

Special

~~Information~~ ~~Security~~



Software Engineering

~~x~~

→ is a branch of computer science which deals with design, developed testing and maintenance of software applications.

→ it apply engineering and knowledge of coding language to build software solution.

→ aim to build a fault free software according to satisfaction and deliver in time within budget.

→ objective of software engineering :-

- ① understand user & conceptual model to develop
- ② to improve design language
- ③ to satisfy user & requirement
- ④ to achieve low maintenance
- ⑤ to provide the software within budget

○ principal of S/W Engineering :-

① modularity :- Breaking software into smaller & reusable component that can be developed and tested

..... Date

Spurts

② abstraction :- Hiding the implementation detail of a component and exploring only necessary.

③ incapsulation:-

④ Reusability :- Creating component that can be reused into multiple project.

⑤ Maintenance :- Updating software regularly within a week or within a month to fix the bug and add new feature of security.

⑥ Testing :- for bug free and compatible with user need.

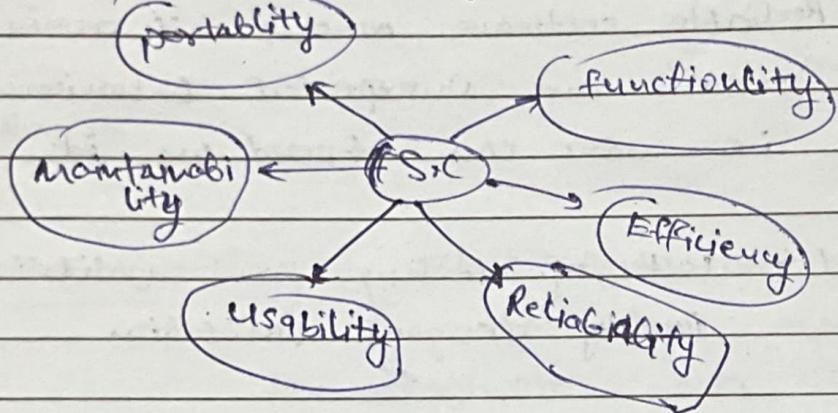
⑦ continuously integration } and development } integrate the changes and deployment into the environment

Software

Software characteristics :-

software = collection of computer program, rule and data

it has six major component :-

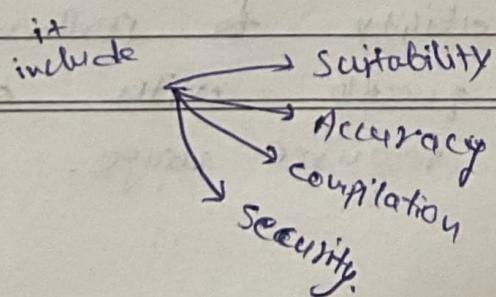


(1) functionality :- it refers ability of software to perform its intended task or functions.

→ it includes features like calculn, data processing, user interface

→ good functionality means which meet user requirement effectively and efficiently.

- Data storage
 - Data processing
 - Data processing and manipulation.
- of functionality



Spices

② Reliability → means the software you have made should be able to perform consistently and predictably under various conditions.

→ Reliable software means it avoids errors and unexpected behaviour i.e. user can depend on it.

c.f. → lack of testing and validation
poorly designed validation

③ Usability → go for easy development of software so that user can easily understand the functionality.

include → aspect like user interface designed migration, accessibility and user experience

→ reduce time in using and understanding the software,

④ Efficiency → ability to perform task quickly with minimal resource usage.

.....
Date

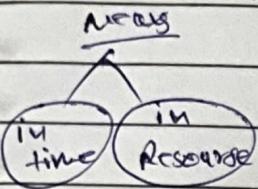
Special

include :- optimising algorithm, utilising system and reducing overhead.

These can effect efficiency → ① poorly designed algorithm and data structure

② unnecessary computation

③ unoptimised code



⑤ maintainability :- Stand for How easy it is to maintain and update the software over its life cycle.

includes → code readability, documentation and use of best practices.

→ Maintainable Software easier to debug/ enhance and scale.

key testability, stability, changeability, openability

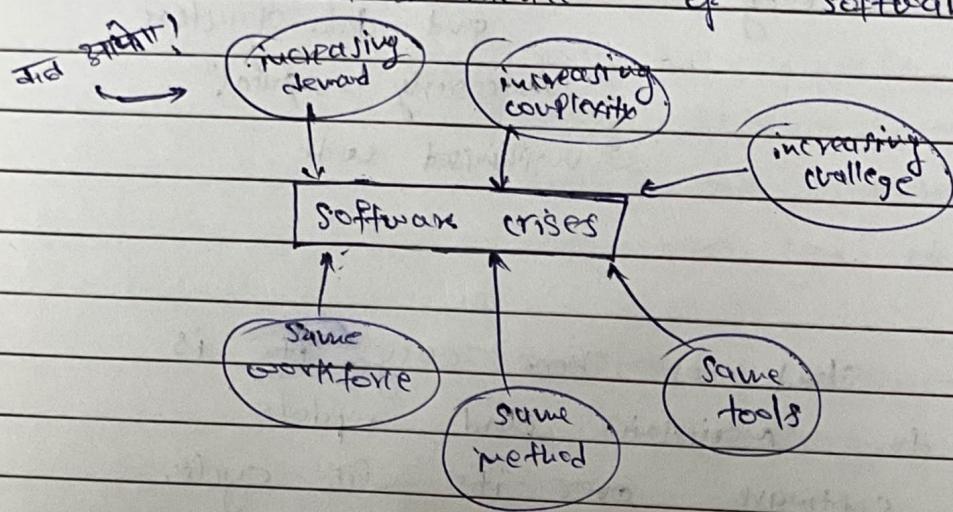
⑥ portability :- It refers to the software ability to run on different platforms.

