**20 forums of BA**

1. **BPM (Business process model):**

* Business process model (BPM) helps the entire organization construct framework for all its process that uses various methods to discover, model, analyze, measure and improve the business process.
* BPM consists of goal, input, resources, output, activities and value of a business.

**2. FEASABILITY STUDY:**

* It is a the possibility and capability of something being done
* Feasibility study is a controlled process for identifying problems and opportunities determining objectives and defining successful outcomes

The five components of feasibility study include:

Economic, marketing, technical, financial and management feasibility

|  |  |
| --- | --- |
| **Business Feasibility:** | Does the solution align with business goals and strategy? |

|  |  |
| --- | --- |
| **Technical Feasibility:** | Can IT/technical teams build or support it? Do we have the right tech stack? |

|  |  |
| --- | --- |
| **Economic Feasibility:** | Is it financially viable? Cost-benefit, ROI, payback period? |

|  |  |
| --- | --- |
| **Operational Feasibility:** | Will end-users accept it? Can we support it operationally? |

**3. GAP ANALYSIS:**

* It is a method of assessing the performance of business unit to determine whether business requirements or objects are being met if not what steps should be taken to meet them
* Gap analysis consist of three components:
* Current state, desired state and gap
* Defining main purpose, assess current performance, analysing root cause problems analyzing where improvements has to be done.
* Gap: considering the market and how customer’s needs may be unmet, identify areas where product supply is not meeting consumer demand can help a company fill that market gap

**Monitor progress and evaluate results:**

After implementing and making changes evaluating results after this conclude a gap analysis

**Example:**

**Define the objective** (e.g. improve customer service response time)

**Assess the current state** (e.g. avg. response time = 48 hours)

**Define the future state** (e.g. target = 12 hours)

**Identify the gap** (what’s causing the delay? people? process? tech?)

**Recommend actions** (automation, training, more staff, etc.)

**4. RISK ANALYSIS:**

* It is a process of identifying, estimating and prioritizing potential risks to an organization and developing strategies to mitigate or eliminate them.

**BA RISKS:**

* Not gathering requirements properly,
* Not choosing proper elicitation techniques.
* Change request has been given in last moment.
* Coordination are not proper between developers
* Lack of communication
* Not having good team management.
* No easy payments.

**PROJECT RISKS**:

* Quality of product is not good
* No proper awareness of product and application
* App features are not understanding properly to end users

**5. FIXED BID AND BILLING**

**FIXED BID:**

* Requirements are fixed at the start of project estimation made based on requirements
* Based on requirements number of resources required at each stage is decided
* Cost of developing the product is estimated once discussed
* Cost may be increased or decreased if change occurs
* Iterations are introduced to improve software quality
* Each stage is defined with timeline cannot be changed

**BILLING BID:**

* Requirement defined at initial stage, requirement may increase during developing the software
* Resource requirement may vary based on change
* Budget may increase if new features are added, timelines are flexible for change request
* Timeline for delivery has no change in the requirement

**6. RACI:**

* A RACI matrix is a tool in project management designed to clarify team roles and responsibilities across tasks.
* Each role is categorized as Responsible, Accountable, Consulted, or Informed, ensuring that everyone understands their role in a project’s success.
* By assigning these roles, project managers prevent confusion and overlapping duties while increasing accountability.
* The RACI framework is especially beneficial for complex projects where clear role distribution is essential for smooth workflows and improved communication.
* Project managers use RACI charts to keep track of team roles and relay those responsibilities to the larger team.

**7. 3-Tier Architecture:**

3-Tier Architecture is a software architecture that consists of three layers they are:

Application layer, business logic layer and data base layer

**Application layer:**

Application layer is atop layer of the architecture and is responsible for presenting, interact with the end-users’ using screens and pages it is also known as user interface layer or client layer. This layer handles interaction between the user and the system.

**Business logic layer:**

This layer is middle of the architecture and contains the business logic of the system it is also known as logic layer or server layer.

This layer manages the application logic, data validation, data processing. it communicate with the application layer and data layer.

Code for implementing business rules, connecting to 3rd party tools-API(application programmed interface).

Example:

Printers, pen-drive, internet

**Data base layer:**

It is bottom layer of the architecture and is responsible for managing the data storage and retrieval, it is also known as data layer or server layer.

This layer is responsible for storing and retrieving data from a database management system (DBMS), it provides an interface for the application layer to access and manipulate data.

