VIVA Questions

Q1 what is problem statement?

Ans A problem statement is a concise description of an issue to be addressed or a condition to be improved upon. It identifies the ~~top~~ gap b/w the current (problem) state & desired (goal) state of a process or product. The first condition of solving a problem is understanding the problem, which can be done by way of a problem statement.

Q2 what are the benefits of writing problem statement?

Ans A problem statement is basically a statement that illustrated a clear vision of the overall method that will be used to solve the problem at hand. Usually used when doing research, a problem statement discusses any foreseeable tangible or intangible problem that the researcher may face throughout the course of the project.

Q3 writing a problem statement, is really a beneficial for you in proceeding project?

Ans Yes, writing a problem statement should help in careful decision-making for project approval. Often, the problem statement will serve as the basis for the introductory section of a final proposal directing the readers attention to the issues that your proposed project will address.

Q4 Explain 5W's can be used to spark the problem)

The five 'Ws are questions whose answers are considered basic in problem solving & information gathering. They are:

* who
* what
* when
* where
* why

The 5 'Ws can be used to spark the discussion about the problem, A problem statement expresses the issue that will be used to keep the effort focused & it should represent a solvable problem.

Q5 What are steps that need to follow while writing problem statement?

Ans Step that we need to follow while writing problem statement are:

1. Define the problem
2. Reason for the problem's occurrence
3. When the problem began or was first noticed.
4. Place of the problem's first occurrence or sighting.
5. The person or thing that the problem affects.
6. The sequence of event that resulted in the problem.

VIVA QUESTIONS

Q1 What are the objective of requirement analysis?

Ans The purpose of the Requirements Analysis phase is to transform the need & high-level requirements specified in earlier phases into unambiguous (measurable & testable), traceable, complete, & stakeholder approved required.

Q2 Define different type of requirent?

Ans According to IEEE standard 729, a requirement is defined as follow:

* A software requirment can be of 3 types
  * Functional requirments
  * Non-functional requirments
  * Domain requirments

Functional Requirements: These are the requirment that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Non-functional requirments: These are basically the quality constraints that the system must statisfy according to the project contract. The priority or extent to which these factor are implemented varies from one project to other.

Domain requirement - Domain requirment are the requirements which are characteristic of a particular category or domain of project. The basic functions that a system of a specific domain must necessarily exhibit come under this category.

Q3 Outline structure of SRS Document?

Ans
1. Introduction
   (i) Purpose of this document
   (ii) Scope of this document
   (iii) Overview
2. General description
3. Functional Requirement
4. Interface Requirments
5. Performance Requirments
6. Design Constraints
7. Non-functional Attributes
8. Preliminary Schedule & Budget
9. Appendices.

Q4 What are benefits of writing SRS document?

Ans
* An SRS establishes the basis for agreement between the customer & the supplier on what the software product will perform.
* An SRS provides a reference for validation of the final product / software.
* A high-quality SRS is a prerequisite to high-quality product / software.
* A high-quality SRS reduces the development cost.

Q5 Define functional & non functional requirement?

FUNCTIONAL Vs NONFUNCTIONAL REQUIREMENT

	FUNCTIONAL REQUIREMENT	NONFUNCTIONAL REQUIREMENT
OBJECTIVE	Describe what the product does	Describe how the product works
O RESULT	Define product feature	Define product properties
FOCUS	Focus on user requirment	Focus on user expectation
DOCUMENTATION	Captured in use case	Captured as a quality attribute.
ENTIALITY	They are mandatory	They are not mandatory, but desirable
ORIGIN TYPE	Usually defined by User	Usually defined by developers or other tech experts
TESTING	Component, API, UI testing etc. Tested before nonfunctional testing	Performance, usability security testing etc. Tested after functional testing.
TYPES	· External interface, authentication authorization levels, business rules etc.	Usability, reliability scalability, performance etc

VIVA QUESTIONS

Q1 Define DFD? what are different level of DFD?

