

THE SOFTWARE PRODUCT AND SOFTWARE PROCESS

C

C

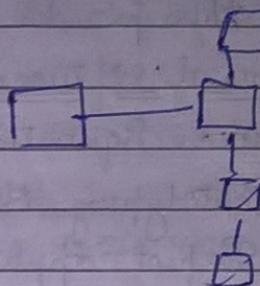
C

C

(1) Software Process -

It is defined as the structured set of activities that are required to develop the software system. The fundamental activities are -

- (1) Specification
- (2) Design and Implementation
- (3) Validation
- (4) Evolution/Evolution



A software process specifies a method of development software. A software project, on the other hand is a development project in which a software process is used and software product are the outcomes of a software project.

(2) Software Characteristics -

- ① Software is engineered, not manufactured.
- ② Software does not wear out.
- ③ Most software is custom built rather than being assembled from components.

(3) Common process framework -

COMMON PROCESS FRAMEWORK			
FRAMEWORK ACTIVITIES			
TASK SETS			
Tasks			
Milestones, deliverables			
SQA points			
UMBRELLA ACTIVITIES			

SQA → Software Quality Assurance.

The Software Process

#### ④ Software Process Models -

It is a simplified representation of software process. Each process model represents the a process from a particular perspective, and thus provides only partial information about that process.

#### Types of software process models -

(1) Linear Sequential Model (~~Waterfall Model~~)

(2) Prototyping Model

(3) Rapid Application Development (RAD) Model

(4) Evolutionary Process Models - (They are iterative. They are characterized in a manner that enables software engineers to develop increasingly more complete version of the software.)

#### ⑤ Software Process Characteristics -

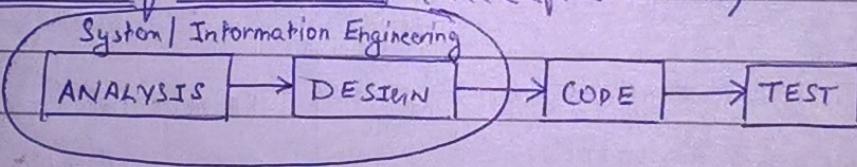
→ Predictability

→ Testability and Maintainability

→ Early defect removal and defect prevention

→ Process improvement.

#### ⑥ Linear Sequential Model (~~Waterfall Model~~)



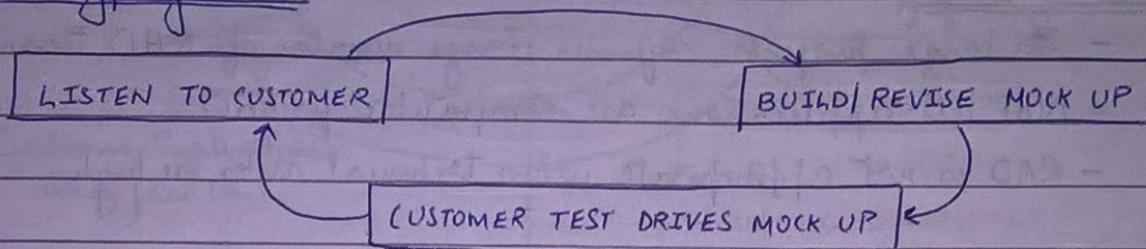
#### Advantages -

- Easy to understand
- for implementation of small systems.
- Identifies deliverables and milestones.
- Document driven
- Work well on mature products and weak teams.
- Widely used and known.

### Disadvantages -

- Idealized, doesn't match reality well
- Unrealistic to expect accurate requirements so early in project
- Software is delivered late in project, delays discovery of serious errors

### ⑦ Prototyping Model -



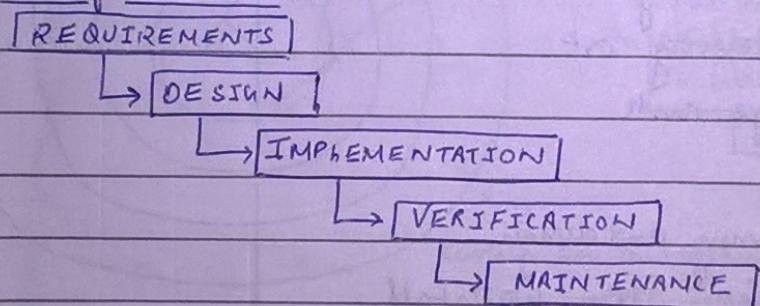
### Advantages -

- Better communication between customer and developer
- Satisfied User requirement
- High Reliability .