Sprial

#

Software crises →

Set of challenges, and issues that arise in the development and maintenance of software system.



Some causes

- → do not complete at demand time
- → Quality was low
- → Software can't fulfill user requirement
- → at the fine software never delivered
- → lack of optimisation
- → lack of communication b/w user and client

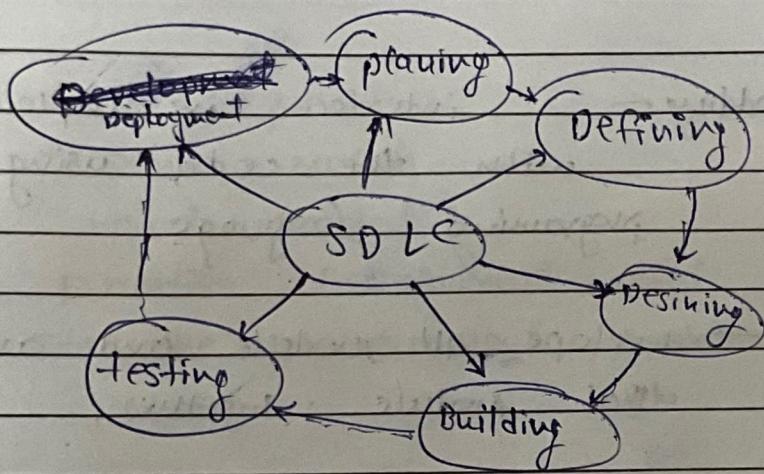
Date

Special

SDLC

* Stand for software development life cycle.
it is "Diagrammatic representation" of software life cycle.

→ SDLC used by software industry to design, develop and test software.



Planning → gather the necessary information from customer
understanding the project

Defining & feasibility → important phase discuss about the profit cost and company loss

→ measure the benefit of project for Organisation

Steps

(+)

Design :- work with logical designing

→ language, database, modular design etc.

→ provide a prototype of final project

(+)

Building & coding :- interface is implemented with the code using programming language

→ develop all module and arrange that module together.

→ largest phase have a big work.

(+)

Testing :- → testing team start testing the functionality of the entire system

→ software is checked for bugs/error if found then send to coder to fix it

→ again re-test. and refix.

..... Date

Spiceball

④ Maintenance :- look over the software use and user's feedback

→ bug fixing update and enhancement of the software

S.D.L.C modules :-

- ① Waterfall model
- ② Prototype model
- ③ Spiral model
- ④ Evolution Development model
- ⑤ Iterative enhancement models.

① Waterfall Model :-

→ here each state depends on the previous state except first state

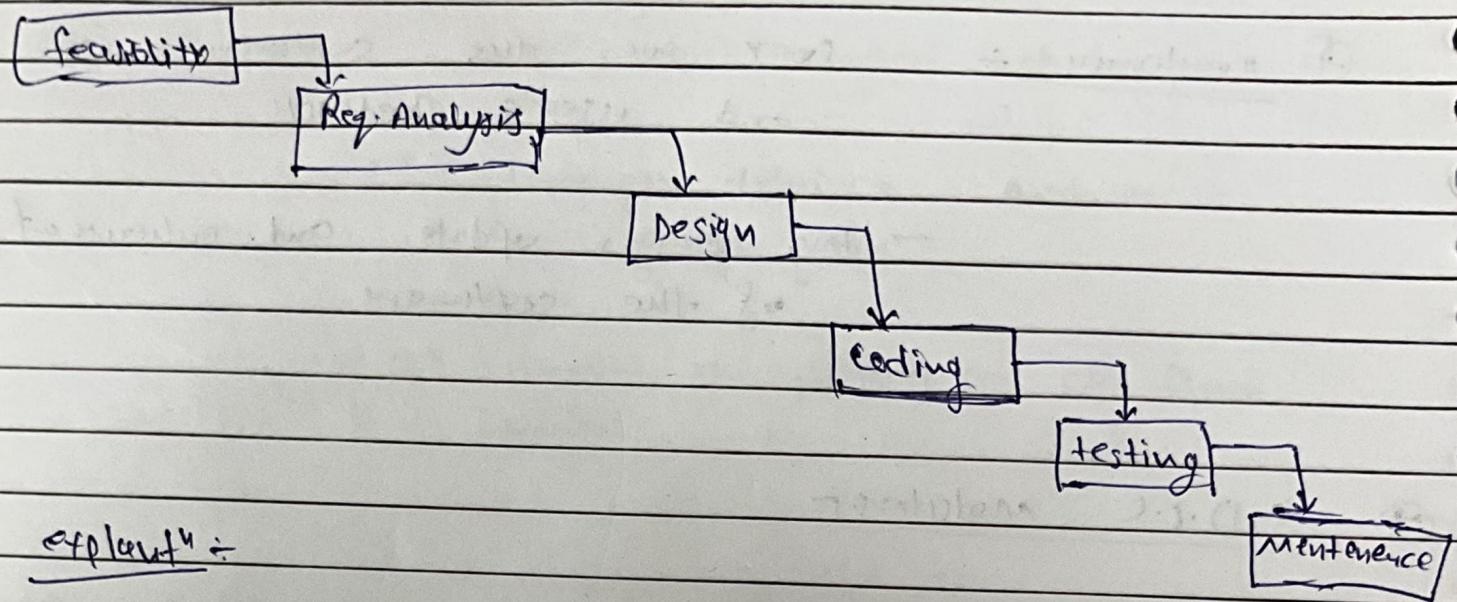
→ because of its diagrammatic representation resembled as like a waterfall