Ans Data flow diagrams are versatile diagramming tools. With only four symbols, data flow diagrams can represent both physical & logical information system. The four symbols used in DFD representation are data flow, data stores, process, & sources/sinks (or external entities).

0-level DFD:

It is also known as a context diagram. Its designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represent the entire system as a single bubble.

1-level DFD

In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level we highlight the main functions of the system.

2-level DFD

2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning.

Q2 Describe symbols used for constructing DFD's?

Ans The four symbols used in DFD representation are data flows, data stores, processes & sources/sinks. (or external entities).

Q3 Distinguish b/w a data flow diagram & a flow chart with example?

Ans The main difference b/w DFD & flowchart is that DFD is graphical diagram that represent the data flow of a system. while flowchart is a graphical diagram that represents the sequence of step to solve a problem.

Q4 Explain structure chart diagram?

Ans A structure chart (SC) in software engineering is a chart which shows the breakdown of a system to its lowest manageable levels.

Q5 Describe symbols used for constructing structured chart diagram?

Ans Module - It represents process or subroutine or task. A control module branches to more than one sub-module. Libary module are re-usable & invokable from any module.

condition - It is represented by small diamond to the base of module. It depicts that control module can select any of sub-routine based on some condition.

Jump - An arrow is shown pointing inside the module to depict that the control will jump in the middle of the sub-module.

Loop - A loop carved arrow represent loop in the module. All sub-module covered by loop repeat execution of module.

Data flow - A directed arrow with empty circle at the end represent data flow.

control flow - A directed arrow with filled circle at the end represent controll flow.

Q6 Explain ER diagram?

Ans An entity relationship model, also called an entity-relationship (ER) diagram, is a graphical representation of entities & their relationship to each other, typically used used in computing in regard to the organization of data within databases or information system.

VIVA QUESTIONS

exp-2

Q1 Explain use cause approach of requirement elicitation)

Ans This technique combines text & pictures to provide a better understanding of the requirement. The use cases describe the 'What' of a system & not 'How'. Hence, they only give a functional view of system. The components of the use case design include three major things - Actor, Use cases, use case diagram.

Q2 Explain term: Use case, use-case scenarios, use-case diagrams?

Ans A use case is a written description of how users will perform tasks on your website. It outlines, from a user's point of view, a system's behaviour as it responds to a request.

A user case scenario is a single path through the use case. Unlike a use case which is a step-by-step enumeration of the task carried out during a process (with the associated actors), a scenario is much more free-form.

A use case diagram can summarize the details of your system users (also know as actors) & their interactions with the system.

Q3 what are Actors & use cases?

Ans An Actor is use case modelling specifies a role played by a user or any other system that interacts with the subject. An actor model type of role played by an entity that interacts with the subject. (eg., by exchanging signals & data), by which is external to the subject.

A use case is a written description of how users will perform task on your website. It outlines, from a users point of view, a systems behaviour as it reponds to a request.

Qy Explain guidelines that should be kept in mind while creating use cases?

Ans Consider the following:

* Single statement per line
* Always have a subject - "User" or "System"
* Be concise - remember, use cases are not requirements you should be demonstrating the interaction between the system & user, but not detailed specifications
* Use an active voice.

Qs Name the person who invented use case approach?

Ans Ivar Jacobson first formulated textual & visual modelling techniques for specifying use cases.

VIVA QUESTIONS

Exp - 5

Q1 Explain class diagram?

Ans A class diagram is a type of statics structure diagram that describle the structure of a system by showing the system's classes, their attributes, operations (or methods), & the relationship among objects.

Q2 Explain four types of relationship used in class diagram?

Ans Inheritance (or generalization)

A generalization is a taxomic relationship b/w a more gehual classifier & a more specific classifier. Each instance of the specific classifier is also an indirect instance of the general classifier.

Association - Association are relationship b/w classes in a UML class diagram. They are represented by a solid line b/w classes. Association are typically named using a verb or verb phrase which reflected the real-world problem domain.

