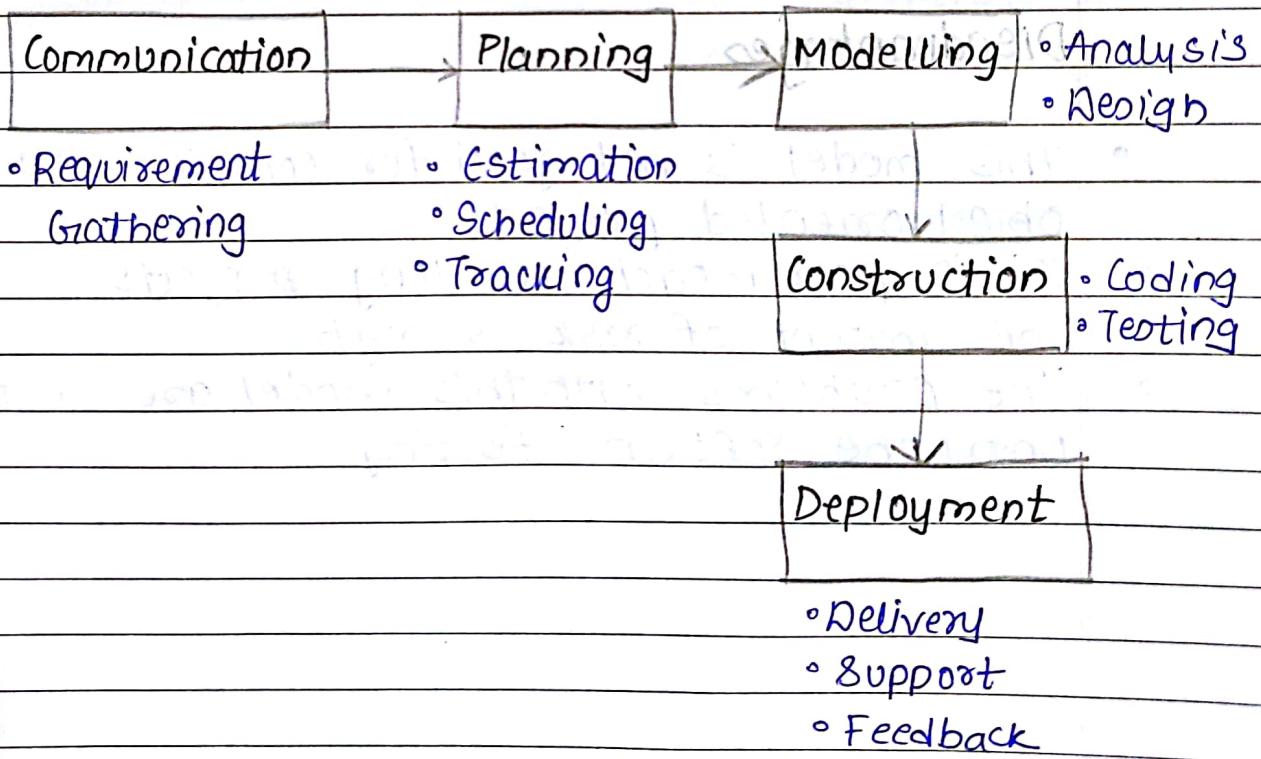
IV] Level 4 : Managed

Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled using detailed measure.

PREScriptive PROCESS MODEL

10. The Waterfall Model

- The waterfall model is also called as 'Linear Sequential model' or 'Classic Life Cycle Model'
- In this model, each phase is fully completed before the beginning of the next phase
- This model is used for small projects
- In this model, feedback is taken after each phase to ensure that the project is on the right path.
- Testing part starts only after the development is complete



Advantages

- The waterfall model is simple and easy to understand, implement, and use.
- All the requirements are known at the beginning of the project, hence it is easy to manage.
- It avoids overlapping of phases because each phase is completed at once.
- This model works for small projects because the requirements are understood very well.
- This model is preferred for those projects where the quality is more important as compared to cost of the project.