→ technologies should be well known or stable

→ project should be of minimal changes are expected during the deployment.

..... Date

Spiral Model



explain :-

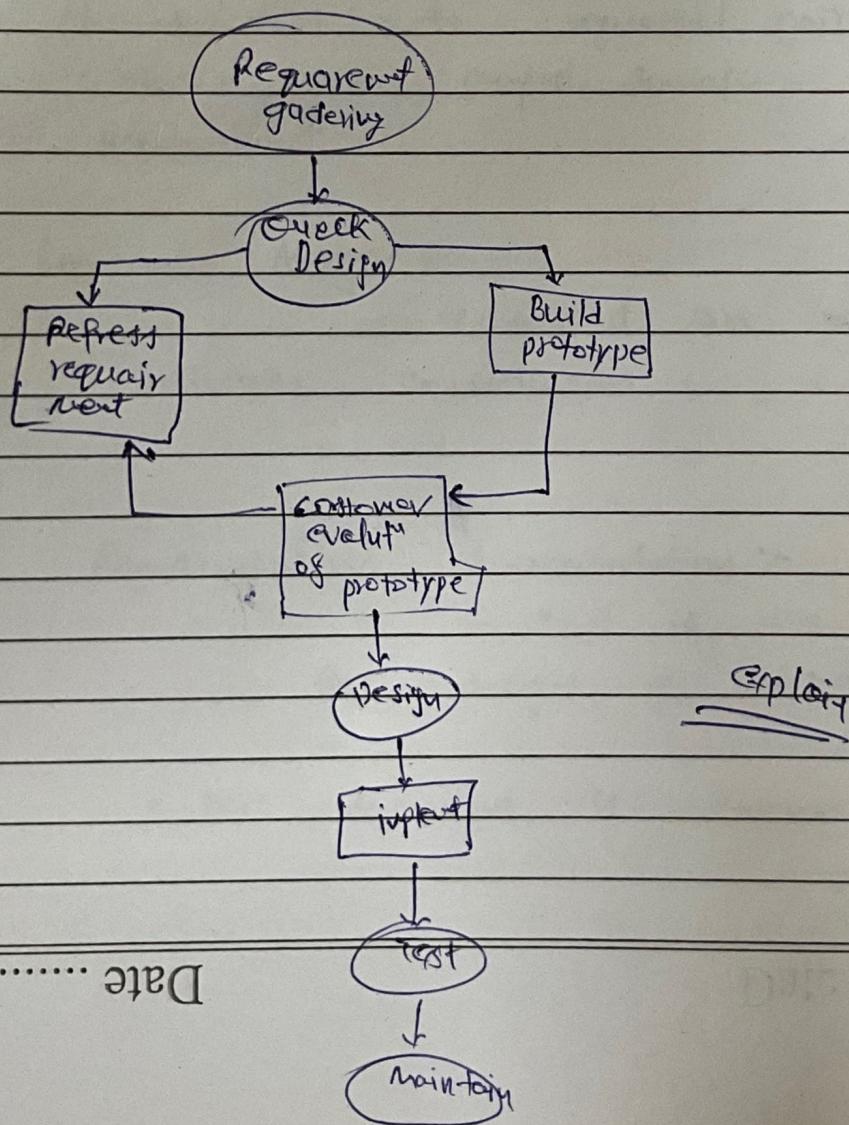
Spicule

PROTOTYPE MODEL

Some or most of the ~~the~~ customer are not sure about the functionality they require from the software.

