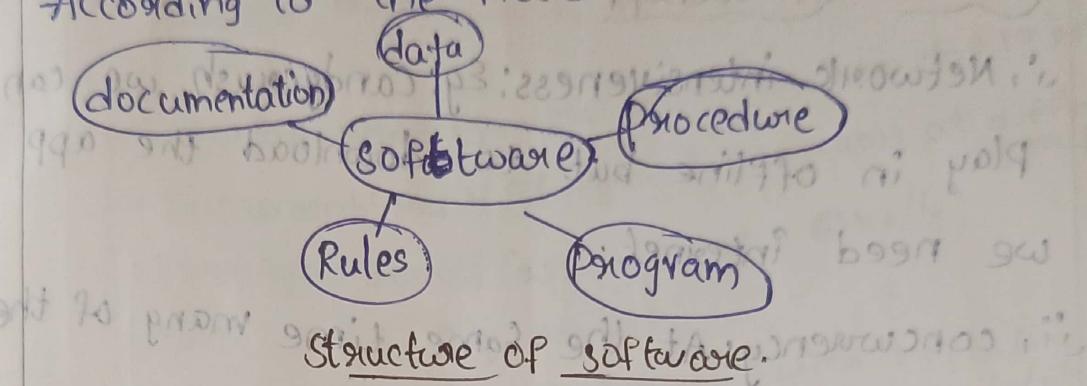


3/6/2023

## UNIT-01: Software and software Engineering

software Engineering: It is a collection of programs which is used to implement a particular application or website etc...

According to the need of user:



characteristics of software:

1. It is logical component      Intranet.
2. It has multiple users      (It is accessed by)
3. Internet (world wide application)      a particular organisation like  
↓  
used to access the software      wifi)
4. Cost is high
5. Software will not die.
6. Software need upgrade (update)

Software Classification:

- i. Generic software
- ii. customised software
- iii. system software

- iv, Application Software
- v, Embedded software
- vi, Product line software
- vii, AI software
- viii, Web software
- ix, Engineering & Scientific software.

### Web apps:

#### Features of webapps:

i, Network intensiveness: Eg: Candy crush we can play in offline but to download the app

we need internet.

ii, concurrency: At the same time many of the users can access.

iii, unpredictable role: The Google will give accurate answers for the respective question asked by the user.

iv, Immediacy

v, Security ~~AES~~ (Advanced encryption Standard)

To encrypt or decrypt the data.

vi, data driven: Eg: shareit, shareme

data transfer, data modification takes place.

## Software Myths:

Management myth

customer myth

Developer myth

Practitioner myth

## Software Process Models:

collection of set of activities

process model

- waterfall Model (process will go from top to bottom)
- Incremental Model (different programs at different groups)
- Prototype Model (assing work actually)
- RAD Model (Big projects)
- Spiral Model (same project with repeating models)

Steps in SPM: 1) Activities

2) Action

3) Task

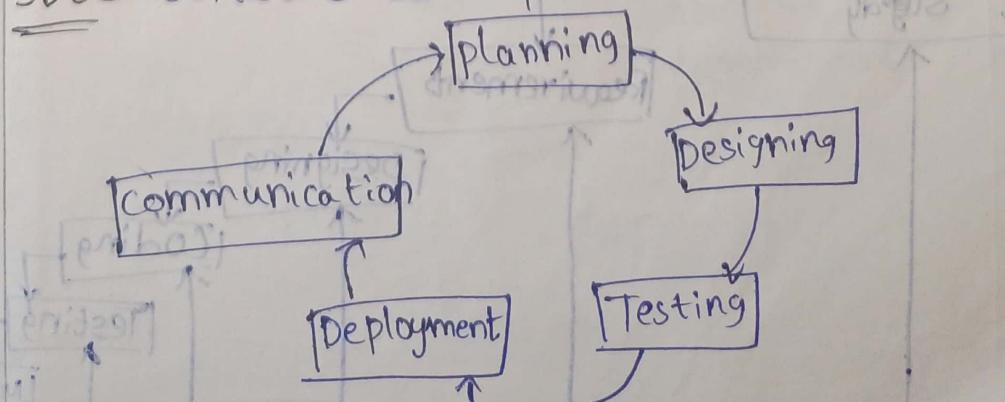
Define → Task

→ Input & output

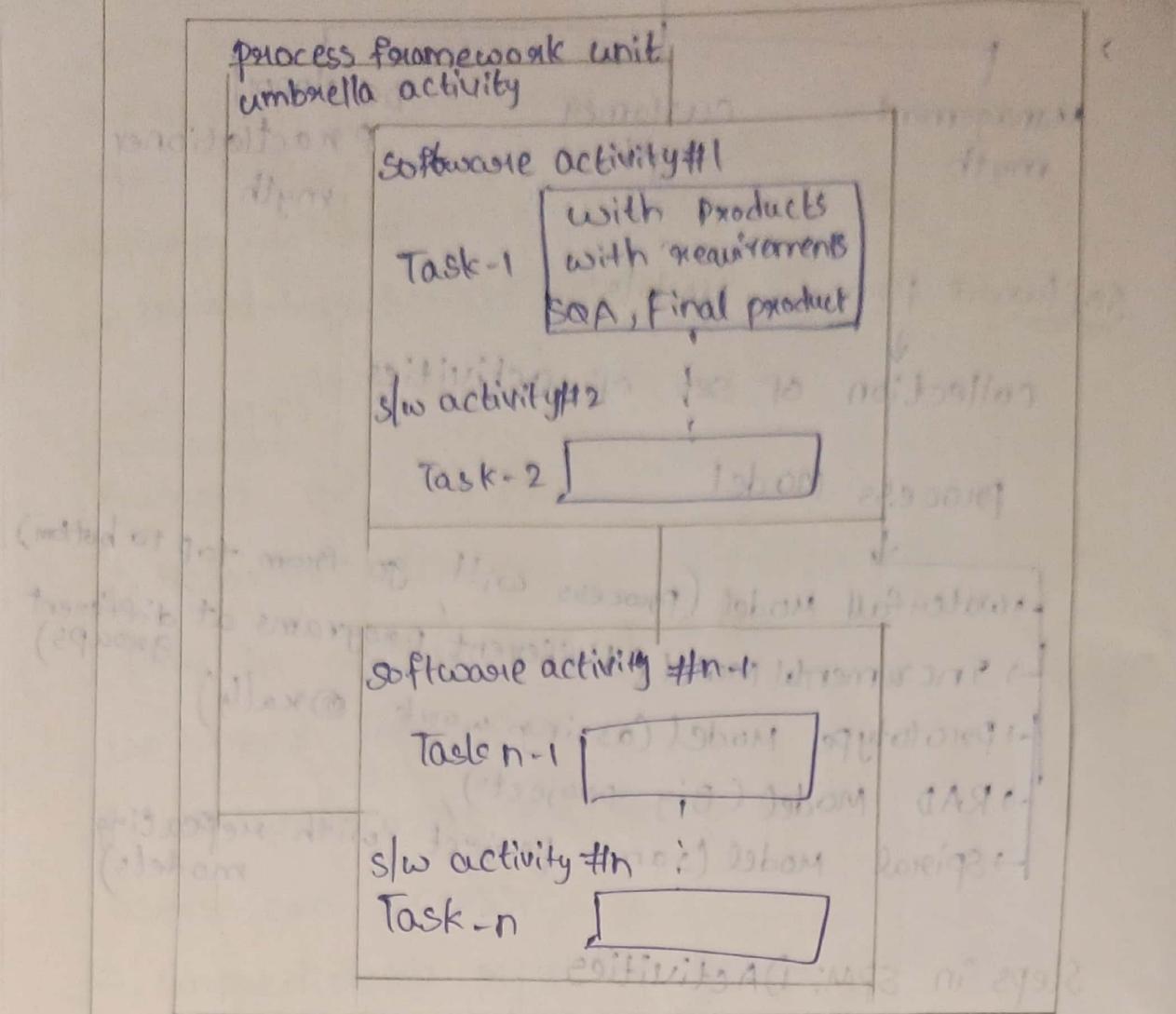
→ pre & post process

→ flow & sequence

## SDLC: Software development life cycle.



# Generic process Model: Framework layout for any process model



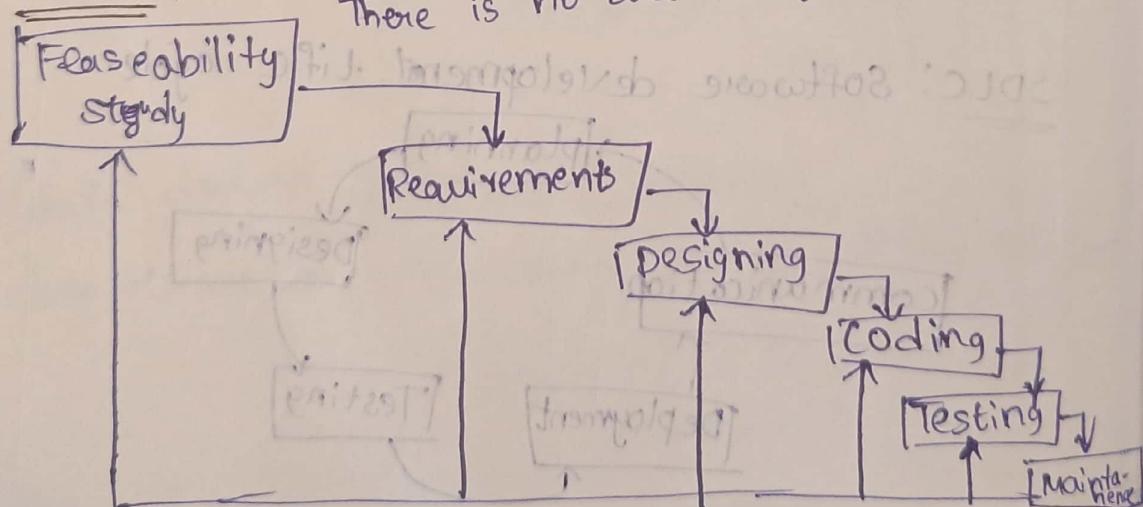
umbrella activity:

1. Requirements

2. Deciding & Modelling.

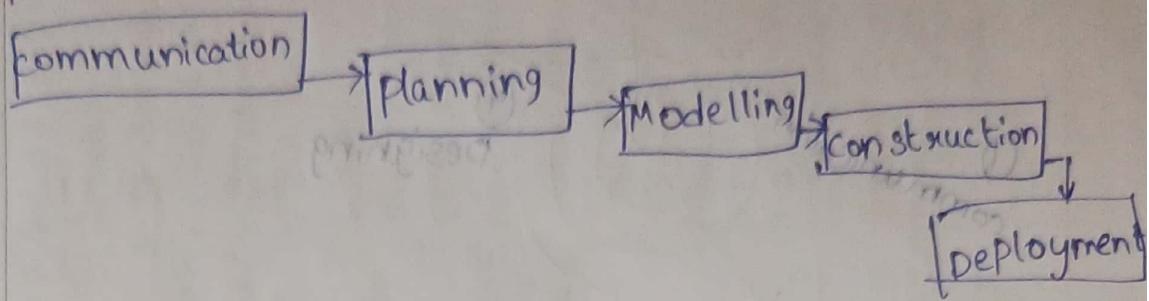
3. SQA (Software Quality Assessment)