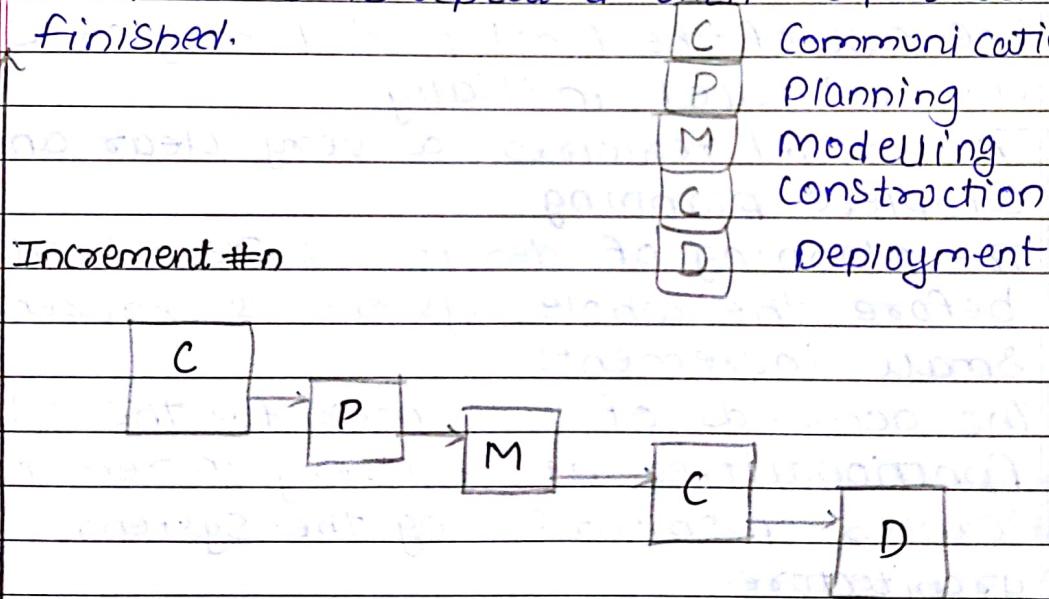
Disadvantages

- This model is not good for complex and object oriented projects.
- It is poor model for long projects.
- The amount of risk is high.
- The problems with this model are uncovered until the software testing.

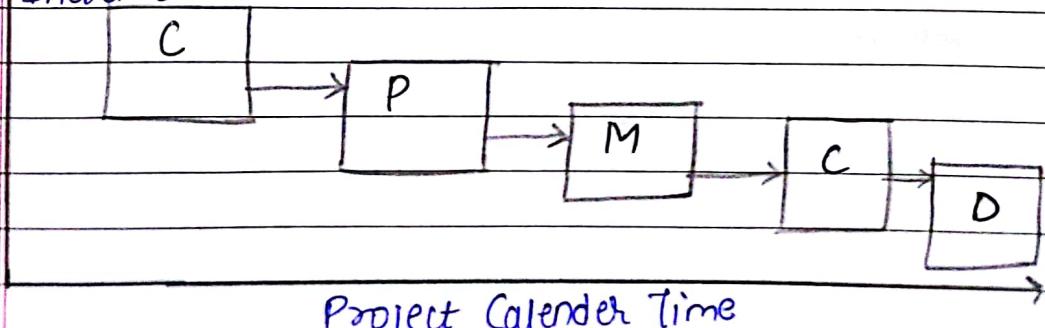
2] Incremental Process Model

- The incremental model combines the elements of waterfall model and they are applied in an iterative fashion
- The first increment in this model is generally a core product
- Each increment builds the product and submits it to the customer for any suggested modifications
- The next increment implements on the customer's suggestion and adds additional requirements in the previous increment
- This process is repeated until the product is finished.

Software functionality and features



Increment 1



Advantages

- This model is flexible because the cost of development is low and the initial product delivery is faster.
- It is easier to test and debug during smaller iteration.
- The software generates quickly and early during the software life cycle.
- The customer can respond to its functionalities after every increment.

