

06/10/22

ASSIGNMENT-3

Q.1 What do you mean by software design? Explain its objectives, principles and design concepts.

- Ans:-
- o Software Design is a process to plan or convert the software requirements into a step that are needed to be carried out to develop a software system.
 - o Software design is an iterative process through which requirements are translated into a - blueprint for constructing a the software.
 - o The blueprint depicts a holistic view of software.

- OBJECTIVES OF SOFTWARE DESIGN
- 1. CORRECTNESS: Software design should be correct as per requirement.
- 2. COMPLETENESS: The design should have all components like data structures, modules & external interfaces, etc.
- 3. Efficiency: Resources should be used efficiently by the program.
- 4. Flexibility: Able to modify on changing needs.
- 5. Consistency: There should not be any inconsistency in the design.
- 6. Maintainability: The design should be simple so that it can be easily maintainable by other designers.

WOW!

O PRINCIPLES OF SOFTWARE DESIGN

Software Designs in which these principles are applied affect the content & the working process of the software from the beginning.

1. Should not suffer from "Tunnel Vision".
→ which means that it should not only focus on completing or achieving the aim but on other effects also
2. Traceable to analysis model.
→ which means it should satisfy all the requirements that software requires to develop a high-quality product.
3. Should not "Reinvent The Wheel"
→ which means it should not waste time or efforts in creating things that already exist.

WOW!

4. Minimize Intellectual Distance
→ The design should reduce the gap between real-world problems & software solutions for that problem.

5. Exhibit uniformity and integrity.

→ Uniformity: uniform throughout the process without any change.
Integration: mix or combine all parts of software.

6. Accommodate Change

→ Software should adjust to the change that is required to be done as per as the user's need.

7. Robust

→ it should work properly even if an error occurs during the execution.

8. Assessed or Quality
- During the evaluation, the quality of the redesign needs to be checked & focused on
9. Review to discover errors
- The design should be reviewed and check if there is an error present or if it can be minimized

10.

o SOFTWARE DESIGN CONCEPTS

1. Abstraction: is the process or result of generalization by reducing the information content of a concept, in order to retain only information which is relevant for a purpose.

2. Refinement: is a process of elaboration. A hierarchy is developed by decomposing a **now!**

macroscopic statement of function
in a step-wise fashion
until programming language
statements are reached.

3. Modularity: Software architecture is divided into components called modules.
4. Software Architecture: refers to overall structure of the software & the ways in which that structure provides conceptual integrity for a system.
5. Control Hierarchy: A program structure that represents the organization of a program component & implies a hierarchy of control.
6. Structure Partitioning: The program can be divided into both horizontally or vertically.

WOW!

6. Data Structure: It is a representation of the logical relationship among individual
7. Software Procedure: It focuses on the processing of each module individually.
8. Information Hiding: Modules should be specified & designed so that information contained within a module is inaccessible to other modules that have no need for such info.

Q.2 Describe Unified Modeling language. Explain its type with proper example.

Ans: Unified Modeling Language (UML) is a standardized purpose visual modeling language in the field of Software Engineering.

- o It is used for specifying, visualizing, constructing & documenting the primary artifacts of the software system.
- o It describes the working of both hardware & software systems.
- o TYPES OF UML:

Q-3

What do you mean by UID? Explain the process of UID with proper diagram.

Ans: UID stands for User Interface Design. It is also known as User Interface Engineering

- UID means the process of designing user interfaces for software & machines like a mobile device, computer, etc. with the aim of increasing usability & improving the user experience.
- The aim of user interface design is to make user experience as easy as possible while still being successful in achieving user goals.
- PROCESS OF UID

Analyses
understand
user activity

Produce paper
based design
prototype

Evaluate
design
with end-user

Design
Prototype

Produce
Dynamic
design
prototype

Evaluate
design with
end-users

Executable
Prototype

Implement
final user
interface

10 User, Task, Environment Analysis

→ Initially, the focus is based on the profile of users who will interact with the system i.e. understanding, skill & knowledge, type of user, etc.