waterfall model: (linear sequential model)  
There is no back propagation

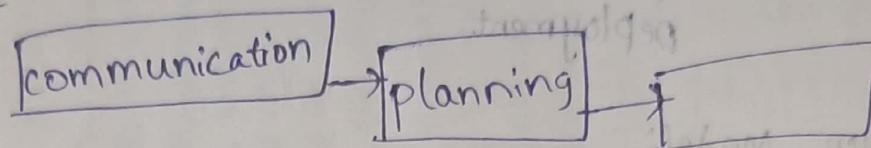


## Incremental Model:

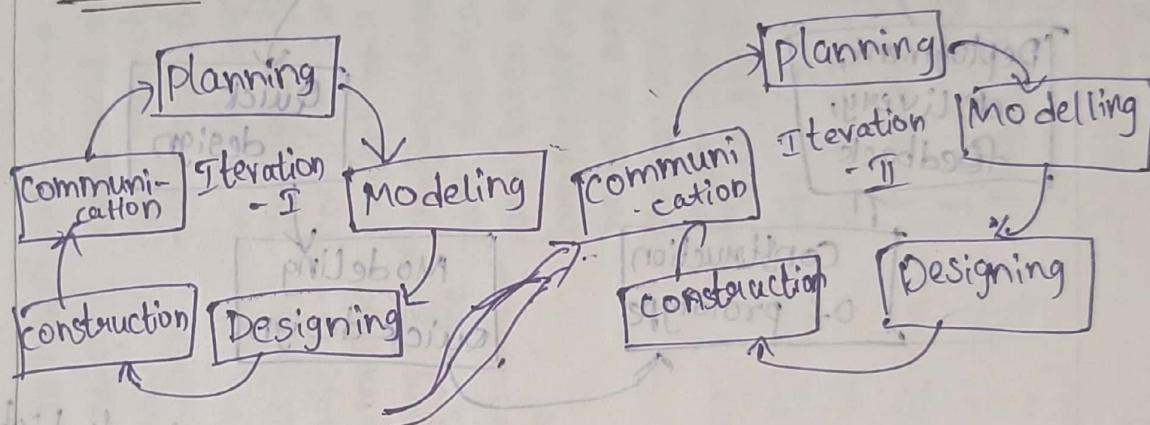
#1



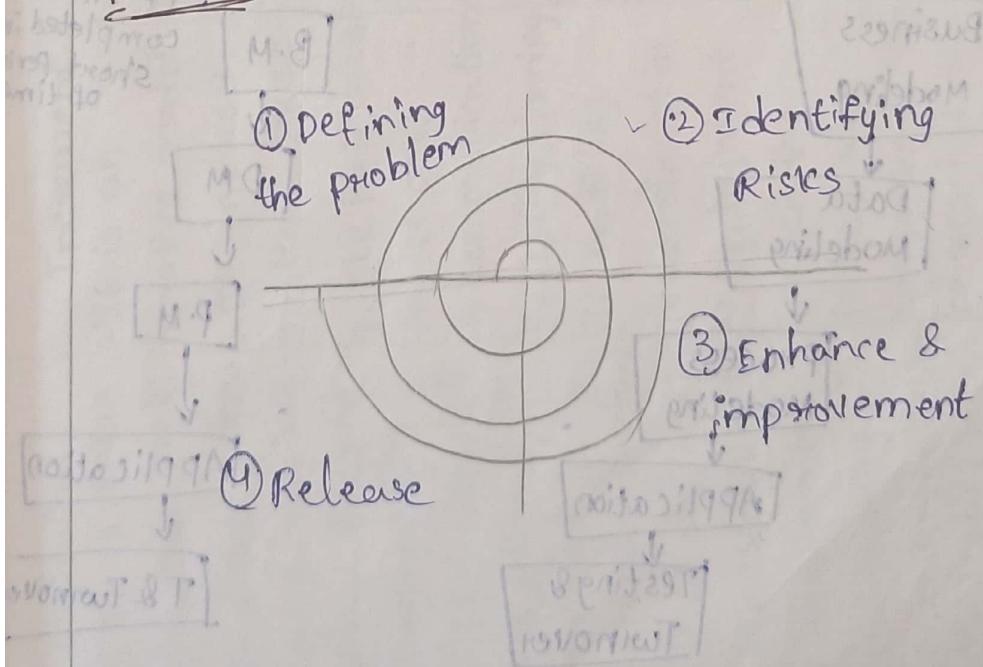
#2

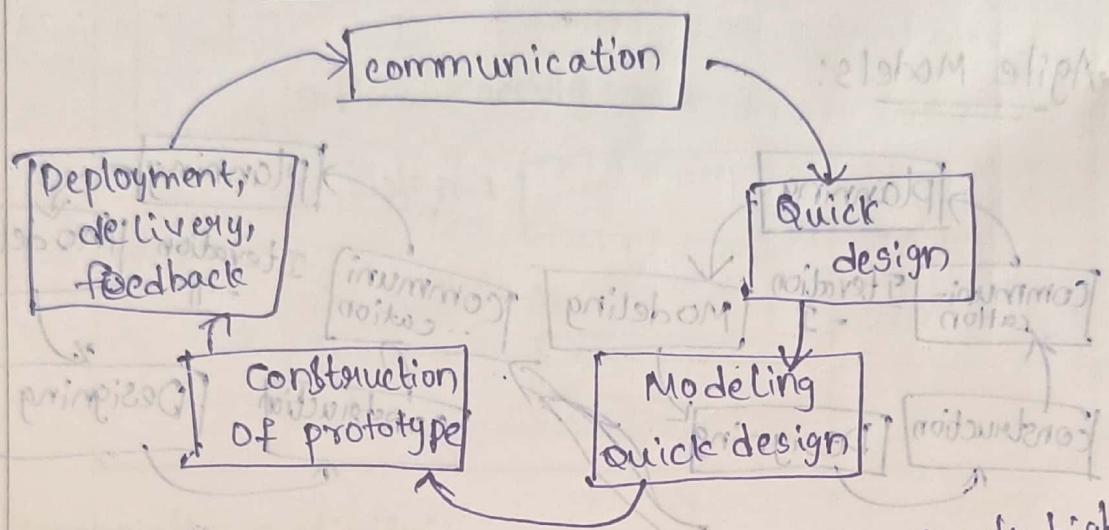
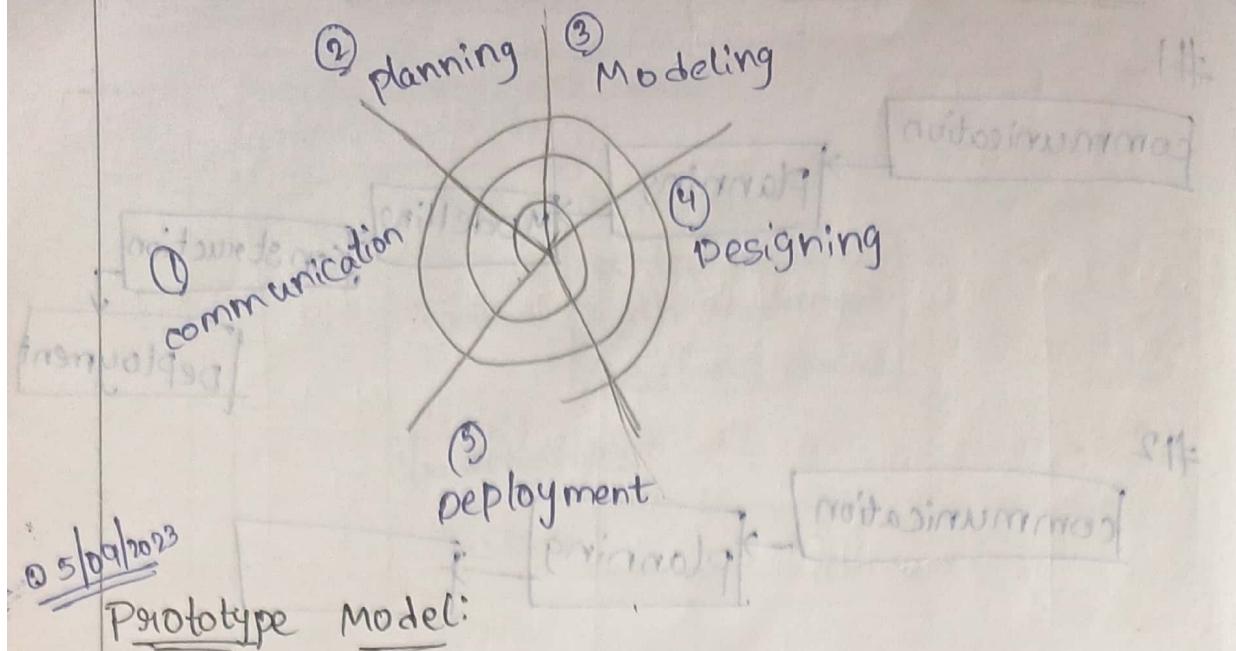


## Agile Models:

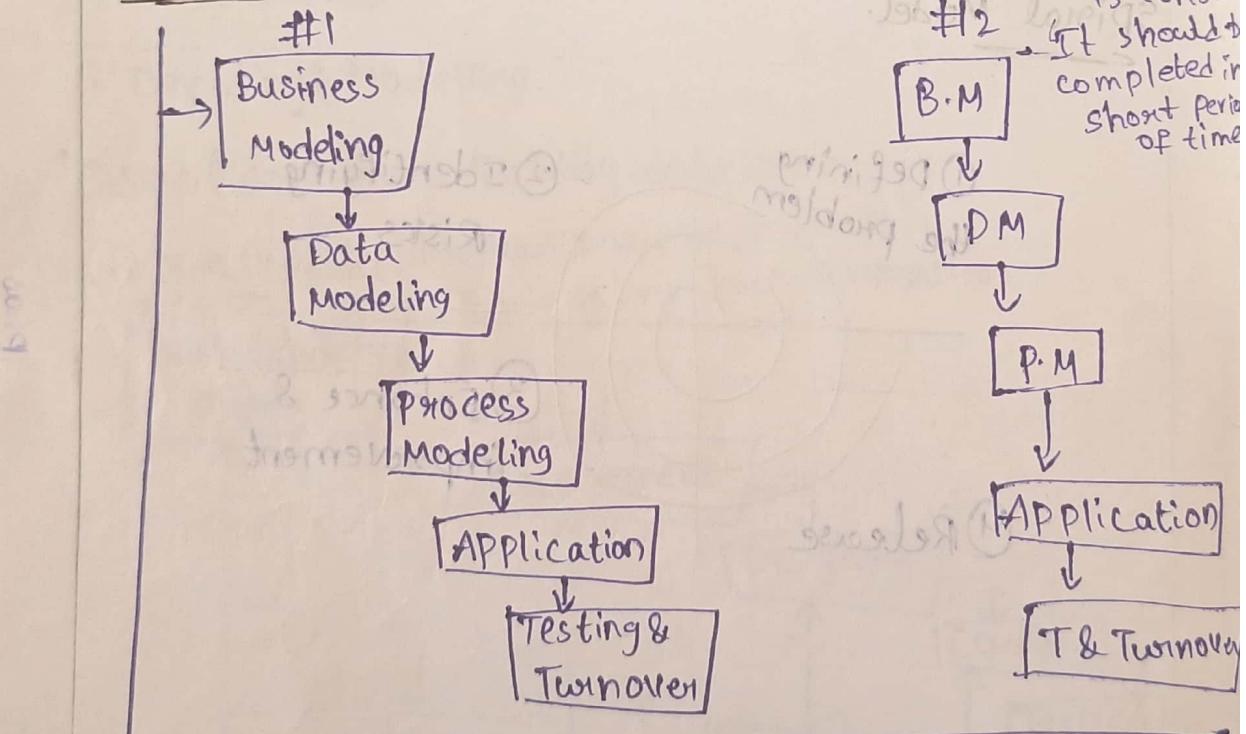


## Spiral Model:





RAD model: (Rapid application development) only high level programming is considered



It comprises more money & more manpower.

Waterfall	Increment	Agile	Spiral.	Prototype	R&D
<p>1. It is linear sequential model.</p> <p>2. complete process will go through only forward propagation</p> <p>3. It is simple &amp; traditional method.</p> <p>4. phases involved in it are</p> <ul style="list-style-type: none"> <li>i, Feasability Study</li> <li>ii, requirement</li> <li>iii, Designing</li> <li>iv, coding / modeling</li> <li>v, Testing</li> <li>vi, Maintenance.</li> </ul> <p>5. It is used for short projects.</p> <p>6. EX: NASA</p>	<p>1. It is an iterative model.</p> <p>2. It includes multiple phases.</p> <p>3. steps involved in it are:</p> <ul style="list-style-type: none"> <li>i, communication</li> <li>ii, Testing / planning,</li> <li>iii, Modeling</li> <li>iv, construction</li> <li>v, Deployment</li> </ul> <p>4. once, the project was completed it will be reached to the customers for deployment phase.</p> <p>5. for deployment phase, once the modifications were observed then again the model was passed through</p>	<p>1. It is also known as more quickly.</p> <p>2. It is used for large projects.</p> <p>3. Large projects are no subdivisions of the project subdivided into small projects.</p> <p>4. In this method, modifying the errors and obtaining the final product was accurate.</p> <p>5</p>	<p>1. It is used for large and medium size projects.</p> <p>2. In this there are no subdivisions of the project.</p> <p>3) the complete large project will be undergo through all the stages and later on if any modifications required again the process will continue from starting to final stage unless the project was completely finished.</p> <p>4) High Risk Projects uses spiral model.</p>	<p>1. working models core created in this method.</p> <p>2. In this designing the hardware model was the major task.</p> <p>3. It is used for big projects</p> <p>4. hardware model will be given as a sample to the customers.</p> <p>5</p>	<p>1. Rapid application development.</p> <p>2. It is used for medium &amp; large applications.</p> <p>3. The phases in it are:</p> <ul style="list-style-type: none"> <li>Business modeling</li> <li>Data modeling</li> <li>Process modeling</li> <li>Application Testing &amp; Turnover.</li> </ul> <p>4. In this, the large project was not subdivided into small projects.</p> <p>5. If depending upon the reviews (yes) modifications, the complete phase will repeat again.</p> <p>next phase until the output was obtained</p>



Perceptive models: waterfall model, increment model

11/09/2023

and RAD model:

Waterfall Model:

Example: NASA

Application: for designing & modeling space crafts.

Increment Model:

Example: Microsoft

Application: updates about windows.

Agile Model:

Example: Spotify

Application: aggregation of songs based upon various factors such as recently released, regional bases (or) language basis

Spiral Model:

Example: Boeing

Application: for modeling and designing aircrafts.

Prototype Model:

Example: Google

Application: for Google Glass.

RAD model:

Example: IBM (International Business Machine)

Application: software products released by IBM.



Scanned with OKEN Scanner

## Process Assessment and improvement:

The process models in software requires continuous monitoring and improvement.

There are some standard techniques. They are:

1. SCAMPI (standard CMMI Assessment method for Process Improvement)
  2. CBAPI
  3. SEPIC
  4. ISO 9001:2000
1. SCAMPI - Standard CMMI Assessment method for Process Improvement.  
CMMI - Capability Maturity Model for Interface.
2. CBAPI - CMMI Based Appraisal ~~integation~~ for Process Improvement.
3. SEPIC - Standard evaluation for process improvement and development.
4. ISO 9001:2000  
↳ Indian Standard organization.

