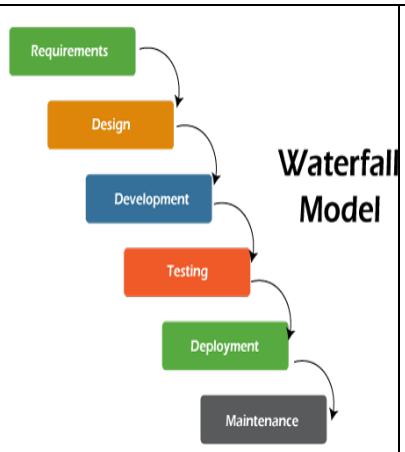
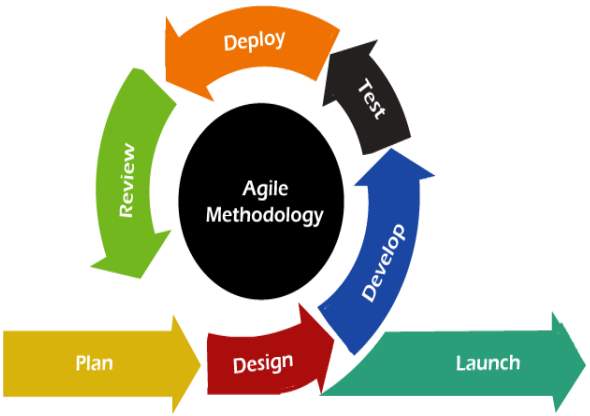
Technical round Ques/ Ans

Que 1: - explain the difference between waterfall and agile methodologies. How do they impact the role of a business analyst?

Ans: -

Basis	Waterfall Model	Agile Model
Definition	• Waterfall model follows a sequential design process.	• Agile model follows a incremental approach, where each incremental part is developed through iteration after every timebox.
Nature	• Waterfall model is rigid as it does not allow to modify once the development process starts.	• Agile model is flexible as it allows to modify after the development process starts.
Team size	• In waterfall model, team may consist more members.	• In agile model, the team size is small so that they can move faster.
Customer Interaction	• Customer interaction is very less in this model. Because the product is delivered to the customer after overall development.	• There is a high interaction with the customers. Because after every iteration, an incremental version is deployed to the customer.
Suitability	• Waterfall model is works well in smaller size projects where requirements are easily understandable. • But this model is not suitable for developing the large projects.	• Agile model is not suitable for small projects. The expenses of developing the small projects using agile is more than compared to other models.

Working Diagram		
Conclusion	In waterfall model, quality is managed through detailed planning and extensive testing at the end of the development phase.	In agile model, continuous integration and testing are employed to ensure the quality throughout the development process.

- **Impact of business analyst on these methods: -**

  In Waterfall model, the Business Analyst's impact is most pronounced during the initial phases of the project, where through requirements and gathering and documentation are essential. The Business Analyst's ability to manage changes effectively is also crucial, and given the rigidity of the waterfall model.

  In Agile model, the Business Analyst's impact is **continuous** throughout the project, with a focus on collaboration, flexibility, and adaptability. Their role in maintaining and prioritizing the backlog, writing user stories, and facilitating. Agile practices directly influence the success of the project in a fast paced, iterative environment.

Que 2: -What is the use case, and how do you create one? can you provide an example from your past projects?

Ans: - Use Case is a tool used to capture and describe how a system, process, or business function interacts with user or other systems to achieve a specific business goal. Use Cases helps in understanding and documenting functional requirements, facilitating communication between stakeholders, and ensuring that the final solution meets the needs of the business.

- **Steps for creating a use case in business analyst: -**

1. **Identify Stakeholders and Actors:**

Determine who will interact with the system or process. Actor can be human user, other systems, or external entities like vendors or regulators.

2. **Understand the Business Goal:**
   Clearly define what the business is trying to achieve with the use case. This goal should be aligned with business objectives and address a specific need or problem.

3. **Gather and Define Requirements:**
   Work with stakeholders to gather and document the functional requirements that the use case will address. Ensure these requirements are clear, measurable, and achievable.

4. **Outline Preconditions and Triggers:**
   Identify any prerequisites that must be in place before the use case can begin. Also, determine what event or condition triggers the start of the use case.

5. **Describe the Basic Flow:**
   Write out the sequence of steps that the actors and the system will take to achieve the business goal. This is the "happy path" where everything proceeds as expected.

6. **Identify and Document Alternate Flow:**
   Consider all the different ways the use case could play out.

7. **Determine Postconditions:**
   Define what the system should look like once the use case is completed. this could be involve updating records sending notifications, or changing the status of a transaction.

8. **Review With Stakeholder:**
   Present the use case to stakeholders to ensure it accurately reflects the business requirements and processes. Make any necessary revision based on their feedback.

9. **Create Visual Models:**
   Depending on the complexity, you might create visual representations like use case diagrams to complement the written use case. This can help in better understanding the relationships and flows.

- **Example of using USE CASE:**
  - Title: - Approve a Loan Application

- Actor: - Loan Applicant, Loan Officer, Credit Bureau
- Preconditions: - The applicant has submitted a completed application.
- Trigger: - The applicant submits the loan application for approval.
- Basic Flow: -
    1. The loan officer reviews the submitted application.
    2. The system checks the applicant's credit score by querying the credit bureau.
    3. The loan officer assesses the credit score and application details.
    4. The loan officer approves or rejects the loan application.
    5. The system notifies the application of the decision.
- Alternate Flows:
    1. If the credit score is below a certain threshold, the system automatically rejects the application.
    2. If the application is incomplete, then the system prompts the loan officer to request additional information from the applicant.

