

11 Jan 2023
Thursday

WQMR
200.

(software has to go through changes & feature may depends on productivity!)

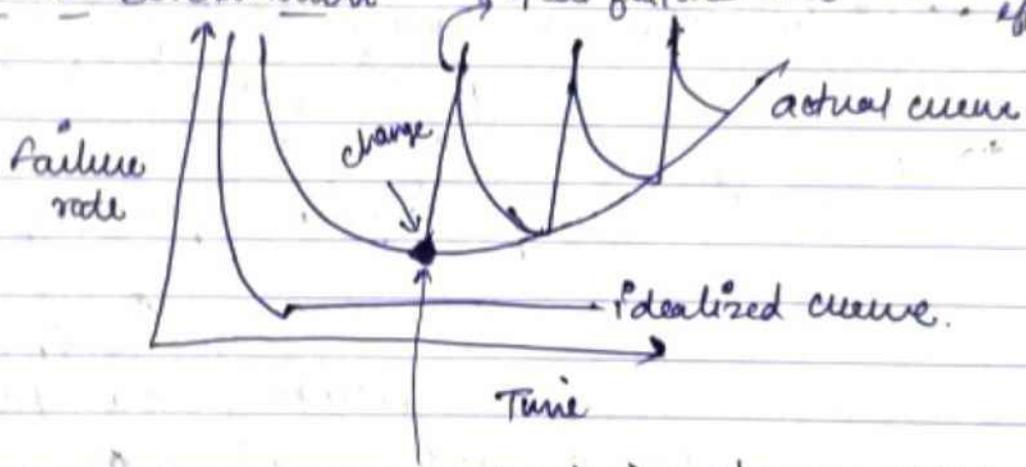
[loc :- 1] :-

- ① risk of writing software w/o proper software pipelines?
- ② not justifying purpose of it, or have lot of bugs, faulty things

③ Practically, software does not have lifetime.

- ↳ can be changed as per need, not bridge.
- ↳ errors can be turned to refinement, not bridge
- ↳ again get bna, saare & jye!

Wear Vs Deterioration :- Red. failure rate due to side effects.



every time change occurs,
software becomes new
software &
failure challenge

Q1 Which engineering principle u will use to ensure that after change (Inevitable) no sha toh ~~the~~ itna steep failure challenge na ho?

Q2 If change happens, how it is absorbed by the organisation and software engineers.

WJMK
= 500.

- Amazon sold camera of 9 lakh in 6K-7K.

↳ what's pro

which problem?

- (1) Hack kiya, value change ki^{get}
- (2) problem in database, kisi tuple se value le li
- (3) problem in testing
- (4) problem in database entry ✓
- (5) All above

(and all problem
in development?)

Yehi kar skta, as thorough testing hoga hai!!
↳ poor security", "

↳ so testing kro which compare values (purchasing price & selling")

ista ↑ difference ~~not~~

nikaalke, alert kro!!

DevOps address such losses & problems.

Q 'return policy' → problem of e-commerce.

↳ if return rate $\geq 20\%$ → loss (%)

↳ bring changes in inventory, taxation.

as GST bhi
gaayi thi, saapis
dilni hoga.

Q what is economical

↳ software?

↳ does not mean money,

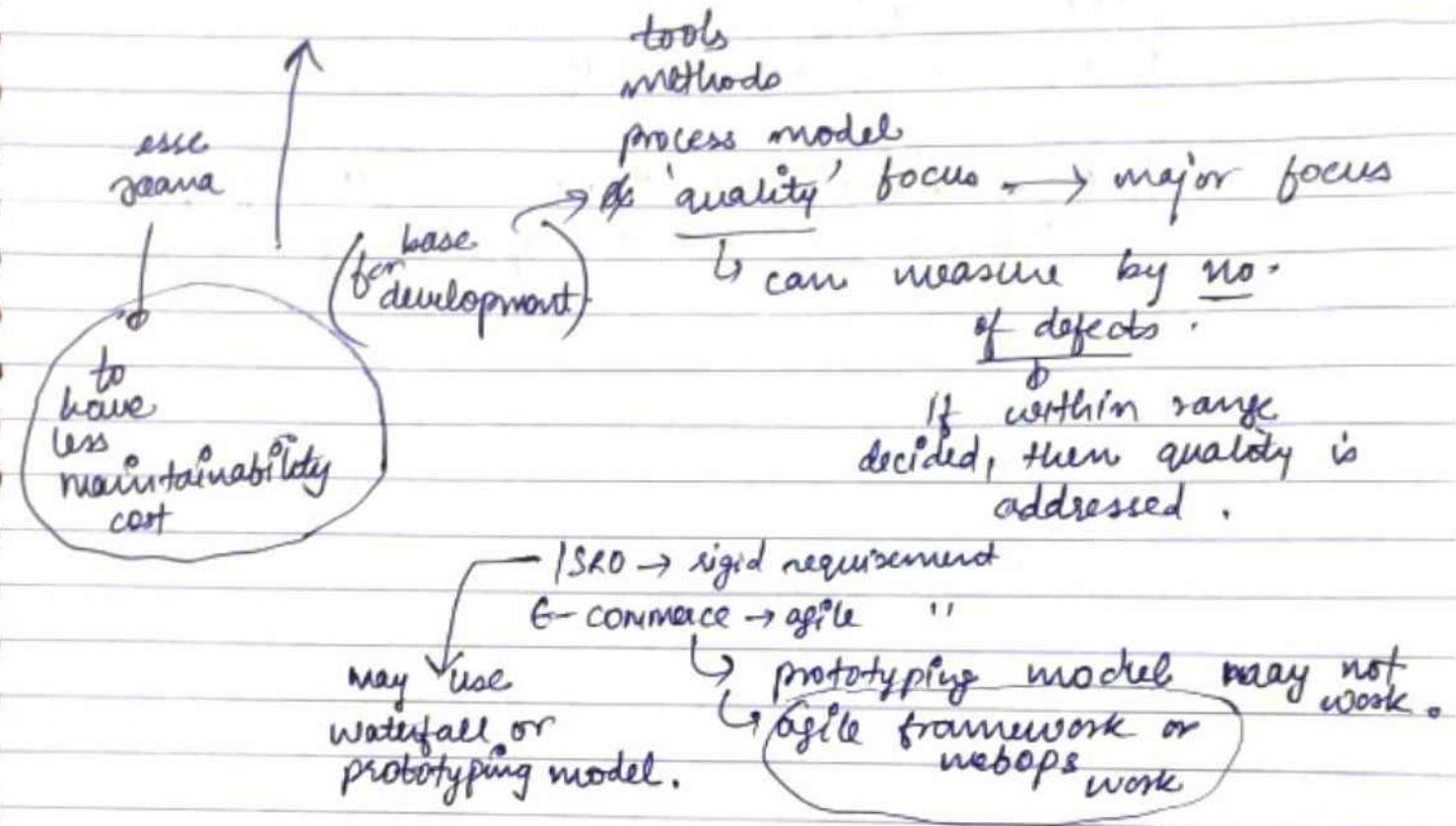
tasking of maintainability & computational efficiency.

→ more than development, maintenance cost will be high

→ payback period: 5-6 years

WAMK ~~2000~~.

A Layered technology:-



tool → integrated set of technology which will be used for development.

① Importance of SE:-

- ① Large software → difficult to handle
- ② Scalability → e.g. IRCTC (Trainter)
- ③ Cost
- ④ Dynamic Nature
- ⑤ Quality Management

$$W.S.M.K = \frac{5}{30} = 0.17$$

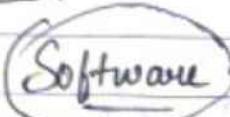
① 4 essential attributes of good software :-

- 1) maintainability
- 2) Dependability & security
- 3) Efficiency
- 4) Acceptability (must be user friendly),
↳ plays very important experience & role.

18 Jan 2023

[lec-2] :-

Thursday

Process Model :- (problem of abstraction)

↳ should be adaptive, futurist, ready for changes,
follows pipeline that may guarantee correct
output, business aims to achieve this!

→ This model is an abstract representation of a process
that presents description of process from some particular
perspective.

SOFTWARE EVOLUTION:-

→ 3 different types of software :-

- ① S-type (static type) → very rigid requirements,
strictly acc. to defined specifications & solutions.
- ② P-type (practical type).
- ③ E-type (embedded type)

↑
business type
solving real-world
problems
[architectural design
perspective]

software with collection
of procedures.
e.g. gaming s/w

SRS documentation: software requirement specifications
documentation.

- ① When cost & time are very strictly defined, then go for prescriptive process model.
- ② agile process model → talk of maneuverability & adaptability

*changes aayengya, kitni jaldi
kaise loge?*

① Five activities of a Generic Process Framework:-

- ① Communication : communicate with customer to understand objectives & gather requirements.
- ② Planning
- ③ Modeling
- ④ Construction
- ⑤ Deployment

→ creates + map defines the work by describing tasks, tasks & resources, work products & work schedule.

Stakeholders → clarifying Software requirement,
 client / user ki problem.

Communication important.

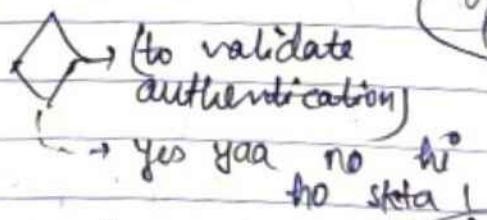
as usko kuchh nahi pta hogi → usko hoga main pay ke shaheen, so give me sol.