### Receptive

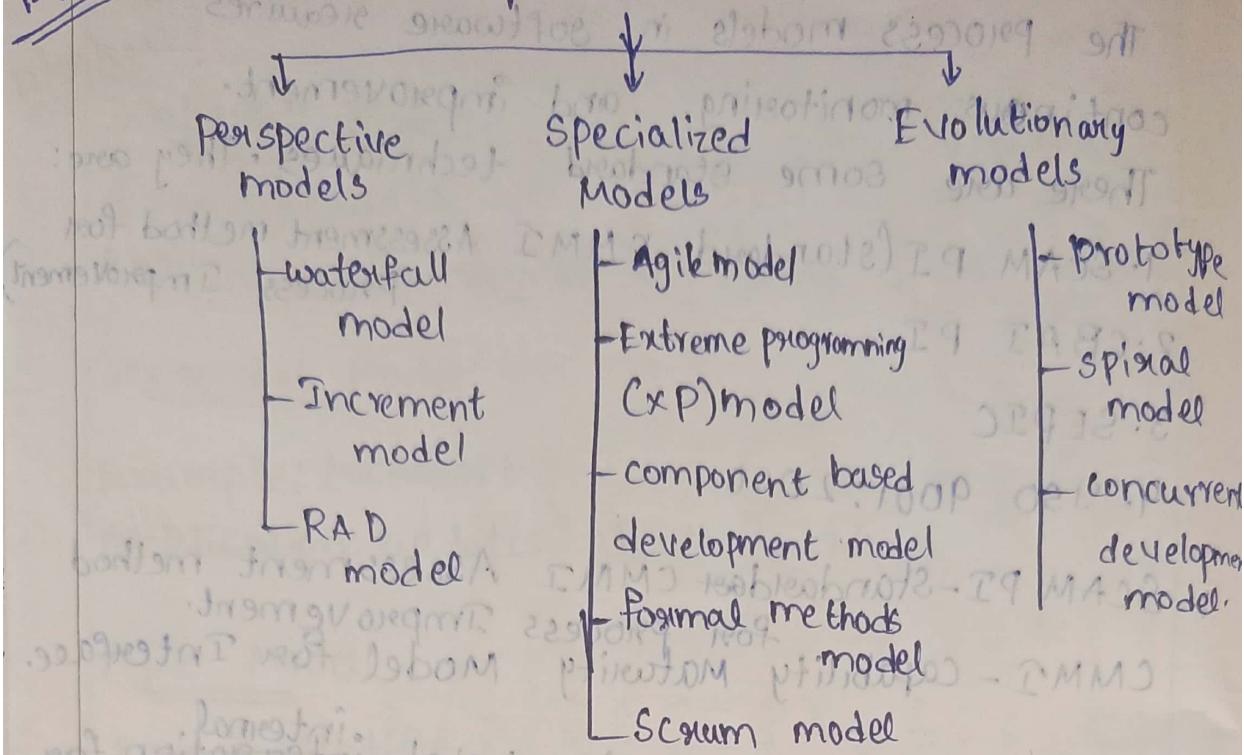
Perspective - Waterfall, Increment, RAD

Specialized -

Evolutionary -

12/09/2023

## process Models

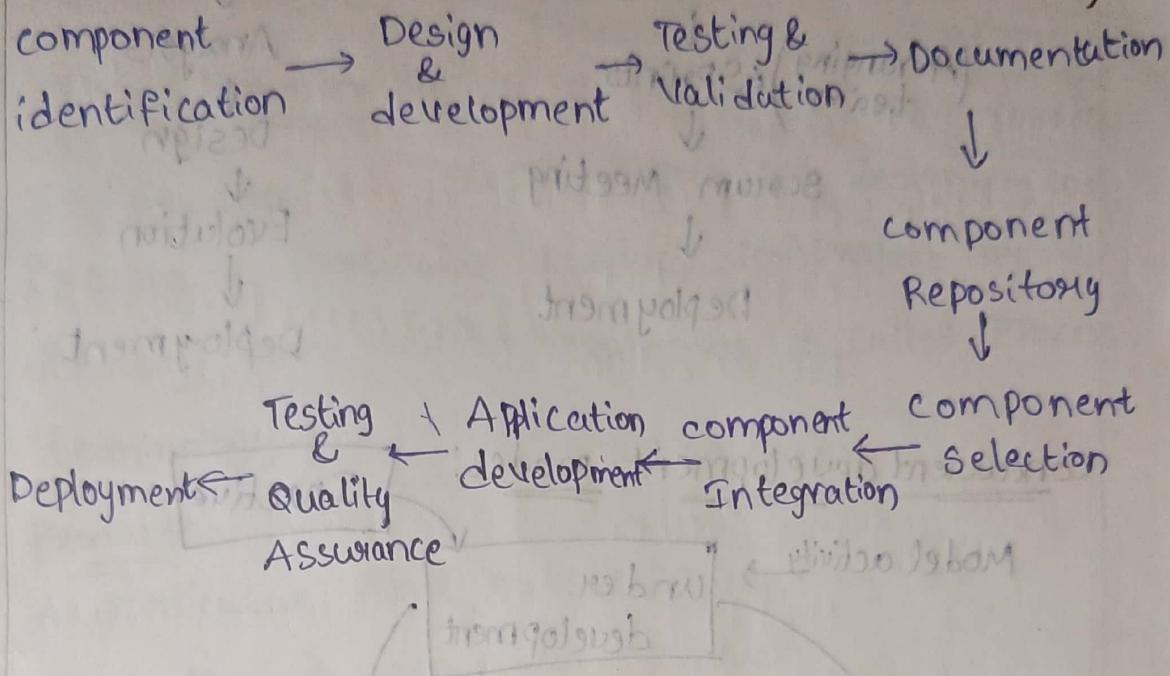


Perspective models: The software models which require a sequential set of operations will fall under this category.

Specialized models: The software models which require an extended activity for already existed ones will fall under this category.

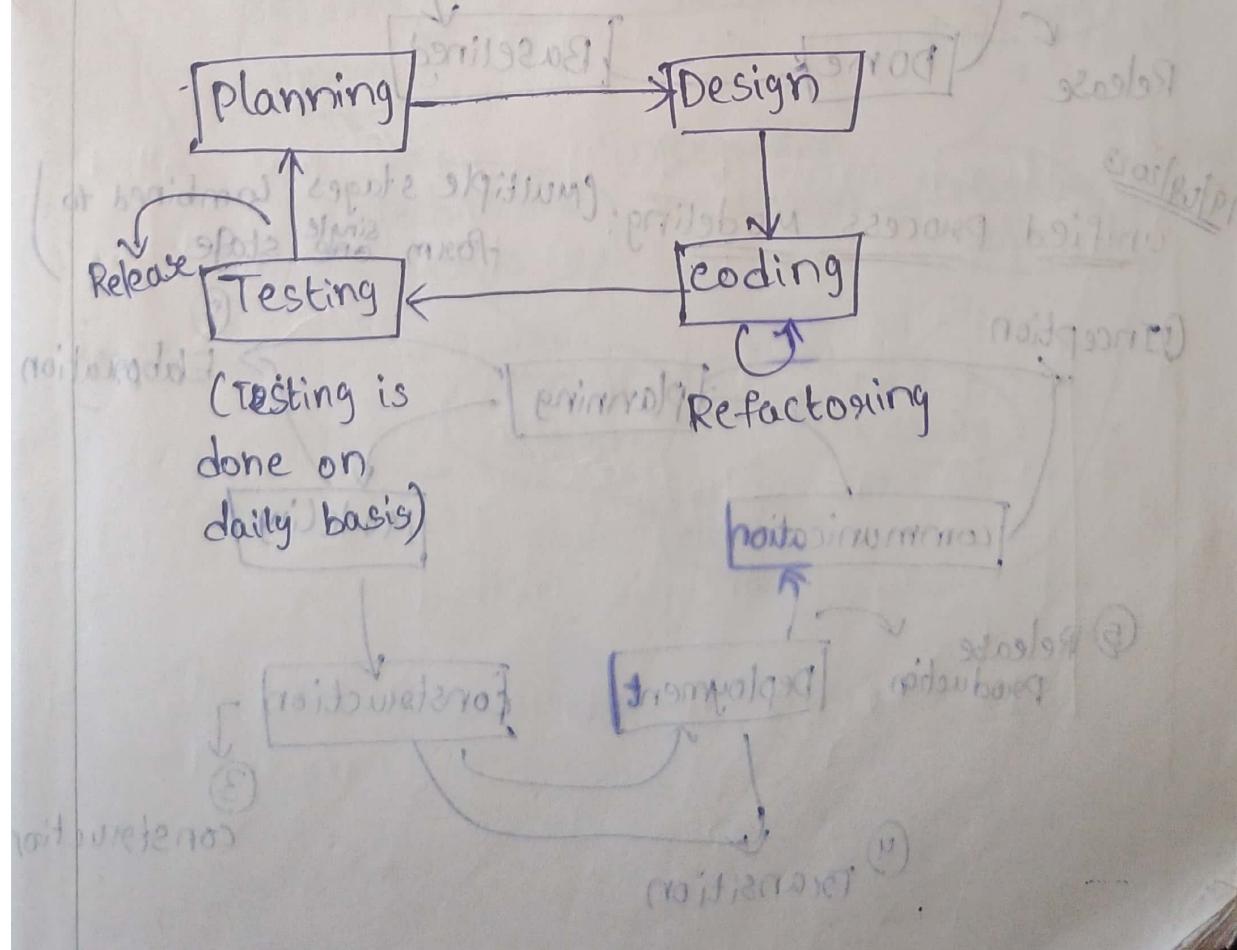
Evolutionary models: The models which are combination of both iterative and incremental methods will fall under this category.

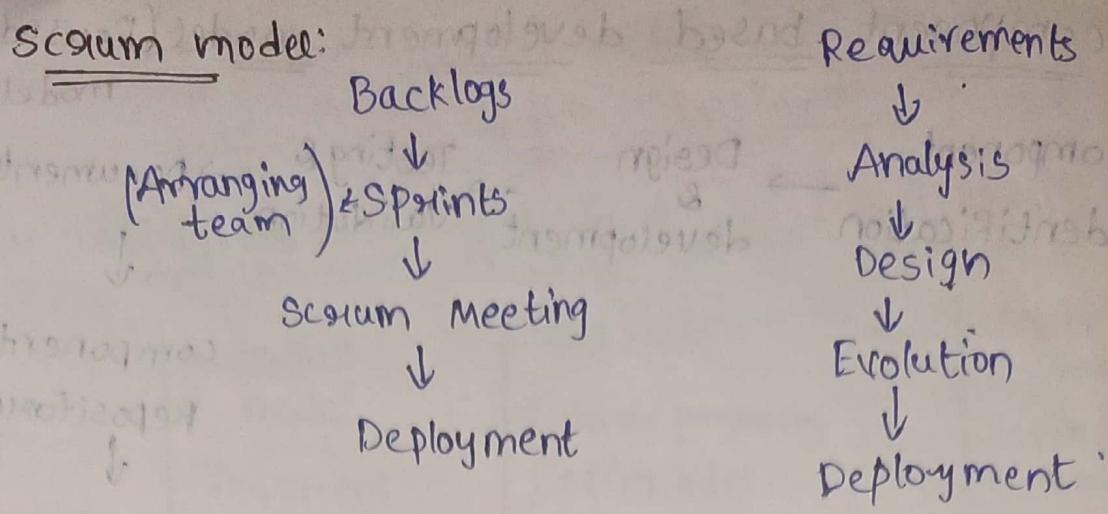
## component based development model (Reused model):



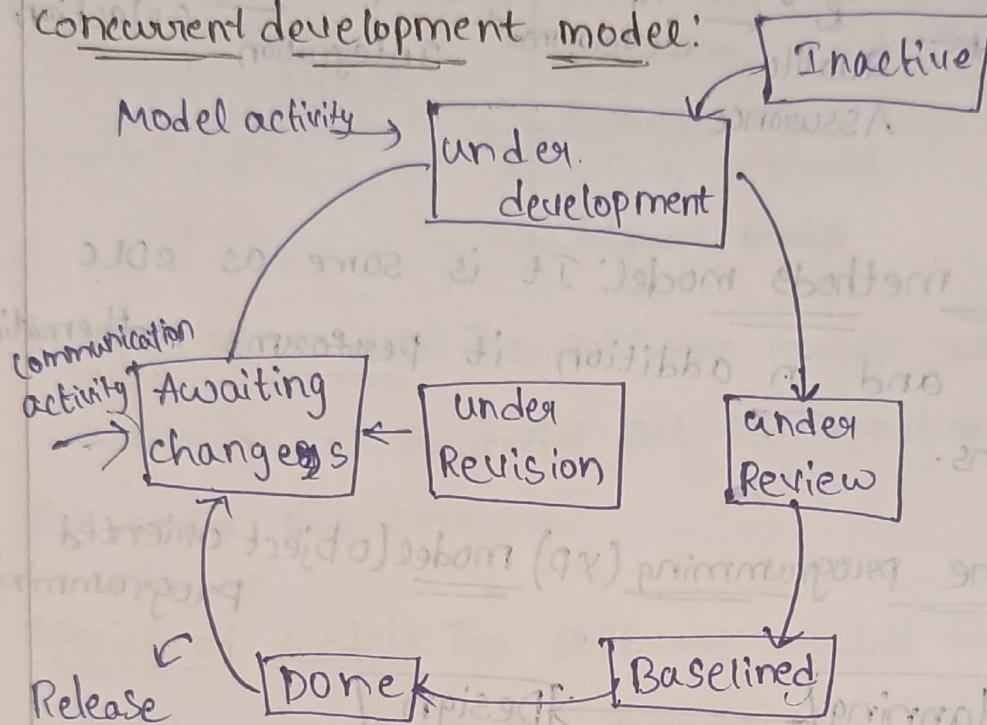
Formal methods model: It is same as SDLC and in addition it performs mathematical operations.

