**Question 1 - write Agile Manifesto**

The Agile Manifesto is statement of Agile values and Principles that summarizes the thinking about agile perspectives and methodologies designed to improve software development by promoting flexibility, collaboration, and responsiveness to change. Created in 2001, it stand up as an alternative to traditional, rigid project management methods that struggled to adapt to fast-changing requirements. The Agile Manifesto encourages teams to focus on delivering functional software, involve the customer throughout the process, and be open to adapting plans when needed.

**Four Core Values of the Agile Manifesto:**

1. **Individuals and Interactions over processes and tools:**

Even though processes and tools are necessary for the project to visually represent but the Individual and Interaction is necessary than processes and tool, because Individual and Interactions adds business needs via Interaction.

1. **Working Software over comprehensive documentation:**

In traditional development, enormous time were spent on documentation and ultimate delivery. This value reminds us of working values that provides business end user values rather than so much of documentation

1. **Customer Collaboration over contract negotiation:**

We want to see collaborative solution to issue that might arise during product development rather than to stick to rigid contract. Working closely with customers ensures that the product meets their needs

1. **Responding to Change over following a plan:**

Needs and requirement are always shifting and Priorities are always Changing. Suggests that the software team should be able to pivot and change direction whenever needed and embrace flexible roadmap versus Locked in plan.

**12 Principles of Agile:**

1. **Customer satisfaction:** Our highest priority is to satisfy the customer through early and continuous delivery of valuable software
2. **Welcome changing requirements**: Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage
3. **Deliver working software frequently:** Deliver working software frequently from couple of months with a preference to the shorter timescale
4. **Business people and developers must work together:** Business people and developers must work together daily throughout the project.
5. **Build projects around motivated individuals:** Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. **Face-to-face conversation**: The most efficient and effective method of conveying information to and within a development team is face-to-face
7. **Working software:** Working software is the primary measure of progress.
8. **Sustainable development:** Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. **Technical excellence and good design**: Continuous attention to technical excellence and good design enhances agility.
10. **Simplicity:** Technical excellence and good design: Simplicity–the art of maximizing the amount of work not done–is essential.
11. **Self-organizing teams:** The best architectures, requirements, and designs emerge from self-organizing teams.
12. **Regular reflection:** At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

**Question 2 – User Stories- Acceptance Criteria-BV-CP Write minimum 40 User stories and their Acceptance Criteria along with their BV and CP**

In Agile, user stories are short, simple descriptions of a feature or functionality that is valuable to the end user. They are written from the perspective of the user or customer and focus on their needs and goals.

A user story typically follows a simple format: As a [type of user], I want [an action] so that [I can achieve a goal].

**Key Characteristics of User Stories:**

1. **INVEST Criteria:** Independent: Can be developed, tested, and delivered independently. o Negotiable: The details of the user story can be discussed and adjusted. o Valuable: Provides value to the user or the business. o Estimable: It should be possible to estimate the effort required to complete the story. o Small: User stories should be small enough to be completed within a sprint or iteration. o Testable: The story should include acceptance criteria that are testable.
2. **Acceptance Criteria:** These are conditions that must be met for the user story to be considered complete and successfully implemented. They provide more detailed descriptions of the behaviour or functionality.
3. **BV (Business Value):** The team would assign a business value to each feature based on how important it is to the business goals.
4. **CP (Critical Path):** The critical path identifies the sequence of tasks that must be completed on time to ensure the project is delivered on time.

**User Story 1:**

|  |  |
| --- | --- |
| **User Story ID: 001**  **Task:** Create login functionality, validate user credentials, integrate authentication system  **Priority:** | |
| **Value Statement** | |
| **BV: 500** | **CP: 5** |
| **Acceptance criteria:**   1. **Enter valid credentials.** 2. **2. Click "Login".** 3. **Redirect to personalized homepage on successful login.** 4. **Display error on invalid credentials.** | |

**User Story 2:**

|  |  |
| --- | --- |
| **User Story ID:002**  **Task: Develop search feature, integrate filter options for cuisine, price, andratings**  **Priority: High** | |
| **Value Statement:** As a user, I want to search and filter food items to find my preferred options. | |
| **BV: 500** | **CP: 8** |
| **Acceptance criteria:**  **1. Enter keywords or select filters.**  **2. Click "Search".**  **3. Display filtered results based on chosen criteria.**  **4. Ensure relevant results are shown.** | |

**User Story 3:**

|  |  |
| --- | --- |
| **User Story ID:003**  **Task: Implement user profile management, add options to updatepersonal details.**  **Priority:** Medium | |
| **Value Statement**: As a user, I want to manage my profile to keep my information updated. | |
| **BV: 100** | **CP: 5** |
| **Acceptance criteria:**  **Basic Flow**  **Alternate Flow:**  **Exceptional Flow** | |