WOW!

based on the user's profile
user are made into categories

- From each category requirements are gathered.
- Based on the requirements developer understand how to develop the interface.

2. Interface Design

- The goal of this phase is to define the set of interface objects & actions i.e. control mechanisms that enable the user to perform desired tasks.
- Specify the action sequence of tasks & subtasks also called a User-Scenario.
It always follows the three golden rules.

3. Interface Construction And Implementation

- The implementation activity begins with the creation of prototype that enables usage scenarios to be evaluated.
- The design process continues a User Interface toolkit that allows the creation of windows, menus, error messages, etc. can be used for completing the construction of an interface.

4. Interface Validation

- This phase focuses on testing the interface. The interface should be in such a way that it should be able to perform tasks correctly & should be able to handle variety of tasks.
WOW!!

Q.4 What is Architecture Design?
Write objectives & types of architecture style.

Ans: The software needs the architectural design to represent the design of software & identify the subsystem of system.

- It is the process of defining a collection of hardware & software components and their interfaces to establish the framework for the development of a computer system.
- The architectural style is a very specific solution to a particular software, which typically focuses on how to organize the code created for the software.

→ ARCHITECTURAL STYLE OBJECTIVES:

1. A set of components that will perform a function required by the system.
2. The set of connectors will help in coordination, communication & cooperation between the components.
3. Conditions that how components can be integrated to form the system.
4. Semantic models that help the designer to understand the overall properties of the system.
5. The use of architectural styles is to establish a structure for all the components of the system.

WOW!!