## Extreme programming (XP) model (object oriented programming)

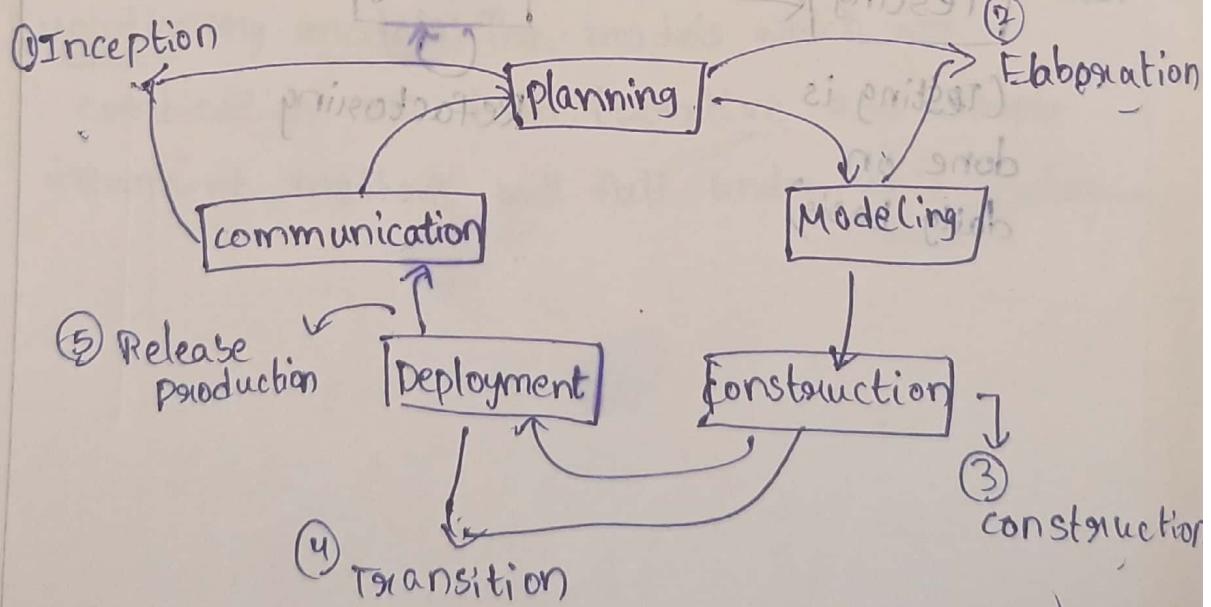




concurrent development model:



unified process modeling: (multiple stages combined to form one stage)



Personal software process and Team software process:

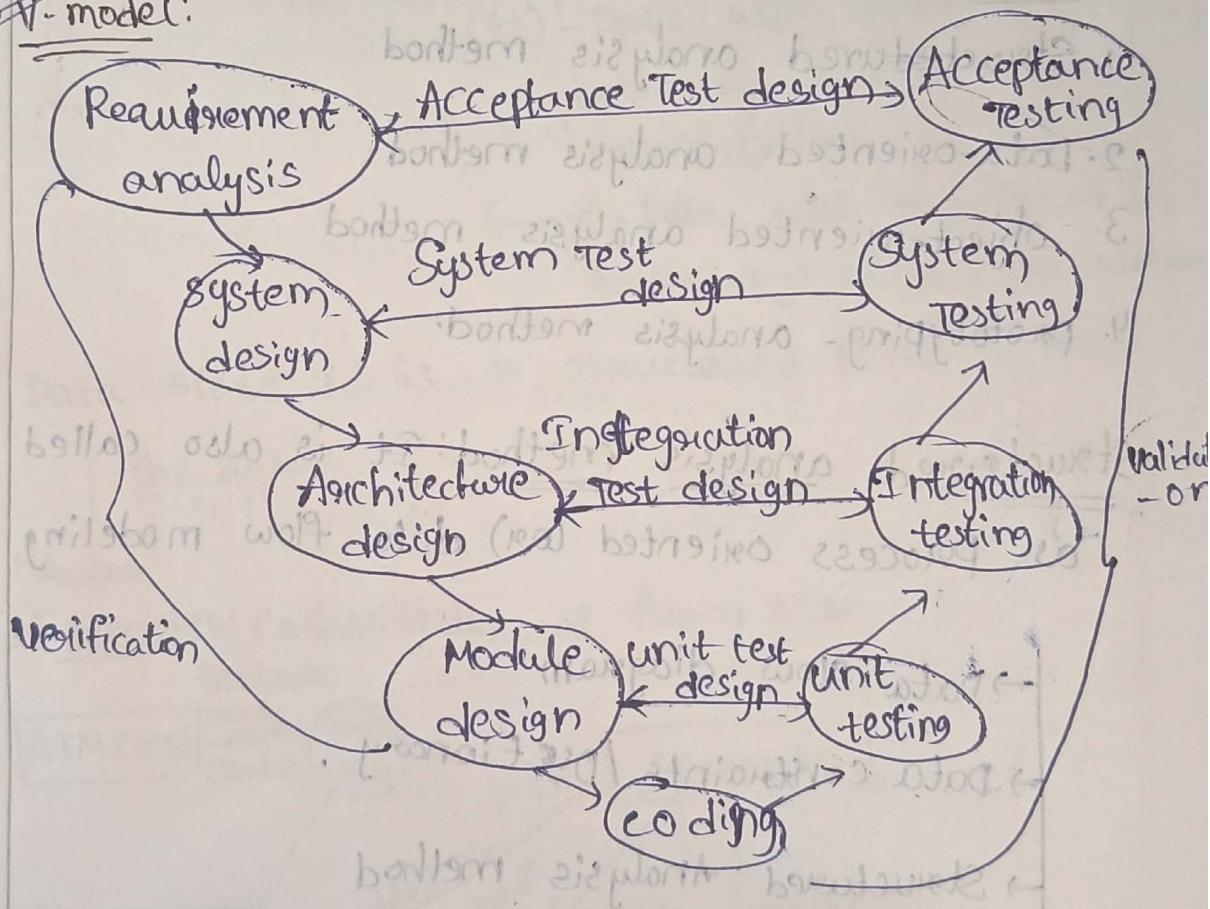
Single person can do the process ↳ Group of members are supposed to do the process.

Process terminology: The sequence of steps follow in designing (or) creating a particular product (or) project by following some rules & regulations is called Process terminology.

Products:

A particular final output given by the organization is called as product.

V-model:



## Unit-II : Requirement Analysis & Specifications

### Software Design

Requirement analysis: It is the set of activities

like classification

organization

Prioritization

constraints

Negotiation

Modeling

Types of requirement analysis:

1. Structured analysis method
2. Data-oriented analysis method
3. Object-oriented analysis method
4. prototyping - analysis method.

Structured analysis method: It is also called  
as process oriented (or) data flow modeling

- Data flow diagram
- Data constraints / Dictionary .
- Structured Analysis method
- Pro's & Con's of Structured Analysis method.

The transformation of data from one end to other end is shown in structured analysis method.

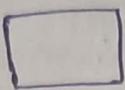
Data flow diagram:

① Process



② Data flow →, ←, -→ some change will occur during coming back

③ Data store



④ Actor



Data flow: The data was being transferred from

input to output.

It is shown by: → - Data was only transferred in one way

↔ - Data was transferred in both ways without any manipulation.

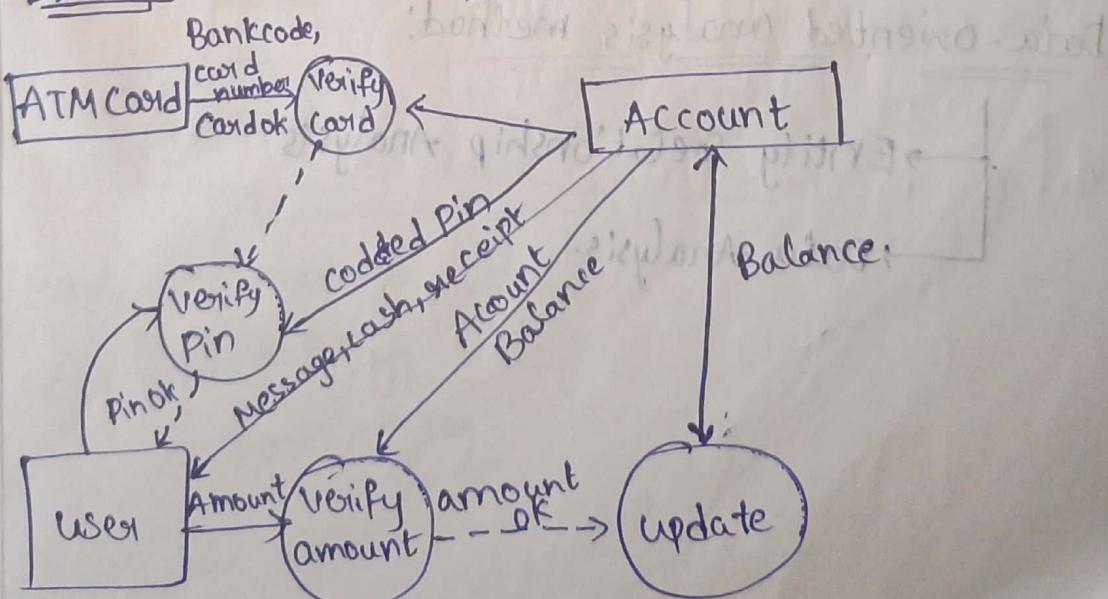
-→ - Data was transferred in both ways after manipulation

Data store: It is a structured place to keep

the data.

Actor:

Examples: Cash withdrawal from ATM



- Data Dictionary: A book to maintain terminology.
- ① + → composition
  - ② - → Repetition
  - ③ | → selection
  - ④ [ ] → Repeated Structure.

### Structured analysis method:

Prepare the context diagram



Construct level-1 DFD



Decomposition of Level-1 DFD

Identifying man-machine Boundary

Prepare data dictionary, data description

↓ Requirements specifications.

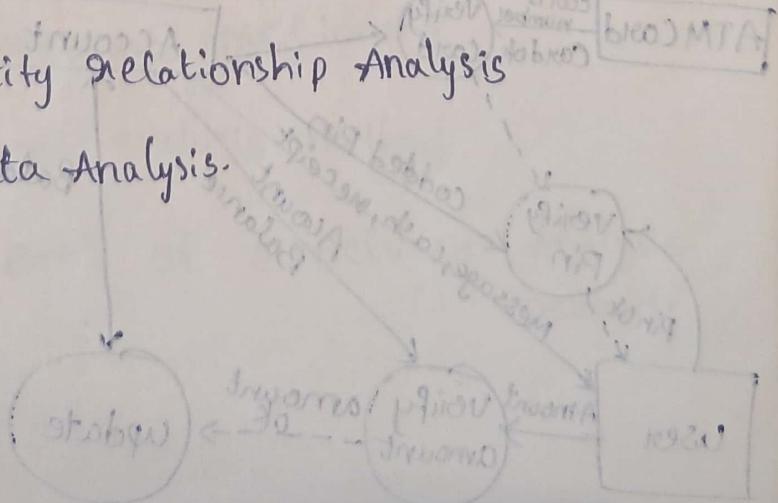
23/09/2023

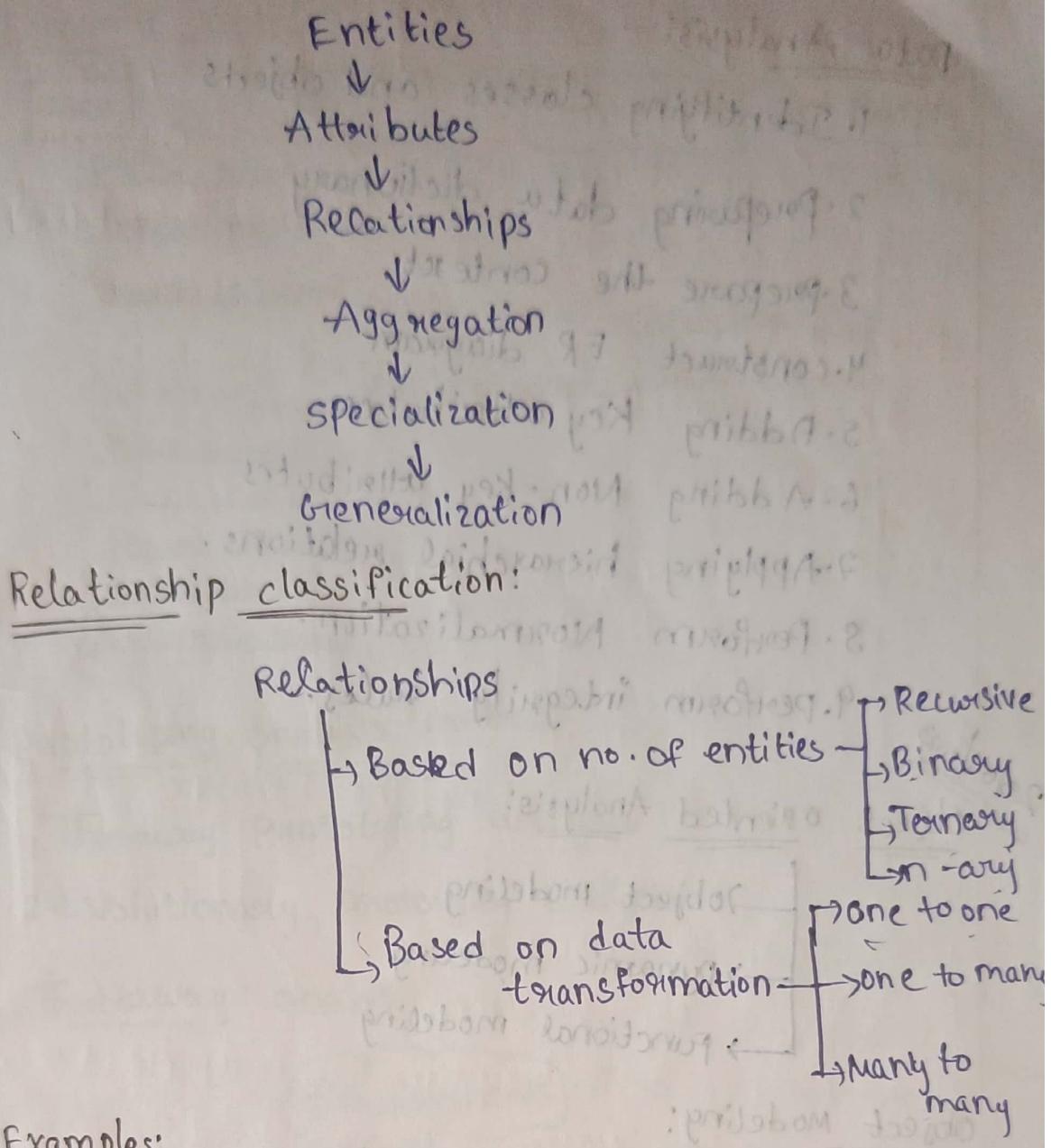
Entity: It is a group of objects (Persons)

Things.