### Disadvantages -

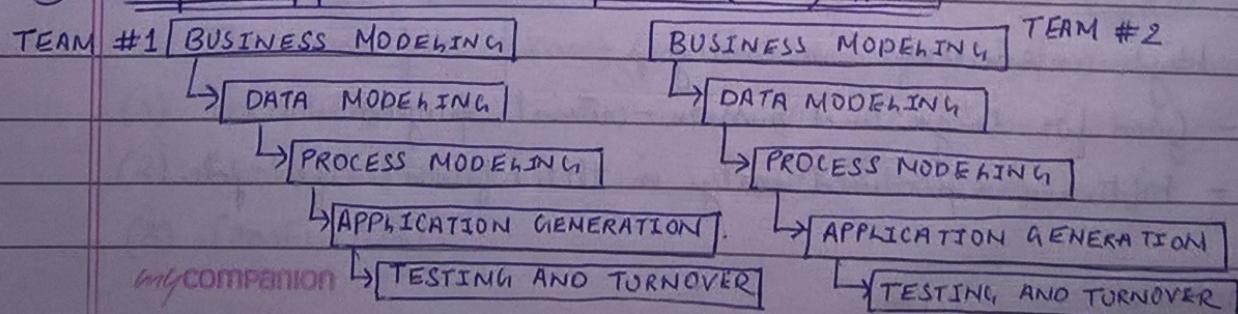
- Time Consuming
- Increase complexity of the system .

### ⑧ Waterfall Model -



Advantages and Disadvantages is same as linear sequential model.

### ⑨ Rapid Application Development (RAD) model -



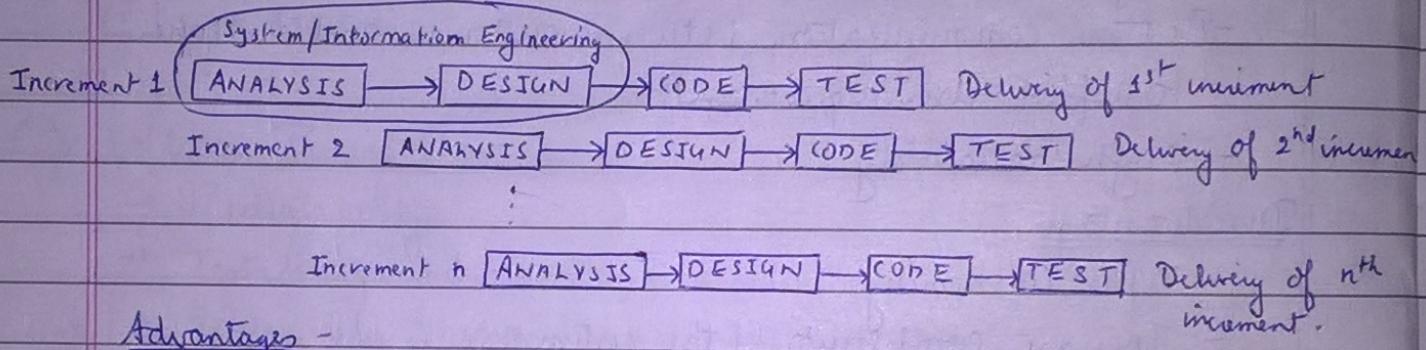
### Advantages -

- less time
- Reliable
- low Maintenance
- Fast Processing.

### Disadvantages -

- For large projects, requires large number of R&D teams
- Not all applications are compatible for RAD
- RAD is not appropriate when technical risks are high.

### (10) Incremental Model -



### Advantages -

- Generates working software quickly
- Easier to test and debug.
- lower initial delivery cost
- Easier to manage risk.

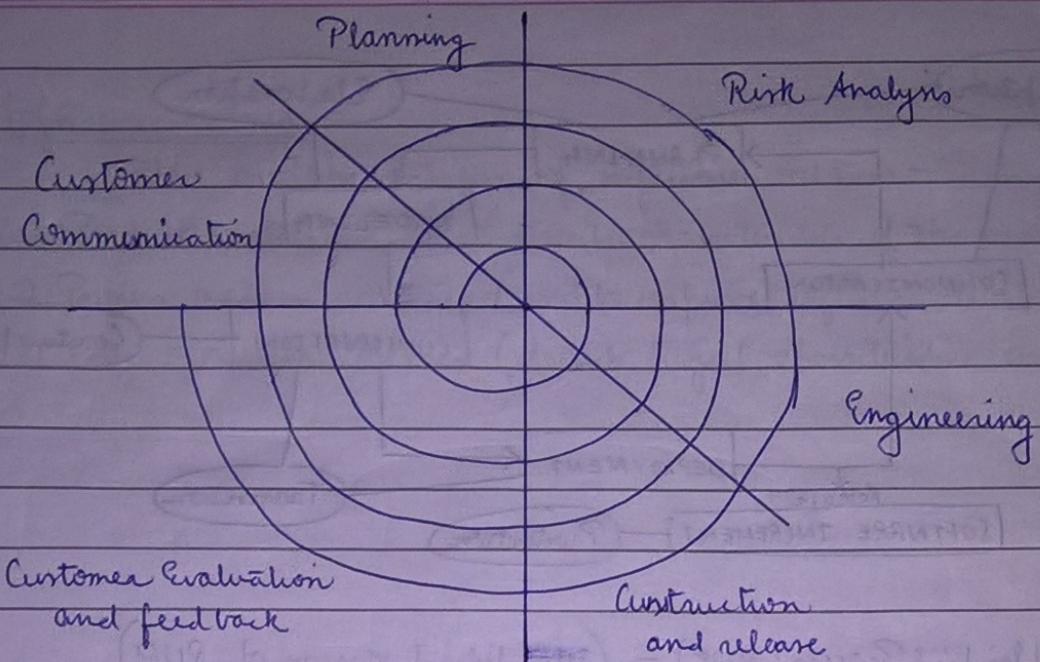
### Disadvantages -

- Need good planning and design
- Total cost is higher than waterfall.

### (11) Spiral Model -

#### Advantages -

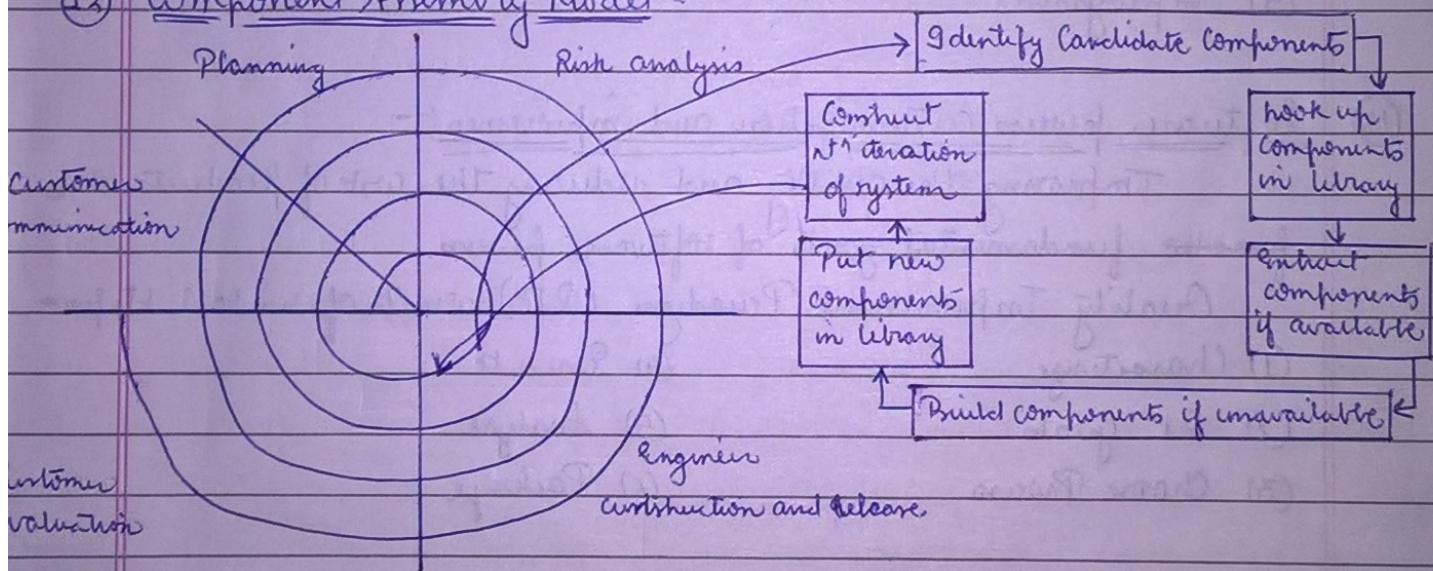
- High amount of risk analysis
- Good for large and mission-critical projects
- Software is produced early in the software life cycle.



### Disadvantages -

- Risk analysis requires highly specific expertise
- Doesn't work well for smaller projects.

### (12) Component Assembly Model -

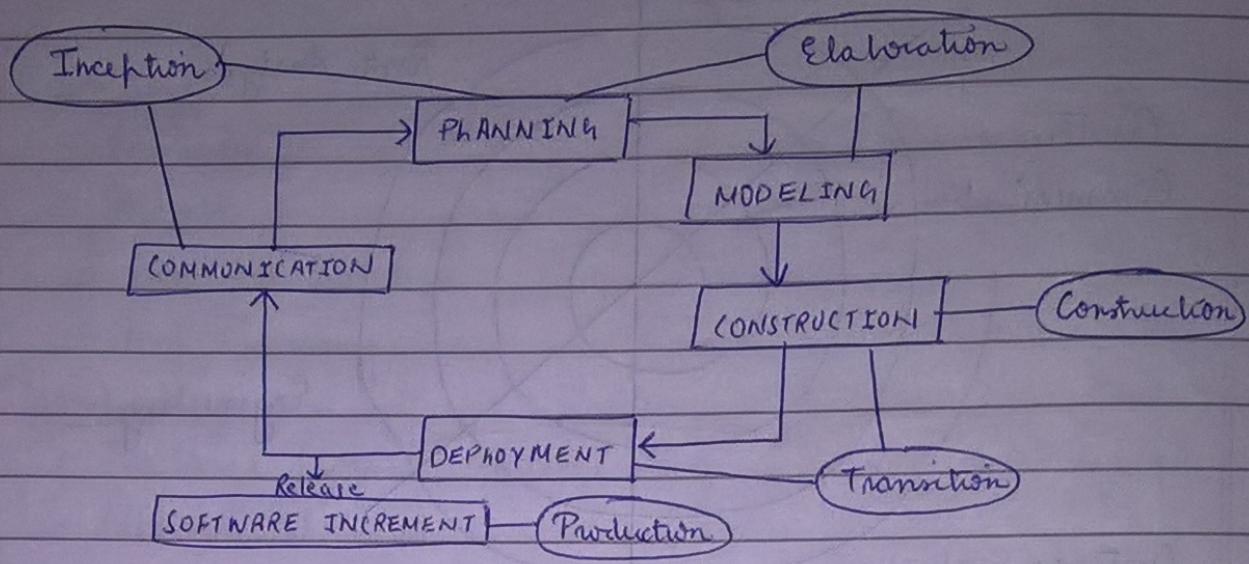


Concept of object-oriented programming is used.

### (13) Rational Unified Process (RUP) - (framework for object oriented models)

There are five phases for Rational Unified process model -

- (1) Inception
- (2) Elaboration
- (3) Construction
- (4) Transition
- (5) Production.



(14) Agile Unified Process (AUP) - (simplified version of RUP).

There are seven phases of AUP -

- (1) Model
- (2) Implementation
- (3) Test
- (4) Deployment
- (5) Configuration Management
- (6) Project Management
- (7) Environment

(15) Software process customization and improvement -

Improving the quality and reducing the cost of products are fundamental goals of software process.

Quality Improvement Paradigm (QIP) consists of six basic steps -

- |                    |             |
|--------------------|-------------|
| (1) Characterize   | (4) Execute |
| (2) Set Goals      | (5) Analyse |
| (3) Choose Process | (6) Package |

(16) Competency Maturity Model (CMM) -

It is used in assessing how well an organization's processes allow to complete and manage new software projects.

Various process maturity levels are -

- |                |                         |                       |                                   |
|----------------|-------------------------|-----------------------|-----------------------------------|
| (1) Initial    | → Disciplined processes | (4) Managed           | continuously improving processes. |
| (2) Repeatable | Standardized processes  | (5) Optimizing        |                                   |
| (3) Defined    | only companion          | Predictable processes |                                   |

(17) Software metrics -

There are three types of software metrics -

- (1) Product metric metrics (describe characteristics of the product e.g.- design)
- (2) Process metric metrics (used to improve software development & maintenance)
- (3) Project metric metrics (describe project characteristics and execution)

17  
32