Date _____ Day _____

WEEK # 05

▷ WIDE BAND DELPHI TECHNIQUE

"A process that a team can use to generate an estimate"

- ROLES
  - Estimation Team: Project Manager chooses an estimation team that include reps from all project areas (manager, developers, QA, UI/UX etc)
    - ↳ Every member should have stake in plan
    - ↳ should understand Delphi process
  - Moderator: someone who understand Delphi Process but has no stake in results
  - Observers: Selected stakeholders or users
    - ↳
- STEPS FOR DELPHI PROCESS

(1) Choose the Team:

→ The PM selects the estimation team and a moderator.

→ Should consist of 3-7 members

Date _____ Day _____

(2) Kickoff Meeting:
→ Ensure all team members understand the Delphi Process
→ Create WBS with 10-20 tasks
→ Agree on a unit of estimation (eg: hours, days, weeks)

(3) Individual Preparation:
→ Each member independently estimates effort for all WBS tasks
→ Identify subtasks if needed for clarity
→ Document effort estimates, assumptions & any missed tasks.

(4) Estimation Session:
→ Team discusses & refines estimates for each WBS task
→ Each member sumbits individual estimates
→ The moderator collects & plots total effort estimates
→ Disagreements are resolved through assumption clarification
→ Multiple rounds continue until consensus is reached

Date _____ Day _____

5. Assemble Tasks :
   → The PM collects final estimates from the team
   → Compiles the final task list, estimates & assumptions.

6. Review Result :
   → The PM reviews the final tasks list with the estimation team.

▷ PROBE :

→ Proxy Based Estimating
→ PROBE is ~~based on the idea that~~ if ~~an engineer~~ an effort estimation technique used in Personal Software Process.
→ It helps developers estimate the effort and time required for software tasks based on historical data.
→ Uses prev project data to predict effort for new tasks
→ Estimates are based on size, complexity and historical performance

Date _____ Day _____

▷ [illegible] :

→ It is a software cost estimation model to predict effort, cost and time for software development based on their size.

→ Constructive Cost Model

→ A set of variables that must be provided as input for model & output of model is a set of size and effort estimates that can be developed into a project schedule.

▷ The Planning Game :

→ It is a software project planning method from XP.

→ It is highly iterative

→ It is a full planning process that combines estimation with identifying the scope of the proje and task req to complete the software.

→ See Work Break Down Structure (WBS) from Slides (Week #5)

Date _____ Day _____

WEEK # 07

"SYSTEM MODELING"

→ System modeling ~~ko~~ means representing a system using some kind of graphical notation

→ System modeling bas system/software ko explain krne k liy bnate hai.

→ It is an abstract representation of system.

SYSTEM PERSPECTIVE:

→ You may develop different models to represent the system from different perspectives. Like

- external perspective → modeling of context & environment of system
- interaction perspective → modeling of interaction b/w sys & env or b/w components of sys
- structural perspective → modeling the org of system or structure of data processed by sys
- behavioral perspective → modeling of dynamic behavior of system & how it respond to events

Date _____ Day _____

→ System modeling k liye ye mainly use hoti hai:
① Activity diag ② Use case diag ③ Sequence diag
④ Class diag ⑤ State diag

Use Of Graphical Models:

→ Agr ~~kisi~~ kisi existing system k baare mein discussion krni hai to incomplete or incorrect diagram bhi chale gi

→ Agr documentation purpose k liye krni hai to correctness hona zaroori hai. Completness not req.

→ Agr detailed description deni hai ~~to~~ (like we generate code in Papyrus) uske liye complete or correct diagram honi chahiye. Like diagram mein arrow heads or unki direction, relationship sab shi honi chahiye.