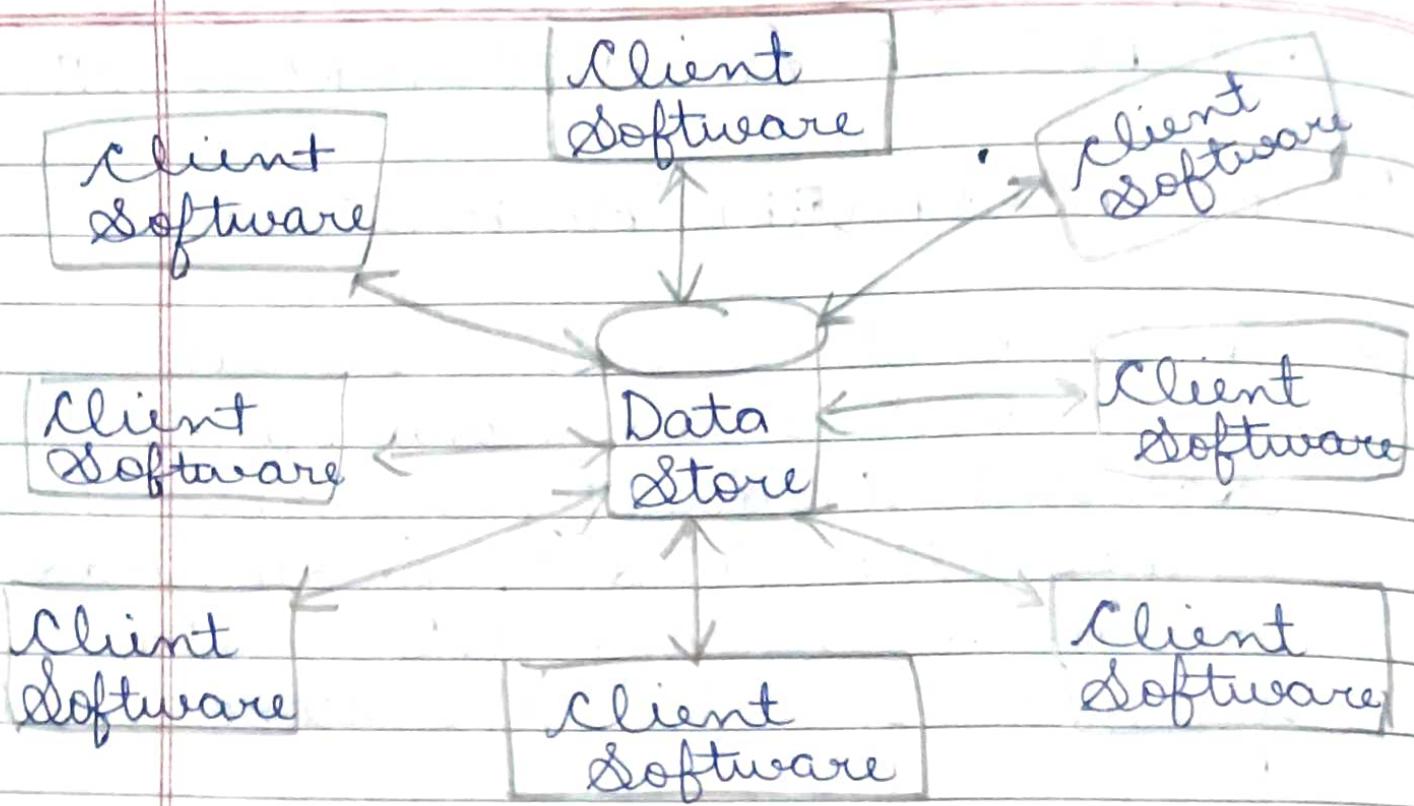
→ TYPES OF ARCHITECTURAL STYLES

1. DATA CENTERED ARCHITECTURES

- A data store will reside at the center of this architecture & is accessed frequently by the other components that update, add, delete or modify the data present within the store.
- The client software access a central repository.
- Promotes integrability which means the existing components can be changed & new client components can be added to the architecture without the permission of other clients.
- D Fig: DATA CENTERED ARCHITECTURE

(P.T.O)

NOW!