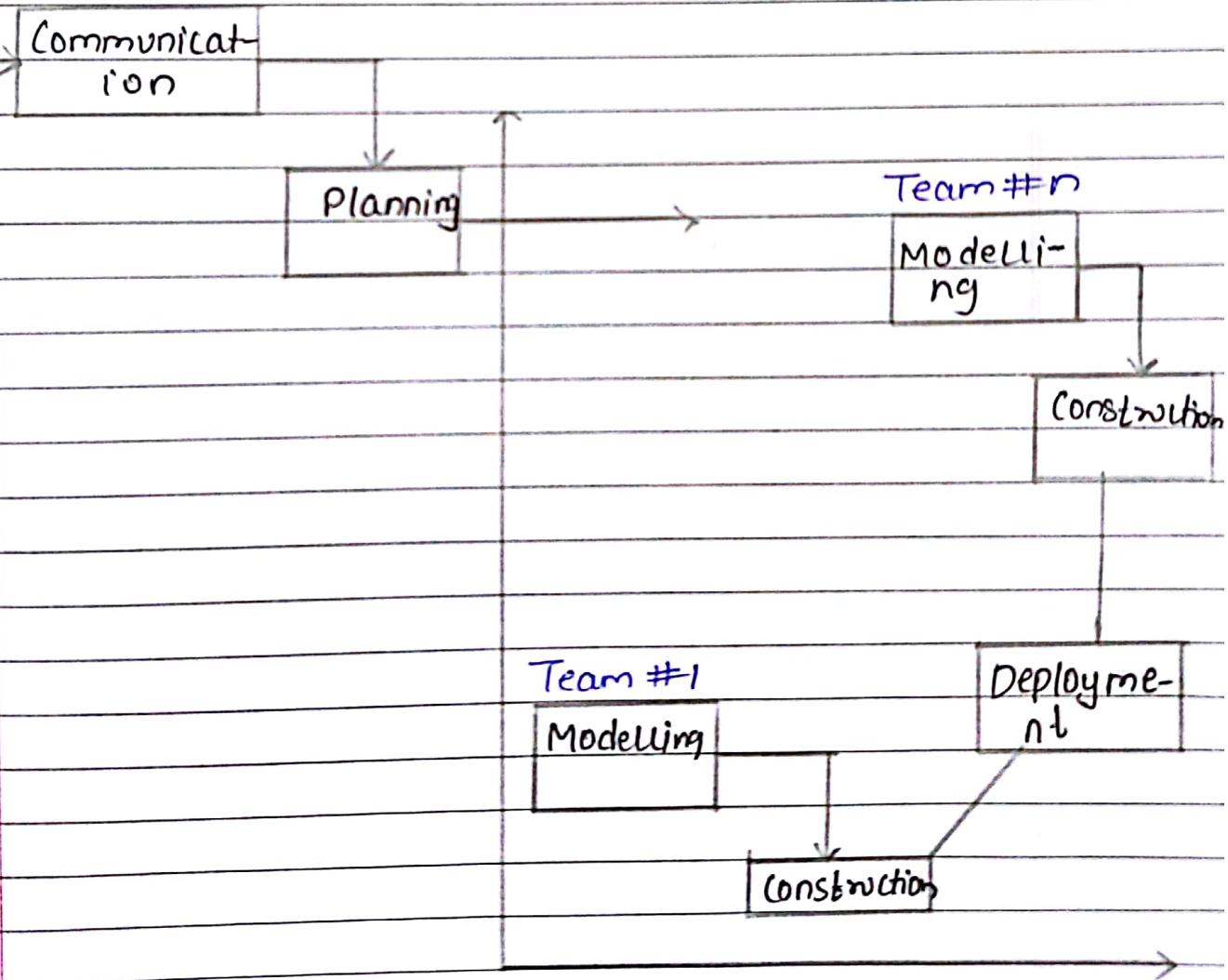
Disadvantages

- The cost of the final product may cross the cost estimated initially.
- This model requires a very clear and complete planning.
- The planning of design is required before the whole system is broken into small increments.
- The demands of customer for the additional functionalities after every increment causes problem during the system architecture.

3] RAD Model

copyright

- RAD is a Rapid Application Development Model
- Using the RAD model, software product is developed in a short period of time
- The initial activity starts with the communication between customer and developer
- Planning depends upon time initial requirements are divided into groups



Advantages

- Reduce Development Time
- Increases Reusability of component

Disadvantages

- Require highly skilled developer or designers
- Cannot deal with cheap project.