Data-oriented analysis method:

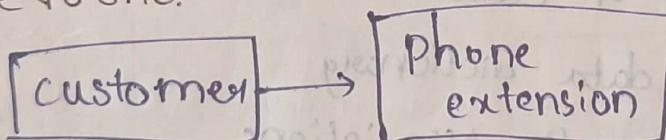
- Entity Relationship Analysis
- Data Analysis.



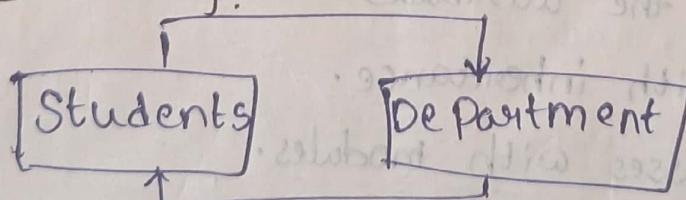


Examples:

One to one:



One to many:



Many to many:

posts & tags

Posts → Tags

## Data Analysis:

1. Identifying classes and objects
2. Preparing data dictionary
3. Prepare the context
4. construct ER diagram
5. Adding Key attributes
6. Adding Non-key attributes
7. Applying hierarchical relations
8. Perform Normalization
9. Perform integrity rules.

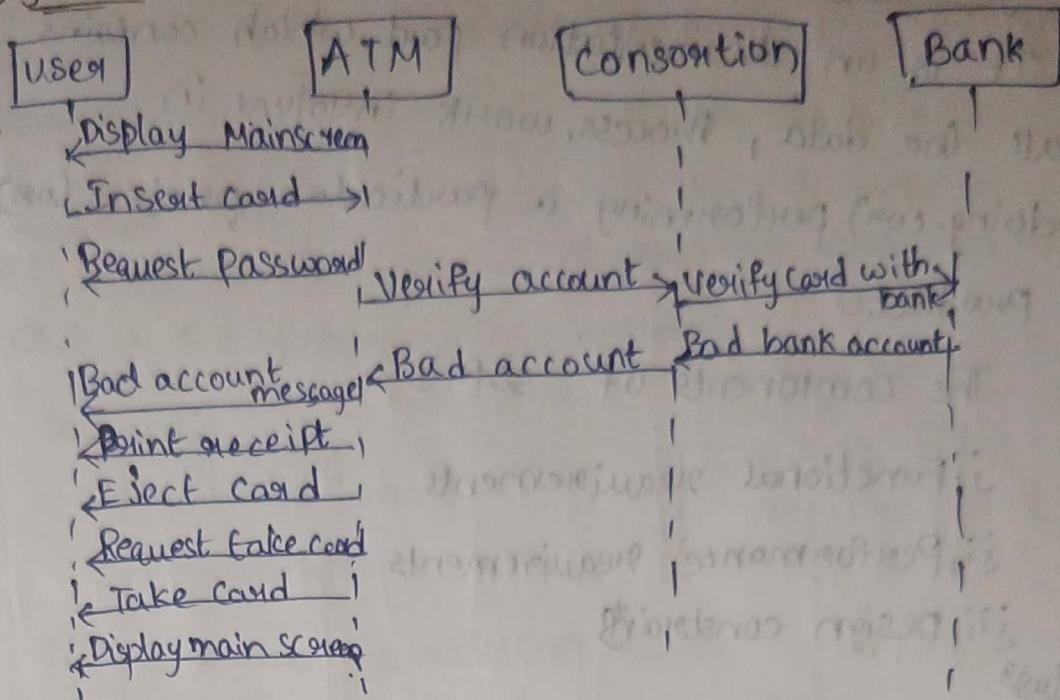
## Object oriented Analysis:

- Object modeling
- Dynamic modeling
- Functional modeling

## Object Modeling:

1. Identification of classes & objects.
2. Preparing the data dictionary
3. Identification of Associations
4. Identifying the attributes
5. Refining with inheritance.
6. Replacing classes with modules.

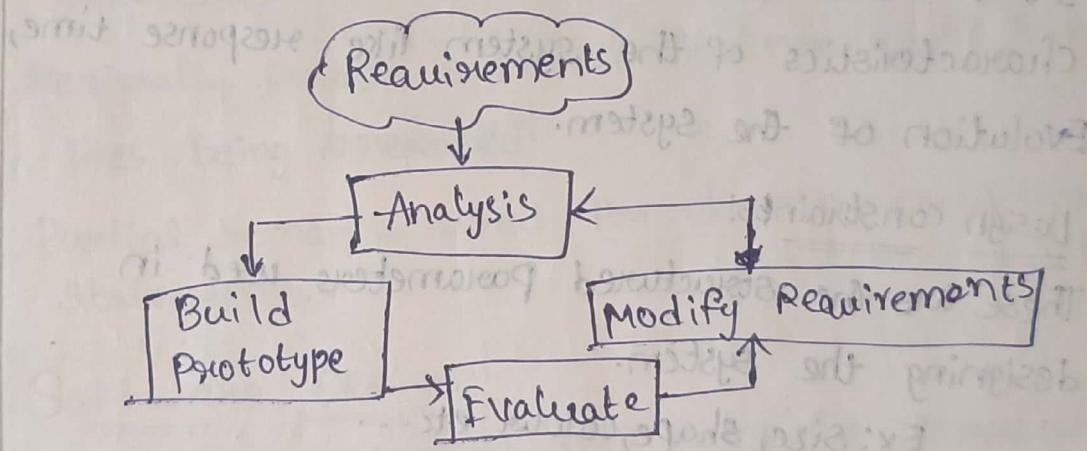
## Dynamic modeling:



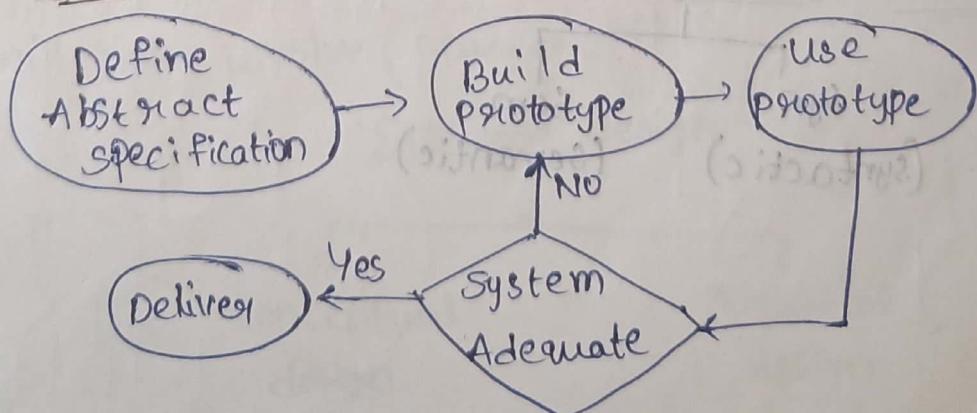
## Prototyping - analysis method:

1. Throwaway prototyping analysis
2. Evolutionary prototyping analysis

### Throwaway analysis:



### Evolutionary analysis:



## System specification requirements

It is an documentation part which contains all the data, process, work involved in doing (or) performing a particular project (or) product.

The components of SSR is :

- i, Functional requirements
- ii, Performance requirements
- iii, Design constraints

26/09/2023

### Functional requirements:

It is used for describing the behavior of the system.

### Performance requirements:

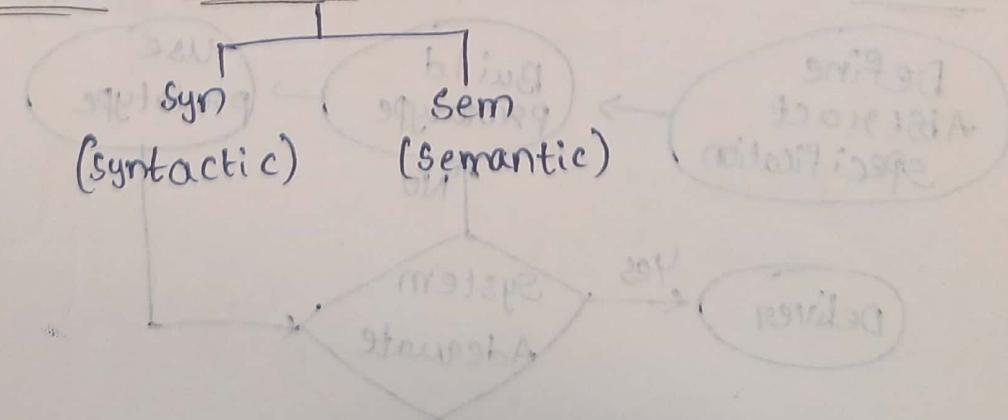
It is used for describing the performance characteristics of the system like response time, evolution of the system.

### Design constraints:

These are the structured parameters used in designing the system.

Ex: size, shape, colour etc...

## Formal specification language:



## Syntactic:

It is an basic alphabets used in programming language.

## Semantic:

It involves the special characters and symbols used in the programming language.

### Semantic

- Linear Semantic
- Branching semantic
- Maximally parallel semantic
- Partial Semantic

Linear semantic: when the program contains sequence of events.

Branching semantic: used for graph oriented structures

Maximally parallel semantic: the number of same lines being repeated.

Partial semantic: used for checking conditional statements.

## Software Design:

It is process where the complete structure of project (or) product will be designed which will not involve analysis of it.

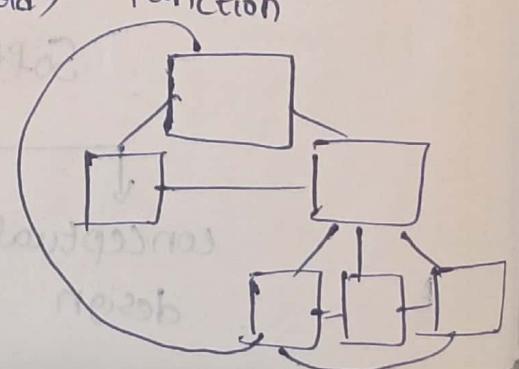
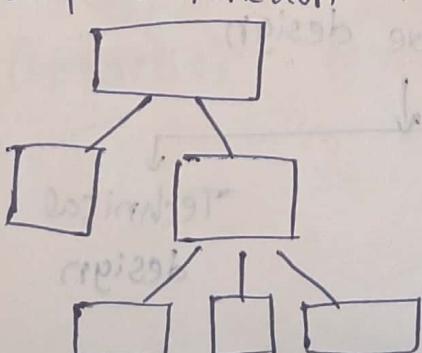
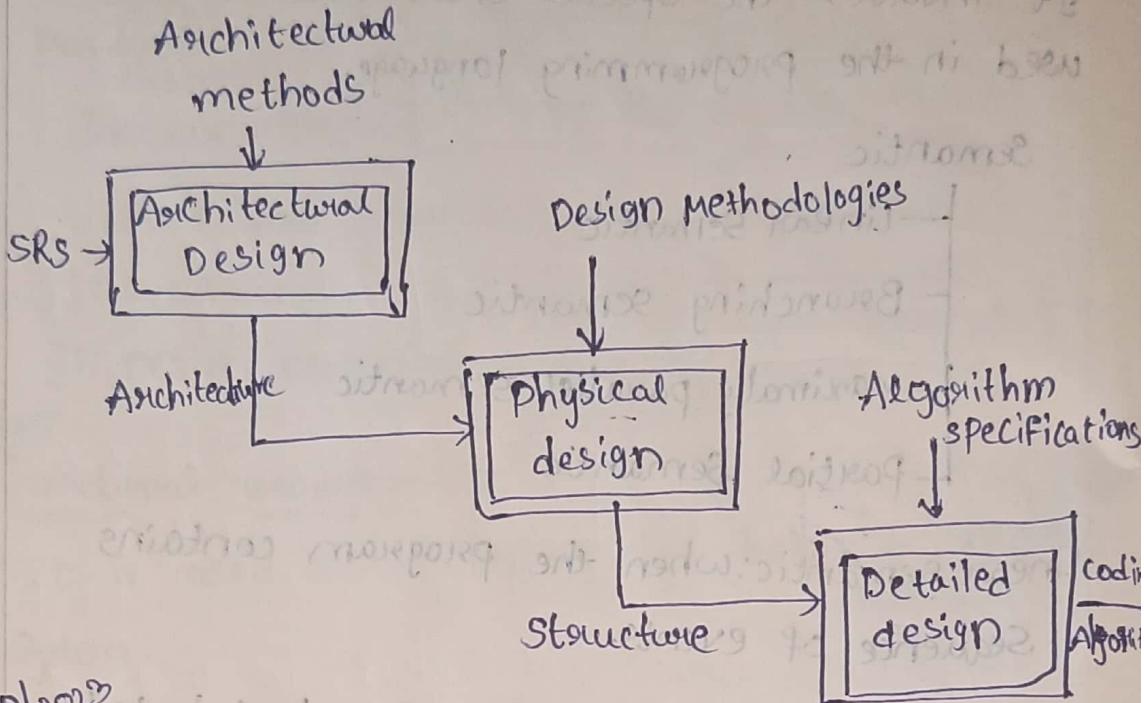
### Software design

conceptual  
design

Technical  
design

The three stages involved

- i, Architectural design
- ii, physical design
- iii, detailed design



cohesion: It is a process of reducing no. of functions in a particular module.

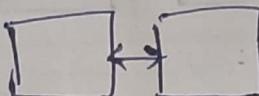
It is performed only on a single module.

It is also known as Intermodule.



coupling: It is a process of interconnection between two modules.