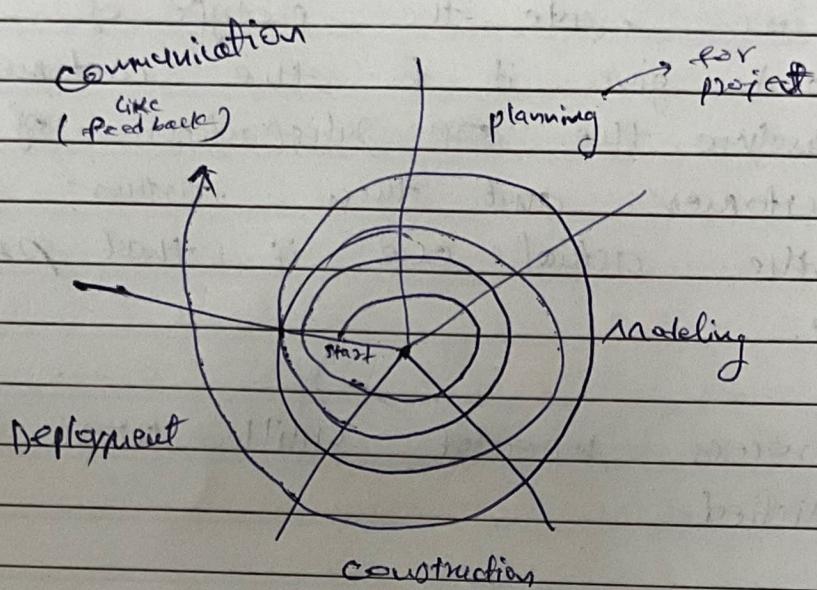
Here we create the prototype of model and give it to the customer and analyse the ~~the~~ satisfaction of the customer and then further build the actual one if that prototype satisfies.

This process repeated until user get satisfied.



Spiral model

Spiral model is a combination of a waterfall model and iterative model known as meta model.



..... Date

Requirement Engineering process :-

- it refers the process of defining, maintaining, designing, documenting requirement in engineering.
 - it provides appropriate mechanism to understand what the customer desires specifying the solution clearly.
- ⇒ Process :-

(1) Requirement Elicitation :-

- gathering of requirement
- if it is related to required fair knowledge about the project domain and requirement.

(2) Requirement Analysis :-

Requirement are analyzed to maintain consistencies

(3) Requirement documentation :-

- it is the foundation for design of software

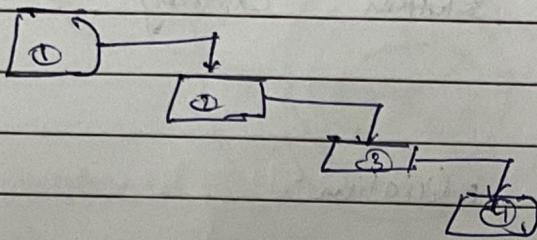
- Here document is known as SRS

Spiral

(4) Requirement review :-

- Review process is carried out to improve the quality of the SRS.
- ↳ (Software Requirement)
System

if it is done only at the end
of production of SRS
also called requirement verification.



Method of requirement elicitation :-

(i) interview :-

- Both parties move like to understand each other
- it is of two type
- open ended → where no preset agenda, free question can ask
 - structured interview → preset agenda

(ii) Brain storming :-

- A kind of group discussion which Date

Spiral

lead to idea very quickly.

all participant are encouraged to say whether which idea come in their mind.

(3) Delphi technique :-

Here some participant are very sri so they write the idea into the paper and distribute them among all.

(4) FAST :- Here customer + developer make a team and build a fast together.

(5) prototyping :- to gather the feed before development

(6) Feasibility study :- Here we specify whether the software product / project is practically possible or not

there should be a valid reason to create the software that accept the user.

..... Date

Notes

→ Diff tool which is being used.

- ① Data flow diagram
- ② control flow
- ③ ER Diagram

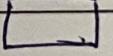
① DFD :-

→ Data flow diagram is a graphical representation of the flow of data within a system.

→ also known as Bubble chart through which we track the flow of data graphically.

→ By the help of this we can easily understand the flow of data.