- Postconditions: - The loan application is either approved or rejected, and the applicant is informed of the outcome.

- Exceptions: - if the credit bureau service is unavailable, the system notifies the loan officer to perform a manual credit check.

Que 3: - How do you ensure that requirements are testable and measurable?

Ans: - There are so many ways to ensure that requirements are testable and measurable.

- One of the ways is to use the SMART criteria. which stands for Specific, Measurable, Achievable, Relevant, and Time-bound. These criteria help to define the requirements in a precise and realistic way, and to avoid subjective terms.
- Next is, BE SPECIFIC. Requirements should be specific and not leave for interpretation or assumption. They should also be measurable with a quantifiable or observable indicator.
- BE CONSISTENT, requirements should use a consistent vocabulary. If different groups within the same business use different terminology for the same ideas, activities, or things, it may be necessary to introduce a standard glossary.
- Consider the testing ability, when implementing a new feature, the development team should consider the testing ability and get input from testers to ensure efficient testing.
- Validate and Verify, Validation means checking that the requirements meet the needs and expectations of the stakeholders.
  Verification means checking that the requirements are feasible, logical, and consistent.

Que 4: - Explain the concept of user stories. How do they contribute to the agile development process?

Ans: -User Stories are a key tool in agile development for capturing user requirements in a simple, actionable way, driving collaboration, and ensuring that the development process remains flexible and user- focused.

User stories are a fundamental component of the agile development process, particularly in frameworks like Scrum and kanban. A user story is a short, simple description of a feature or functionality from the perspective of the end-user or customer.

- Structure of a story user: -
  User story follows a simple template.
  As a [type of user], I want [an action/feature], so that [a benefit/value].

- Contribution of User Story: -
  1. **Focus on user needs: -**
     User stories ensure that development is driven by the needs and experiences of the end user, leading to more user-centered products.

  2. **Incremental development: -**
     User stories are typically small and manageable, allowing teams to deliver features incrementally, which aligns with the agile principle of iterative progress.

  3. **Collaboration: -**
     Writing and discussing user stories fosters collaboration between developers, testers, product owners, and other stakeholders, ensuring everyone has a shared understanding of what needs to be built.

  4. **Flexibility: -**
     User stories can be easily adjusted or reprioritized based on feedback or changing business needs, allowing the team to remain agile and responsive.

  5. **Continuous improvement:** -
     As a team completes user stories and receives feedback, they can continually improve the product and process, which is a core aspect of Agile.

Que 5: - what is the purpose of data flow diagram (DFD), and how do you use it in the context of requirements analysis?

Ans: - DFD is a valuable tool for visualizing and analyzing how data is processed and exchanged within a system, which aids in gathering, validating, and refining requirements. It helps in understanding and analyzing the data processing and interactions between different components of a system.

- **Purpose of DFDs:** -
    1. **Visual Representation:** - DFDs provide a clear visual representation of the system's data processes, data storage, and data flow. This makes it easier to understand complex processes and interactions.
    2. **Communication:** - They facilitate communication between stakeholders, such as business analysts, developers, and clients, by providing a common understanding of how data moves through the system.
    3. **Requirement Analysis:**- DFDs help in identifying and clarifying requirements by showing how data is processed, where it comes from, and where it goes. This can help in spotting gaps, redundancies, or inconsistencies in the requirements.
    4. **System Design:** - They are useful in the design phase to ensure that the system's architecture will meet the specified requirements by mapping out how different components interact.

- **DFDs as in Requirements Analysis:** -
    1. **Identify Entities:** - Start by identifying external entities that interact with the system, such as users or other systems.
    2. **Define Processes:** - Determine the processes that transform input data into output.
    3. **Determine Data Flow:** - Arrows are used to represent the flow of data, which helps in understanding the sequence and dependencies.
    4. **Establish Data Stores:** - Identify and represent data stores where information is held within the system.
    5. **Create Hierarchical Diagram: -** Start with a high-level DFD and then break it down into more detailed diagrams to represent more specific aspects of the system.
    6. **Review and Validate:** - Use the DFDs to review and validate the requirements with stakeholders. This ensures that all requirements are captured and correctly understood.
    7. **Refine Requirements: -** Based on feedback, refine the requirements and adjust the DFDs accordingly. This iterative process helps in capturing accurate and comprehensive requirements.

Que 6: -create ER Diagram for a flight booking app?

Ans: -Creating an Entity-Relationship (ER) Diagram for a flight booking app involves identifying the key entities involved in the booking process, their attributes, and the relationships between these entities.

Here's how the ER Diagram might look conceptually

- Entities and Attributes:

1. **User (Customer)**
    - User_id (PK)
    - Name
    - Email
    - Phone_number
    - Password
2. **Flight**
    - Flight_id
    - Flight_number
    - Airline_name
    - Departure_time
    - Arrival_time
    - Departure_Airport
    - Arrival_Airport
    - Duration
    - Status
3. **Airport**
    - Airport id
    - Name
    - City
    - Country
4. **Booking**
    - Booking id (PK)
    - Booking date
    - Total Price
    - Payment status
5. **Payment**
    - Payment id (PK)
    - Payment date
    - Amount
    - Payment method
    - Payment status
6. **Ticket**
    - Ticket id (PK)

- Price

7. **Passenger**
    - Passenger id (PK)
    - Name
    - Passport number
    - Gender
    - Age

Now we have a EER Diagram of flight booking app: -