It is also called as Intermodule



Types of cohesion:

1. Functional cohesion

2. Sequential cohesion

3. Communication cohesion

4. Temporal cohesion

5. Procedural cohesion

6. Logical cohesion

7. Coincidental cohesion

05/01/2023

Types of coupling:

1. Message coupling

2. Stamp coupling

3. Data coupling

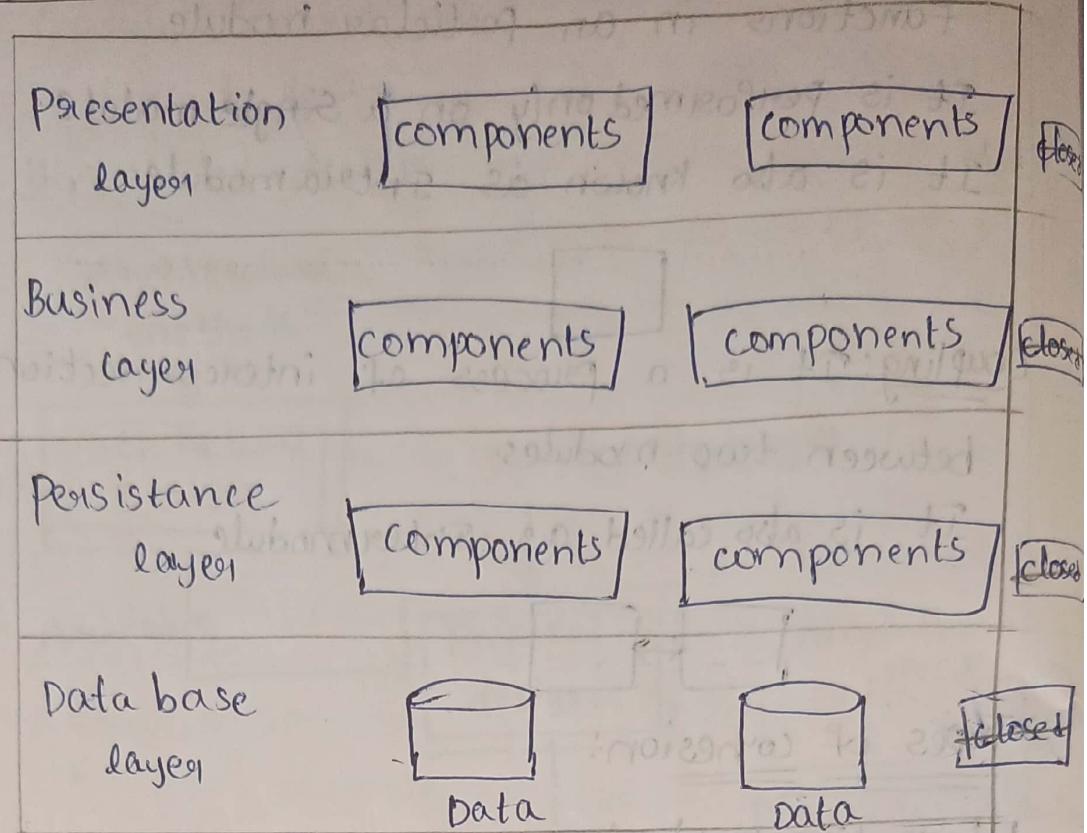
4. Control coupling

5. External coupling

6. common coupling

7. content coupling

## Layered architecture



→ It is an n-tier layer Model.

**Presentation layer:**

→ It is used for only exhibiting (or) displaying the final output but it will not have any access to the data whereas business layer is used for aggregation of data.

→ Data base layer is used for storing the data whereas persistence layer is used for accessing the data.

**Isolation pattern:** It is used for creating open modules for reducing the no. of interconnections.

Presentation layer	Components	Components	closed
Business layer	Components	Components	closed
Service layer	Components	Components	open
Persistence layer	Components	Components	closed
Database layer	Data	Data	closed

17/10/2022

Requirement Gathering: It is the 1<sup>st</sup> phase where the organization is received an order from the customer will go through the primary data collection phase where it involves 1. Interview, 2. Questionnaires 3. Study existing mode 4. Analyse the problem

Requirement Analysis: The relationship between the main attributes and entities will be analysed clearly in this phase.



10/10/2023

## UNIT-03:

Structured analysis: It is used by the analyst to know the relationship between the attributes in an logical way through graphical representation of the data.

Orthogonal views in Structured analysis:

1. Functional view
2. Data view
3. Dynamic view

Functional View: The D.F.D which are used to represent the function between each attributes.

Data view: The entity relationship models will fall under this category.

Dynamic view: This state transitions involved in the model will fall under dynamic view.

Tools for Structured Analysis:

1. DFD
2. Data dictionary
3. Decision Trees
4. Decision Tables
5. Structured English
6. Pseudo code

## Characteristics of Structured Analysis:

1. It is used for graphical representation of the relationship between each modules.
2. It is used for logical analysis.
3. It is used for finding the relation between each attribute in an high to low correlation.
4. It is used by the analyst for better understanding of the entity relationship.

## Data flow diagram:

1. Entity -
2. Process -
3. Data flow -
4. Data Store -

what data flow can do and cannot do:

can	cannot
1. Entity to process	1. Entity to entity
2. process to entity	2. Entity to store
3. process to process	3. store to entity
4. process to store	4. store to store.

## Structured design:

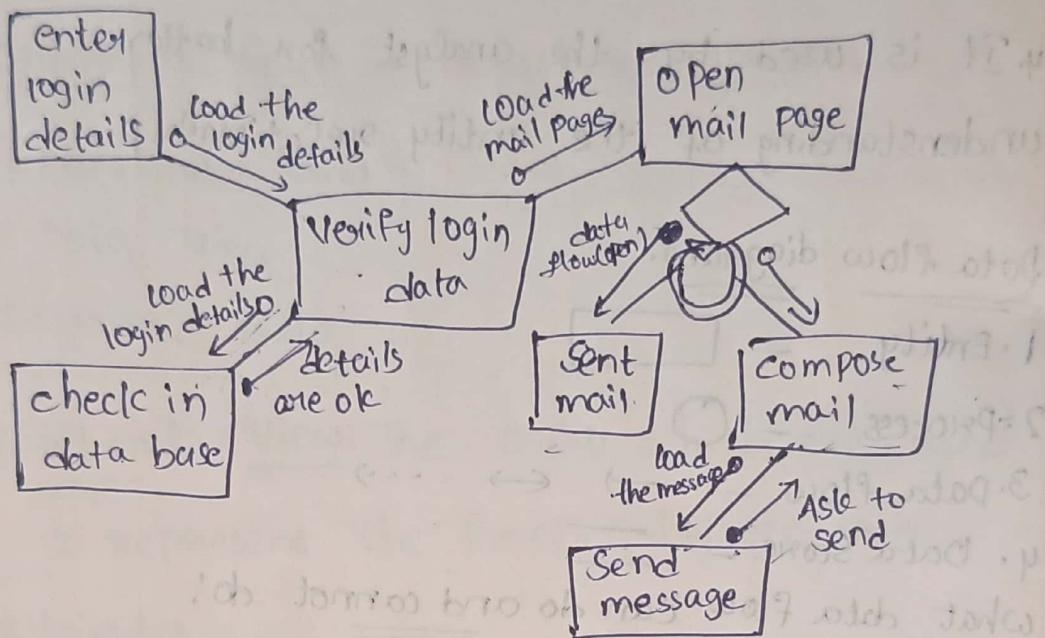
The process of converting the complete program or data into structured chart

## Structured chart:

1. Module -
2. conditional call. -

3. Loop/Repetitive call - 
4. Data flow - 
5. Control flow - 
6. Physical store - 

### Example:



12/10/2023

Detailed design: A complete documentation part prepared for the process which has been done until now and which is used for further applications in future.

### Design Review:

The three stages of review are

- i. Preliminary review
- ii. Critical review
- iii. Program review

Preliminary review: This is the first stage or review where the data, software tools and customer satisfaction were reviewed here.

The members involved in it are

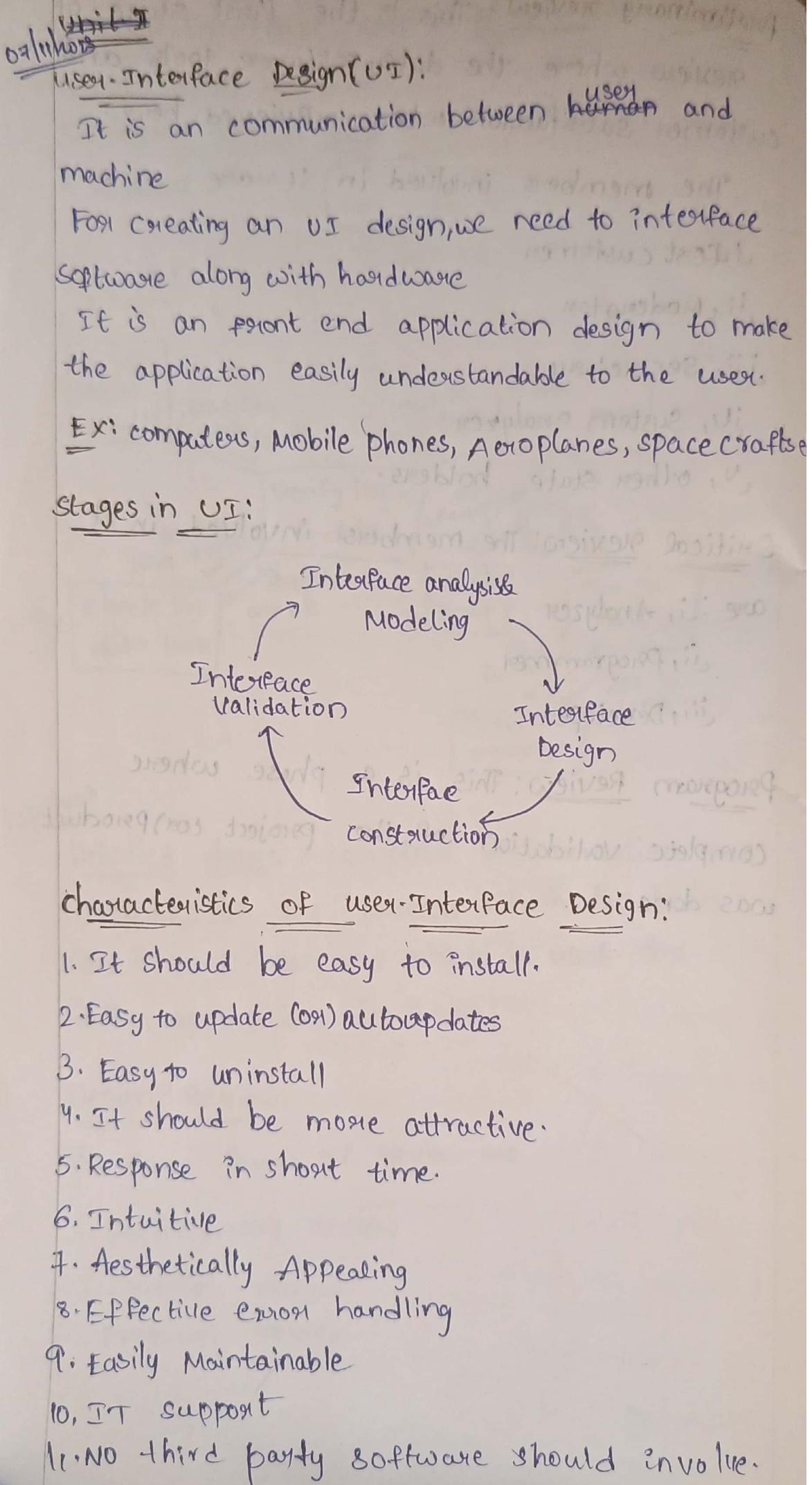
- i, customer
- ii, Moderator
- iii, Secretary
- iv, System analyzer
- v, other state holders.

Critical review: The members involved in it

are :

- i, Analyzer
- ii, Programmer
- iii, Developer.

Program Review: This is a phase where complete validation of the project (or) product was done.



## Types of UI:

1. command line Interface(CLI)
2. Graphical user interface(GUI)

CLI: It is an syntax (or) text entered by the user to perform any action within that application.

Ex: Scientific applications will be designed using

CLI

The keyboard is used for performing every action in CLI

components:

1. command prompt

2. cursor

3 command

command prompt: The display visible to the user in the command line environment is called command prompt.

cursor: The pointer used for entering the text. It can only be access through keyboard.

command: The syntax (or) instruction provided by the user.

GUI: It is an environment where double quote developer will represent every action in pictorial and text format which is easily understandable to the user.

Here, the user defines  
mouse is an important tool used for accessing  
(or) performing the actions.

## Components:

1. Windows
2. Tabs
3. Menu
4. Icons
5. Cursor.

Tabs: The space where all the menus were placed.

Menu: The list of options which are placed in the tab by the developer to make the user perform any actions.

Icons: It is an symbolic (or) textual representation used by the user to perform actual function.

cursor: It is used for both <sup>entering</sup> and selecting the text.

## Fundamentals of GUI:

1. Aesthetically Appealing
2. clarity
3. compatibility
4. configurability
5. comprehensibility
6. control
7. consistency
8. Directness
9. Efficiency
10. Familiarity
11. Flexibility
12. Forgiveness
13. Predictability
14. Recovery
15. Responsiveness
16. Transparency



Aesthetically Appealing: Developer should not compromise upon the structure of the front-end clarity: The front-end part should contain a bright structure which pleases the user's eye

compatibility: It should be easily accessible in every device.

configurability: The size of the application should not exceed the maximum range.

comprehensibility: The text data provided should be easily understandable to everyone.

control: can be accessed for multiple tabs.

consistency: Maintaining the stability of the system after updation.

Directness: Every picture (or) text represented should convey a direct meaning.

Efficiency: Software should have high quality & high response irrespective of the environment.

Familiarity: The application created should not contain much changes.

Flexibility: Easily adopting the changes in environment.

Forgiveness: Debugging of errors automatically without user involvement.

Predictability: The generation of errors was done automatically.

Recovery: The updation of error debugging task will be done here.

Responsiveness: Responding in short time for every action.

Transparency: The only the front-end should be visible to the user.