Umbrella Activities

(work space for team collaboration)

IDS → Intrusion detection system
 intrusion

Cybot → chatbot



- ② personal insulin pump → software must do & then categorize them, then think of interaction b/w hardware & software & then decode problem.
- functions / Tasks :-
- firstly list down the tasks :-
- software must do!

dependent
 independent
 multiple &
 conflict

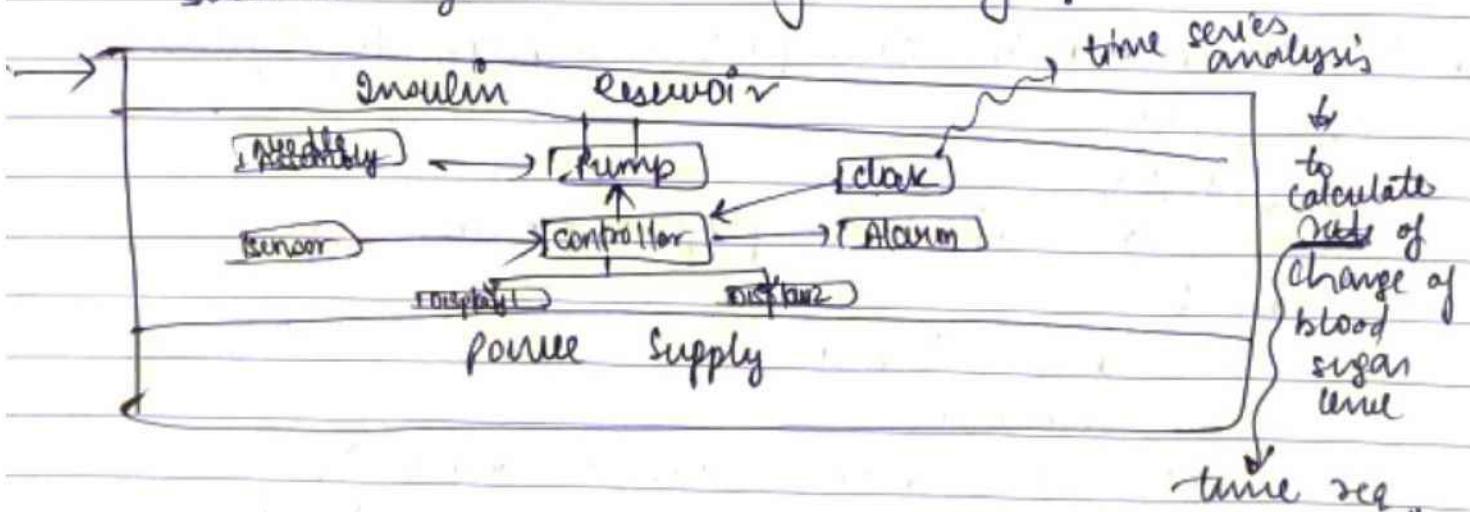
work \rightarrow product \rightarrow major module

constraints :- ① software must have calculating material to calculate insulin.

↳ very critical,
life threatening
↳ very high risk

diktat, \uparrow prdP to bhp dikaat.

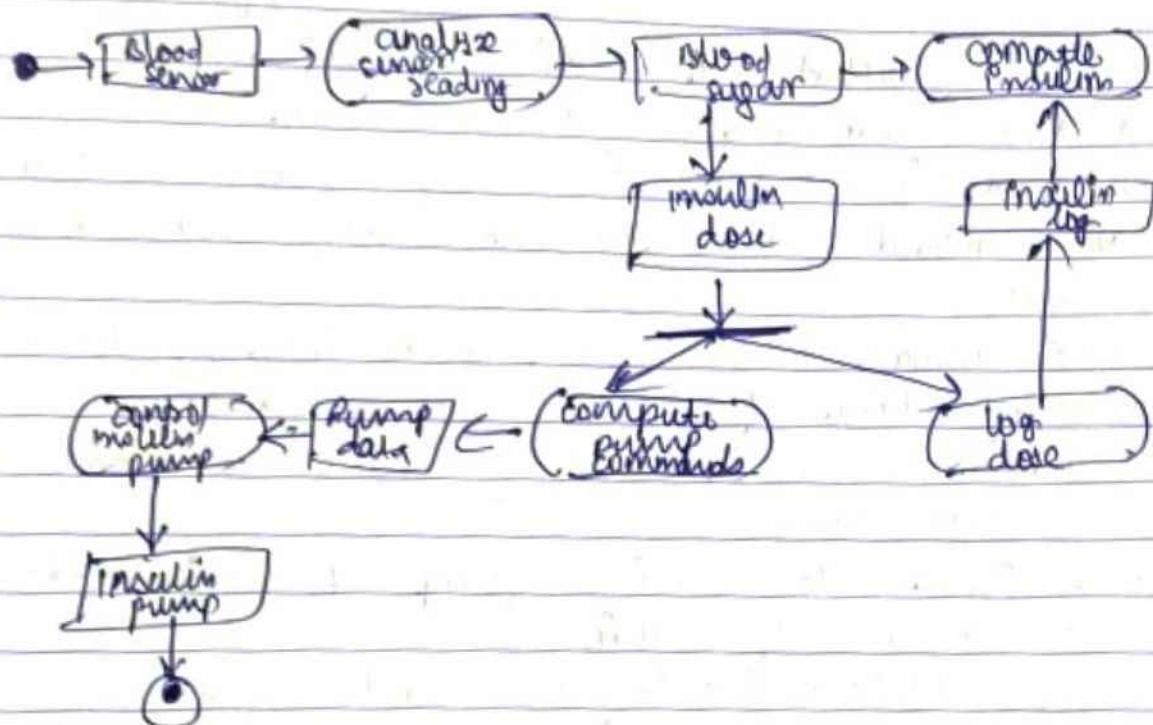
- ② Discover insulin level trends over days
- ③ must spot highs & lows i.e. must show all your data combined onto charts to identify areas for discussion
- ④ must calculate insulin req. to be injected after collecting data from blood sugar sensor.
- ⑤ calculation based on rate of change of blood sugar levels.
- ⑥ sends signals to an micro-pump to deliver the correct dose of insulin.
- ⑦ safety critical system as low blood sugars can lead to brain malfunctioning, coma & death, high blood sugar levels have long-term consequences such as eye & kidney damage!



WORK ³ ₂ M.

→ here can be failure → as something implanted in body,
that's so difficult, then life threatening.

Activity Model :-



Agile → means flexible

① A mental health case patient management system :-

(S-type) X

(Embedded) ✓
so architecture plays important role

Tasks:-

① login | sign up

② Database

③ Users: Administrators, Doctor, Patient

(3rd re diag-2
task Echo)

W3MK
= \sum MM.
=

Administration:-

- register patient
- keep record
- view history
- assign doctor
- notify doctor / patient about appointment
- assign ward / bed to patient if required
- generate report
- inform doctor about test results

Doctor:-

- Medical Examination
- request for test
- update patient reports
- assign medicines
- schedule next meet up

CSP file

- EEG, PRM report
- projecting file in form of amplitude.

- 3 parameters very imp. in this system:-
 - size of database must be scalable.
 - security of data imp.
 - scalability in terms of catering the patients.
[\because online solⁿ, many clinics]
- Pop / push data based on availability of network communication.
- think of ^{dynamic} scalable / optimized solⁿ after identifying constraints

25 Jan 2024

Thursday

Loc-3 :-

① constraints in project Model:-

- ① Time
- ② Resources

- ③ method of development

- ④ how much company & you know the requirements

- ⑤ what kind of software you are developing.

formal model :-

4 types :-
① linear model ③ Revolutionary process
② iterative model ④ parallel Model

10-40% in development process flow
60-90% in maintaining model

Types of process model :-

- ① Universal → adopting formal model based on assumptions set before you.
- ② Wordly → functional & non-functional seq, you are going to set.
- ③ Atomic

(Linear) pipeline

for development where changes not required or not acceptable.

drawback: change not possible, but change very important.

Task Set: defines actual work to be done to accomplish objectives of software engineering action.

WJMK JP
MM

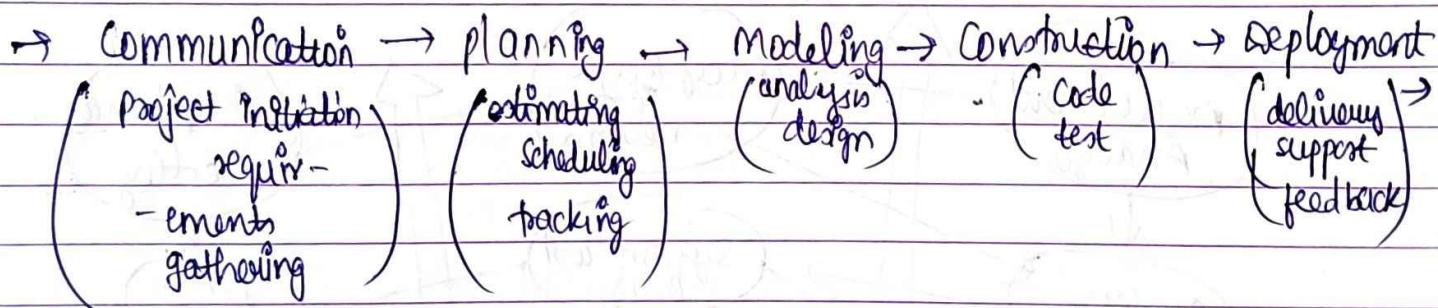
• SDLC (Software Development Life cycle) :-

systematic approach that ensures quality output

- ① Feasibility study stage
↳ whether software we are going to design / problem we will solve, is feasible or not.
- ② Requirements analysis & specification
- ③ Design → UML modelling approach (use abstraction, to represent problem in most appropriate way)
- ④ Coding ⑤ Testing ⑥ Maintenance.

when to use?

⑦ Waterfall Model :- → changes X
→ requirements known at starting only.