(Implementation)

or

Implementation

Implementation

or

Implementation

Implementation

Implementation

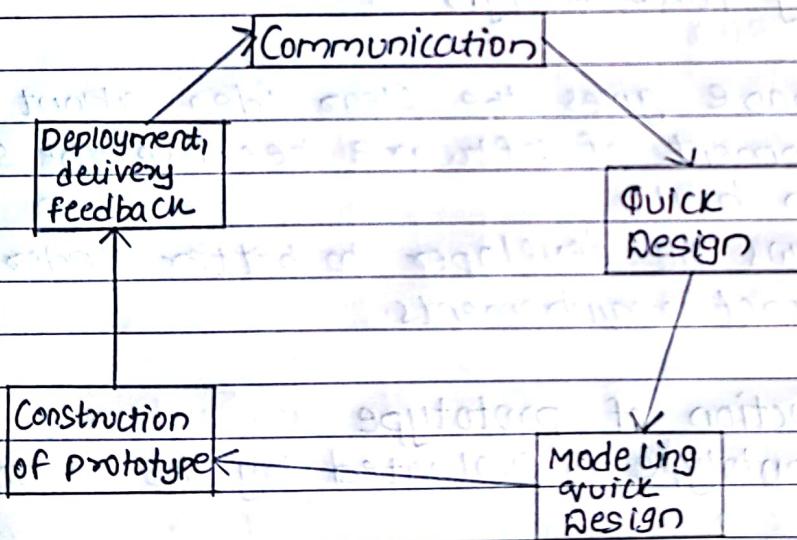
Implementation

Implementation

Evolutionary Process Models

1] The prototyping Model

- Prototype is defined as first or preliminary form using which other forms are copied or derived
- Prototype model is a set of general objectives for software
- It does not identify the requirements like detailed input, output
- It is a software working model of limited functionality
- In this model, working programs are quickly produced



1] Communication: In this phase, developer and customer meet and discuss the overall objectives of the software.

2] Quick Design:

- Quick Design is implemented when requirements are known.
- It includes only the important aspects like input and output format of the software.
- It focuses on those aspects which are visible to the user rather than detailed plan.
- It helps to construct a prototype.

3] Modeling Quick Design

- This phase gives the clear idea about the development of software because the software is now built.
- It allows the developer to better understand the exact requirements.

4] Construction of prototype

The prototype is evaluated by the customer itself.

5] Deployment, Delivery, Feedback

- If the user is not satisfied with current prototype then it refines according to the requirements of the user.
- The process of refining the prototype is repeated until all the requirements of users are met.

- When the users are satisfied with the developed prototype then the system is developed on the basis of final prototype.

Advantages

- Prototype model need not know the detailed input, output, processes, adaptability of operating system and full machine interaction.
- In the development process of this model users are actively involved.
- The development process is the best platform to understand the system by the user.
- Errors are detected much earlier.

Disadvantages

- The client involvement is more and is not always considered by the developer.
- It is a slow process because it takes more time for development.
- Many changes can disturb the rhythm of the development team.
- It is a thrown away prototype when the users are confused with it.

2) The Spiral Model

- Spiral model is a risk driven process model
- It is used for generating the software projects
- In spiral model, an alternate solution is provided if the risk is found in the risk analysis, then alternate solutions are suggested and implemented
- It is combination of prototype and waterfall model
- In one iteration all activities are done, for large projects the output is small