• CONTEXT MODEL:

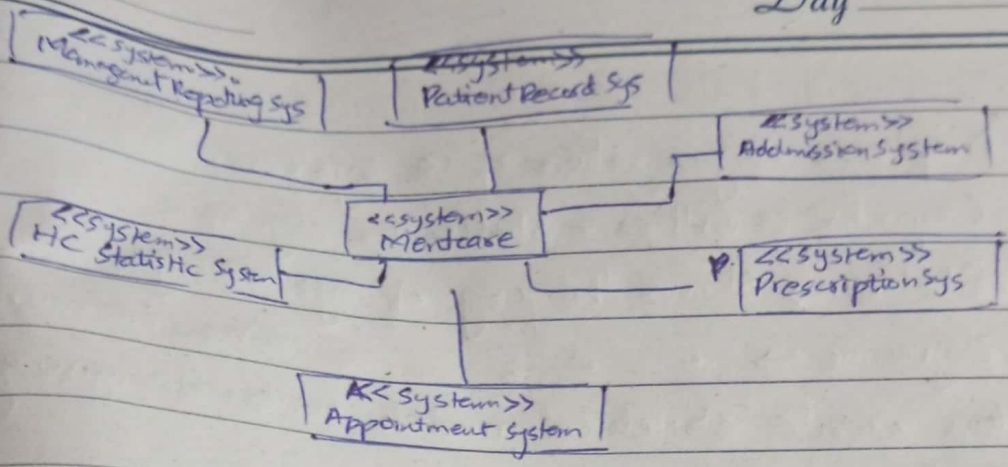
→ Context models are used to illustrate the operational context of a system

→ They show what lies outside the system boundaries

→ System boundaries are established to define what is inside & outside the sys.

→ They show other system that are used or dependent on the system being developed

Date Day



Content Model of Mentcare System

- **PROCESS MODEL:**
  - → Process model reveal how the system being developed is used in broader business processes.
  - → Activity diagram is used for this
  - → See page # 13 (Week# 7)

- **INTERACTION MODELING:**
  - → A system interaction can be:
    - ↳ User interacting with Sys
    - ↳ Two Sys interacting with each other
    - ↳ Components of a Sys ~~interfacing~~ interacting with each other
  - → ① Usecase diag ② Sequence diagrams are used for interaction modeling

Date _____ Day _____

STRUCTURAL MODELS:

→ Displays organization of a system.