**8. Requirements traceability matrix (RTM):**

* A **Requirements Traceability Matrix (RTM)** is a document that links requirements throughout the validation process.
* It helps ensure that all requirements defined for a system are tested and met during the development lifecycle.
* It is a crucial tool in systems engineering, software development, and quality assurance.

**PURPOSE OF RTM:**

* Ensure each requirement is covered by at least one test case.
* Trace requirements forward (from requirement to design/code/test) and backward (from test case to requirement).
* Identify missing functionality, redundant requirements, or invalidated requirements.
* Assist in impact analysis during requirement changes.

**9. DB Schema and Entity Relationship Diagram (ERD):**

* DB schema is a blueprint that outlines the structure of a database, including, its tables, fields, relationships, constraints and other characteristics.
* ERD is a visual representation of the relationships between entities in a database.
* It depicts the entities such as tables, attributes and relationships between them.

**10. Project closure document:**

A project closure document also known as project closure report –is a formal document that summarizes the key outcomes, lessons learned and final details of completed project.

Points to be included in the project closure document are:

* Project overview
* Achievements
* Lessons learned
* Quality assurance
* Resource utilization
* Risk management
* challenges

**11. WHAT IS DIFFERENCE BETWEEN BV AND CV?**

Business value (BV): Business value measures the benefit or impact a feature or task deliveries to the organization or end users.

It focuses on the importance and outcome of completing a task.

Complexity points (CP):

Complexity points estimate the effort and technical difficulty required to complete a feature or task.

They focus on how challenging it is to implement.

**12. Explain product backlog and sprint back log:**

Product backlog:

Product backlog is a key artifact in the scrum framework, representing a prioritized list of work items that the scrum team needs to complete to deliver a product or project.

It serves as the single source of truth for all the features, enhancements, bug fixes, technical requirements and other deliverables are required for achieving the products goals.

**Key characteristics:**

1. Dynamic and evolving:

The product backlog is not static; it evolves over time based on changing requirements, market needs or feedback from stakeholders.

2. Prioritized:

Items are ordered by priority, with the most important and valuable items placed at the top. These are usually addressed first in sprints.

3. Refined:

Items in the backlog are regularly refined or groomed to ensure they are ready for future sprints.

4. Owned by the product:

The product owner is responsible for maintaining the backlog ensuring it aligns with thr product vision and stakeholder needs.

5. Detailed as needed:

Items near the top are more detailed, while those further down can remains less defined until they approach implementation.

**Sprint backlog:**

The sprint backlog is a key artifact in the scrum framework that represents a subset of the product backlog items selected for a specific sprint, along with the plan to deliver them it is the teams working to-do-list during the sprint includes all tasks required to meet the sprint goal.

**Key characteristics**:

1. Subset of the product backlog:

The sprint backlog is delivered from the product backlog during sprint planning.

It contains only the items the team commits to completing during the current sprint.

2. Owned by the development team:

The development team is responsible for managing and updating the sprint backlog throughout the sprint

3. Detailed and clear:

Each backlog items is broken into smaller tasks with sufficient detail for the team to understand and work on them.

4. Dynamic but stable:

The sprint backlog may evolve slightly as tasks are refined , but the scope of the sprint goal should not change.

5. Linked to the sprint goal:

Every item in the sprint backlog contributes directly to achieving the sprint goal.

**13. Explain about product grooming?**

Product grooming also known as backlog refinement is a collaborative agile practice aimed at preparing and prioritizing the product backlog to ensure that it is well organized, up-to date and ready for upcoming sprints.

The goal includes clarity requirements, prioritize work, estimate effort, remove obsolete items, and break down large items.

Benefits of product grooming include improved clarity, smoother sprint planning, increased productivity, focus on value.

**14. Explain sprint size and scrum size:**

Sprint size is refers to the amount of work that scrum team commits to completing during a single sprint. It typically measured in terms of:

1. **Story points:**

A relative measures of complexity, effort or size for backlog items.

1. **Hours:**

The estimated time is required to complete the tasks.

1. **number of tasks/user stories:**

A count of items the team plans to complete

**Scrum size:**

Scrum size refers to the composition and capacity of the scrum team, encompassing the number of team members, their roles and their collective ability to deliver work within a sprint.

While there is no strict rule for scrum team size, certain guidelines exist to ensure efficiency and collaboration.

**15. Explain DOR and DOD:**

Definition of ready (DOR):

It is asset of criteria that a user story or product backlog item must meet before it can be considered ready for the team to start working on it during a sprint.