Planning
Estimation, Scheduling
Risk Analysis

Communication

Modeling

Deployment
Delivery
Feedback

Construction

Code Generation
Testing

Advantages

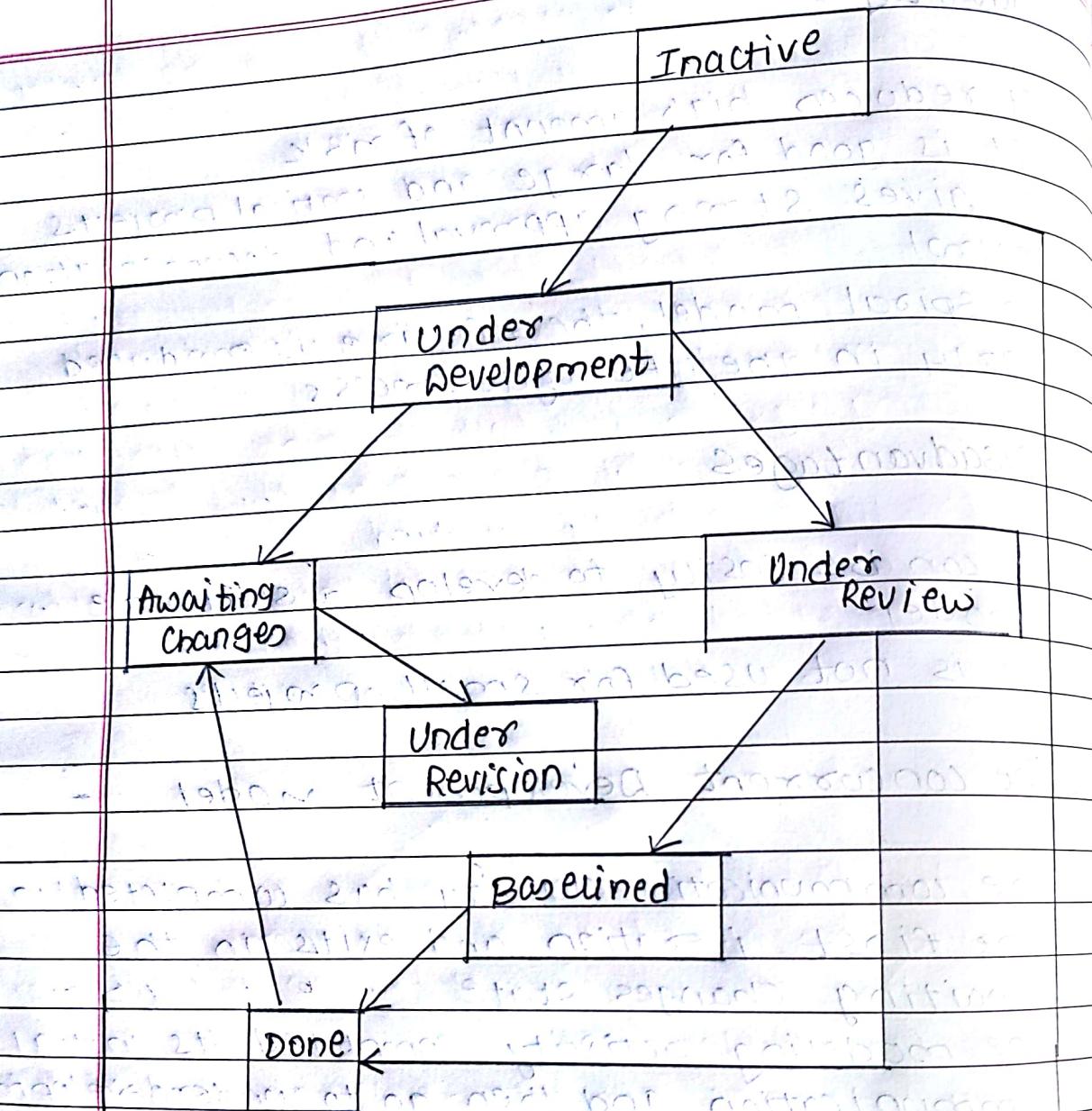
- It reduces high amount of risk
- It is good for large and critical projects
- It gives strong approval and documentation control
- In spiral model, the software is produced early in the life cycle process

Disadvantages

- It can be costly to develop a software model
- It is not used for small projects

3.] The concurrent Development model

- The communication activity has completed in the first iteration and exits in the awaiting changes state
- The modeling activity completed its initial communication and then go to underdevelopment state
- If customer specifies the change in the requirement, then the modeling activity moves from the underdevelopment state to awaiting state
-



Advantages

- This model is applicable to all types of software development process
- It is easy for understanding and use
- It gives immediate feedback from testing
- It provides an accurate picture of current state of a project

Disadvantages

- Requires to remember the status of the different activities
- Needs better communication between the team members. This may not be achieved all the time