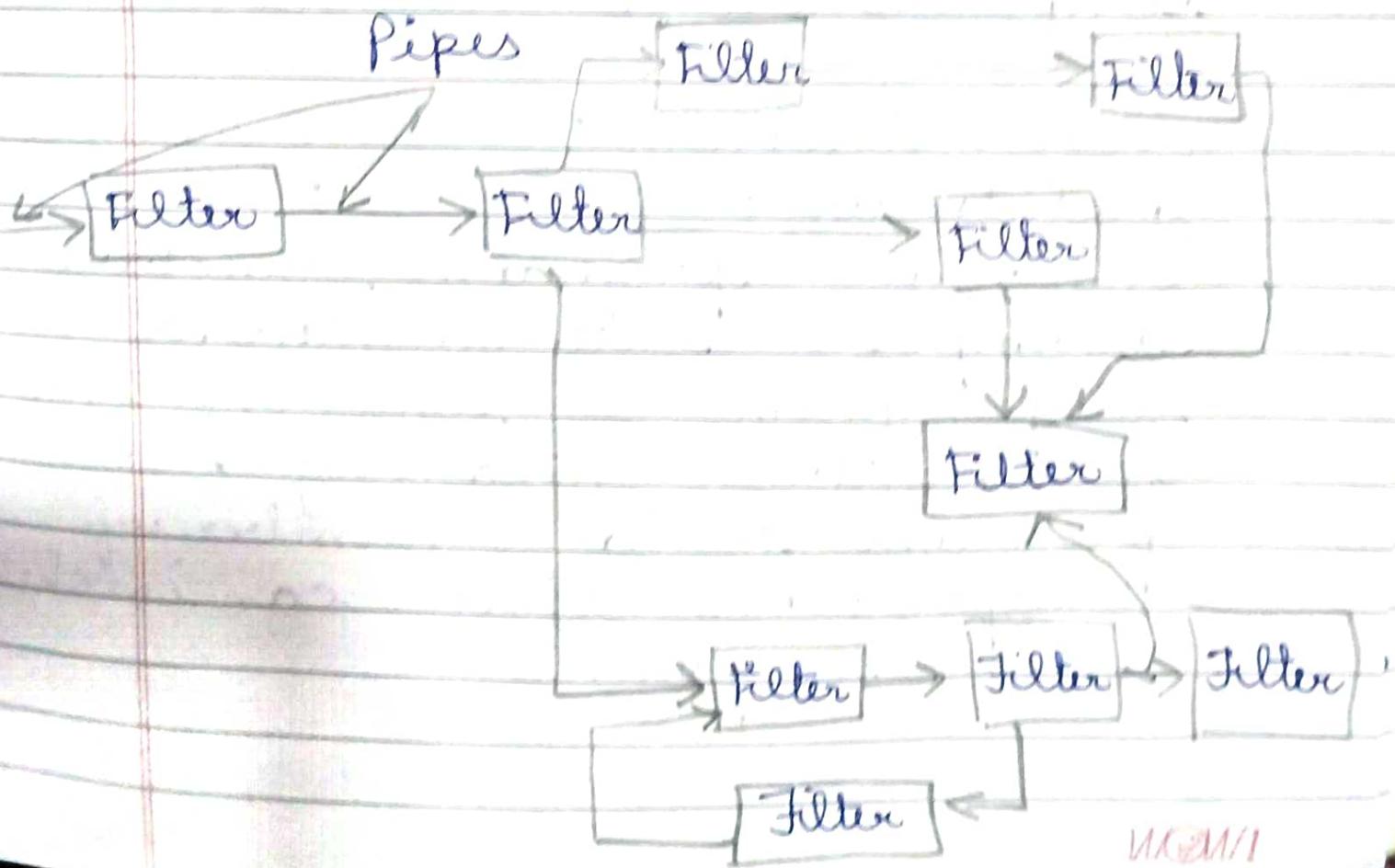


2. DATA FLOW ARCHITECTURE

- This kind of architecture is used when input data to be transformed into output data through a series of computational manipulative components
- The figure represents pipe-and-filter architecture since it uses both pipe & filter & it has a set of components called filter connected by pipes.

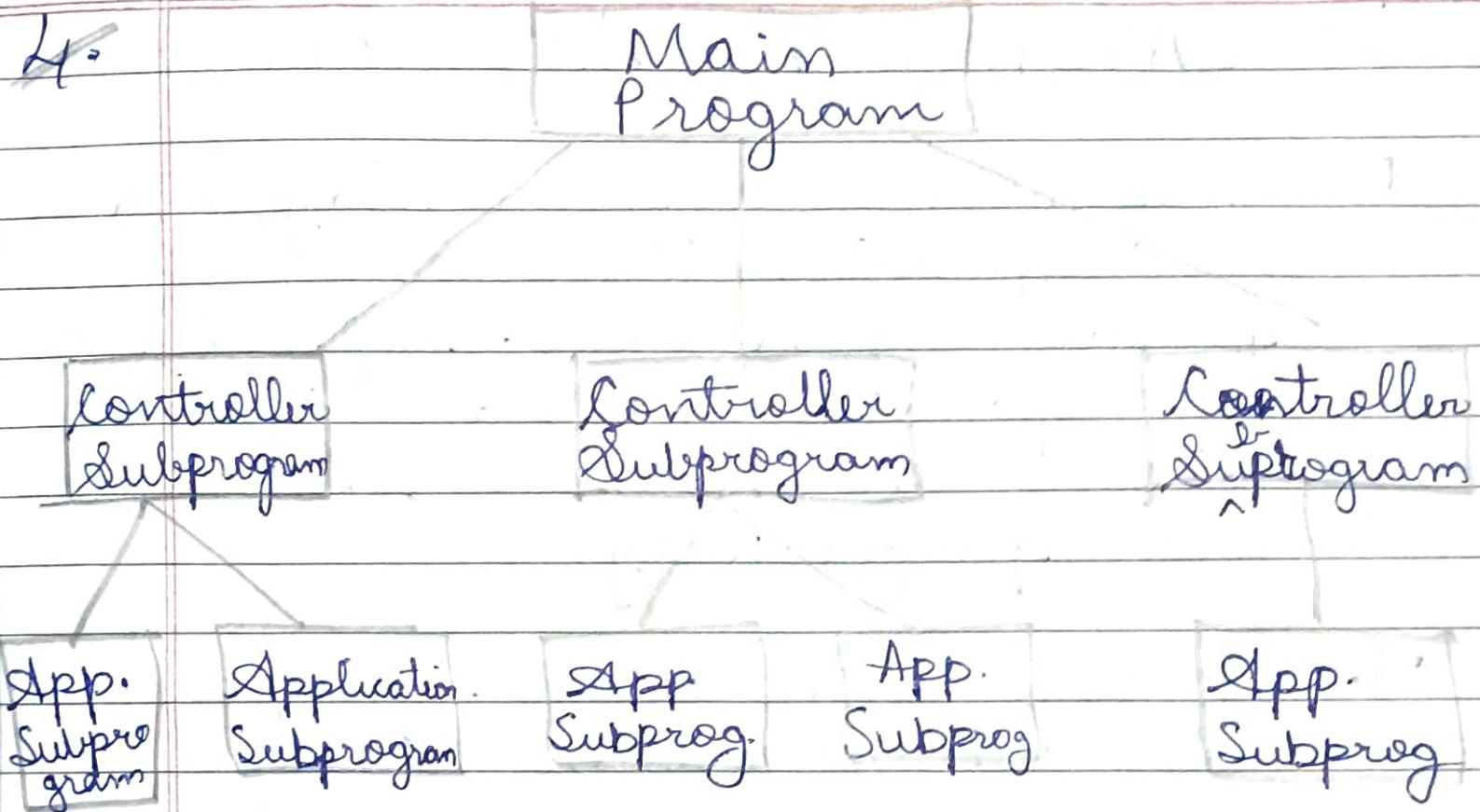
E) Pipes are used to transmit data from one component to the next.