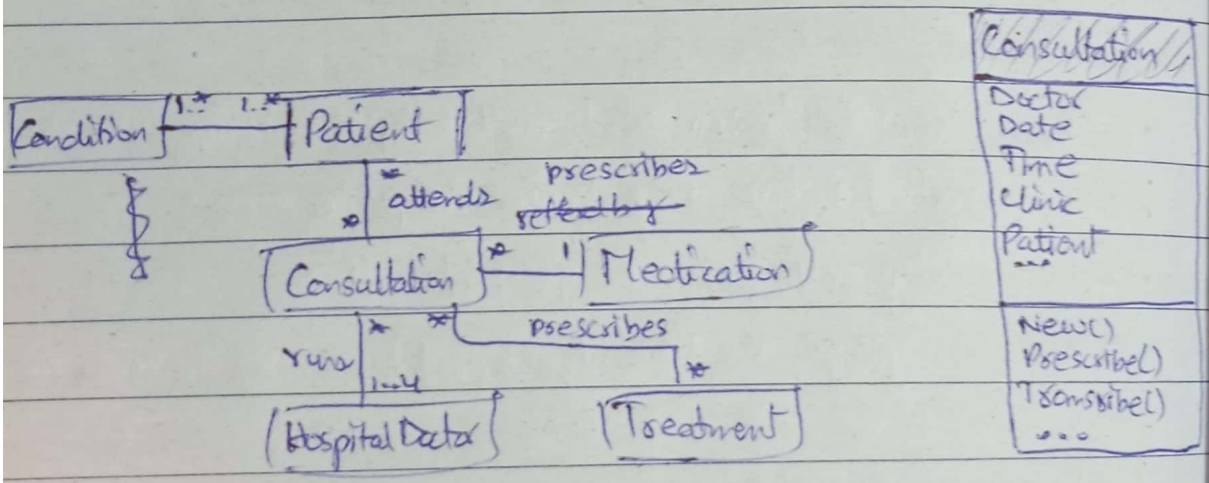
→ Can be either static or dynamic

Static → shows the org. of sys design

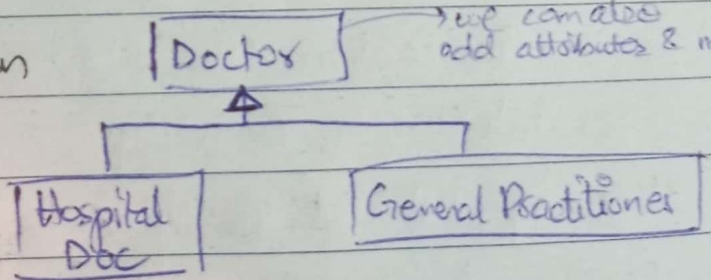
dynamic → " " " " " when executing

→ You create structural models when designing system archi.

→ Class diagrams are used for static structures



→ Generalization



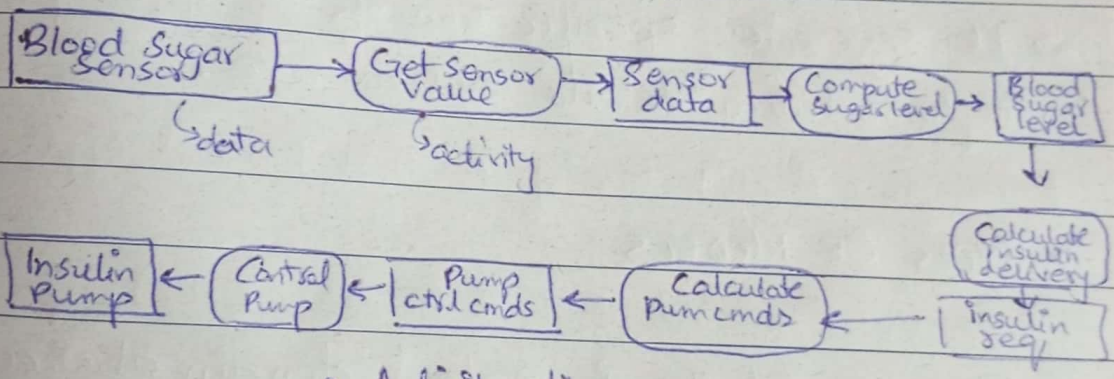
→ We can also show aggregations

Date

Day

- BEHAVIORAL MODELS:
  - → They show what happens or what is supposed to happen when a sys responds to a stimulus from its environment.
  - → Stimuli can be
    - ↳ Data
    - ↳ Event

- Data Driven Modeling:
  - ↳ Can be shown thru Data Flow Diagram (Activity diagram) or Sequence Diagram



▲ Activity diagram

- Event Driven Modeling:
  - ↳ State diagrams are used for this

Date _____ Day _____

- Model Driven Engineering (MDE)

→ It is an approach to software development in which a system is represented as a set of models that can be automatically transformed to executable code.

- Model Driven Architecture (MDA)

→ It is an approach to software design & dev that emphasizes the use of models at various stages of the development process.

→ It seperates specification of the system from implementation details such as code.

→ TYPES OF MODELS

① Computation-Independent Model (CIM)
These model the important domain abstractions used in a system

② Platform Independ Model (PIM)
These model describe the system without consideri the underlying platform

③ Platform specific Model (PSM)
These models map the PIM to a specific platfo or technology.