Realization - Realization is a relationship b/w the blueprint class & the object containing its respective implementation level details. This object is said to realize the blueprint class.

Dependency - An object of one class might use an object of another class in the code of a method. If the object is not stored in any field, then this is modelled as a dependency relationship.

Q3 Explain term classes, interfaces. collaboration & dependency?

Ans classes - A template for creating objects & implementing behaviour in a system. In UML, a class represents an objects or a set of object that share a common structure & behaviour. They're represented by a rectangle that include rows of the class name, it attribute & its operating.

Interface - Interface are model elements that define set of operations that other model elements such as classes, or components must implement. A model element realized an interface by overriding each of the operations that the interface declares.

Collaborations - It is used to show the relationship b/w the object in a system. Both the sequence & the collaboration diagram represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features -

Dependency An object of one class might use an object of another class in the code of a method. If the object is not stored in any field then this modeled as a dependency relationship.

Q4 Explain object diagram?

Ans object diagram represent an instance of a class diagram. The basic concept are similar diagram & object diagrams. object diagram also represent the static view of a system but this static view is a snapshot of the system at a particular moment.

Q5 Explain symbols used in it?

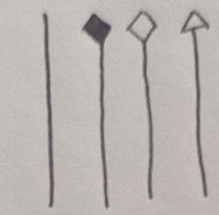
Object Names - Every object is actually symbolized like a rectangle, that offers the name from the object & its class underlined as well as divided with a colon.

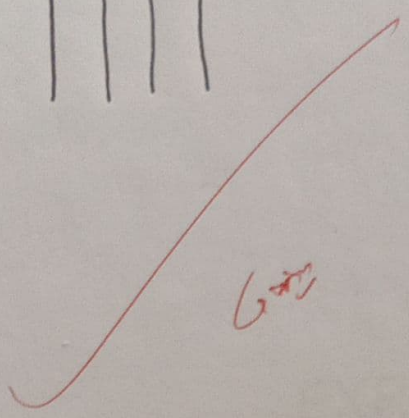
object : class

Object Attributes - Similar to classes, you are able to list object attributed inside a separate compartment. However, unlike classes, object attributes should have values assigned for them.

object : class
attribute = value

Links - Links tend to be instances associated with associations. You can draw a link while using the lines utilized in class diagrams.





VIVA QUESTIONS

Q1 How activity diagram explain behavioral view of system?

Ans An activity diagram is a behavioral diagram. ie it depicts the behaviour of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

Q2 Explain steps followed to construct activity diagram?

Step 1 - Figure out the action step from the use case

Here you need to identify the various activities & actions your business process or system is made up of.

Step 2 Identify the actor who are involved

If you already have figured out who the actor are, then its easier to discern each action they are responsible for.

Step 3 Find a flow among the activities

Figure out in which order the actions are processed - Mark down the conditions that have to be met in order to carry out certain processes, which actions occur at the same time, whether you need to add any branches in the diagram. And do you have to complete some actions before you can proceed to others?

Step 4 Add swinlanes - You have already figured out who is responsible for each action. Now it's time to assign them a swinlane & group each action they are responsible for under them.

Q3 How state chart diagram explain behavioural view of system?

Ans State chart diagram defines that states of a component & these changes are dynamic in nature. Its specific purpose is to define the state changes triggered by events. Event are internal or external factors influencing the system.

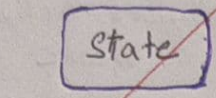
Q4 Explain step followed to construct state chart diagram)

Ans → Identify the final state of the final terminating states.
→ Identify the possible state in which the object can exist (boundary values corresponding to different attributes guide us in identifying different states).
→ Label the event which trigger these transitions.

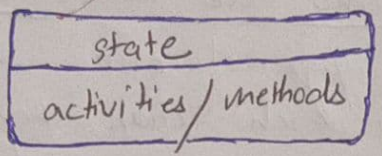
Q5 Explain symbols used to construct these diagram?