Examples :-

- ① NASA Space Program
- ② Manhattan project (World War-2)
- ③ Microsoft Windows OS
- ④ Automated Teller Machine (ATM)
- ⑤ Hubble Space telescope

WJM
=

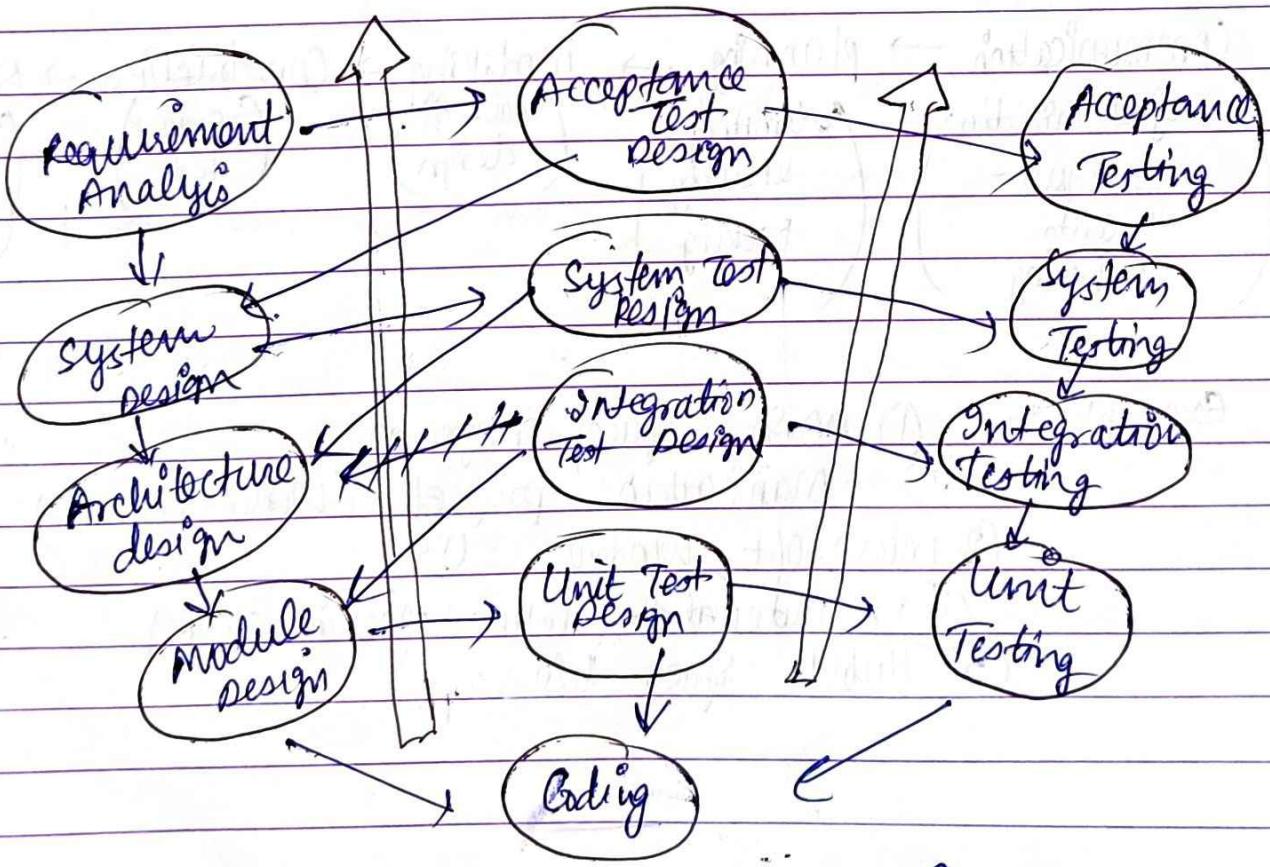
→ ensures stability of resources.
→ requirements known, start at the beginning, &
some " { are evolved in discussion with
stakeholder.

If all known at beginning → as exact o/p
known as: backward approach

↳ then waterfall model is good.

as business landscapes are changing

↳ there waterfall is not good choice.



V-model of SDLC

$$WOMK = \frac{S}{M}$$

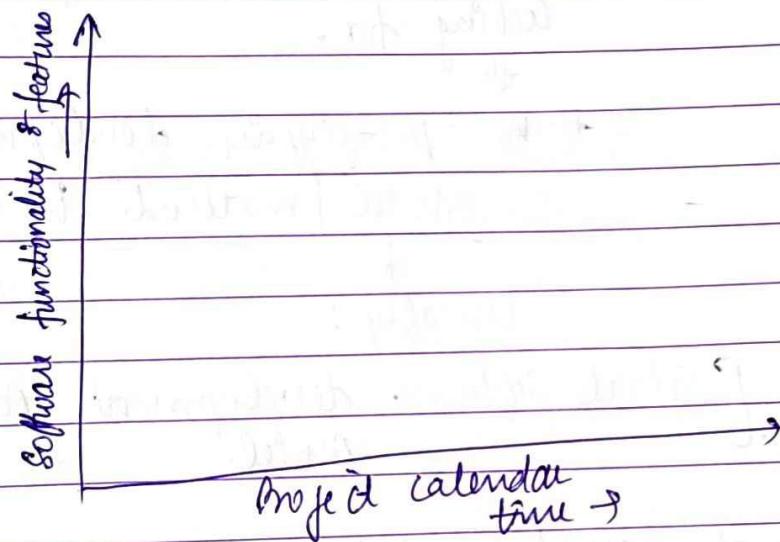
$$WOMK = \frac{S}{M} =$$

After deployment \rightarrow we measure user alignment.

① The Incremental Model :-

↑ why & where to choose it?

what are we incrementing?
functionality & features



- ① scaling not known at starting point.
- ② when phase - 2 in software will grow.

- ③ can't swap the stages.

ek van requirement belli, tot requirement to deployment \rightarrow saare phases kroge linearly

bcz changes should not bring defect ... ~~the~~ effects must get addressed properly.
 \rightarrow implications na aaye, changes se in our model.

~~WEEK~~ = ~~SEM~~

- eg:-
 - (1) OS development.
 - (2) Game development.
 - (3) Web Applications development.
 - (4) Mobile App development.
 - (5) Embedded Systems.

→ clients know what they want, but don't know how much their software can grow.

(1) Evolutionary Model :-

↳ clients does not know what they are looking for.

then prototyping development
Model / method is used !

Usually :

{ Spiral software development formal
Model

talks of
philosophy of secured ~~project~~ software model.

4 Quadrants of Spiral :-

(1) 1st :-

- i) Requirement analysis
- ii) (Risk)

means reability of software.

SQL → risk → SQL injection

- iii) either remove risk or find alternative solution.

↳ prototyping :-

prototype 1, 2, ...,

until objective is
achieved.

WOMK
= 5 mm.

- (2) 2nd :- Alternative solns are evaluated to select best
- (3) 3rd :- develop & verify next level of product.
- (4) 4th :- deployment of system.

→ don't know requirements.

→ experience not required.

↳ as it is black box approach

(means nothing)

to know at
beginning)

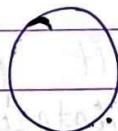
Reusability &
requirements

stable
funds

→ Opting right model is sound engineering principle.

Reuse
components

Tight
Project schedule



→ activity

○ → OR flow

● → AND flow

getRepository

→ Refinement

→ Control flow

→ Data flow

RFP : Request for proposal

↳ based on this → development companies will give
proposal ← we can accept or
reject them !!.

SOW

→ Project Management.

WORK

Apart from code, other
many documents go hand over
kone honge :-

- ① SRC document
- ② Design Document
- ③ Testing Document
- ④ Product guideline document
Procedural or

feasible study

↳ means sol'n u r looking for is technically
feasible to achieve or not ??

Idea u r thinking ← is it possible with ur
skills or datasets u r having or
with technology you are using.

① choice of Right Model :-

→ depends on resources u r going to use.

Software fails

when some resources na ho yaa calculation
glt kii thi,
maintenance ke liye & hi nahi hain !

WOMK
= = =
WMO
= = =

1 Feb 2024

Thursday

Lec-4

- critical thinking very important.

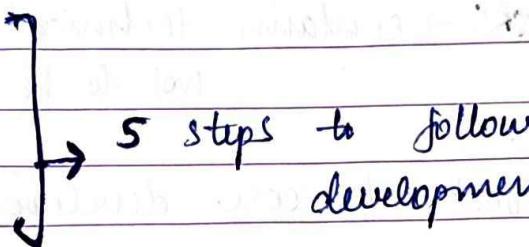
Requirement

Design

Construction

Deployment

Maintenance



5 steps to follow in SDLC development.

- Q. How to build good SRS? What team expect from this SRS?
who is going to be user of SRS? How they should use SRS?

How this document will be used by design, testing, stakeholder, deployment, maintenance team?

A. [Software Requirement Specification Document - By Krazy Tech]

- IEEE template for SRS.

Introduction section: what company, business, stakeholders are?
(ska brief, purpose of company).

document convention section: aligned to company
(can use naming convention of document).

intended audience & reading suggestions

① who will read, how they should read.

② security concern ← stolen up na ho jaaye by
hackers

so mark as internal document

means freely share na kro yeh
document, users ko yeh
pta cheta!

WDMR
= 50
= 100

- agar public \rightarrow joh webserver par must be available.
- agar Top secret

kisi ke saath no share.

SRS \rightarrow contains technical info of company,
not to be shared with hackers.

SRS \rightarrow orange book for core developers!

How to define boundaries of software??

by its features & function
 \rightarrow elaborated in : "system features"

Input date ham \leftarrow Stakeholders

Scenario based scoping can be used.

① SRS - will be cross referred with many other documents in company.

refer with "Password Policy"

detailed description of password :-

① kitne din baad bدلنا?

② kya - 2 nahi ekh skte in password

to
avoid
dictionary/
brute force
attack.

WORK
IS
M

hundred of major cities
means web based solution

① good way of defining products → modelling

can use Class Diagrams,

as well as ER diagram.

• Scenario based model

↳ Use case

↳ variables / attributes value blocks

↳ good for testing team to create test cases &
- test criteria

• Behavioral model

↑ ↑

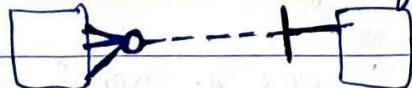
↳ defines each step as representation in modelling
state diagram (to define states)

↳ but direction of flow of info may be

uni / bi directional

So

② flow models :- arrows to define link b/w &.



• How to test robustness of any use case?

→ after 15 days, password should be changed.

→ User Experience very important parameter in testing & for
good software.

• Swimlane Diagrams

WORK

• Troubleshooting / maintenance team refers any document to update from ~~any~~ to protect it from glitch

2.4 OPERATING ENVIRONMENT

↑
we'll talk of dependencies,

① Localised constraints? → eg: • heap allocation in OS
→ processing street of User & Hardware constraint).

② System features:-

requirements

↓ Non-functional

functional

(wo features of software,
jо software ke paas
ham!)
eg: ticket booking

(~~too~~ features
of system)
eg. authentication,
SQL - injection, to
overcome penel. ke liye
Input validation.

Requirements Engineering :- (Stages)

↑
7 distinct stages in SRS & requirement engineering.

→ list of questions must be asked by team making SRS.
(first slide)

• Early the defect is diagnosed, better is the ~~cost~~
money required for resources to be
used in systems/ software ??

WJMk
30
MM

study done to ensure that
Idea \rightarrow desired output dega
gyada acche way
se yaa nahi!

① 4 types of Feasibility :-

(1) Technical

(Technology readiness index)

ML | AI

(2) Economic

↳ measure cost benefit ratio

Itna
kyu
koi?

(3) Schedule

↳ Idea to measure the readiness /
completion of project.

kyu
resources

(e.g. Mass man
mission)

(4) Operational

available,
processing cost etc,

wgira thik karn?

↳ totally dependent on external
factors!

↳ technology you are talking of,
kya law has embedded it
or not?

WORK 50%

- Component based model. → taking components from other company.

8 Feb, 2023

Thursday

[lec-5] :-

risk → technical error, platform change,
↓
effect software quality.

. Planning v/s management
↓ pthle yeh fu yeh → manage / avoid risk & errors.
deciding budget, interface,
theme of product etc.
(everything)

① SDLC: phases:-



WORK ^C _{MON.}

• Planning is present in every phase of SDLC.

B-version

↑
For testing purpose, ke cige software den to some people →

they will give feedback → agar eise
cige → so usko avoid ke company

↑
This comes Under Maintenance.

• Software project management

aimed to ensure that software is delivered on time,
with budget & schedule constraints to satisfies client
req.

(go decide kya
during planning
phase)

→ This management different from others as →

- ① software. not tangible
- ② all process are relatively new & still "under trial"
- ③ larger software projects are usually "one-off" projects

④ computer technology evolves very rapidly.

(task-specific)

can't adapt same software
for different purpose, u need to modify.

WJMK
= 5
= 0 0

- [ChatGPT → Generative AI].
[BharatGPT → to create Images by description].

ethical considerations
natu holi,
iske content
men

• Management Activities :-

- 1) Writing Proposals
- 2) Planning project
- 3) Scheduling "
- 4) Estimating " cost
- 5) Monitoring & reviewing project's progress
- 6) Selecting, hiring, evaluating personnel
- 7) Writing reports & giving presentation.

what will be funding, kitne modules?, har module ko kaise bnaayoge? kitna budget chahiye? team mein kitne log chahiye?

(Project leader will manage it)

bisse experience wale
Chahiye?

• Project plan (Should be ~~start~~ started by beginning of project)

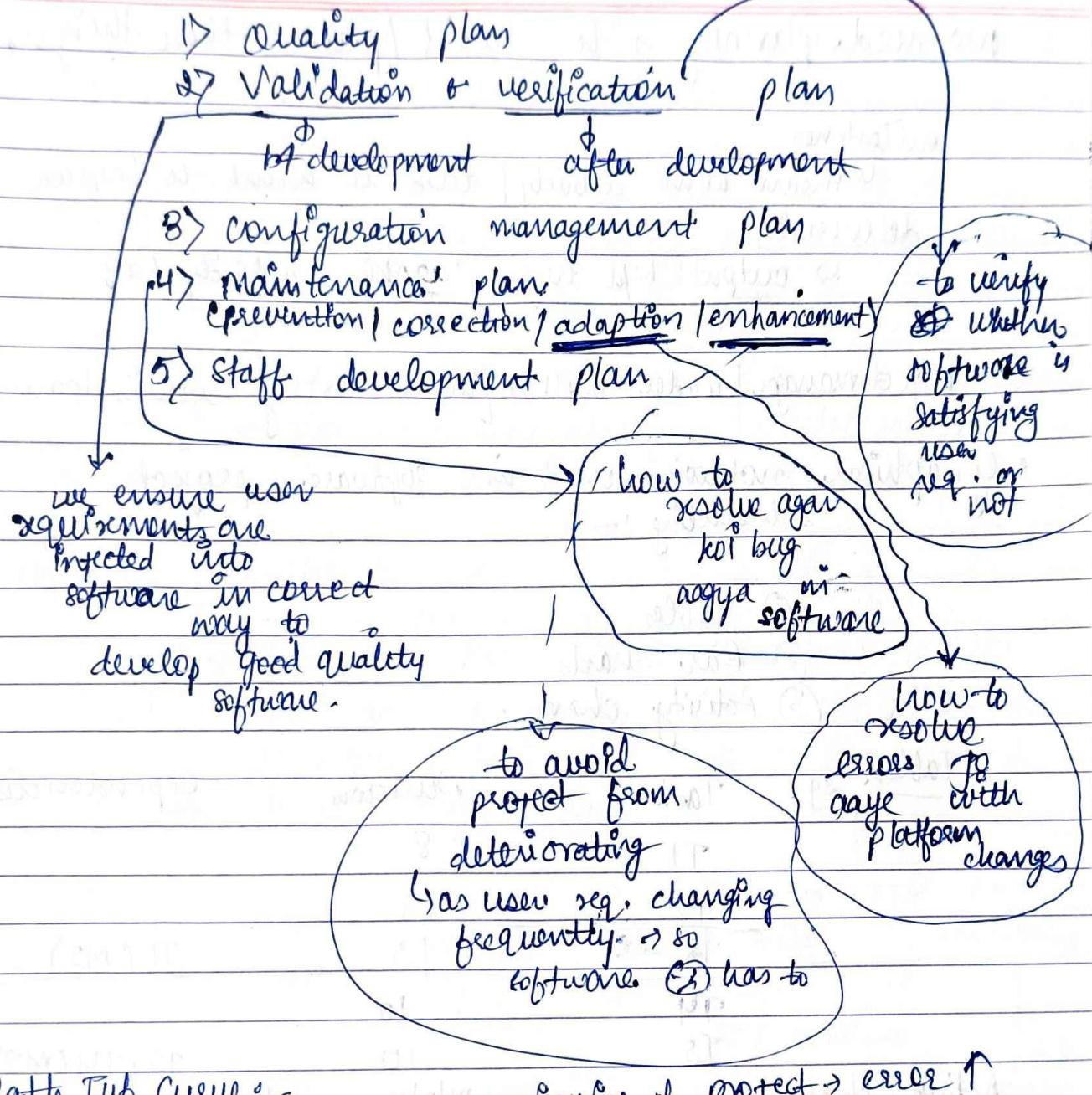
↑
should be drawn at start of project

it defines project & needs to be continuously adjusted.

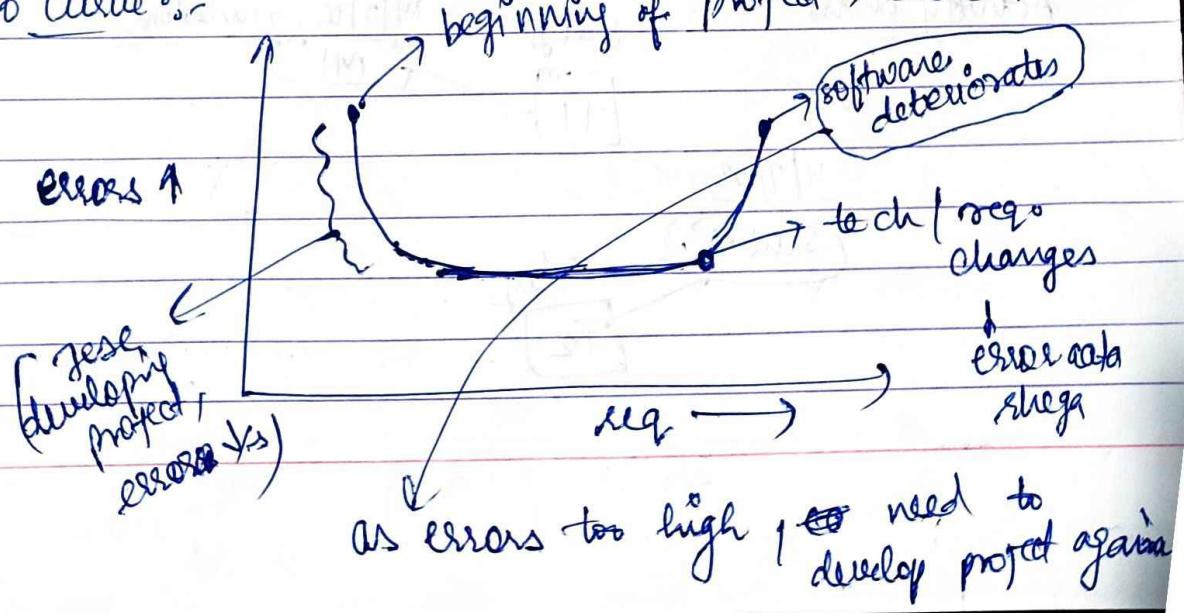
• role of project manager is to anticipate possible problems & be prepared with sol's for these problems.

Other plans to be developed :-

WAMK
3MM



Bath Tub Curve :-



WJ MK $\frac{C}{35}$
 $\underline{\underline{M}}$

we need planning \rightarrow to avoid / prevent these things.

milestones

- ↳ means konki activity / task u want to perform deliverable,
- ↳ output kya hoga aapki activity ka?

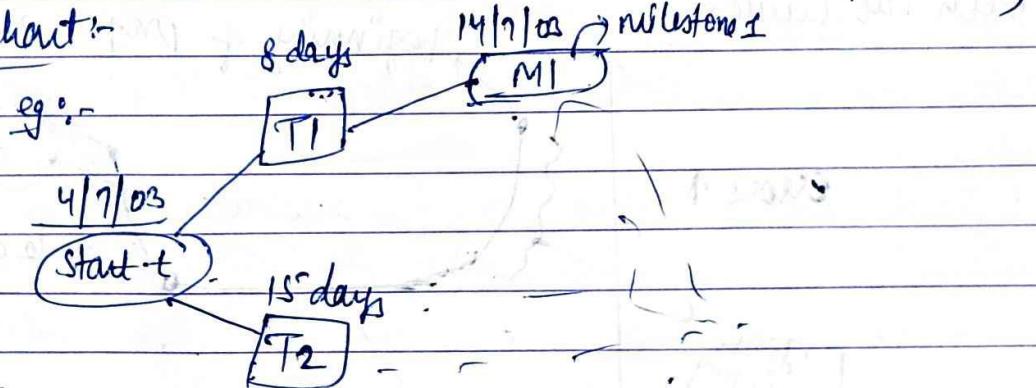
- ② Project manager / leader will define hierarchy within team.
- Graphical notations used in software project scheduling :-

- ① Tables
- ② Bar charts
- ③ Activity charts.

Tables: eg:

Task	Duration	Dependencies
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2, T4 (M2)

Activity chart:-



WJMK
= 35
= CM.

(
α - testing
β - testing
black box testing
)

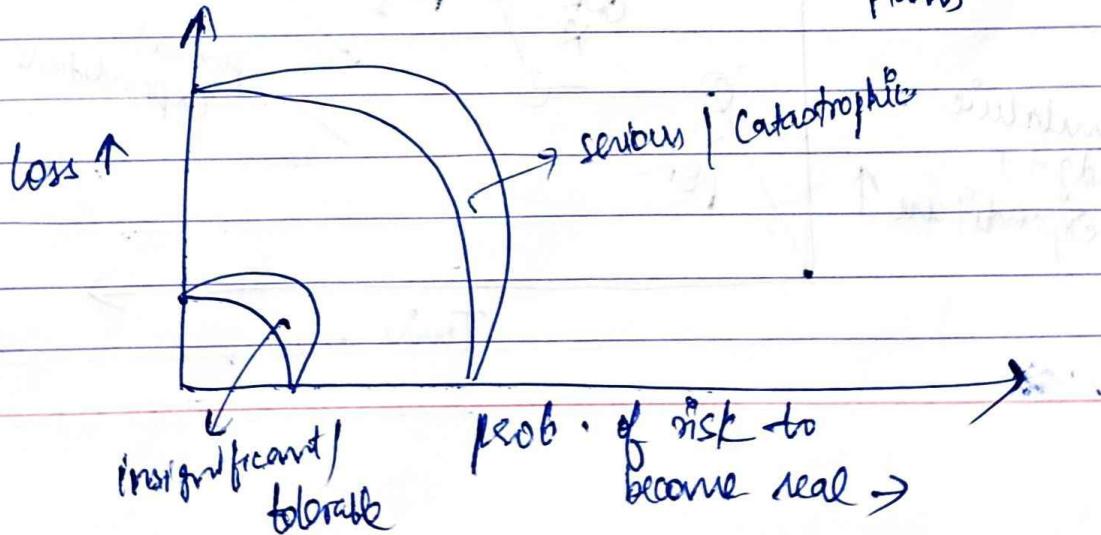
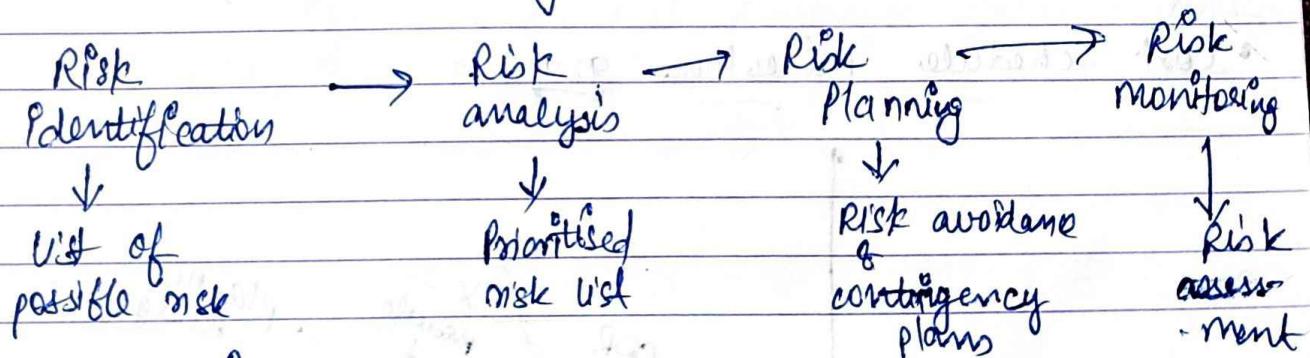
Bar chart:

① Risk → unusual event that negatively effects product, software & business.

- ① Project risk → threaten project to deliver it on time & within budget
- ② Product risk → threaten quality & timeliness
- ③ Business risk → threaten buy ability of project.

Hardware Unavailability

jis hardware par software bnaaya, maanto
kuch time bad upgrade hogya wo hardware,
then software install kene mera user → compati-
bility issues may arise.



WJMK
= 3
= 3
=

① Risk Planning :- prepare strategy to deal with each of risks Identified.

strategies

Avoidance

Minimization

contingency

agari thi kya nahi
ke paa she risk,
toh kya keein?

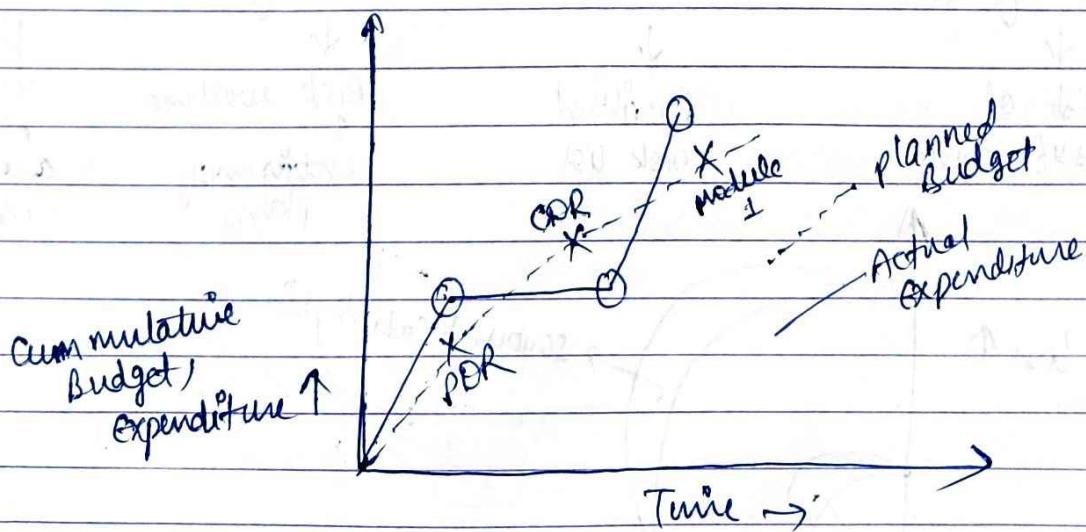
② Risk Monitoring

- ↳ frequently re-assess the risk.
- ↳ any change in risk ranking?

• Project Monitoring Plans :-

documents that can guide & must be executed.

• Cost Schedule, Milestone graph &

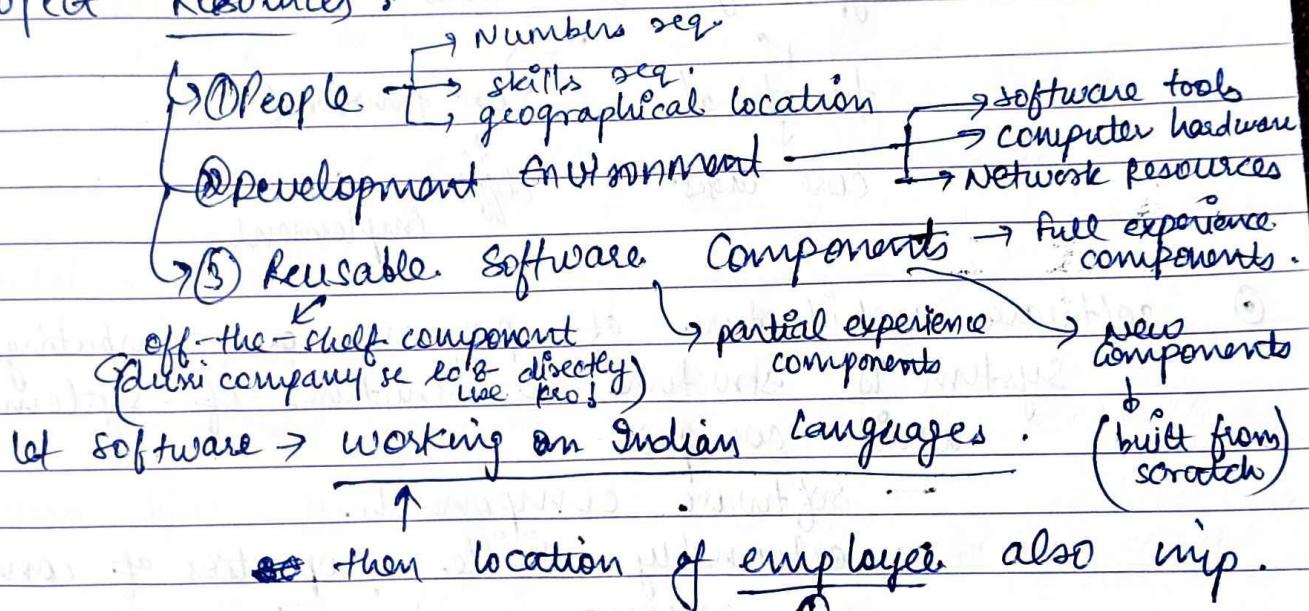


WMB

• OBSERVATIONS ON ESTIMATION :-

→ final pt analysis

• Project Resources :-



has language wale employee haun & unko
acche se aati hoo!

(Conflicting
req. w/o
so can't be
achieved ex
saatthi!)

WJMK =
 Σ MM.

Non-functional

↳ very difficult to
deliver
eg: flexibility, reusability,
security etc.

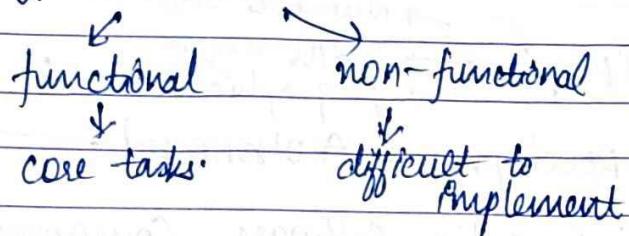
15/feb/2024

Thursday

[lec-6] :-

Software architecture & design :-

- ① SDLC, many software models, extreme programming, SRS, types of requirements



- ② software architecture of program or computing system is structure or structures of system which comprise

- software components.
- externally visible properties of components
- \approx^n among components

- ③ software architecture encompasses the set of significant decisions about organization of software system,

- ④ Architecture → not single structure -- no single structure

↳ is "collection of entity bhi ho skti hai, single entity bhi ho skti hai!!"

- ⑤ When to think about architecture ??

↑
after SRS document, have to think.

W.M.K. \rightarrow M.M.O.

acc to SRS \rightarrow subjected to changes (naturally)
 if choosing correct architecture

to purpose fulfil
 koga, no extra headache
 to take in future.

against requirements
 pchle pta hui nahi

\hookrightarrow toh architecture agar fir kya \rightarrow then
 extra efforts bad mein honge!

Distributed architecture (Scalability)

many hardwares connected together, incorporate as many users as possible, \rightarrow behavioural / utilization aspect.

- Difference b/w architecture & Design ??
- Architecture
- \rightarrow Hardest to change
 - \rightarrow most critical to get right
 - \rightarrow static properties
 - \rightarrow represents set of earliest design decisions
- Design
- \rightarrow only entry & exit
 - \rightarrow (status hair, lift hair)
 - \rightarrow (5 floors, cage low rise, peach high rise)
 - \rightarrow works on based of architecture
 - \rightarrow can be revised / modified / dynamic / behavioural property of systems.
- (lift 3rd floor onward works only!)

⑥ so changes aata hain, so how to chose correct architecture at very beginning???

\hookrightarrow as requirements always evolving

so it is very difficult to

stick to one architecture positively \rightarrow so most critical decision.

W3Mk²
To MM.

→ Attribute is first design artifact where a system's quality attributes are addressed

System Quality Attribute :-

can be addressed early after correct architecture we are taking.

from end user's pt. of view :-
(they want :-)

→ Performance

→ Availability → expect website 24x7 → but, nothing available 100%
→ Usability

→ Security

① Developers view :-

→ Maintainability

→ Portability

→ Reusability

→ Testability (Imp.)

IRFC

4 hours
down period in
midnight
data !!

② Business community view / organizational view

→ Time to Market

→ Cost & benefits

→ projected life time

→ Targeted market

→ Integration with legacy system

→ Roll back schedule.

Project
charlie
bis !!)

eg. 1 sec. time,
then good.

test here → agar har page appearing within {1s → then
(good) software !!})

↳ It gives focussed object

eg. responsive website charlie

↳ tab focus nulla !

① Architecture Drives :-

- Architecture not only serves as blueprint for system but also project :-
- Team structure
- Document organization
- Work breakdown structure
- Scheduling, planning

non-functional requirements ← → architecture

functional requirements ← → design

② Typical descriptions of software Architectures :-

- client-server model
- distributed, object-oriented approach (many machines are connected)
- { → pipeline
- { → general network architecture
- { doing step by step // phase by phase.
 ↳ ek step ka output, dusre ka input
 peer to peer
- ↳ network kind management,
 everyone prolated & doing their job.

③ Architecture design level of Software ?? ?

have total
view

$$WDMk = \frac{50}{mm} =$$

wave focused
concern.

Architecture

→ Interactions among
parts

→ structural properties

→ declarative

→ mostly static

→ system-level performance

→ outside module
boundary

→ composition of
subsystems

Programs

→ Implementations of
parts

→ computational

→ operational

→ dynamic

→ algorithmic

→ Inside module
boundary

→ copy code or
call libraries

both are equally
important for
software

① Benefits of Architectural modelling :-

- ① clarify design intentions
- ② provide basis for analysis in design
- ③ Improve maintenance
- ④ Address hard questions
- ⑤ permits system-level analysis

Correct architecture → so yeh sahi

Correct design

Code

Test

Maintenance

life easy.

WJMK
= 300

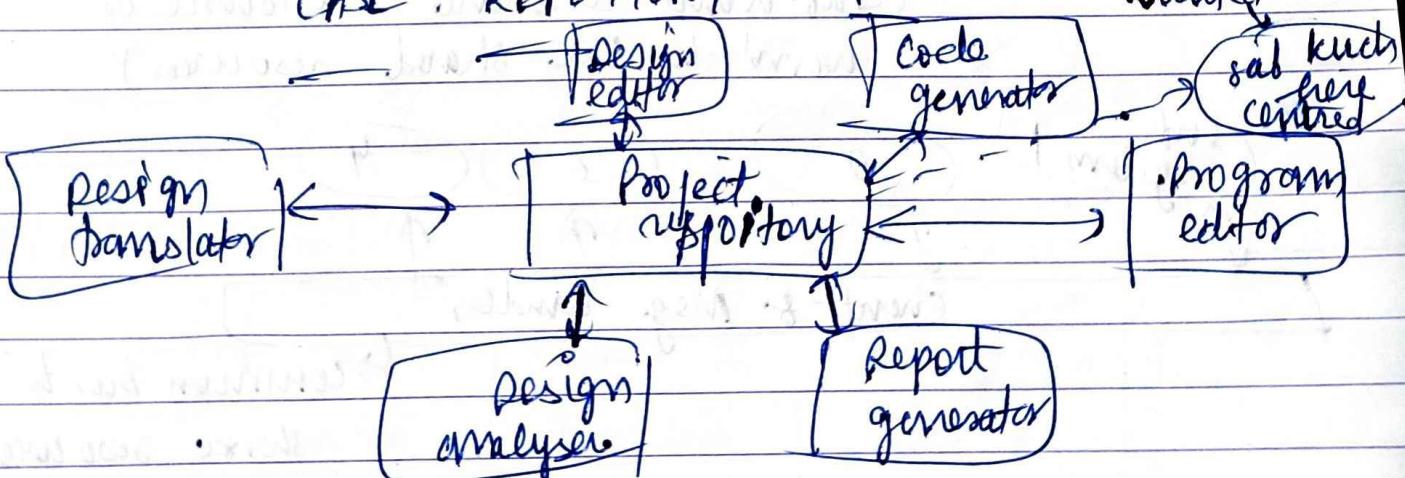
① Architectural structures :-

- ① components → computational elements
e.g. various servers, controllers
- ② ports → Interface points for components.
- ③ connectors
- ④ roles
- ⑤ systems

② Architectural Design Process :-

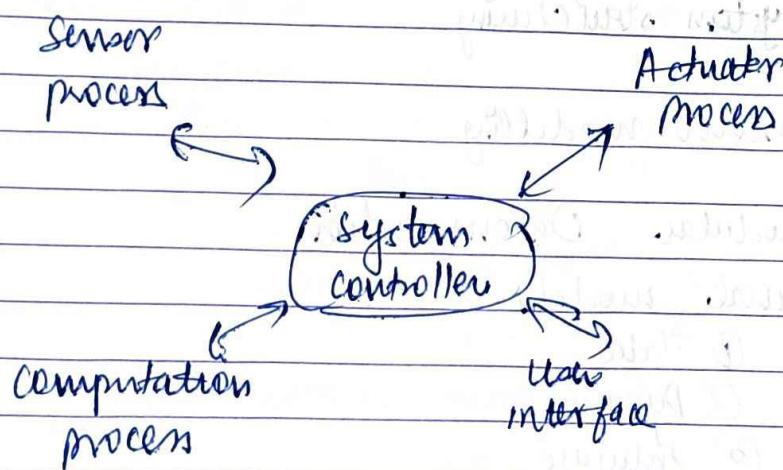
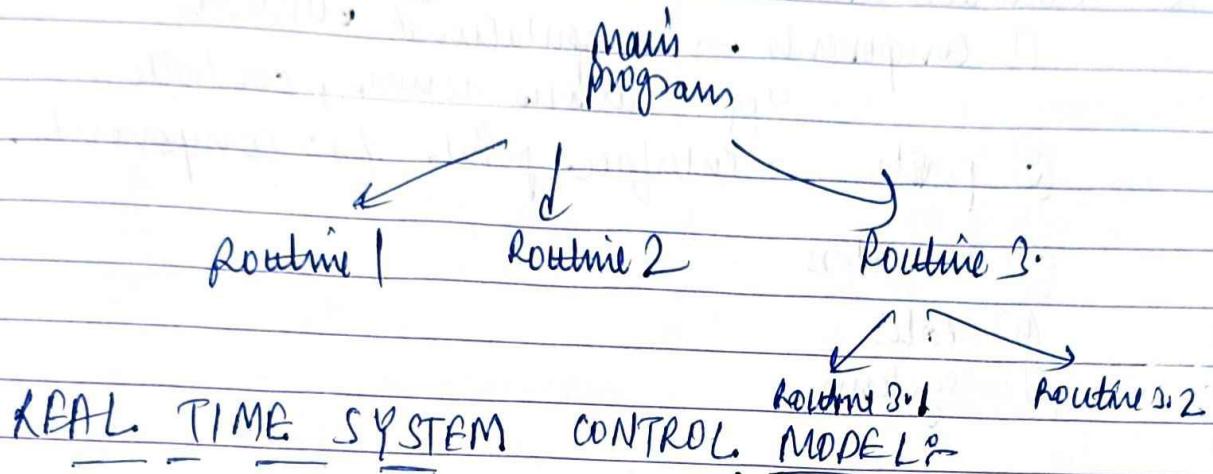
- ① System structuring
 - ② control modelling
 - ③ Modular Decomposition
- ## ④ Architectural models
- ① static
 - ② dynamic
 - ③ interface
 - ④ relationships

CASE . REPOSITORY MODEL :- (Centralised architecture)



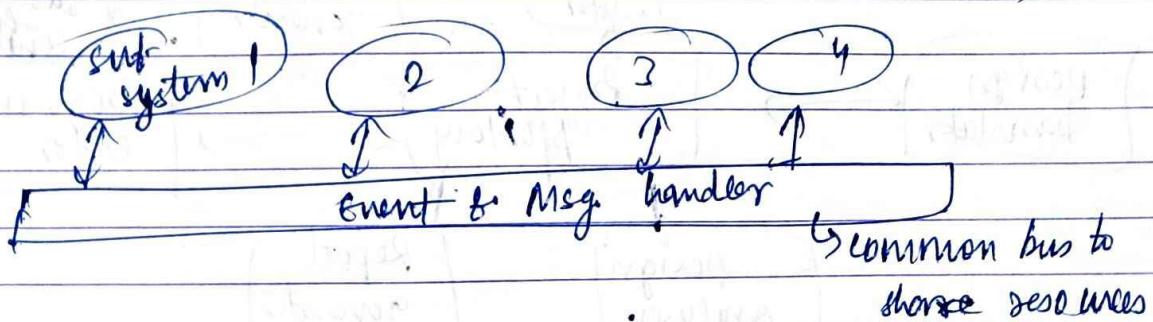
WDMK
= 30
= MM.

CALL RETURN MODEL :- (Hierarchical architecture)



SELECTIVE BROADCASTING MODEL :-

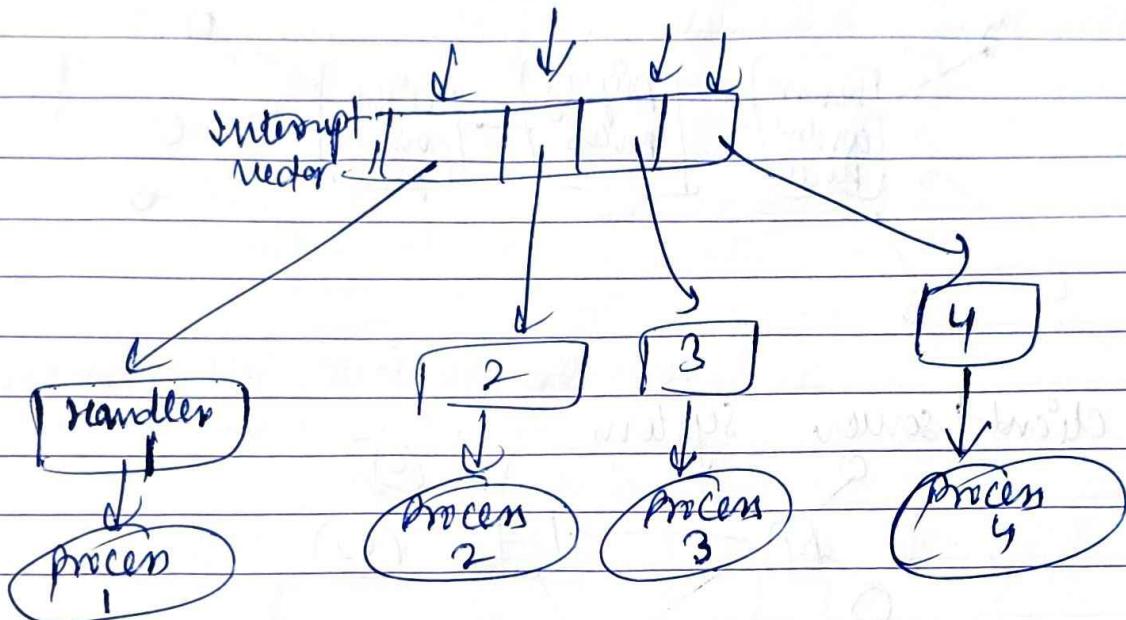
(2 ab want to share resource or want to use shared resources)



W~~ORK~~
IN
~~PROGRESS~~

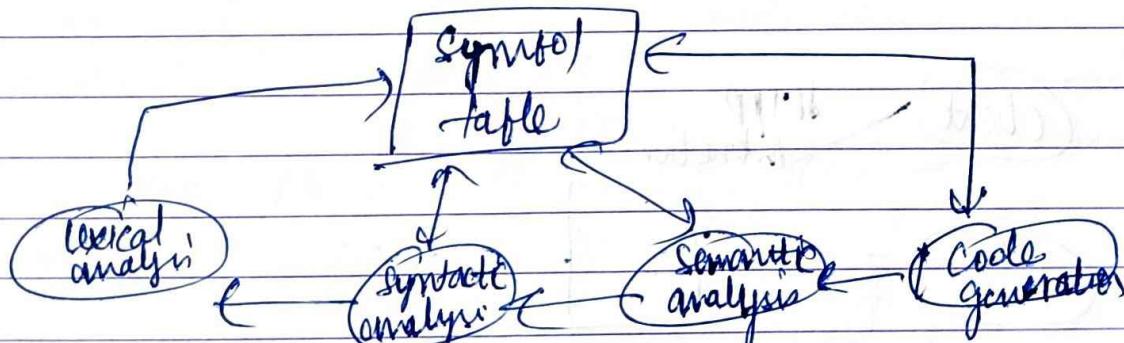
INTERRUPT-DRIVEN CONTROL MODEL → (Hierarchical architecture)

Interrupts

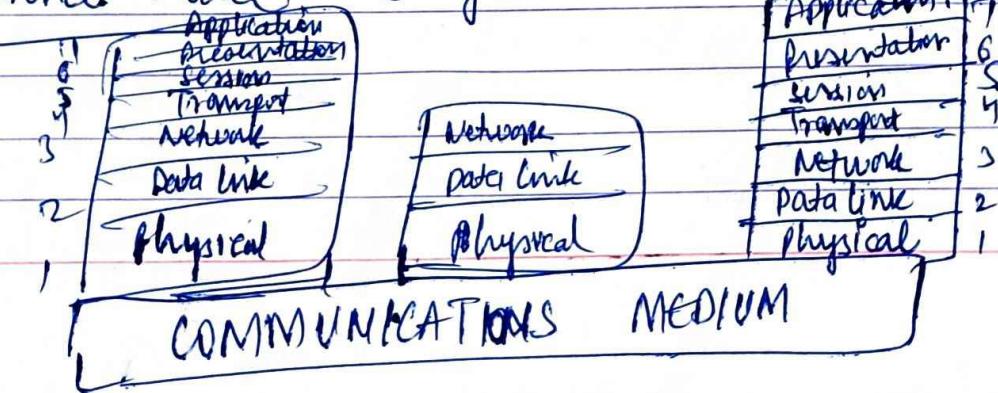


→ very much possible to scale

Compiler Model (Pipe & filter architecture)



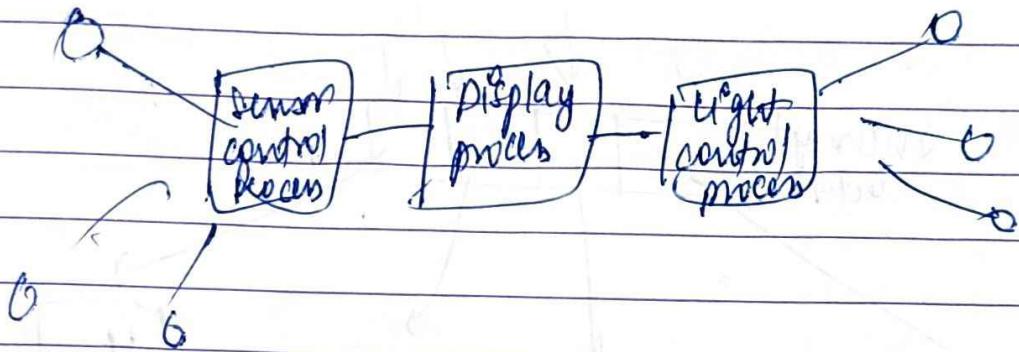
OSI reference model → (layered architecture)



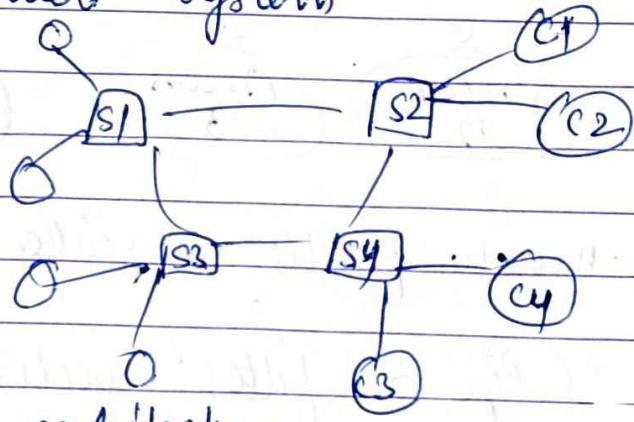
W3MK
= 50
= 00.

① distributed system

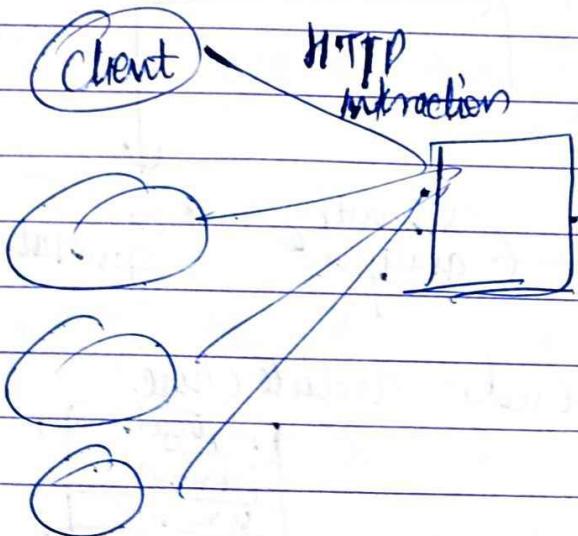
② ~~distributed~~ Multiprocessor Traffic control system :-



③ client / server system



④ Three-tier architecture

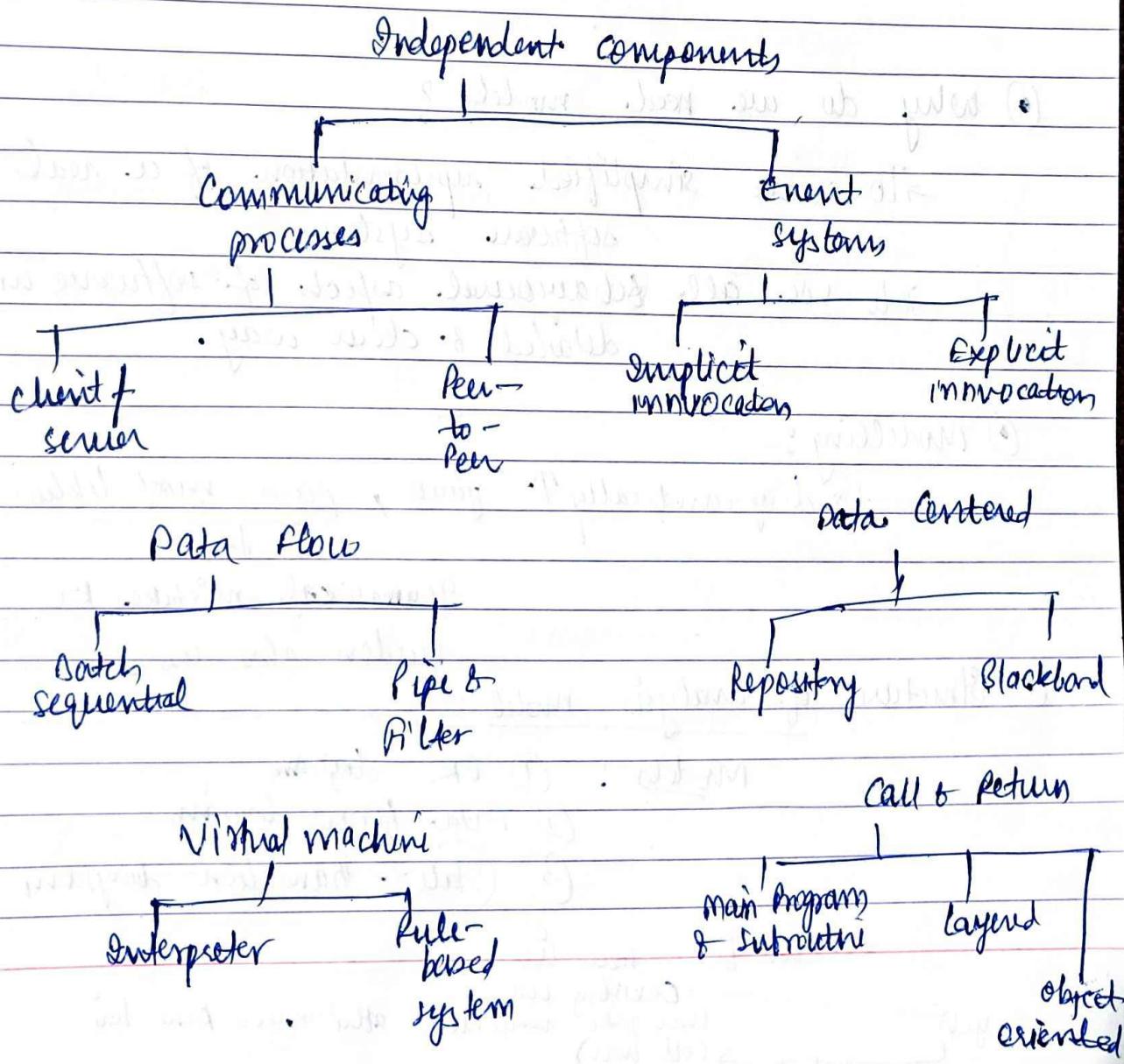


W.M.K
= P.M.

① Distributed object architecture from Sommerville :-



② Taxonomy of architectural styles:-



WJMK
= 50
= 00

① Types of architecture :-

- ① Business
- ② Technical
- ③ Solutions
- ④ Enterprise
- ⑤ product line

↓
how your
product will look
like

→ how we're facilitating the
requirements,
giving alternate
ways if
problem nahi
koi.

① why do we need models ?

→ To create simplified representation of a real software system.

→ to see all behavioural aspects of software in detailed & clear way.

① Modelling :-

↳ diagrammatically → ghaao, para mat likho.

grammatical mistakes ka
burden also less.

① Structure of analysis model :-

- Models :-
- ① ER diagram
 - ② Data flows diagram
 - ③ State - transition diagram

can be new user,
existing user,
user info membership atu° review keri hui
user (old user)
get on state with time

WAMK
C
S
M.

Data flow diagram

bstaat hai kitne vaar table / database ko
update kena hogा.

① Unified Modelling Language

Dynamic
View
↓
Sequence

9 of diagrams :-
UML

static View

Use Case

Collaboration

Activity

Statechart

Model

class

object

component

deployment

kyi vaar also
depicts Dynamic
View.

Use Case:-

actor | Relationship

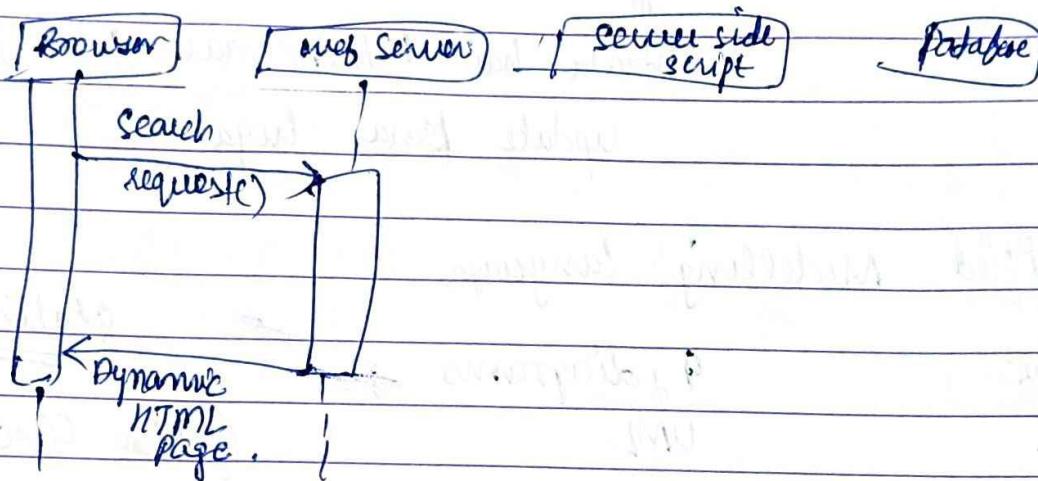
Use case

systems

object

WJM&

Sequence Diagram :-



Collaboration Diagram ?? ?

* class Diagram bnaana hi. bnaana hoi!

→ It depends on criteria, aapko chahiye kya diagram
se → uspe depend keta kaise diagram you
want to make. !!