Page No. _____

Date _____ Day _____

WEEK #08

"DESIGN CONCEPTS & ARCHITECTURAL DESIGN"

▷ DESIGN MODELS :

→ The design model has the following layered elements

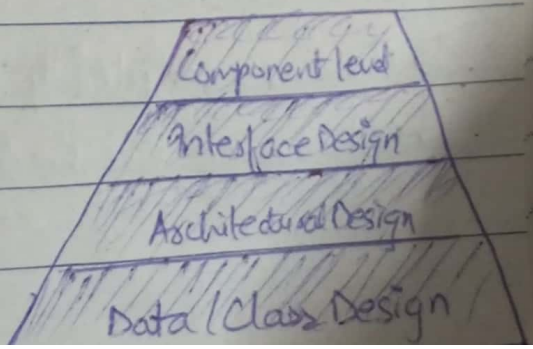
① Data / Class Design
↳ Creates a model of data and objects that is represented at a high level of abstraction

② Architectural Design
↳ Depicts the overall layout of the software

③ Interface Design
↳ Includes user interface, external interfaces and internal interfaces

④ Component Level Design
↳ Describes the internal detail of each software component by way of datastructures, algorithms and interface specifications.

⑤ Deployment-level design
↳ Indicates how software functionality & subsystems will be allocated with the physical computing env.

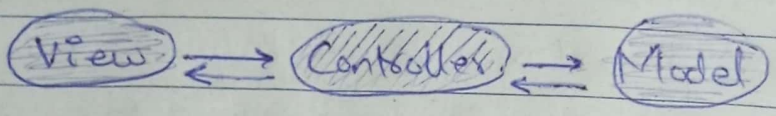


Date _____ Day _____

▷ Architectural Patterns

1. MVC Pattern

→ Model View Controller
→ Model Component manages the system data
→ View " " how data is presented to user
→ Controller " " user interaction & passes these interactions to the View & Model.



Example: Architecture of a Web based application

- Advantages:
  - ↳ Promotes modularity (Seperation of concerns)
  - ↳ Code reusability
  - ↳ Team can work on diff components simultaneously
  - ↳ MVC makes it easier to scale by adding or modifying components without affecting entire app.

- Disadvantages:
  - ↳ Increases complexity – more files & layers harder to manage for small projects
  - ↳ Performance Overhead – Extra processing due to multiple layers can impact speed

Date _____ Day _____

- Preferable For
  - ↳ Large scale Applications
  - ↳ App req multiple views – same data needs to be displayed in diff formats / users
  - ↳ Team based development
  - ↳ Web Apps

Layered Pattern

→ Used to model the interfacing of sub systems
→ Organises the system into set of layers.
→ A layer provides service to the layer above it

- Common Layers
  - Presentation Layer – handles UI & input
  - Business Logic Layer – Contains core funct. & rules
  - Data Access Layer – Manages DB interactions
  - Database Layer – Stores & retrieves data

- Advantages
  - ↳ Seperation of concerns
  - ↳ Mantainablity – easier to update or modify specific layer without affecting others
  - ↳ Reusability
  - ↳ Scalability

- Disadvantages
  - ↳ Difficult to implement
  - ↳ A higher level layer may need to interact with lowest level directly
  - ↳ Extra layers may slow down the system

Date _____ Day _____

- Prefferable For
  - ↳ Large enterprise apps needing modular structure (eg: banking, healthcare)
  - ↳ System requiring scalability & sepration of concerns

Avoid for high performance apps.

(3) Repository Architecture

→ It is a design pattern where a central data store is used for diff subsystems components to share & access data.

→ Components interact with repository rathes than communicating directly with each other.

- Advantages
  - ↳ All components access a single data src, reducing redundancy
  - ↳ Loose Coupling – Components remain indep as they only interact with repo
  - ↳ Changes to data management logic are centralized in one place
  - ↳

- Disadvantages
  - ↳ Single point of failure → repo
  - ↳ All components depend on one data src, so scaling can be difficult
  - ↳ Complex implementation

Date Day

- Prefferable For
  - ↳ Large scale app. needing a shared data model (eg: compilers, databases)
  - ↳ Software with heavy data processing like banking & enterprise app
  - ↳ AI & ML sys where multiple components needs to access to training data.