Component of DFD :-

① processes :- functionality that are perform in data it represented by  or 

② Data stores :-  or 

③ Data flow :- → denotes flow of data between processes, and direction,

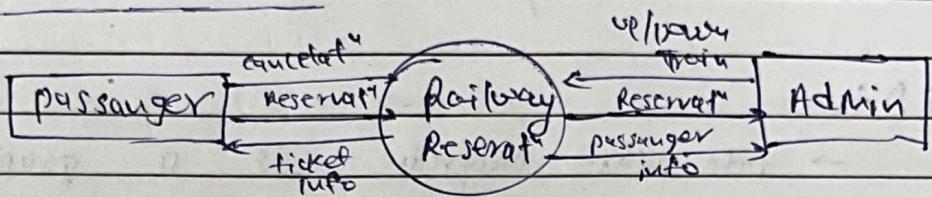
④ External Entities :- Represent source

..... Date or denoting data outside the system,

Spicule

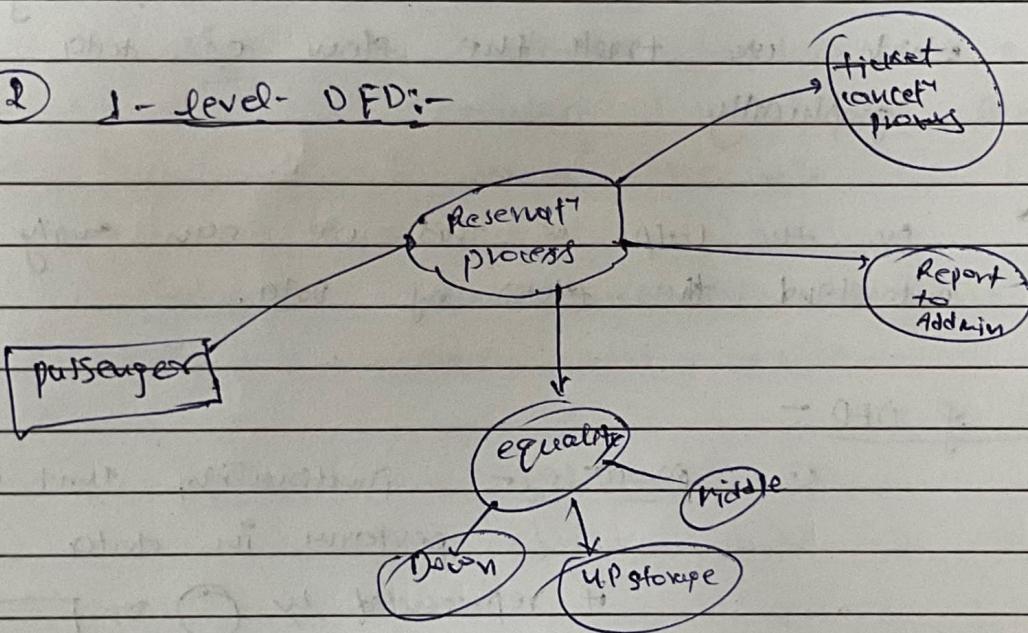
④ Level of DFD → understand with this example

