TOPIC NAME : ____________ DAY : ____________
TIME : DATE : / /

Software Architecture

+ How a system should be designed

How components are
- organized
- communicate
- constraints

+ input of dev phase of SDLC

Advantage :

1. communication bridge between stakeholders and dev team.

2. Easy analysis of non functional requirements

3. Reuse the diagram in diffrent model.

Design :

1. components — decompose into subcomponents

Login — web login
— phone sms login
— code

2. structure of system

3. Distribution across diffrent hardware
   - └ server locations
   - └ device compatibility

4. strategy of operations
   + communication of components

5. structure for non func requirements
   - [bank app - security
   - └ game - fast load priority

(4) Views

Each diagram shows one view

1 diagram can't show multiple views

1. Logical view :

How classes interact

↳ class diagram

2. process view : (Activity diagram)

Flow chart (process activity)

3. Developmnt view : (component diagram)

How system is decomposed into components

4. physical view : - Deploy software

- Hardware comp distribution

connecting pcs

TOPIC NAME : ______ DAY : ______
TIME : DATE : / /

Layered Architecture

System layers :

1. presentation : [UI
   screen, inputs

2. Busness : Data validation
   calculations

3. Persistance : - R/W data
   - SQL

5. Database : [location
   structure > database

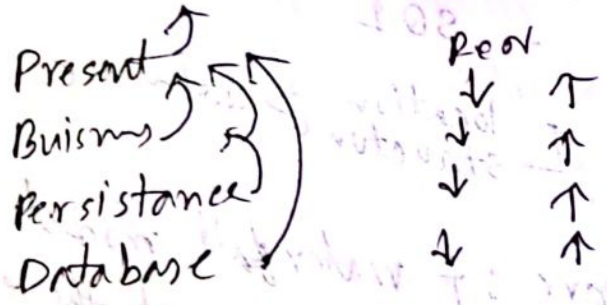
Monolithic software : - waterfall model
- end user at the end gets it
- No seperation comp

problem: - No seperation comp so changing one affect another

2.

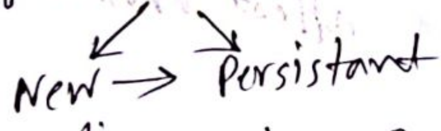
Layered Archi

- Organize system into seperate layers
- seperate changes
- Diffrent layer diffrent functions
- A layers can provide service to the layer above it.



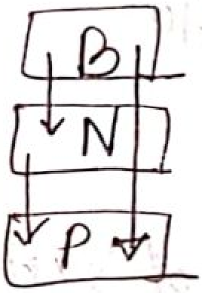
closed layer : Only the above layer can access it. Isolated.

Open layer : If buisness sometims go to



new and sometims to persistant.

:- New layer : open (can be bypassed)



Use : ① New func on a exiting system

② When dev team is seperated

③ multilevel security needed.

Advantage :

① Easy add / remove layers

② Testing is easy

Disadv :

① can't access directly one layer to another

② Performance reduce as many layers pass each time.

③ Restart app after each change.

data distributed to diff subsystems

TOPIC NAME :

DAY :

TIME : DATE : / /

Repository

- storage → distributed sub system data
  - shared ~~Ads~~ data in a central repo → accesable by all subsystem
  - accesable to all components of system
  - components interact through repo

Use :

1. large data generated
2. Store data for long time.

Adv :

1. Components can be independent
2. Manage all data consistently at the same time
3. changes can be shared to all componts

Dis :

1. Single point failure (all at one place)
2. Distributing to several computers is difficult.

→ one computer fail - problem

Client Server Pattern

server

- Provide service to client

clients

- reqr service to server

ex: I reqr a page ingoogle
- gogle server provides me page

Use : ex : email, doc share, banking, video library

1. Data has to be accessed from diff locations
2. server can be cloned to acomodate more people

Advantage :

1. server can be across a network accessable
2. general service accessable to all client

Dis :

1. Performance depend on network
2. server owned by diff organization - prob
3. single point failure - each server .

Client Server Pattern

server

- Provide service to client

clients

- reqr service to server

ex: I req a page ingoogle

- gogle server provides me page

Use : ex: email, doc share, banking, video library

1. Data has to be accessed from diff locations
2. server can be cloned to acomodate more people

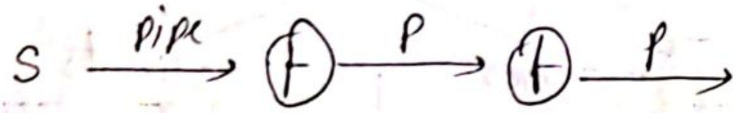
Advantage :

1. server can be across a network accessable
2. general service accessable to all client

Dis :

1. Performance depend on network
2. server owned by diff organization - prob
3. single point failure - each server.

Pipe filter pattern



Data is passed thru pipes

Data is processed by filter component

+ Use when process a stream of data

+ each filter is descreate, carries one type of data

~~Users~~:

Adv: ① easy to understand flow
② workflow similar to business process
③ straightforward ④ sequential or concurrent (parallel)

Dis ① format of data has to be set same for all comps.
② each transformation has to check for format/ data structure.

TOPIC NAME : ______ DAY : ______

TIME : DATE : / /

Ans to Q no C

Since for the longerity of the product codebase has to be well documented. I'll use waterfall model. Waterfall model is a linear sequential model and it is well documented. Also I would like to keep scrum at the same time combining with waterfall. After each sprint I can have my team document the progress of project. In waterfall even though it is well documented. the requirements are fixed and not changable so ~~It~~ i would like to combine scrum with it.

TOPIC NAME : DAY:
TIME : DATE : / /

Ans to Qr 3(A)

The software pattern i'll use is client - server pattern. Because here a server is reqwired to process this remote real time gameplay.

Advantage : server can be accessed acrossed a network.

Disadvantage : The performance depends on network. Since diffrent organization can own the servers sometimes it's not trustable.

Ans to 3 (B)

Monolothic software atchitecture is seqruential and not capable of handeling complex projects like this one For example, this project might need diffrent components independent of each other to develop proprly but monolothic architecture makes it all connected so it becomes single point failure.