Avoid for highly distributed systems, real time apps needing fast direct data access.

4 Client - Server Architecture

→ It is an archi. where clients request service and a central server processes these request & responds accordingly

→ the client & server communicate over a network

→ Functionality of system divided into service

- Advantages
  - ↳ Scalability - More clients can be added without major changes
  - ↳ Easier Mantainance - updates & mantainance is done on server not on each client
  - ↳ Server can be distributed accross the network
  - ↳ Easier to add a new server & integrate it with other system

Date _____ Day _____

- Disadvantages
  - ↳ Each service is a single point of failure, suspectg to DoS attack
  - ↳ High traffic can slow down process
  - ↳ Managerial problems if servers are owned by diff organizations
  - ↳ Performance unpredictable as it depends on network & system.

- Prefferable For
  - ↳ Web apps
  - ↳ Enterprise apps needing centralized data managment
  - ↳ Used when data in a shared database has to be accessed from diff locations.
  - ↳ Multiplayer games
  - ↳ Cloud based systems

(5) Pipe and Filter Architecture

→ It is a model where data is processed through a series of components (filters) each performing a transformation

→ The data flows through these filters via pipes

→ Not really suitable for interactive systems

Day

- Advantages
  - ↳ Each filter is indep & reusable
  - ↳ Filters can be executed ||ly
  - ↳ Easy to understand & maintain

- Disadvantages
  - ↳ Data format must be compatible b/w filters, adding complexity
  - ↳ Sequential processing may cause delays

- Preffered For → Limited user interactions involved.
  - ↳ Data Processing system like compilers, video/audio processing or image pipelines
  - ↳ Batch processing app.
  - ↳ Unix / Linux cmd line utilities

Avoid for apps with complex data dependency & shared states.

▷ APPLICATION ARCHITECTURES

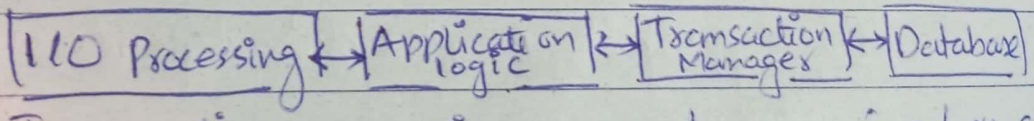
- Data Processing Applications
  - ↳ Data driven applications that process data in batches without explicit user intervention during the processing

Date _____ Day _____

- Transaction Processing Applications
  - ↳ Data centered app. that process user req & update info in a database
- Event Processing Systems
  - ↳ App. where sys actions depend on interpreting events from the system's env
- Language Processing Systems
  - ↳ Applications where the user's intentions are specified in a formal language that is processed by the sys.

(1) Transaction Processing Systems

↳ Users make asyn req for services which are then processed by a transaction manager



→ Transaction processing may be organized as a pipe & filter archi

(2) Information Systems

→ Info sys allows controlled access to large bodies of info eg such as records of patients in a hospital

→ They are almost always web based sys.

Date _____ Day _____

They are organised as a layered archi
- The user interface
- User communications
- Information retrival
- System database

→ These are often implemented as a multi tier client server archi.

③ Language Processing Systems
→ Accept a natural or artificial language as input and generate some other representation of that language.
→ Like compilers translate programing lang into machine code
→ Natural language translator
→ It is a composite of repository & pipe&filter architectural pattern

Date _____ Day _____

▷ ARCHITECTURAL VIEWS :

- Logical View : which shows key abstraction in the system as objects or object classes.
- Process View : which shows, how at runtime the system is composed of interacting process
- Development View : which shows how the software is decomposed for development
- Physical View : which shows system hardware & how software components are distributed accross the processors in the system.