① 0-level DFD :-

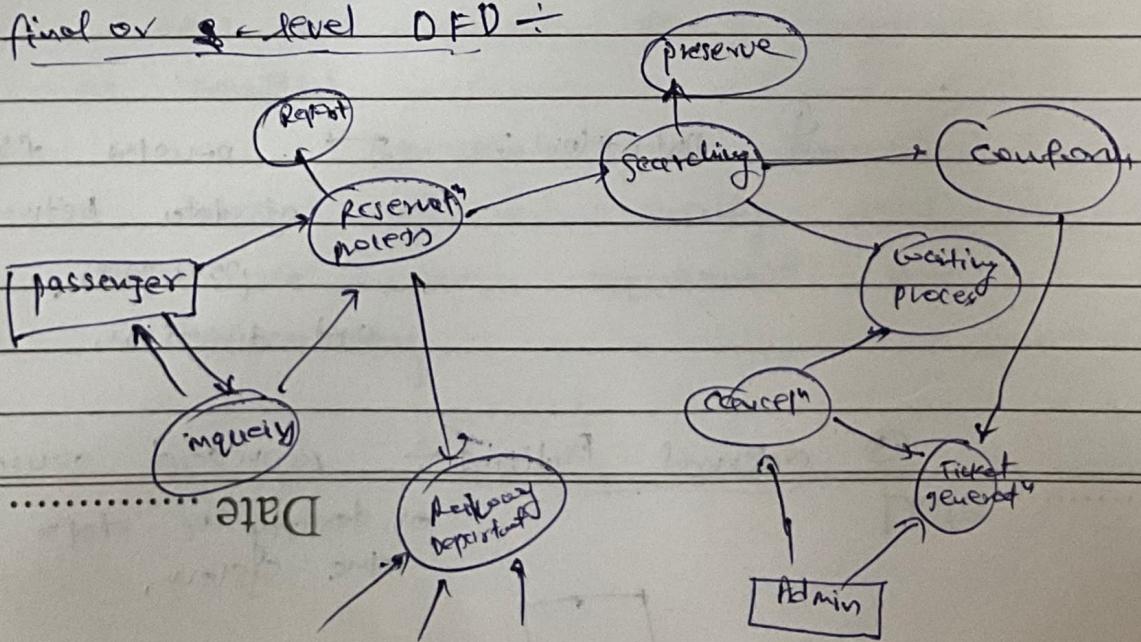


Now convert into
1-level DFD

② 1-level DFD:-



③ Final or 2-level DFD :-

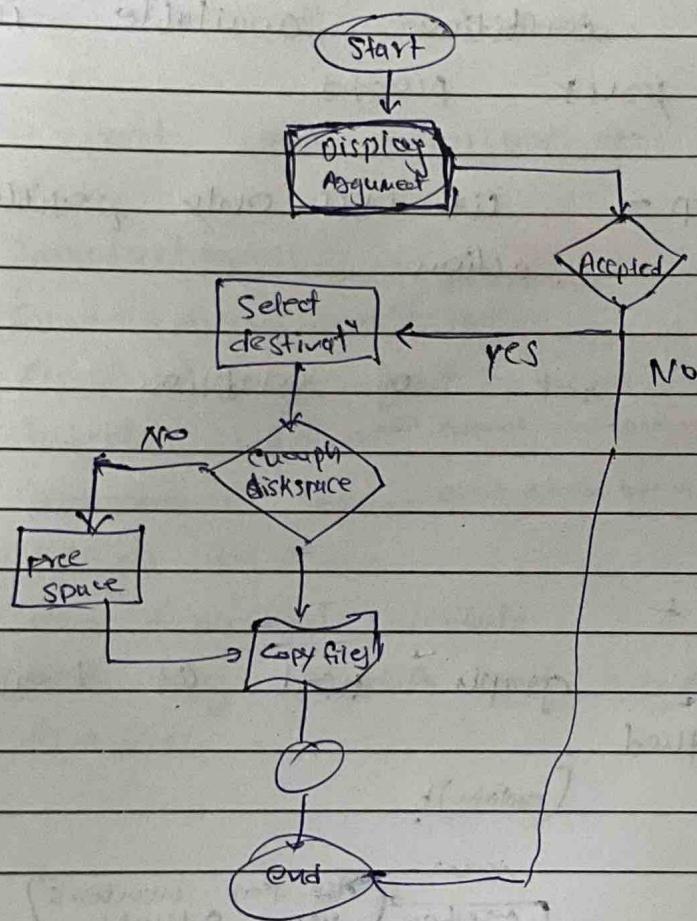


Special

Control Flow Diagram

A flow chart is the graphical representation of how control flow during the execution.

(f) Flow chart of software installation



ER Diagram = entity Relatⁿ Diagram, as I know

..... Date

Spreads

#

Design Table :-

- Tabular Represent" of action and condition
- used when logic is very complicated

⑤ cond step :- list out all the condition available in your process

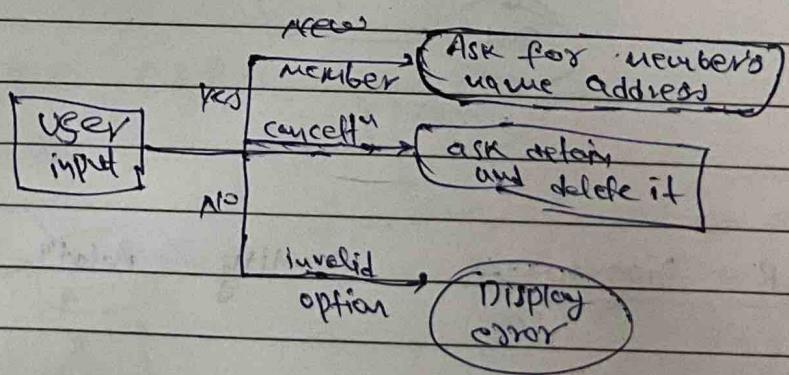
⑥ act step :- list out only possible action

⑦ Rule :- set of condition

#

Design Tree :-

graph used as branching method
(explain) t.

Example

Date

Spec SRS Document

(Software Requirement Specification)

It describe the nature of a project software or application
purpose of SRS is to provide the detail overview of our software product.
it describe the user interface and software requirements.

its part and structure →

- ① introduction
- ② General description ↗ characteristics
features
- ③ Functional requirement ↗ description
cost and schedule
- ④ interface requ → how software communicate with each other (GUI, CLI)
- ⑤ performance requ → event driven type of work
- ⑥ design constrain
- ⑦ non-functional attribute
- ⑧ preliminary schedule and budget
- ⑨ Appendices

chapter 07

* Basic principle of software project scheduling

- (i) compartmentalization
- (ii) Interdependency → all tasks can be done independently
- (iii) Time allocation → each task has time
- (iv) Effort validation → effort should be validated according to task
- (v) Defined responsibilities → define each person's responsibility
- (vi) Defined outcomes.
- (vii) Defined milestones.

* the project which software organization encounter

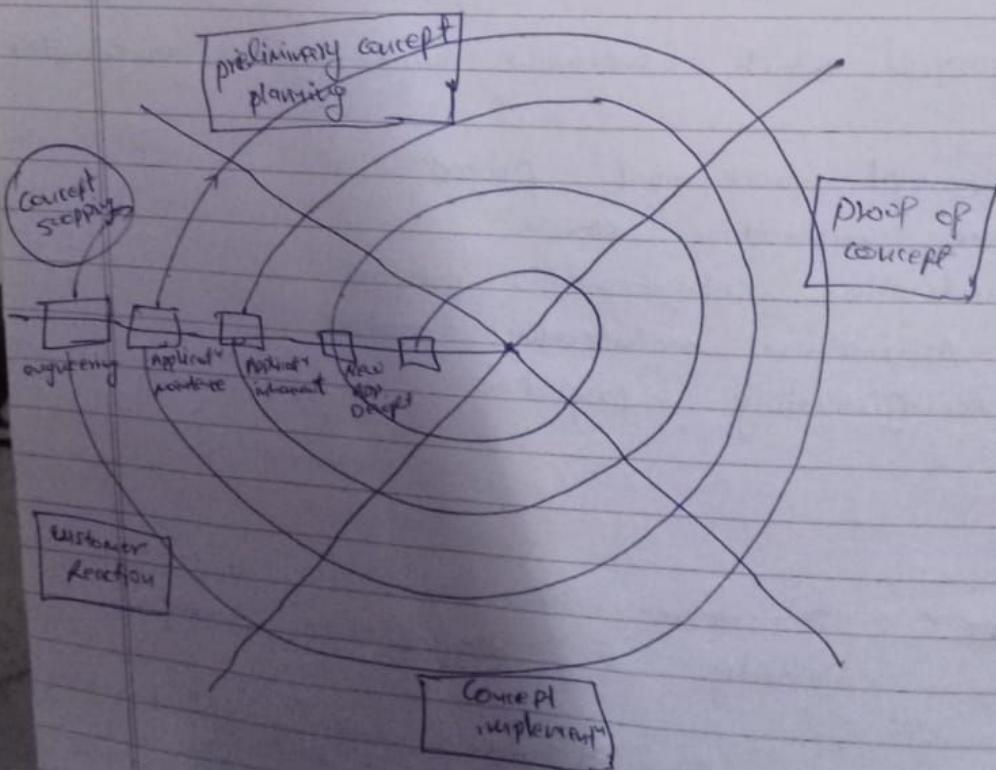
- ① Concept development project
- ② New application develo
- ③ Application enhancement project
- ④ Application maintenance project
- ⑤ Reengineering project

* Rigor :- it decide how seriously and carefully a project is handled

+ low rigor :- means • less formal, fewer rule
• startup making mobile app
• focus:- speed and flexibility

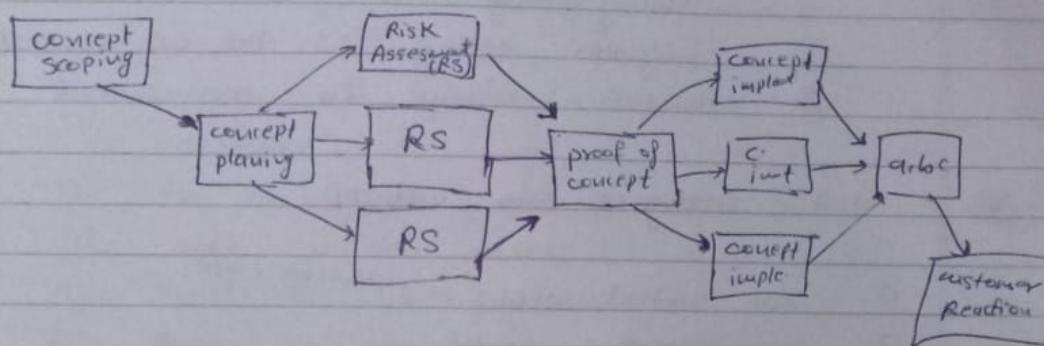
- ② High Rigor :-
- very formal with detail rule and process.
 - software for spaceship or a hospital system
 - focus:- safety, reliability.

③ Selecting Software engineering Task :-





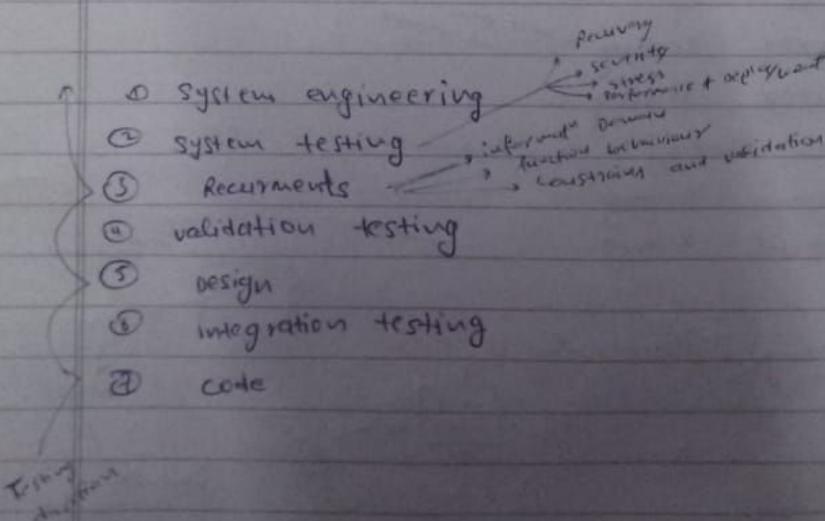
Task Network defining :-



chap 17

Software testing strategy :- (in hand)

write it in spiral order in exam.



* Test strategies for Object-Oriented Software

- ① unit Testing in the OO context
- ② integration testing in the OO context
- ③ validation testing in OO context

relative complexity
complexity

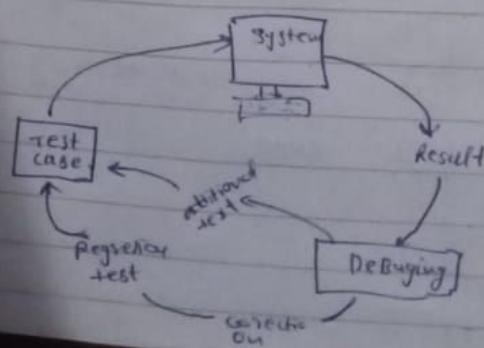
* Test strategies for webapp.

- ① the content model → to uncover error
- ② the interface model
- ③ the design Model → to uncover navigation error
- ④ test in different environment
- ⑤ security test
- ⑥ performance test

* System testing -

- ① Recovery Testing → must recover from fault and resume
- ② Security Testing
- ③ Stress Testing
- ④ Performance Testing
- ⑤ Deployment Testing

* the Debugging process -



~~chapter 8~~

⑦ Software Quality Review guideline -

- ① review the product not the producer
- ② set an agenda and mention it.
- ③ take written notes.
- ④ limit the no. of participant.
- ⑤ Develop a checklist for each product.
- ⑥ schedule time for FTR.

⑧ Preparation of SQA plan/ ^{Activity} for a project

- ① evolution to be performed
- ② audit and review to be performed
- ③ Standard that are applicable to the project
- ④ procedures for error Reporting
- ⑤ document to be produced

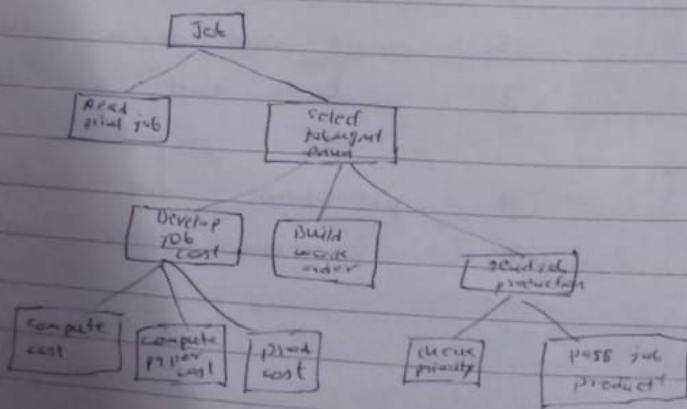
Chapter 10 :- Computer-Based System

system element which is used to make computer based system.

- ① software
- ② hardware
- ③ people
- ④ database
- ⑤ Documentation
- ⑥ procedures.

⑦ component - it is a modular building block for computer software

⑧ traditional view :-



④ Basic Design principle of class Based Component

- ① open close principle - component should be open for extension but closed for modification.
- ② Liskov substitution principle - subclasses should be substitutable for their base class.
- ③ dependency inversion principle (DIP) - depend on abstraction do not depend on concretions.
- ④ common closure principle (CCP) - classes that change together belong together
- ⑤ the common reuse principle (CRP) - classes that are not reused together should not be used together.

- ⑥ cohesion - component or class encapsulates only attributes and operations that are closely related to one another.

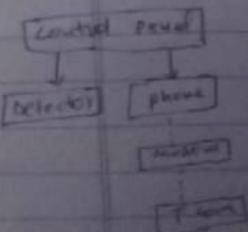


chart of which is not in note-

* software application

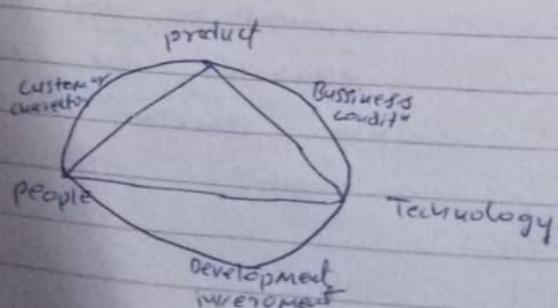
- ① system software
- ② real time software
- ③ business software
- ④ engineering and scientific
- ⑤ personal computer software
- ⑥ AI software
- ⑦ web based software

* component Based Development

figure is same as spiral model

see book once
which is written in box
page no. 42 5th edition.

* process Matrix and software project improvement.



MP-10

Coupling :- is a qualitative measure of the degree to which classes are connected to one another.

① control coupling :-

② common coupling :- occurs when all make use of a global variable.

③ control coupling :-

④ stamp coupling

⑦ Diff b/w black and white box testing

white

involve testing the intended structure, logic and code of software. Tester need to have programming knowledge and access to source code.

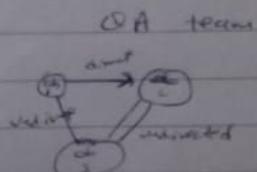
validate the internal logic control flow, loop and code coverage

programmer team.

black

focus on testing the functionality of the software without knowing it's internal structure, design or code.

here person treated software as a "black box" no internal access actually



⑧ Diff b/w verification and validation

verification

→ static practice of verifying

→ does not involve executing
the code

→ use log in type of
activity

validation

Dynamic mechanism of validating
and testing

involves executing the
code

use black/white box testing

1. functional
2. non-functional
3. system
4. integration
5. regression
6. user acceptance
7. performance
8. security
9. compliance
10. accessibility

11. 12