1/1/2023

Design Model: It is a next phase in designing a software after SRS documentation which contains the structure of data, program and model.

There are 4 types:

1. DDM - Data Design Model
2. ADM - Architectural Design Model
3. UIDM - user interface Design Model
4. CLDM - component level Design Model.

UID Principles:

1. user familiarity
2. Recoverability
3. consistency
4. Minimal Surprise
5. user diversity
6. user guidance

User familiarity: The basic content which was already been existing shouldn't be changed.

Eg: In OS - windows, desktop, folder, power etc..

Recoverability: User may make some mistakes while using the application which should be retrieved or modified easily.

Eg: undo, rename, restore etc..

consistency: The data or content placed in an software application must be of same form, size, color etc.,

Minimal surprise: The contents placed in same software applications should be same.

Eg: In MS Office - File → Open, New file, Save, Save as etc.

User diversity: Various options provided to the user for using the application

Eg: Privacy settings.

User Guidance: The guidelines report provided to the user to make them familiar to the application.

Golden rules of UI:

1. It places the user in control
2. It helps in reducing user's memory load
3. Makes the interface more consistent which helps the user to easily access the software.

UI Issues:

There are 4 issues in UI:

- i. Error handling
- ii. Response time
- iii. Health facilities
- iv. Application accessibility

15/11/2023

UID Process:

1. UI requirement gathering
2. User Analysis
3. Task Analysis
4. UI design & Implementation
5. Testing

GUI Methodology:

1. Examine the use case model
2. Task and user modeling
3. Metaphor selection
4. Interface implementation and rough layout.

5. Detailed presentation and demonstration
6. GUI construction
7. user diversity & usability

### Task Modeling:

1. Agents
2. Goal
3. Input
4. Output
5. Method

16/11/2023

### Coding & Testing

coding: It is an programming language used to communicate with the computer

Code review: The phase where developers will execute the code for checking the errors before passing it through testing phase. There are 3 stages

1. code walkthrough
2. code Inspection
3. pair programming.

code walkthrough: The reviewers are:

technical writer,

skilled person in entire team,

General head (or) manager,

Quality assurance officer,

any other interested members.



Code Inspection: It is done within the team where evaluation of the code will be done. The participants involved in it are: Programmer, Tester and Designer.

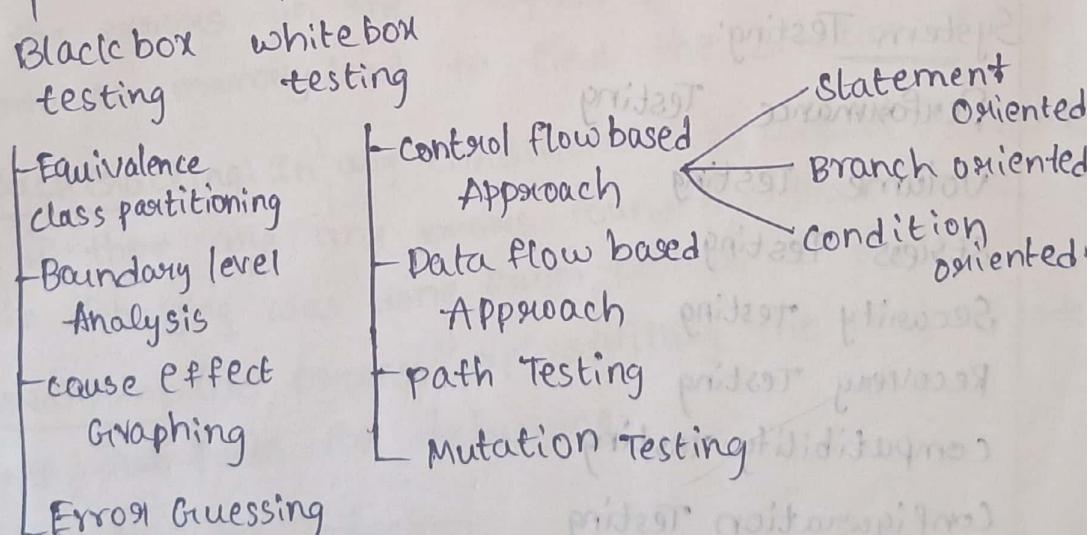
Pair Programming: Only the developers who are involved in it. Here, after developing a code by one individual person, the developer can take suggestions from another programmer for execution and debugging of code.

18/11/2023

## Software documentation:

1. Requirements
2. end user
3. Technical
4. Architecture/Design
5. Marketing.

## Testing:



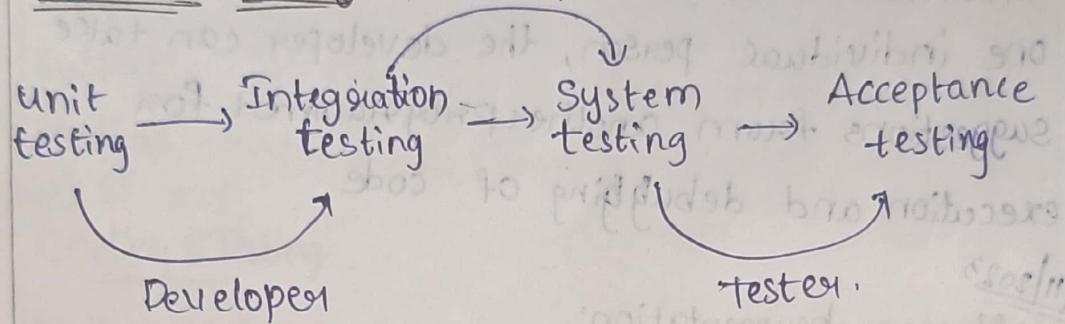
Mutation Testing: It is a process of testing a changed program.

The program which has been changed is called mutated program

The process of changing the program is called mutation.

Program being changed is called mutant

Levels of testing: developer / tester



20/10/23

Integration Testing:

- Bang Bang Approach
- Top- Down Approach
- Bottom-up Approach
- Sandwich Mutation Approach.

System Testing:

- Performance Testing
- Volume Testing
- Stress Testing
- Security Testing
- Recovery Testing
- compatibility testing
- Configuration Testing
- Installation Testing
- Documentation Testing

## Acceptance Testing

Alpha Testing

Beta Testing

Shadow Testing

Benchmark Testing

## Software Testing Techniques:

1. Usability Testing

2. Regression Testing

3. Smoke Testing

Debugging: The process of eliminating or removing the errors (or) bugs is called debugging

1. Brute Force

2. Backtracking

3. Breakpoint

4. Debugging by Induction

5. Debugging by Deduction

6. Debugging by Testing

Brute Force: It is an old fashion technique where the errors in the program (or) code were dumped into memory to find the solutions for it.

Backtracking: In any conditional (or) control statements, if there are any errors found in those bodies, the debugging was done from bottom to top.

the debugging was done from bottom to top.

Breakpoint: The process of splitting the entire code into sub parts for debugging.

Debugging by Induction:

# Programming Analysis Tools:

1. Static
2. Dynamic

24/11/2023

## Software Reliability & Quality Assessment

Software Reliability: A detecting and fixing an errors after gathering reduce from customers.

### Reliability Metrics:

1. Transient Failure - Failure on some inputs
2. Permanent Failure - Failure on all the inputs present
3. Recoverable Failure - Temporary failure of the system
4. Unrecoverable Failure - Permanent failure on the system may be repaired by some professionals
5. Computing Failure - Data will be lost temporarily
6. Non-computing failure - Permanent loss of data.

### Measurement of Reliability Metrics:

1. POFOD - probability of failure on demand
2. ROCOD - Rate of occurrence on demand
3. MTTF - Mean time to failure
4. MTTTR - Mean time to rectify
5. MTBF - Mean time between failure.

POFOD: It describes the no. of failures occurs in an particular period of time.

POFOD rating should always be small

Ex: If POFOD was 0.01 it indicates 1 error

(0.01) failure over using the software for 100 times

ROCOD: It indicates the no. of failures occurred while the

Ex: If POFOD 0.01 for 100 times of overall usage of the software then ROCOD is

$$\text{POFOD} = \frac{\text{No. of failures}}{\text{Total usage period no.}} \rightarrow \text{ROCOD}$$



$$0.01 = \frac{\text{ROCOP}}{100}$$

$$\boxed{\text{ROCOP} = 1}$$

MTTF: The value of time period which describes the average value of time for the occurrence of failure in a specific period.

MTTR: The time period taken to rectify a particular failure in the given specific period of time.

MTBF: The time period between each consecutive failures.

$$\text{MTBF} = \text{MTTF} + \text{MTTR}$$

27/11/2023

System Availability: The availability of the hardware specifications required for proper usage of the software.

Statistical Testing: Faulty conditions were created willingly to check the performance of the software when an unexpected error occurs.

Steps involved in Statistical Testing:

1. Creating test case data
2. Analyzing faulty conditions
3. calculating the fault rate
4. Generating algebraic expressions if necessary
5. calculating Stability rate value

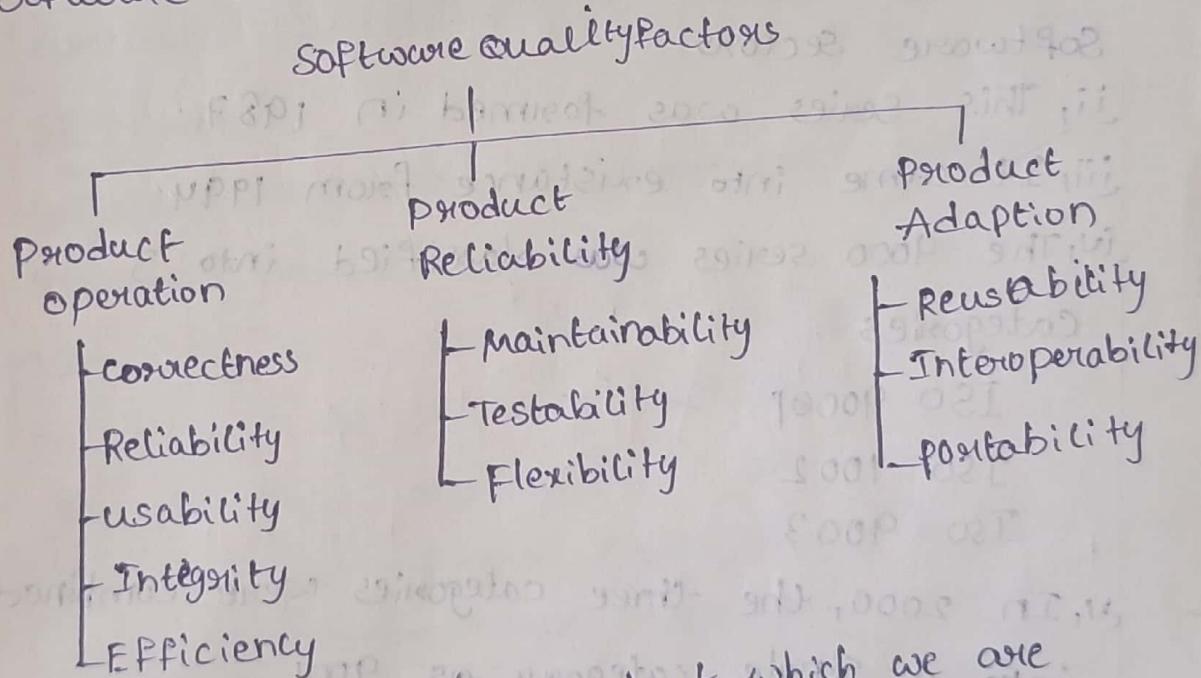
\* Software quality: The process of checking the performance of the Software is called as Software Quality Assessment.

The major components involved in SQ are:

1. satisfying the customer
2. The software should have higher characteristics capability.
3. It should have good internal and external design.
4. It should be within the desired cost
5. It should follow the development standards.

### Software Quality factors:

The components on which the quality of an software product depends



Verification: whether the product which we are designing by following the standard rules

Validation: whether the designed product was constructed correctly or not.

30/11/2023

## Software Quality Management



- \* ISO: The organization which is used for certifying the products (or) organizations etc.
  - i, ISO 9000 was the series which is used for Software sectors
  - ii, This series was formed in 1987.
  - iii, It became into existence from 1994.
  - iv, The 9000 series was classified into 3 categories
    - ISO 90001
    - ISO 9002
    - ISO 9003
  - v, In 2000, the three categories were combined into a single category as 9001
  - vi, Later depending upon the updatations in 2008, the finalized draft of the series were done.

The parameters involved in this are

C - customer focus

L - leadership

I - Involvement of people

P - Process approach

S - System approach to Management

C - continual improvement

F - Factual improvement to decision making

M - Mutual beneficial relationship

04/12/2013

### ISO9000 Process:

1. Proposal stage
2. Pre Assessment stage
3. Document Review & Adequacy Audit
4. Compliance Audit
5. Registration
6. continued surveillance.

### SEI CMM:

Software engineering Improvement - capability maturity Model.

→ In 1986, CMU in USA has given some standard policies for the software application.