STATE CHART DIAGRAM

State

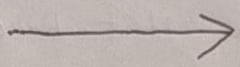


A simple state



A state with internal activities

Transition



Transition

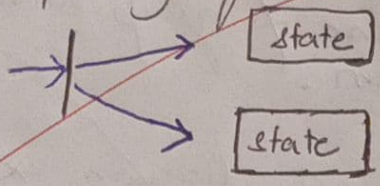
Initial State

Initial state

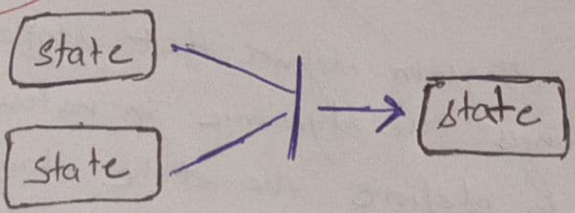
Final State

Final state

Synchronization / splitting of control

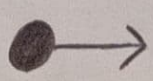


Fork



Join

Initial point or start point



Start Point / Initial point

Viva Questions

Ans 1. Explain use of sequence diagram?

Sol^n Following scenarios are ideal for using a sequence diagram:

→ Usage scenario
→ Method logic
→ Service logic
→ Sequence diagram Visio

Ans 2. Explain steps to draw a sequence diagram.

Sol^n 1. Identify the class instances (objects) by putting each class instance inside a box.

2. If a class instance sends a message to another class instance draw a line with an open arrowhead pointing to receiving class instance; place the name of message/ method above the line.

3. Optionally, for important messages, you can draw a dotted line with an arrowhead pointing back to originating class instance; label the return value above the dotted line

Ans 3. Explain the need of collaboration diagram?

Sol^n 1. Modelling collaborations, mechanisms or the structural organization within a system design.

2. Providing an overview of collaborating objects within a OO object oriented system.

3. Exhibiting many alternative scenario for same use case

4. Demonstrating forward and reverse engineering

5. Capturing the passage of information b/w objects

6. Visualizing the complex logic behind an operation

Ans 4. Explain steps to draw collaboration Diagram

Sol^n By simply pressing combination of keys, we can design collaboration diagram from sequence diagram.

Ans 5. Explain terms - entity object, interface objects and control object:

Sol^n Entity Object
Objects that encapsulate the business model, including rules, data, relationships and persistance behaviour.

Interface objects
Objects which provides all details on how 1 process component exchanges data with another.
Eg- Communication mode and data Structure

Control Objects
These are widget or gadgets and can be used in window dialog box. They cannot exist outside a window or dialog box. You can define controls.

VIVA QUESTIONS

Ques 1. Explain component diagram.

Ans Component diagram or UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double check that every aspects of system's required function is covered by planned development.

Ques 2. Component diagram explains which view of system?

Ans Component diagram describes the static view of the system which includes the organisation of components at a particular instant.

Ans 3. Explain steps to draw component diagram.

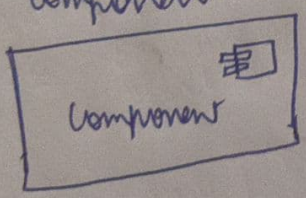
1. Take stock of everything required to implement planned system. For eg. a simple e-commerce system, you'll need components that describe products, orders, and customer account.
2. Create a visual for each of the components.
3. Describe the organization and relationships between components using interfaces, ports and dependencies.

Ans 4. What is benefit of drawing component diagram?

Ans Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the components used to make those functionalities.

Ans 5. Explain symbols used to draw components diagram?

Ans Component → Logical unit block of the system, a slightly higher abstraction than classes. It represented as a rectangle with a smaller rectangle in the upper right corner with tabs or the words written above the name of the component to help distinguish it from a class.



Interface → An interface (small circle or semi-circle on a stick) describe a group of operations used (required) or created (provides) by components.

○ → an interface created or provided by component

) → interfaces requiring person's input.