D) Each filter will work independently. It is designed to take data input of a certain form & produces data output to the next filter of a specified form. The filter don't requires any knowledge of the working of neighbouring filters.



3. Call and Return Architecture

- It is used to create a program that is easy to scale & modify.
- It has substyles:
 - Remote procedure architecture : This component is used to present in a main program or sub program architecture distributed among multiple computers on a network.
 - Main Program or Subprogram architecture : The main program structure decomposes into number of subprograms or functions into a control hierarchy. Main program contains number of subprograms that can invoke other components.



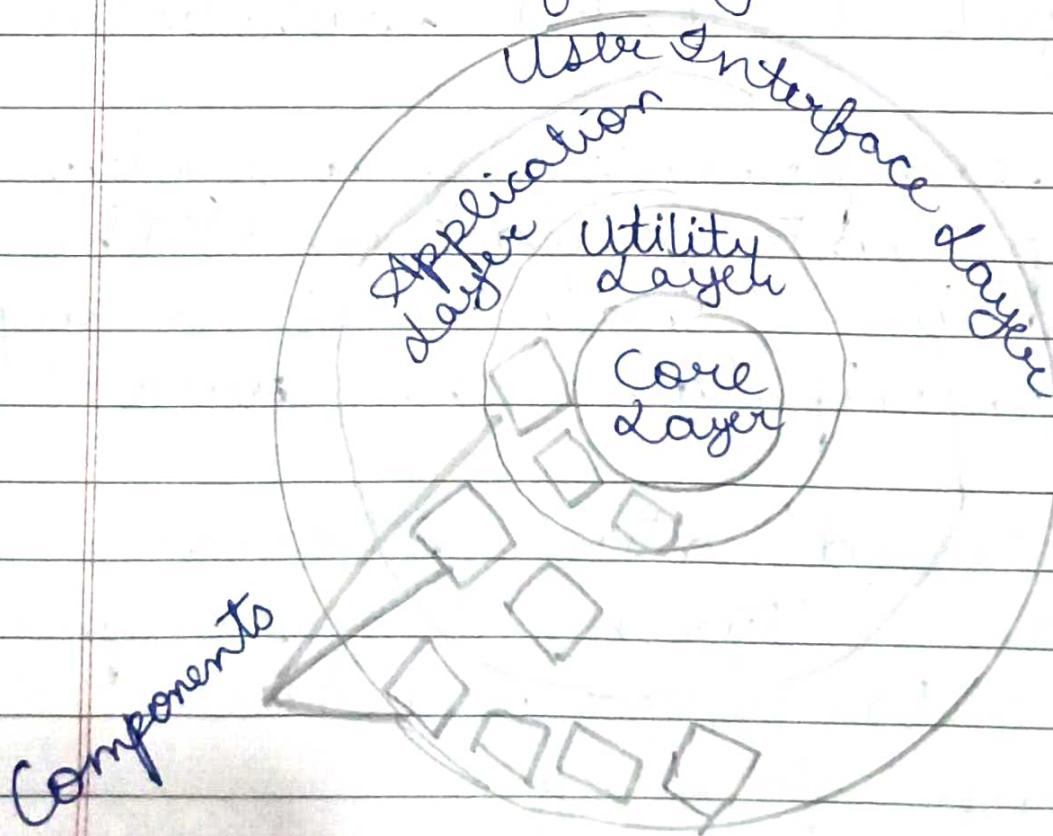
4. Object Oriented Architecture

- The components of a system encapsulate data and the operations that must be applied to manipulate the data.
- The coordination & communication between the components are established via the message passing.

WOW!!

S. LAYERED ARCHITECTURE

- A number of different layers are defined with each layer performing a well-defined set of operations, that becomes closer to machine instructions set progressively.
- At the outer layer, components will receive the user interface operations & at inner layers, components will perform the OS interfacing.



Q.5 Explain 3 golden Rules of UID in detail.

Ans: The following are the 3 golden Rules that must be followed during the design of the interface:

Rule 1: Place the User in Control

- Define the interaction modes in such a way that does not force the user into unnecessary or undesired actions.
- Provide for flexible interactions : different people will use different interactions, mechanism, etc. Hence all interaction mechanisms should be provided.
- Allow user interaction to be ~~sterile~~ interruptable & undoable

- Streamline interaction as skill level advances & allows the interaction to be customized.
- Hide technical internals from casual user.
- Design for direct interaction with objects that appear on the screen.

Rule 2: Reduce the user's memory load

- Reduce demand on short-term memory
- Establish ~~memory~~ meaningful defaults
- Define shortcuts that are intuitive
- The visual layout of the interface should be based

WOW!

on real-world metaphor

- Disclose information in a progressive fashion.

Rule 3: Make the Interface Consistent

- Allow the user to put the current task into a meaningful context.
- Maintain consistency across a family of applications.
- If past interactive models have created user expectations do not make changes unless there is a compelling reason.

Q.6 Difference between functional oriented & object oriented design.

FACTOR	FUNCTIONAL ORIENTED DESIGN	OBJECT ORIENTED DESIGN	Decomp -ose	Dec fun le
--------	----------------------------	------------------------	-------------	------------

Abstract The basic abstraction - tion, which are given ion are not to the user, are real world function real world function

Function Functions are grouped together together on the by which a higher basis of the level function is data they operate obtained.

Execute Carried out using structured analysis & structured design i.e. data flow diagram.

WOW!!

Approach It is a top down approach. It is a bottom up approach.

Decomp
ose Decompose in function / procedure level

Decompose in class level

Use This approach is mainly used for computation sensitive application.

This approach is mainly used for evolving system which mimics a business or business case.

Q.7 Write short notes on:

1. Architecture Views

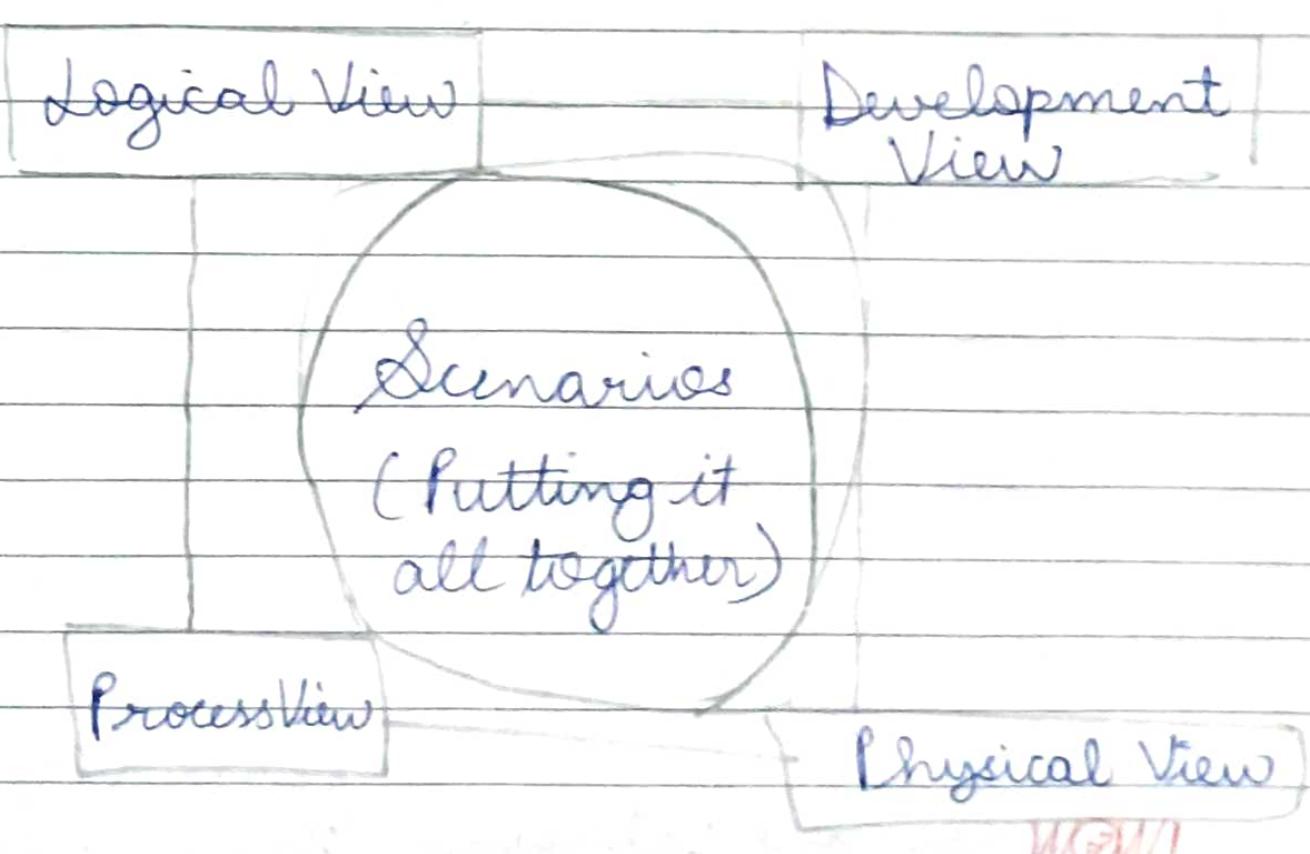
- It is an architecture verification method for studying and documenting software architecture design & covers all the aspects of software architect -ure for all stakeholders
- It provides 4 views:

- ① Logical view: the services that the system provides to end-users depicted with UML model types: Class Diagram, Communication Diagram, Sequence Diagram
- ② Development View: also known as the implementation view. It uses the UML Component Diagram to describe system components.

WOW!

③ Process View: deals with the dynamic aspects of the system, explain the system processes & how they communicate & focuses on the runtime behaviour of the system.

④ Physical View: depicts the system from a system engineer's point of-view. It is concerned with the topology of software components on physical layers as well as physical connection.



2. User Interface Design Models

- User Interface Design is an iterative process where each iteration elaborate and refines the information developed in preceding step.
- General steps for UID
 - 1. Using information developed during user interface analysis
 - 2. Define events that will cause the state of the user interface to change.
 - 3. Depict each interface state as it will actually look to the end user
 - 4. Indicate how the user interprets the state of the system from information provided through the interface **HOW!**