Essentially it is a checklist that ensures the backlog item is sufficiently clear, detailed and understood by the team so that they can begin development without any ambiguity or uncertainty

**Common criteria** for DOR includes clear acceptance criteria, well-defined user story, dependencies identified, estimate, prioritized, no blockers, stakeholder availability

**Definition of Done (DOD):**

It is asset of criteria that must be met for a product backlog item

E.g.: user story, features or an entire sprint to be considered complete.

The DOD ensures that all work is fully finished, meets quality standards and is ready for release or deployment.

It provides clarity on what constitutes done, reducing ambiguity and ensuring consistency in quality across the scrum team.

**Common criteria** for DOD include code complete, peer review, automated tests, manual testing, no critical bugs, integrated, documentation complete performance standards met, no pending tasks, reviewed by product owner, ready for release.

**16. Derive boundary classes, controller classes, entity classes**

**Boundary class:**

* A line that separates classes in a dataset or to a class that is on the edge of a system
* A class boundary is the dividing line between two adjacent classes in a dataset
* These are important for data analysis and interpretation.
* It interacts with actors outside the system as well as objects within the system.

|  |  |
| --- | --- |
| **Boundary class-(use cases)actors speak to the system for authentication** **information**   * Combination of 1 actor and use case 1 boundary class * Combination of 2 actor and use case 2 boundary class * Combination of 3 actor and use case 3 boundary class | **Customer registration**   * Customer login * Bank server login * Customer logout * bank server logout |

**Controller class:**

Managing the flow of data between different parts of the system, including user interactions, data access and business rules, without directly storing data itself

|  |  |
| --- | --- |
| **Controller class-(handles users(primary actors)input and processes the data**   * Use case will consider as controller class system | **Registration controller**   * Login controller * Payment controller * Credential controller |

**Entity class:**

* It is a class that represents a table in a database
* It also be a collection of entities that share the same attributes

|  |  |
| --- | --- |
| **Entity class-all actors**   * Each actor will be considered as an entity | **Actors**   * Customer * Bank server * Cash * Card * Net banking |

**17. List the challenges faced in projects for BA**

* Lack of training
* Change management
* Obtaining sign-off on the requirement
* Conducting meetings
* Co-ordination between developers and testers
* Driving clients for UAT completion
* Making sure status report is effective
* Making sure that the project is going on right track and delivered as per the timelines without any issues
* Unable to understand what stakeholder is trying to convey
* Scope creep-change in requirement or scope of the project during the project lifecycle can lead to scope creep
* Managing the stakeholder with conflicting interest can be a difficult task for BA
* BA may face difficulties in understanding the requirements if the domain is not familiar to him
* Poor communication between stakeholder and BA can affect the process of gathering the information
* Technical complexity

**18. Write the difference packages and sub-systems**

**Packages:**

* It is a group of classes or use cases that are used to organize model elements.
* Packages can be nested within other packages
* These are used as containers to organize elements
* It is very useful to represent system architecture

**Subsystem:**

* It is logical grouping of related components.
* It is collection of classes, packages, libraries and other sub systems that work together to deliver a specific set of functionalities

**19. What is camel-casing and explains where it will be used**

Camel- casing refers to the naming convention of variable, parameters or properties. Here, multiple words are combined together.

In camel-casing, the starting letter of first word starts with small letter and other words first letter starts with capital letters.

Ex-first Name, last Name

In BA, camel-casing is used in requirements documentation

In requirement documentation, BA often use camel-casing to name the entities like use case, features, user stories like validate customer details, calculate interest rate etc..,

Business rules which should be satisfied by the system use camel-casing.

While documenting business process or work flows, camel-casing can be used to individual in steps.

This will help maintain consistency in the document.

The database tables name also uses camel-casing.

Requirement naming-camel casing is used in requirement document also, to name the functional and non-functional requirements.

By using camel-casing in the documents, it helps to maintain consistency in the entire document and also increase readability.

**20. What is Data Mapping?**

The database contains multiple tables in it

Scenarios like, where we need to map the data from on e table to another.

Data mapping is necessary in cases where we want quick manner.

Data mapping is nothing but a process to establish connection between multiple data sources.

The purpose of data mapping is to ensure that the data is accurately transferred or converted into different format.

**The main purpose of data mapping is:**

**Data integration:** while combining the data from different sources, it ensures that the data is properly matched.

**Data migration:**

While migrating the data from legacy system (source) to the new system (destination), the data elements are mapped accurately into the new system

Required technique techniques are applied to convert the data into the format that is required by the new system.

**Data transformation:**

Data transformation means converting the data from one format to other, in data mapping, data transformation plays very important role which ensures that the data of legacy system (source) is mapped correctly to the data in new system (destination)