VIVA QUESTIONS

Ques 1. What is deployment diagram ?

Ans Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagram are used to describe the static deployment view of a system. Deployment diagrams consists of nodes and their relationships.

Ques 2. Deployment diagram explains which view of system ?

Ans Deployment diagram are used to describe the static deployment view of a system.

Ques 3. Explain steps to draw deployment diagram ?

Ans To create a deployment diagram :

1. Identify component
2. Add shapes
3. Connect nodes
4. Format arrows

Ques 4. Explain symbols used to draw deployment diagram ?

Ans Communication path : Straight line that represents communication between two device nodes.

Artifacts : A box with headers '<>' and then the name of the file

Package : A file-shaped box that group together all the device nodes to encapsulate the entire deployments.

Nodes : 

there are 2 types of nodes in a deployment diagram. device nodes and execution environment nodes.

Ques 5. What is the benefit of drawing deployment diagram ?

Ans System engineering mainly consume deployment diagrams. These diagrams help us to describe the physical components like hardware involved, participant nodes, their distribution and how they are interconnected. Deployment diagram could be assumed as the hardware components where the software components reside.

Viva Questions

exp - 10

Ques 1. Explain the role of testing in software development?

Ans Software testing comes into play at different times in different software development methodologies.
→ Waterfall
→ Agile
Testing modular functions and testing help us find bugs quicker, easy to make changes which is part and parcel of agile software development methodology and reduces the risk of errors/faults.

Ques 2. How much testing is sufficient? Is it possible to do exhaustive testing of the software?

Ans It is impossible to exhaustively test software or prove absence of errors, no matter how specific your test strategy is. Exhaustive testing doesn't ensure no errors or have discovered it all so there is no accurate measure to say this much testing is best but it should cover major and trivial factors.

Ques 3. Why developers shouldn't test the software they wrote?

Ans Developer make poor tester. Here are the reasons why:
→ They try to test code to make sure that it works, rather than testing all the way in which it doesn't work.
→ Since they wrote it themselves, developers tends to be very optimistic about the software and don't have the correct attitude needed for testing: to break software

Ques 4. What is functional testing?

Ans Functional testing is a form of black-box testing. As the name suggest, it focuses on software's functional requirements rather than it's internal implementation. A functional requirement refers to required behaviour of the system, in terms of I/O.

Ques 5. What is non-functional testing?

Ans It's tests for non-functional requirements, which refers to an attribute or quality of the system explicitly requested by the client. These include - security, scalability and usability.

VIVA QUESTIONS

Ques 1. Explain 5 functional units of FPA (functional Point Analysis)?

Ans
- No. of External Inputs (EI) → Input screens and tables
- No. of External Outputs (EO) → Output screens and reports
- No. of External Inquiries (EQ) → Prompts and interrupts
- No. of External files (ILF) → Databases and directories
- No. of External Interfaces (EIF) → Shared DB and shared routines

Ques 2. Explain special features of FPA?

1. FPs of an application is found out by counting the number and types of func used in applications
2. FP characterizes the complexity of software system and hence can used to depict the project time and manpower requirements
3. The effort used to develop project depends on what software does.
4. FP is language independent.
5. FP is used for data processing systems like Information systems.

Ques 3. Explain weighing factors (3) of functional units.

Ans It's the weight given to data points do assign lighter or heavier importance in a group. It usually used for calculating a weighted mean, to give less (or more) importance to group members

	Low	Medium	High
EI	3	4	6
EO	4	5	7
EQ	3	4	6
ILF	5	10	15
EIF	7	7	10

Ques 4. Explain term unadjusted function point count (UFP)?

Ans for each function there is weight associated. The sum of weights quantifies the size of information processing and is reffered to as UFP (Unadjusted Function Points)

Ques 5. what are uses of Function Point (FP)?

Ans
- The FP is a "unit of measurement" to express amount of business functionality an information system. (as a product)
- FP is used to compute FSM (functional size management)
- The cost (in dollars or hours) of single unit from past projects.