Q1. What do you interpret from the term “4+1” view model?

Sol: Software architecture deals with abstraction, with decomposition and composition, with style and esthetics.

To describe a software architecture, we use a model composed of multiple *views* or perspectives. In order to

eventually address large and challenging architectures, the model we propose is made up of five main views

• The logical view, which is the object model of the design (when an object-oriented design method is

used),

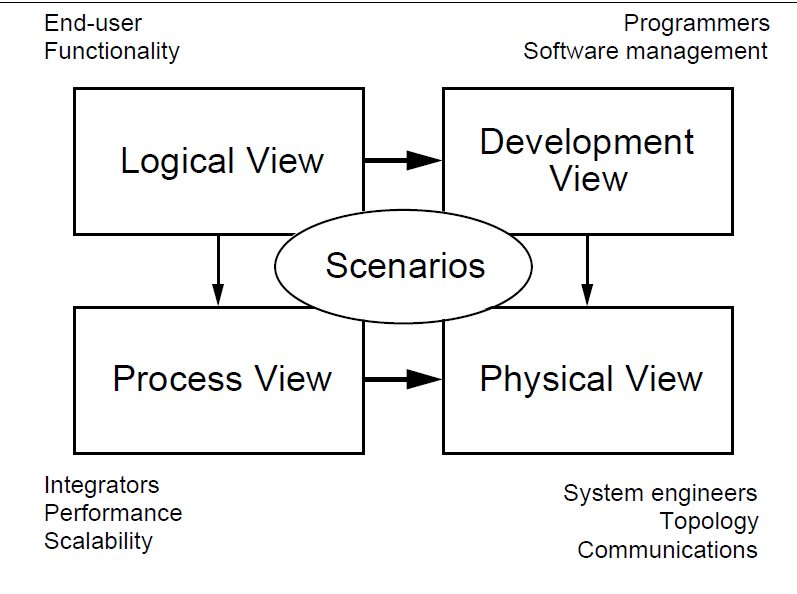
• the *process* view, which captures the concurrency and synchronization aspects of the design,

• the *physical* view, which describes the mapping(s) of the software onto the hardware and reflects its

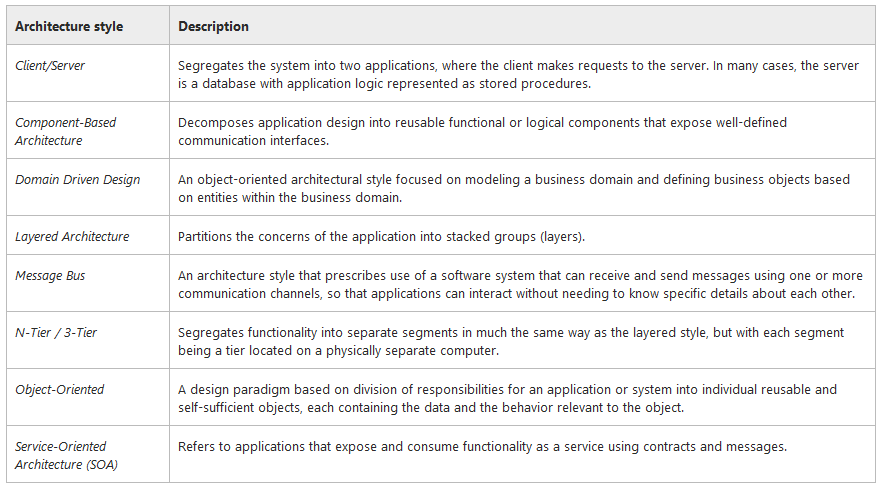
distributed aspect,

• the *development* view, which describes the static organization of the software in its development

environment.



Q. List common architectural styles



Q. Point out the pros and cons of Pipes and Filters. Give an example.

Sol: Adv: ex: Compiler design

* Overal I/O is a composition of the behavior of independent filters
* Support reuse
* Maintainability, improvement
* Allow analysis (throughouput, deadlock analysis)
* Support concurrency

Disadv

* Often lead to a batch organization of processing
* Not good at handling interactive applications
* Quite complex
* Not so performant

Q. What are the three major parts of Blackboard model?

**The knowledge sources.** The knowledge needed to solve the problem is partitioned

into ***knowledge sources,*** which are kept separate and independent.

**2. The blackboard data structure.** The problem-solving state data are kept in a global

data base, the ***blackboard.*** Knowledge sources produce changes to the blackboard

which lead incrementally to a solution to the problem. Communication and

interaction among the knowledge sources take place solely through the blackboard.

**3. Control:** The knowledge sources ***respond opportunistically to changes*** in the blackboard system

Q. Show the strength and weakness of Repositories

Strengths:

* Efficient way to share large amounts of data
* Data integrity localized to repository module

Weaknesses:

* Subsystems must agree (i.e., compromise) on a repository data model
* Schema evolution is difficult and expensive
* Distribution can be a problem

Q. Show the strength and weakness of layered architecture

Advantages

* Abstraction (deals with complexity)
* Modifiability Changing one layer influences only the two adjacent layers
* Reuse Different implementations easy to substitute
* Interfaces

Disadvantages

* Not all systems suitable for this
* Performance may require other coupling
* Abstraction quite hard

Q. Summarize the properties of model view controller

Q. What is heterogeneous architecture? Classify their styles.

Q. What is the significance of SOA

Q. List the activities involved in creating software architecture

Sol:

  **Identify your architecture goals at the start**. The amount of time you spend in each phase of architecture and design will depend on these goals. For example, are you building a prototype, testing potential paths, or embarking on a long-running architectural process for a new application?

 **Identify who will consume your architecture**. Determine if your design will be used by other architects, or made available to developers and testers, operations staff, and management. Consider the needs and experience of your audience to make your resulting design more accessible to them.

 **Identify your constraints**. Understand your technology options and constraints, usage constraints, and deployment constraints. Understand your constraints at the start so that you do not waste time or encounter surprises later in your application development process.

Q. “Architecture is high level design”-Justify

or

Describe the technical importance of software architectures. Further elaborate on the fact that architecture is a vehicle for stakeholder communication.

Q. Give the three broad types of decision that architecture design involves

1. **Identification of the Architecture Objectives**. Clear objectives help you to focus on your architecture and on solving the right problems in your design. Precise objectives help you to determine when you have completed the current phase, and when you are ready to move to the next phase.
2. **Define Key Scenarios**. Use key scenarios to focus your design on what matters most, and to evaluate your candidate architectures when they are ready.
3. **Look for Application Overview**. Identify your application type, deployment architecture, architecture styles, and technologies in order to connect your design to the real world in which the application will operate.
4. **Identify Key Issues**. Identify key issues based on quality attributes and crosscutting concerns. These are the areas where mistakes are most often made when designing an application.
5. **Propose Candidate Solutions**. Create an architecture spike or prototype that evolves and improves the solution and evaluate it against your key scenarios, issues, and deployment constraints before beginning the next iteration of your architecture.

Q. Differentiate functional and non functional requirements.

Sol: Simply put, the difference is that **non-functional requirements describe how the system works**, while **functional requirements describe what the system should do**.

Q. Software architecture is often compared to building architecture. What are the strong points of this comparison? What is the correspondence in buildings to software architecture structures and views? What is the weakness of this comparison? When does it breakdown?

Q. What is Software Process? List different software processes.

Sol: A structured set of activities required to develop a software system.

The waterfall model -Plan-driven model. Separate and distinct phases of specification

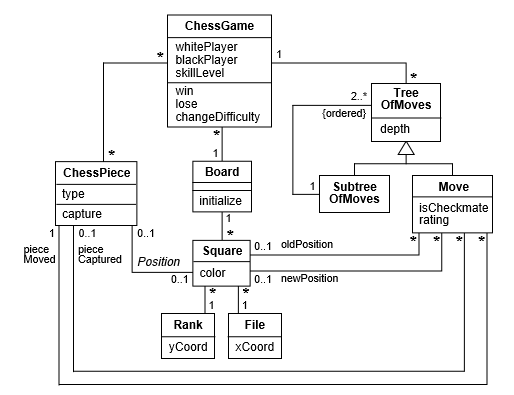
and development.

Incremental development - Specification, development and validation are interleaved. May

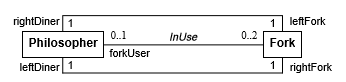
be plan-driven or agile.

Reuse-oriented software engineering - The system is assembled from existing components. May be plan-driven or agile.

Q. Create a class diagram having at least 6 relationships using the following set of classes: chess, piece, file, square, board, move, tree of move. Also show associations and show multiplicity

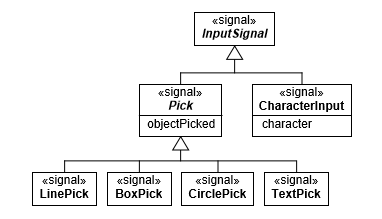


Q. create a class diagram for dining philosopher problem



Q. place the following signal classes into generalized hierarchy:

Pick, character input, line pick, circle pick, box pick, text pick, input signal



Q. given the following scenario: create a sequence diagram:

Purchase items

Customer brings items to the counter.

Cashier scans each customer item.

Cashier totals order, including tax.

Cashier requests form of payment.

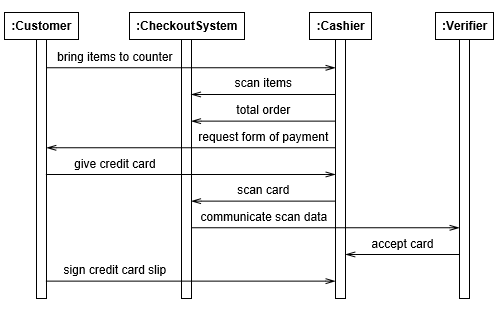
Customer gives a credit card.

Cashier scans card.

Checkout system communicates scan data to verifier.

Verifier reports that credit card payment is acceptable.

Customer signs credit card slip.



Q. Choose the correct alternative

What does conceptual modelling represents ?  
a. Responsibility  
b. Attributes  
c. Important relationships between them  
**d. All of the above**

2. What are the sequence of steps for conceptual process ?  
a. Add Classes  
b. Add Attributes  
c. Add Association  
**d. All of the mentioned**e. None of the mentioned