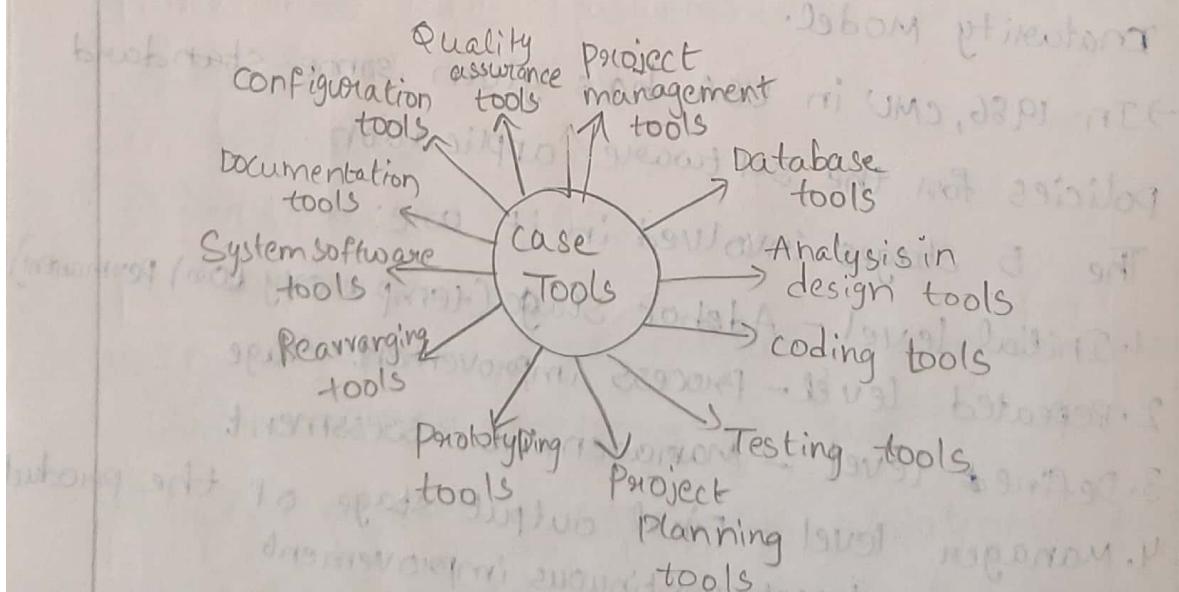
The 5 stages involved in it are:

1. Initial level - Adhoc Stage (temporarily or permanent)
2. Repeated level - Process improvement stage
3. Defined level - Management assessment
4. Managed level - Final output stage of the product
5. Optimized level - continuous improvement

Table for SEICMM regarding KPA (key process area):

level	FOCUS	KPA
1. Initial level	adoch stage	
2. Repeated level	process management	Requirements planning, Quality assurance, process management
3. Defined stage	Project Standardization	Training Program Design Improvement Planning
4. Measured level	Project improvement	Quantitative Assessment Quality Management
5. optimized level	continual improvement	Defect management updates.

\* Case:



CASE: computer Aided Software Engineering.

07/12/2023

## Unit-5: Software Maintenance

### & Software Reuse

Software Maintenance: The process of correcting the errors and improving the performance of the system after delivering the product to the customer.

#### Activities in Software Maintenance:

1. Error correction
2. Deletion of obsolete capabilities
3. Enhancement of capabilities
4. Optimization.

Error correction: Correcting of defects present in the system after using the product for some days.

Deletion of obsolete capabilities: The process of deleting the old features and introducing the new features into the system to improve its performance.

Ex: Windows update, Android update etc.

Enhancement of capabilities: The process of adding new features for already existing applications without deletion of old features.

Ex: Spell check in MS Word.

Optimization: Checking the response time of a system after installing the updates.

## Types of Software Maintenance.

1. corrective Maintenance
2. Adaptive Maintenance
3. perfective Maintenance
4. preventive Maintenance
5. Emergency Maintenance.

corrective Maintenance: The process of debugging the errors by placing a patch at the specified error location

Ex: Instagram - Reporting the error bug to the developer.

Adaptive maintenance: The process of updating the system without affecting the current application

Ex: Windows Update, Android update etc..

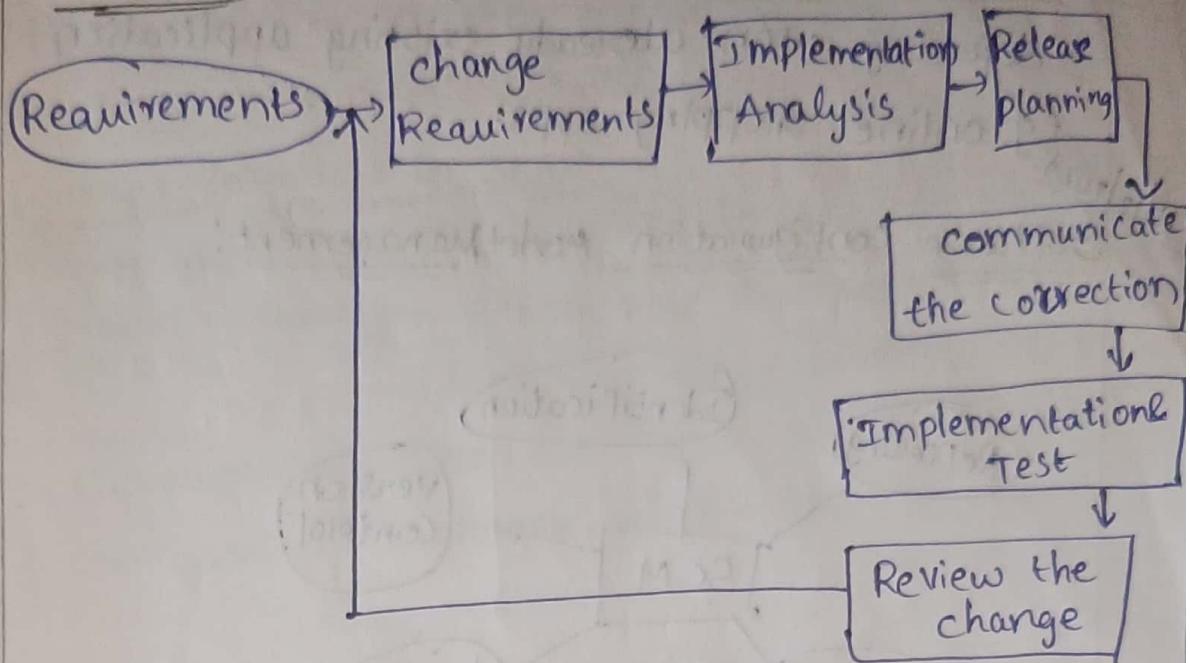
perfective Maintenance: The process of updating some features in any application to make the system work 100% efficient

Ex: updation of apps

Preventive Maintenance: The process of preventing the defects or errors present in the system before releasing it into the market.

Emergency Maintenance: When there is an heavy load on any particular application, that particular system will be in halt condition for some period of time.

## Maintenance Process:



→ The process of collecting the requirements and performing maintenance task so that the performance of the particular product will be improved.

### Maintenance process Models:

1. Quick Fix Model
2. Osbrone's model
3. Iterative enhancement model.
4. Full Reuse model.

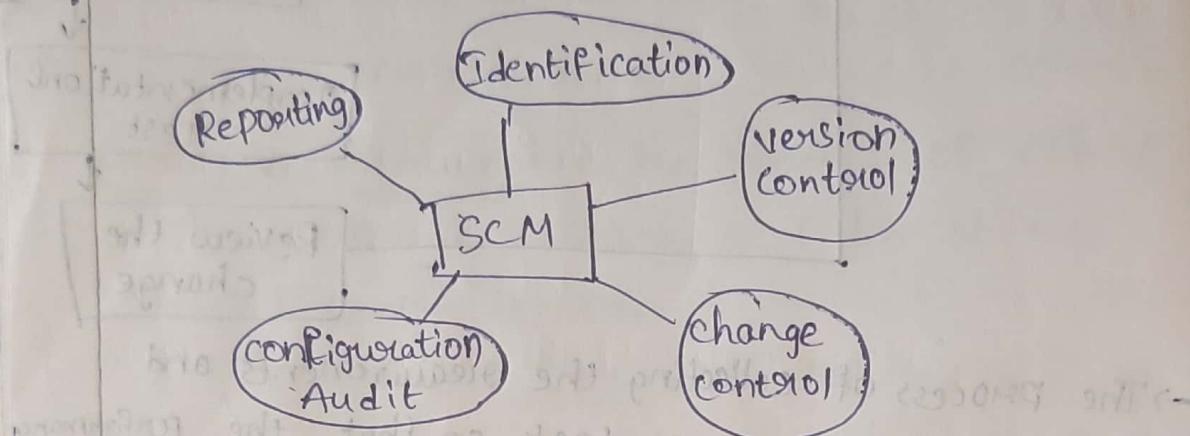
Quick Fix Model: Stopping the application for certain period of time in case of emergency  
Ex: Any e-commerce application like security aspects.

Osbrone's Model: Maintaining the entire system for improving the performance.  
Ex: Online Banking applications.

Iterative enhancement model: Implementing new features in various sub parts of the system.  
Ex: phonepe

Full Reuse Model: updating new features depending upon already existing application  
e.g. online shopping apps.

11/12/2023  
Software Configuration Model Management:



SCM: The process of adding (or) deleting the features in the existing models within the SDLC

Reporting: It is a stage of identifying the issues or implementing any new characteristics within the system.

Identification: The process of checking whether the new features can adapt to the existing version or not.

Version control: The phase where either adding (or) deleting of features will affect the existed version.

Change control: Final phase from the management side, to check whether the modified version was being adapted in the system or not.

Configuration Audit: Reviewing the updated version by passing it to the consumers (or) stateholders (or) some organizations etc...

### Persons involved in SCM:

There are 4 people involved in it. They are:

1. Project manager - Responsible for reporting & identifying the features.
2. Developer - Responsible for version control.
3. Configuration manager - Responsible for checking the performance of the system
4. User - Responsible for auditing the new version.

### Maintenance

#### Maintenance cost:

In the overall cost, the maintenance cost will

cover 80-85%.

→ It is calculated through ACT (Annual change traffic) invented by Barry Boehm

ACT: The software update done either by adding or deleting or modifications in source code within an year.

$$ACT = \frac{kloc \text{ add} + kloc \text{ del}}{kloc \text{ total}}$$

COCOMO

Basic

Intermediate

complete

14/12/2023

Software Reuse: The process of using the data from an already existing model to create a new design.

Software Reuse components: The components that can be collected from an already existing model. The major components involved are:

1. User Manual
2. Software Documentation
3. Software Requirements Specifications
4. Design/Interface
5. Source code

Steps involved in Software Reuse:

1. Requirement Specification - Identifying the features present in the system required for designing a new model.
2. Component Analysis: The process of collecting the data regarding the features that can be used for designing a new model from the existing model.
3. Requirement updatations/modifications: If there are any components present in the existing model that which can be used in the new model.
4. Reuse System Design: using the system design of the existing model.
5. Integration: checking of the new model which has been designed by the components collected from the existing model by integrating it with any hardware component.

6. Validating: The process of validating the model by passing it through testing (or) hand overing the product to some customers

### Advantages of Software Reuse:

1. Time Saving
2. Minimizes the cost
3. Team coordination will be improved.
4. Product Quality will be improved.

### Factors affecting Software Reuse:

1. component creation
2. component indexing and sorting
3. component search.
4. component understanding
5. component integration
6. Repository maintenance.

16/12/2023  
In an given project the number of lines of code are 20,000. After modification during maintenance period 1000 lines of code were added to the existing code by deleting 500 lines of code. calculate the maintenance cost.

Given

$$\text{KLOC Total} = 20,000$$

$$\text{KLOC add} = 1000$$

$$\text{KLOC delete} = 500$$

$$ACT = \frac{1000}{20,000}$$

$$= 0.05.$$

In the following software project the total estimated lines of code were 20,000. By considering  $(a, b) = (2.4, 1.05)$  as multiplicative and exponential factors estimated from COCOMO model where the estimated multiplication and exponential factors were  $(c, d) = (2.5, 0.38)$ . calculate the project completion time?

$$MT = M \cdot F \times (\text{effort})^{E/F}$$

$$\text{Effort} = M \cdot F \times (\log \text{in } k)^{E/F}$$

Given:  $(a, b) = (2.4, 1.05)$

$$(c, d) = (2.5, 0.38)$$

$$LOC = 20,000$$

### Blanks

A software company needs to develop a project i.e., estimated as 1000 function points and is planning to use Java as a programming language whose approximate lines of code per function point is accepted as 50. Considering  $a=1.4$  and as multiplicative factor,  $b=1.0$  as exponential factor for the basic COCOMO effort equation  $c=3$  as multiplication factor,  $d=0.33$  as exponential factor for the basic COCOMO duration equation, approximately how long does the project take to complete?

Given,

$$(a, b) = (1.4, 1.0)$$

$$(c, d) = (3, 0.33)$$

$$\text{Gitter L.O.C} = 50 \times 1000 \\ = 50,000$$

$$\begin{aligned}\text{effort} &= \boxed{1.4 \times (50,000)} \\ &= 70,000 \\ M.T &= \boxed{\begin{array}{l} (3) \times (70) \\ 3 \times 4.0633 \\ = 12.1899 \end{array}} \quad \boxed{\begin{array}{l} 1.4 \times 70,000 \\ = 70,000 \\ 3 \times (70,000) \\ 3 \times 39,708 \\ = 119.12 \end{array}} \\ &\quad \begin{array}{r} 0.33 \\ \hline 33 \\ 100 \end{array}\end{aligned}$$

21/12/2023

### DevOps:

In entire software engineering the complete team was characterized categorized into two parts:

i, Developing team

ii, operational team

The functions of developing team are:

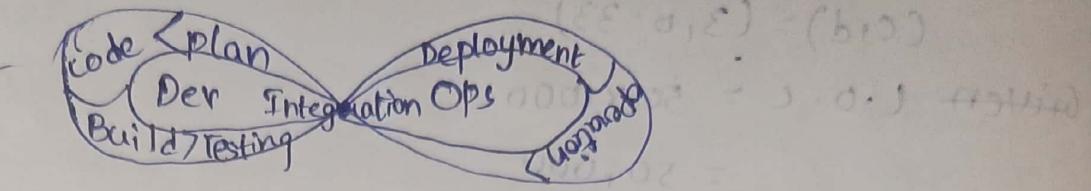
Planning, Designing the product and testing for errors.

They are also involved in debugging process whereas the functions of operational team are: Deploying the product into the market, collecting the reviews from the customers (or) users and preparing a documentary report to then handover them to the developing team.

This entire cycle will may take a long time, to avoid that DevOps was introduced.

DevOps was the process where both developing & operational team will work together

The functions of DevOps were:



## Architecture of CASE Environment:

