

## Software Design Engineering

What does software?

It is a collection of executable program/code associated with library and documents.

What does engineering?

Engineering is a application of science, tools & methods to find out the cost effective solution.

Definition of SDE,

It involves architect, requirements, design, implementation and validation.

It is a process of

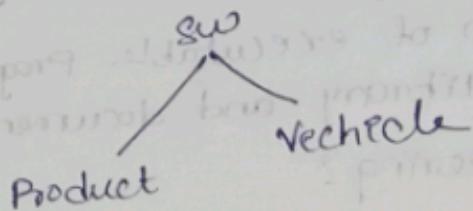
- ① Analyzing user need
- ② Designing
- ③ coding
- ④ testing the end user application.

Fundamental steps for SDE

- ① requirements analysis
- ② designing
- ③ coding
- ④ testing (in done in two ways (i) manually  
(ii) automatic)
- ⑤ implementation and maintenance.

## Evolving the role of software

software takes place in dual role



as a product it is embedded by computer hardware and network component.

### Vehicle (process)

As a vehicle it is an information transforming process like producing, managing, modifying, displaying or transmit the information single bit or multimedia.

### Characteristics of Software

- ① software is developed or engineering not manufactured.
- ② the ~~surface~~ software does not wear out (not purchasable) There is no expiry date for software.
- ③ software is a custom built.
- ④ software delivers the most important product of information.

(19) IT tra  
(19) IT P  
information  
theory

1990 or

Oshorne  
toffier  
still C

1990

Toffier

Hammer

2000's

Johnson

GENE

- A soft

Techno

T - T

M - M

P - P

a -

- (ii) It transform the personal data.  
(iii) It provide the gateway of worldwide information.

History :-

1990 or 1980.

Oshorne (1<sup>st</sup> version of software)

Toffler (conveying the information)

Stoll (interchange the knowledge to wwi)

1990

Toffler (convert to the powershift)

Hammer (provide the information based on)

2000's

Johnson (interchange the information)

GENERIC VIEW OF SOFTWARE / Layered Technology

A software engineering is called as A layered technology in terms of 4 layers.

T - Tools

M - Methods

P - Process

Q - Quality

## Tools

Providing supports to methods

## Methods

How the software Engineers plan like

## Process

Foundation of software (or) framework of software.

## Quality

It is a base for software. It is a backbone.

## def of Alayered

Software engineering use of sound engineering principle in order to obtain software economically i.e. reliable & work efficiently on real-time machines.

→ The foundation for Software engineering is called as "Process layer" it defines the framework for effective delivery of software.

⇒ Then basement is called as quality it is used to measure the process is called as "bedrock"

- Methods supports for process
- Tool it support for methods it may be automatic / semiautomatic

## Process Assessment

### Process Pattern

Pattern: software process is defined as collection of pattern. Pattern provide the template. Template is further used of assessment.

### Different types of pattern

#### ① Process template

- (i) Patent name
- (ii) Intent
- (iii) type
  - \* task
  - \* stage
  - \* phase

#### ② Initial contact

Before using we need to define

#### ③ problems

#### ④ solutions

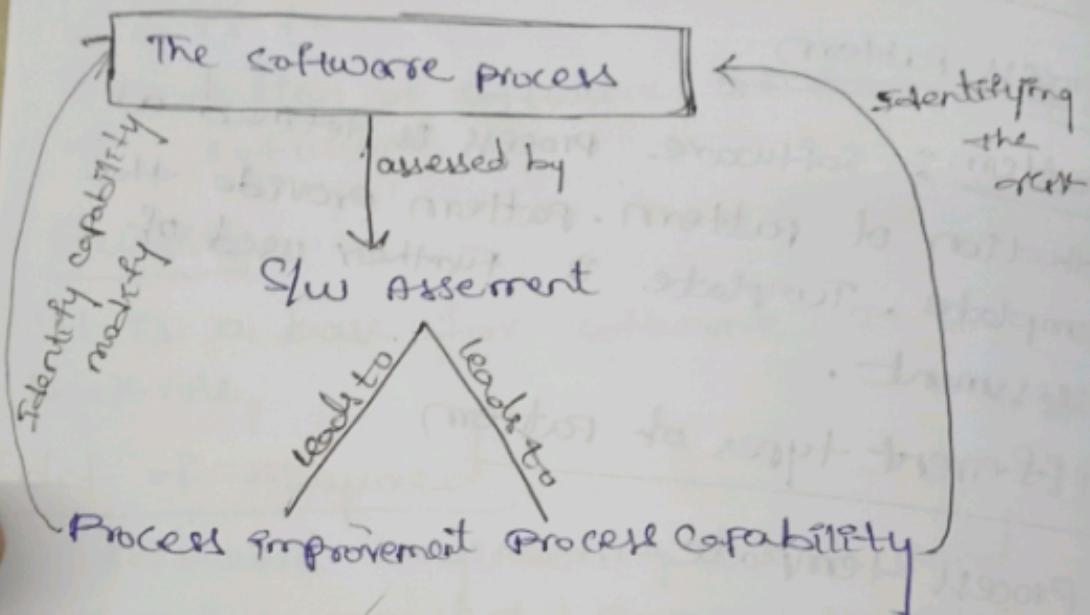
#### ⑤ Result

#### ⑥ Related Problem

## Process Assessment

Process Pattern + SW Project  $\rightarrow$  new SW

This entire process should continuously assess  
It is called as Process Assessment



## Techniques in Process Assessment

### ① SCAMPI (standard CMMI Assessment)

Method for Process Improvement  
done in 5 ways

- (i) Initiating
- (ii) diagnosing (Test / implement the project)
- (iii) establishment
- (iv) Action (To work on project)
- (v) learning (we need to checkout)

## ② CBA IPI

(CMMI Based Appraised Internal Process Implementation)

CBA → in order to do Assessment

## ③ SPICE (ISO)

It is a set of requirement process for Software Assessment

## ④ ISO 9000 / 2000

ISO having PDCA cycle

(Plan do check act)

## CMMI

[capability Maturity Model integration]

→ CMMI is a procedure for software development model it is assessed in organisation and software development process. This used bust up the development process & reduces the risk in proj.

→ CMMI is not a software process model it is used to measure the Maturity of project

→ (CMMI) if is used to analysis the existing process & classify the flow & strength

→ To measure the assessment to convert weakness to the strength.

→ It represent in two ways

i) continuous model

ii) stage model.

(i) <sup>mix</sup> continuous model describes the dimension of the project.

(ii) stage model used to measure the whole organization.

LEVELS OF CMM

1) incomplete [level 0]

2) performe

3) managed

4) define

5) Quantitative Management

6) optimization.

Process model

→ process model define different set of activity action & work for the Project that are required to built a high quality of software.

→ It provide road map for software problem.

→ It display how each state in SDLC (Software Development life cycle) work & what are the process required for each & every step in SDLC.

→ software Engineering have choose Process framework have the following activities.

- al of the organisation.
- 1) communication
  - 2) planning
  - 3) modeling
  - 4) construction
  - 5) deployment.

### Types of SDLC

Generally there are different types

- 1) Waterfalls model
- 2) Incremental Process model.
- 3) Evolutionary Process model.
- 4) RAD model.

### SDLC

- It is very important for software development process.
- It involve the entire process raw material / scrabed to development phrase.

### Example :

constructing a new house.

There is 6 phases in SDLC

- 1) requirement
- 2) analysis
- 3) Design
- 4) implementation / development.
- 5) Testing
- 6) Deployment and Maintenance.

## 1) Requirement phase

- This is important phase in SDLC
- It collects all the requirement from the user (client business need & document is required for business)
- document (may be differ) is like SRS (Software requirement specification)  
BRS (Business " )  
BS (Business specification)

## 2) Analysis

- Once the requirement is done the next step is to define document Product requirement & get approval from the customer
- It consists all the product requirement to be designed & developed during the project lifecycle.
- The key people are involved in this space like project manager, business analyst & senior manager.

## ③ Designing

Having two level

1) HLD [high level design]

2) LLD [Low level design]

### ① HLD

- It describes architecture of the project.
- This design is done by senior manager or architect.

### ② LLD

- It describes how each & every product should work & how every each & every component is working.

### ④ Implementation / and development

- All the coding comes under this phase

→ In this case we start building the software & writing the code based on the software. The outcome of this phase is source code document and development product.

### ⑤ Testing:-

- When the software is ready send to the testing theme to check it manually & automated.

→ In testing team they make sure the Quality Assurance for the product

→ The outcome of this phase is quality tested and testing artifact

### ⑥ Deployment

- After successful tests the product delivered to the end customer and deployment done by the engineer.

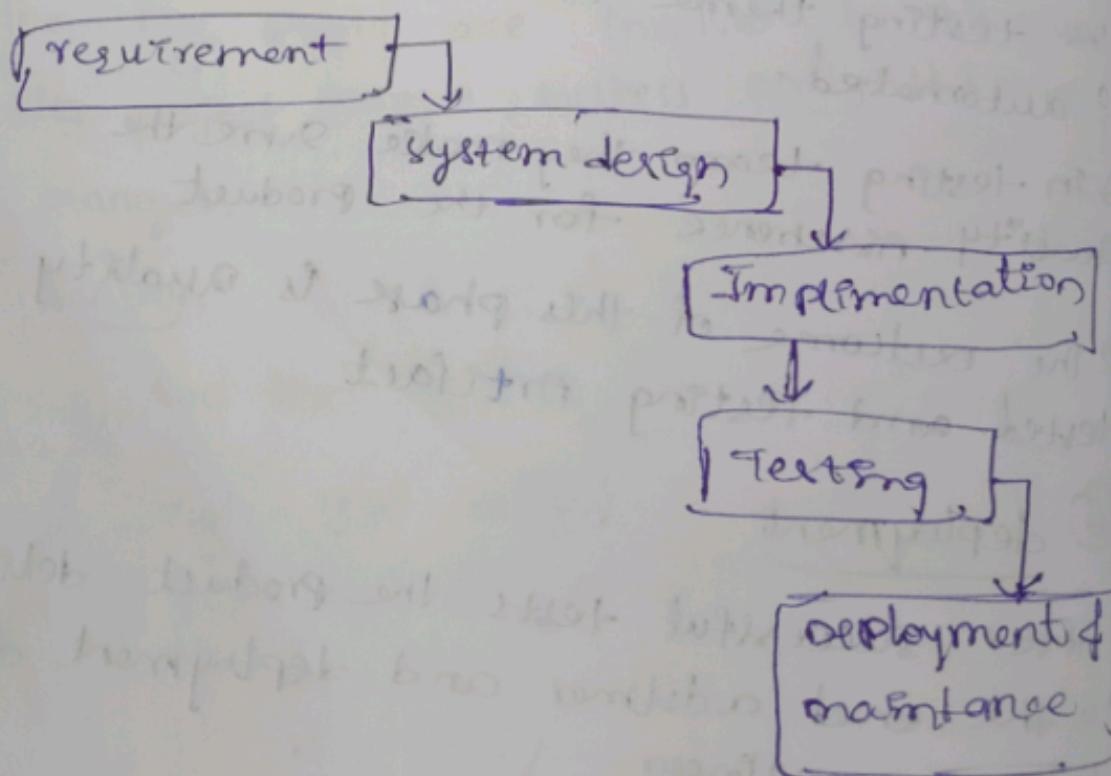
## Maintance

→ Once the customer start using the product if the problem it will come, we need a service re called as maintenance.

## Waterfall Model

- This is a first model in SDLC widely used in software development process.
- The whole software development process is divided into following stages. This is also called as classic life cycle. So, it is a systematic, sequential approach to software development process.

### Stages



## ① requirement analysis:-

The requirement for user need & expectation to be documented.

## ② system design

The system requirement & software component and overall system architecture is done by the system design.

## ③ Implementation

get a input from system design, here system create the small program called as unit.

## ④ Testing

→ testing is done by manual & automated

→ The each individual unit is tested is called as unit testing

## Integration

It combine all the units.

## ⑤ deployment & Maintenance

If all functional and nonfunctional are done the product is ready to delivered.

## Maintenance

If the product deliver to the customer in future if customer need any service, the product fare is called as maintenance.

## The problems in waterfall model

- ① real-time project rarely follow the waterfall model for sequential flow that model proposes.
- ② It's a linear model can't account the iteration indirectly.
- ③ It's difficult for customer to state all the requirement in the beginning of the project.
- ④ the waterfall model <sup>blocking, holding</sup> state which means, the some of the team members must wait for other team member should completed so, it is called as dependent task.

## \* Incremental Process model

- ① It's combine all the element of waterfall model applied for iteration.
- ② In this model the whole Project is divided into multiple Path. This path be called as module.

③ Incremental process model applied for system development process into no. of increments.

④ The certain no. of increments

⑤ Increment is also called is iteration

⑥ The iteration consist of same step in the project

so, the 6 steps are

① communication

② planning

③ modeling

④ constructing

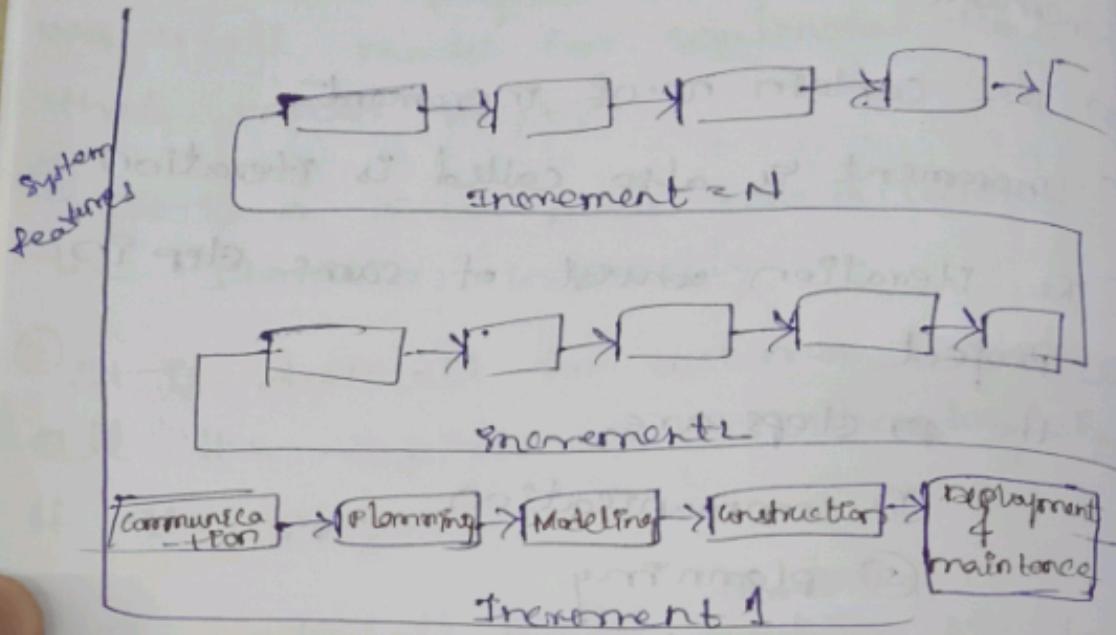
⑤ deployment

⑥ Maintenance.

→ so, the requirement phase is very important without requirement we can't develop the process

→ Incremental model is based on developing on project for initial implementation, exposing. The different versions until the system accepted as been completed.

- This is also called as linear sequence
- Incremental model apply linear sequence



- The basic requirement are identified
- The plan for the project is modification of the product betterment to meet the customer needs / requirement & delivers the additional functionality
- This process is repeated a until the completion of the project.

Advantages

- ① It project
- ② It
- ③ It
- ④ It

Disadvantages

- ① It
- ② It

## Advantages of Incremental Process

- ① It is suitable for the real-time project.
- ② It is more flexible.
- ③ It is very less cost compared to waterfall model.
- ④ The customer / client respond to built the product they can give the feedback if any changes is required.

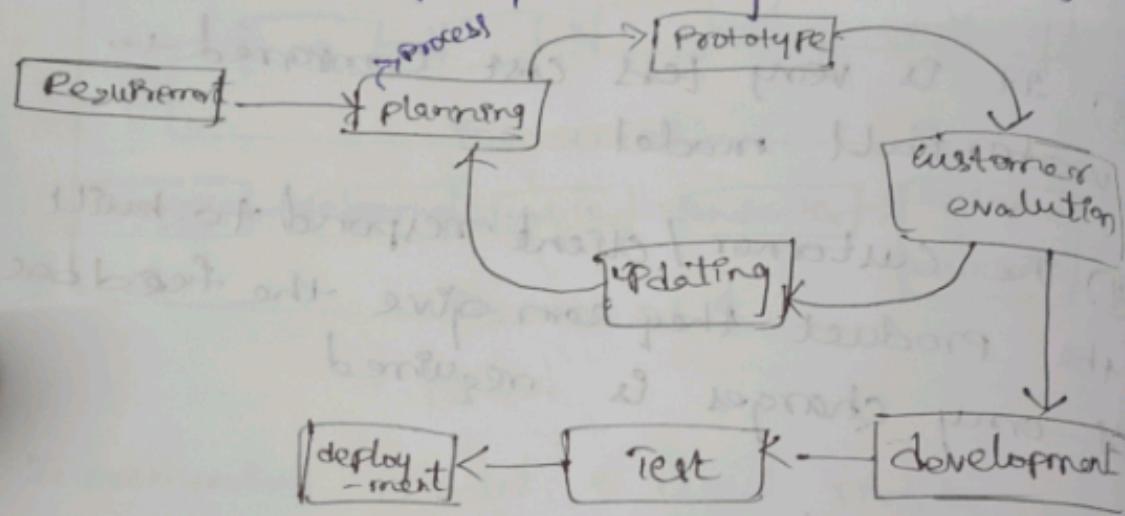
## Disadvantages

- ① The product cost is very high.
- ② They need good plan.

\* Evolution Process model,  
The model is called as Iteration Process

Prototype :-

- The customer define general objective
- for software process it does not identify the detailed input, processing and output



- Prototype model begin with customer requirement
- to define an overall objective for a software process.
- In this model planned quickly and modify.
- The design is focus on requirement of customer

Advantages :-

- This Prototype model does not need any detail input, processing, output

→ the customer actively involved in development process.

## Disadvantages:-

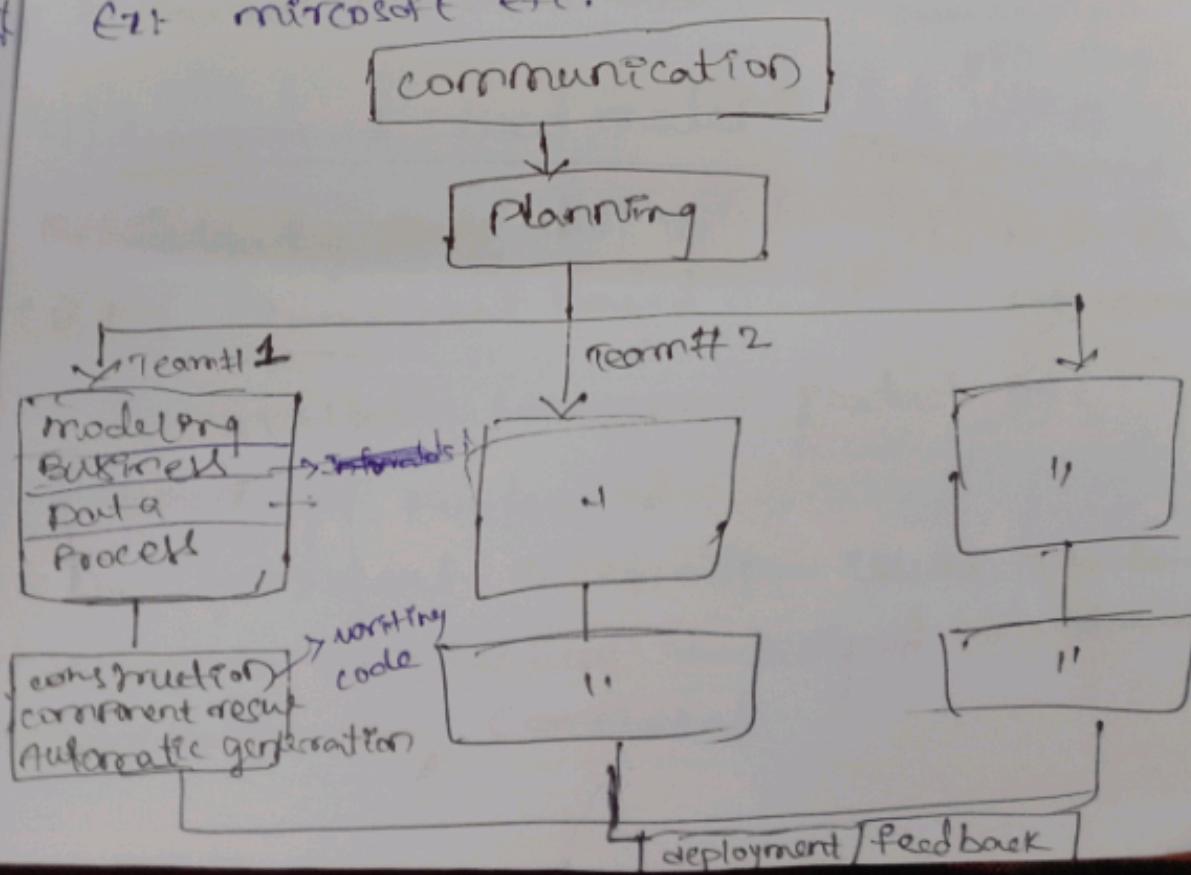
- It is impossible to know the project time
  - It very slow process.

## \* RAD Model

# Rapid Application Development model

- RAD Process is an incremental software process for a short development cycle (60-90 days)

- It is a high speed which is the rapid development achieved by using component based communication.



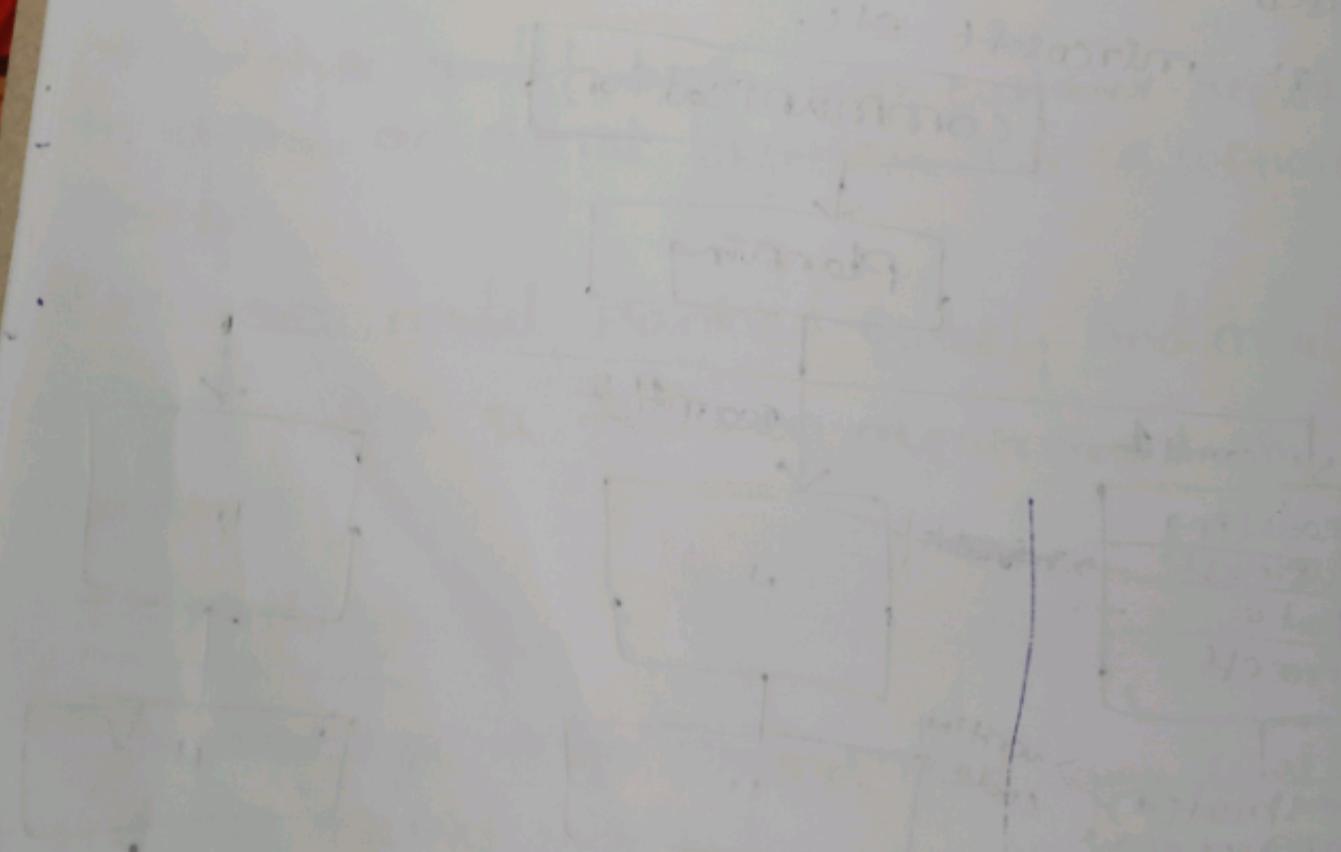
### Advantages :-

- flexible to change
- The development project time is very less
- Requirement reuse increases
- The model can adapt to any rules and technology.

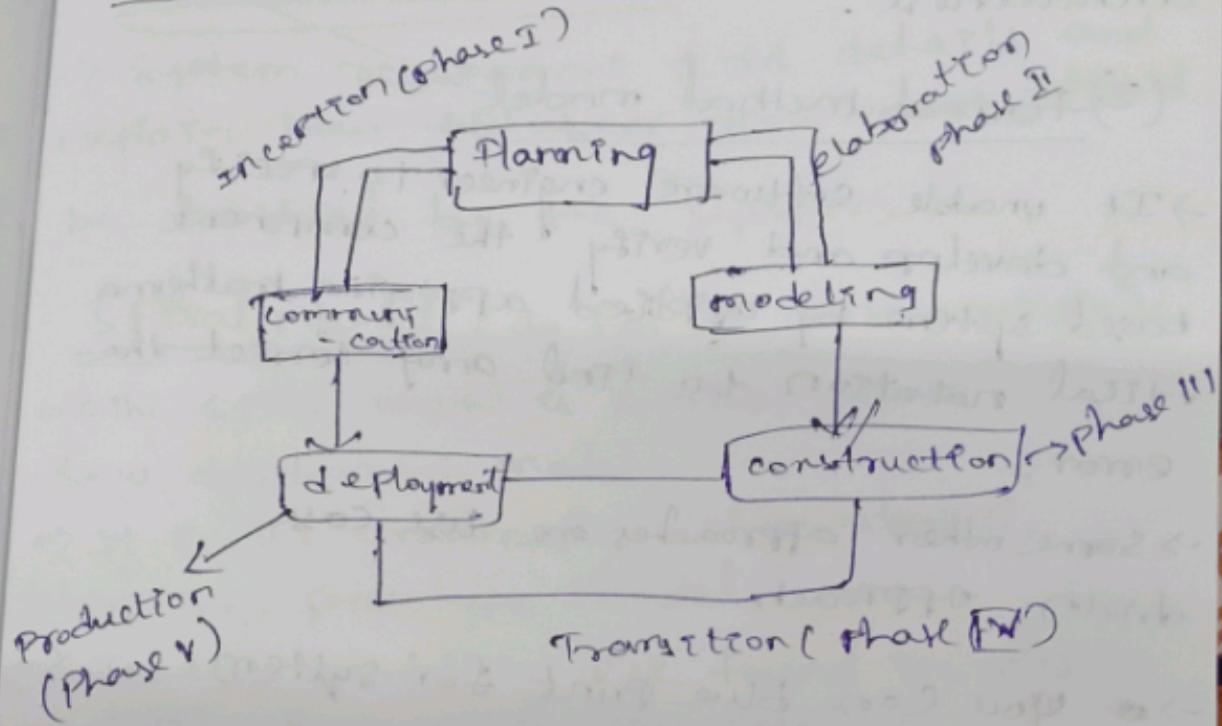
### Disadvantages :-

- The project required more team members to complete the project.
- high performance issue

## Unified Process Model



## Unified process Model



## Specialized Process Model

1) Component - based model  $\rightarrow$  (CBM)

2) Formal method model.

(i) Component based model

the following steps used in CBM (Component based model) to build the software.

- The available component product are Research or Evaluated
- The component integration issues considered
- Software architecture is designed to accommodate the component

→ Component are integrated into architecture.

## (2) Formal method model

- It enable software engineer to specify and develop and verify the component based system by applying mathematical notation to find any correct the error.
- Some other approaches are use case driven approach
- You can blue print for system process

## Advantages of unified process model

- It cover the complete software development life cycle
- Best support for development process

## Disadvantages

- complex implementation
- cost of the project is high

## ~~System Requirement~~

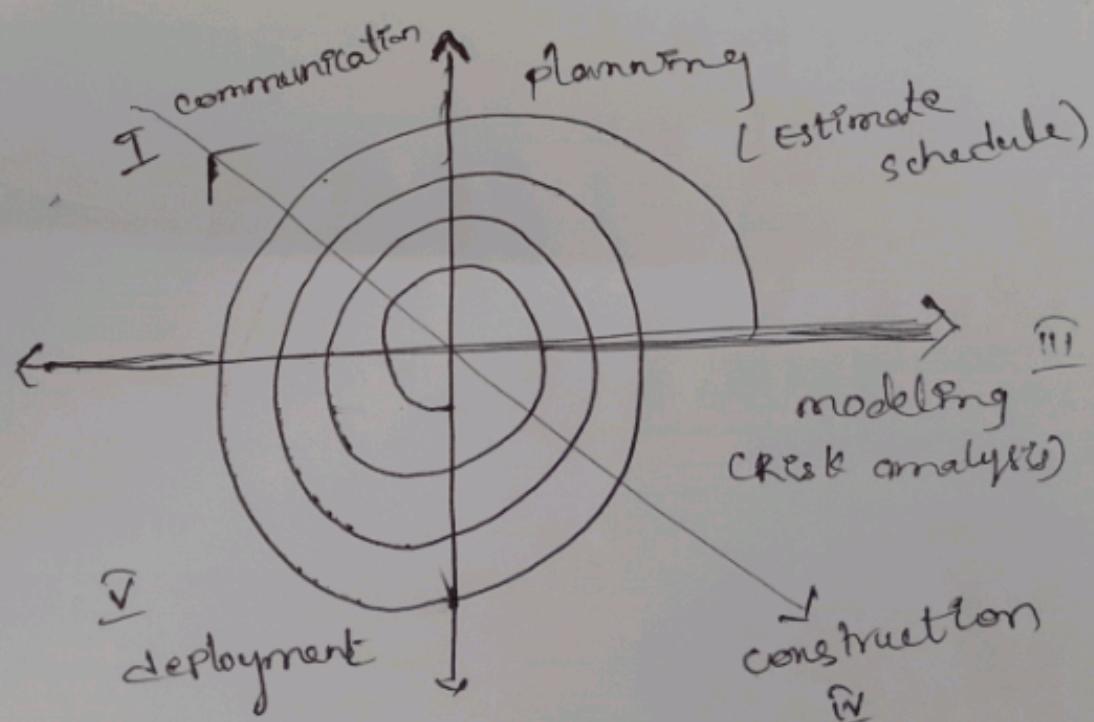
→ System requirement add details and explain how the user requirement should be provided by the system.

## Spiral model (do for the risk analysis)

- The spiral model is a risk driven software development model.
- It is the combination of waterfall, iterative, prototype model.
- In this model the risk found in risk analysis then alternative solution implemented.

→ It is also called as "Meta model"

## Work flow



## Phase I

communication or requirement gathering

→ Identification of requirements

## Phase II

Planning: we estimate the schedule of planning & prototype

## Phase III

modeling → further risk analysis

## Phase IV

construction: writing code for the Project

## Phase V

Deployment - all the feedback & support



## System requirement s-

- It is expanded version of user requirements.
- System requirement used by software engineer as the starting point for system design.
- It is a part of implementation. The system should be complete & consistent state.
- It is the part of contract for the implementation section of system it should be complete & consistent state.
- Natural language or processor language.
- It is used to write the system requirement.

like :

If feel some complexity like

1) Some word for some content.

2) The reader must find out when the requirement in different activities.

<u>Notation</u>	<u>Description</u>
1) Structured Notation	→ This is standard form of template to express requirement specifications.
2) Design description	→ It is used to specify the requirement by defining operational model of system
3) Graphical Notation	→ Defines the functionality of systems.
4) Mathematical Specification	→ Used the maths concept such as Finite State System (or) set

## ④ SRS :-

(Software requirement Specification)

- It is an official statement it should include both user requirement and system requirement
- It is a document mentioned the complete description about a system expected to perform like behaviour of system, functional and nonfunctional requirement

## Structure of the document

### IEEE STD Structure

#### 1) Introduction

- i) purpose of product
- ii) scope of
- iii) definition & abbreviation
- iv) References
- v) overview.

#### 2) General description

- i) Product perspective
- ii) Product function
- iii) User characteristic
- iv) General constraints
- v) Assumption

#### 3) Specific Requirement

#### 4) APPENDIX

#### 5) Index

## => Need of SRS Document

→ It provides structure and format

→ It exactly meet customer specification

orientated  
menu  
Soft  
It is a  
station of  
engineers  
→ this is  
by system

- ① Fun
- ② Nor
- ③ Use
- ④ Sys

① Fun  
→ desire  
System  
→ It +  
develop

→ FS

FSR  
detail  
the sys  
State

→ minimise the project workload.

## Software Requirement :-

It is a process of finding out, analysis, documentation and constraints is called as requirement engineering.

→ This is description of the services provided by system and operational constraints

① Functional requirement

② Non functional requirement

③ User requirement

④ System requirement

① Functional requirement

→ describe the functionality of system, what system should do for project.

→ It totally depend on type of system being developed

→ FSR (functional ~~System~~ Specification Requirement)

FSR describe the system function in detail like input & output device the system should go complete and consistent State

## ② Non-functional requirement

- It does not describe the functionality of System
- These one constrain ~~and~~ services or the functional offer by the system
- It defines the quality constraint and also include the timing constraints on development process not directly related to system functionality.
- It is related to ~~re~~ reliability and response time and also storage capacity.

## ③ User

- Types of Non-functional requirement

① Product requirement

② Business requirement

③ External requirement

④

User Requirement

The user requirement describes the functional and non-functional requirement.

- It is understandable by the customer without technical knowledge.

such as Linux, the need of one arising in the course of their decisions, etc.

→ For writing the requirement we face some problems they are

- 1) lack of clarity & precision
- 2) Requirement confusion
- 3) requirement combination.

→ Guidelines for writing the user requirement :-

- 1) Invent standard format
- 2) use language consistency
- 3) use text highlights
- 4) Try to avoid special character/word.