**User Story 4:**

|  |  |
| --- | --- |
| **User Story ID: 004**  **Task: Enable password reset via email verification.**  **Priority:** High | |
| **Value Statement:** As a user, I want to reset my password if I forget it. | |
| **BV:** 500 | **CP:** 2 |
| **Acceptance criteria:**  1. Click "Forgot Password".  2. Enter registered email.  3. Receive verification email.  4. Follow link to set a new password | |

**User Story 6:**

|  |  |
| --- | --- |
| **User Story ID: 006**  **Task: Set up notifications for order status updates.**  **Priority:** High | |
| **Value Statement:** As a user, I want to receive notifications for my order status to stay informed. | |
| **BV:** 500 | **CP: 3** |
| **Acceptance criteria:**  1. Place an order.  2. Receive notifications at each stage (order placed, dispatched, delivered).  3. View notifications in the app notification panel. | |

**User Story 7:**

|  |  |
| --- | --- |
| **User Story ID: 007**  **Task: Integrate payment gateway for secure transactions.**  **Priority:** High | |
| **Value Statement:** As a user, I want a secure way to make payments for my orders. | |
| **BV:** 500 | **CP: 3** |
| **Acceptance criteria:**  1. Add items to cart.  2. Proceed to checkout.  3. Choose a payment method.  4. Enter payment details and complete transaction.  5. Receive confirmation of successful payment | |

**User Story 8:**

|  |  |
| --- | --- |
| **User Story ID: 008**  **Task: Implement a referral system for users to invite friends.**  **Priority:** Medium | |
| **Value Statement:** As a user, I want to refer friends to earn rewards. | |
| **BV:** 100 | **CP: 3** |
| **Acceptance criteria:**  1. Access referral section.  2. Copy referral link or enter friend's email.  3. Friend receives invitation and registers.  4. User receives reward on friend's first order. | |

**User Story 9:**

|  |  |
| --- | --- |
| **User Story ID: 008**  **Task: Add cart functionality to store selected items before checkout.**  **Priority:** High | |
| **Value Statement:** As a user, I want to add items to my cart and review them before placing an order. | |
| **BV:** 500 | **CP: 2** |
| **Acceptance criteria:**  1. Access referral section.  2. Copy referral link or enter friend's email.  3. Friend receives invitation and registers.  4. User receives reward on friend's first order. | |

**User Story 10:**

|  |  |
| --- | --- |
| **User Story ID: 010**  **Task: Implement order history for users to view past orders.**  **Priority:** Medium | |
| **Value Statement:** As a user, I want to view my past orders for reference and reordering. | |
| **BV:** 100 | **CP: 3** |
| **Acceptance criteria:**  1. Access order history.  2. View past orders with details (date, items, amount).  3. Option to reorder items from history. | |

**User Story 11:**

|  |  |
| --- | --- |
| **User Story ID: 011**  **Task: Enable users to track delivery in real-time on a map.**  **Priority:** High | |
| **Value Statement:** As a user, I want to refer friends to earn rewards. | |
| **BV:** 500 | **CP: 3** |
| **Acceptance criteria:**  1. Place an order.  2. Access order tracking.  3. View delivery vehicle location on map.  4. Show ETA based on live location updates. | |

**User Story 12:**

|  |  |
| --- | --- |
| **User Story ID: 012**  **Task: Add restaurant rating and review functionality.**  **Priority:** Medium | |
| **Value Statement:** As a user, I want to rate and review restaurants after my order is complete | |
| **BV:** 100 | **CP: 3** |
| **Acceptance criteria:**  1. Complete an order.  2. Access review section for the restaurant.  3. Rate on a scale of 1-5 and add comments.  4. Submit review and view it on the restaurant's page. | |

**User Story 13:**

|  |  |
| --- | --- |
| **User Story ID: 013**  **Task: Develop a sorting feature to sort restaurants by ratings, distance or popularity.**  **Priority:** Medium | |
| **Value Statement:** As a user, I want to sort restaurant listings based on ratings, distance, or popularity. | |
| **BV:** 100 | **CP: 3** |
| **Acceptance criteria:**  1. Select sorting criteria.  2. Display sorted results based on chosen criteria.  3. Verify that results match selected sort option. | |

**User Story 14:**

|  |  |
| --- | --- |
| **User Story ID: 014**  **Task: Implement push notifications for special offers and discounts**  **Priority:** Medium | |
| **Value Statement:** As a user, I want to receive notifications about offers and discounts to save on orders | |
| **BV:** 100 | **CP: 5** |
| **Acceptance criteria:**  1. User opts into notifications.  2. Receive timely notifications for offers.  3. Tap notification to view offer details in the app. | |

**User Story 15:**

|  |  |
| --- | --- |
| **User Story ID: 012**  **Task: Create delivery partner registration and verification process**  **Priority:** High | |
| **Value Statement:** As a delivery partner, I want to register and verify my account to start delivering orders. | |
| **BV:** 500 | **CP: 5** |
| **Acceptance criteria:**  1. Fill out registration form.  2. Upload verification documents.  3. Receive approval notification on successful verification.  4. Access delivery partner dashboard. | |

**User Story 16:**

|  |  |
| --- | --- |
| **User Story ID: 016**  **Task: Implement earnings tracker for delivery partners**  **Priority:** Medium | |
| **Value Statement:** As a delivery partner, I want to view my earnings to track my income. | |
| **BV:** 100 | **CP: 3** |
| **Acceptance criteria:**  1. Access earnings section.  2. View daily and weekly earnings.  3. Breakdown of earnings per order.  4. View total income for selected date range. | |

**User Story 17:**

|  |  |
| --- | --- |
| **User Story ID: 017**  **Task: Add order assignment feature for delivery partners**  **Priority:** High | |
| **Value Statement:** As a delivery partner, I want to receive assigned orders to start the delivery process | |
| **BV:** 500 | **CP: 5** |
| **Acceptance criteria:**  1. Receive new order notification.  2. Accept or reject order.  3. Confirm order pick-up and start delivery.  4. Mark order as delivered upon completion. | |

**User Story 18:**

|  |  |
| --- | --- |
| **User Story ID: 018**  **Task: Set up customer support contact options (chat, email,phone)**  **Priority:** Medium | |
| **Value Statement:** As a user, I want various contact options for customer support in case of issues. | |
| **BV:** 100 | **CP: 2** |
| **Acceptance criteria:**  1. Access support section.  2. Choose chat, email, or phone.  3. Initiate conversation or contact support.  4. Receive response confirmation. | |

**User Story 19:**

|  |  |
| --- | --- |
| **User Story ID: 019**  **Task: Implement estimated delivery time display for each restaurant**  **Priority:** High | |
| **Value Statement:** As a user, I want to see an estimated delivery time for each restaurant to plan my order. | |
| **BV:** 500 | **CP: 5** |
| **Acceptance criteria:**  1. View estimated time on restaurant page.  2. Confirm accurate time calculation based on location.  3. Time updates based on real-time conditions | |

**User Story 20:**

|  |  |
| --- | --- |
| **User Story ID: 017**  **Task: Create profile management for restaurants to update menus and information**  **Priority:** Medium | |
| **Value Statement:** As a restaurant owner, I want to update my restaurant details to ensure accurate listings. | |
| **BV:** 100 | **CP: 2** |
| **Acceptance criteria:**   1. Log into restaurant profile. 2. Edit menu items, hours, and contact details. 3. Save updates. 4. View confirmation message after successful updates. | |

**User Story 21:**

|  |  |
| --- | --- |
| **User Story ID: 021**  **Task: Implement loyalty program for returning users.**  **Priority:** Medium | |
| **Value Statement:** As a user, I want to earn loyalty points on each order to redeem rewards. | |
| **BV:** 100 | **CP: 2** |
| **Acceptance criteria:**   1. Place an order. 2. Earn points on order completion. 3. Track points in profile. 4. Redeem points for discounts on future orders. | |

**User Story 22:**

|  |  |
| --- | --- |
| **User Story ID: 022**  **Task: Enable order cancellation option before food preparation starts**  **Priority:** High | |
| **Value Statement:** As a user, I want to cancel my order if needed, as long as food preparation hasn't started. | |
| **BV:** 500 | **CP: 5** |
| **Acceptance criteria:**   1. Access order status. 2. Cancel order if prep hasn't begun. 3. Receive refund confirmation. 4. View updated status as "Cancelled." | |

**User Story 23:**

|  |  |
| --- | --- |
| **User Story ID: 023**  **Task: Add user profile for storing preferences and order history**  **Priority:** Medium | |
| **Value Statement:** As a user, I want a profile to save my food preferences and view my order history. | |
| **BV:** 100 | **CP: 2** |
| **Acceptance criteria:**   1. View profile section. 2. Access food preferences and order history. 3. Edit preferences. 4. Confirm changes saved successfully. | |

**User Story 24:**

|  |  |
| --- | --- |
| **User Story ID: 024**  **Task: Set up delivery fee calculations based on distance**  **Priority:** High | |
| **Value Statement:** As a user, I want the delivery fee to be calculated based on distance to understand the cost. | |
| **BV:** 500 | **CP: 5** |
| **Acceptance criteria:**   1. Enter delivery address. 2. View calculated fee based on distance. 3. Confirm accurate fee added to total. 4. Complete order with correct fee applied | |

**User Story 25:**

|  |  |
| --- | --- |
| **User Story ID: 025**  **Task: Provide real-time delivery status updates to customers.**  **Priority:** High | |
| **Value Statement:** As a user, I want to receive real-time status updates on my order delivery. | |
| **BV:** 500 | **CP: 5** |
| **Acceptance criteria:**   1. Place an order. 2. Receive real-time notifications for each stage (e.g., preparation, dispatch). 3. Access delivery status updates. 4. View notification when the order is delivered. | |

**User Story 26:**

|  |  |
| --- | --- |
| **User Story ID: 026**  **Task: Integrate payment gateway for secure transactions.**  **Priority:** High | |
| **Value Statement:** As a user, I want secure payment options to complete my order transaction | |
| **BV:** 500 | **CP: 3** |
| **Acceptance criteria:**   1. Choose payment method. 2. Enter payment details securely. 3. Receive confirmation of successful payment. 4. View updated order status after payment. | |

**User Story 27:**

|  |  |
| --- | --- |
| **User Story ID: 027**  **Task: Allow multiple addresses to be saved in user profile.**  **Priority:** Medium | |
| **Value Statement:** As a user, I want to save multiple addresses to quickly select for future orders | |
| **BV:** 100 | **CP: 2** |
| **Acceptance criteria:**   1. Add new address in profile. 2. Save multiple addresses. 3. Select address during checkout. 4. Confirm correct address is applied to the order. | |

**User Story 28:**

|  |  |
| --- | --- |
| **User Story ID: 028**  **Task: Create an order summary with detailed breakdown.**  **Priority:** Medium | |
| **Value Statement:** As a user, I want to view a detailed order summary before confirming my order. | |
| **BV:** 100 | **CP: 2** |
| **Acceptance criteria:**   1. Review items in the cart. 2. View itemized cost breakdown. 3. Confirm order summary matches items. 4. Complete purchase after reviewing. | |

**User Story 29:**

|  |  |
| --- | --- |
| **User Story ID: 029**  **Task: Add feature for restaurant owners to manage their menu.**  **Priority:** High | |
| **Value Statement:** As a restaurant owner, I want to manage my menu to keep offerings up-to-date | |
| **BV:** 500 | **CP: 3** |
| **Acceptance criteria:**   1. Log into restaurant profile. 2. Edit menu items and prices. 3. Save updates. 4. View updated menu on customer side. | |

**User Story 30:**

|  |  |
| --- | --- |
| **User Story ID: 030**  **Task: Implement feedback collection post-delivery for improvement.**  **Priority:** High | |
| **Value Statement:** As a user, I want to provide feedback after delivery to help improve service. | |
| **BV:** 500 | **CP: 5** |
| **Acceptance criteria:**   1. Receive prompt to rate order after delivery. 2. Provide feedback and submit. 3. View confirmation of feedback submission. 4. Feedback stored in system for review. | |

**User Story 31:**

|  |  |
| --- | --- |
| **User Story ID: 031**  **Task: Enable multiple payment methods (credit card, debit card, UPI,wallet)**  **Priority:** High | |
| **Value Statement:** As a user, I want multiple payment options for convenience in completing transactions | |
| **BV:** 500 | **CP: 3** |
| **Acceptance criteria:**   1. Select payment method at checkout. 2. Complete payment using chosen method. 3. Receive transaction success notification. 4. View updated order status. | |

**User Story 32:**

|  |  |
| --- | --- |
| **User Story ID: 032**  **Task: Integrate GPS for accurate restaurant and customer locationtracking..**  **Priority:** High | |
| **Value Statement:** As a user, I want GPS-enabled location tracking to ensure precise delivery locations. | |
| **BV:** 500 | **CP: 3** |
| **Acceptance criteria:**   1. Enable GPS on device. 2. Confirm restaurant and customer locations on map. T 3. Track delivery location in real time. 4. Receive accurate ETA based on live location.. | |

**User Story 33:**

|  |  |
| --- | --- |
| **User Story ID: 033**  **Task: Set up order history with filter options (date, status).**  **Priority:** Medium | |
| **Value Statement:** As a user, I want to view my order history and filter results to easily find past orders. | |
| **BV:** 100 | **CP: 2** |
| **Acceptance criteria:**   1. Access order history. 2. Apply filters (date, status). 3. View filtered list. 4. Confirm accurate results based on selected filters | |

**User Story 34**

|  |  |
| --- | --- |
| **User Story ID: 034**  **Task: Add option for delivery instructions (e.g., leave at door).**  **Priority:** Medium | |
| **Value Statement:** As a user, I want to add special delivery instructions for added convenience. | |
| **BV:** 100 | **CP: 3** |
| **Acceptance criteria:**   1. Enter special instructions at checkout. 2. Confirm instructions appear on delivery partner's app. 3. View confirmation message that instructions were saved. | |

**User Story 35**

|  |  |
| --- | --- |
| **User Story ID: 035**  **Task: Implement customer referral program**  **Priority:** Low | |
| **Value Statement:** As a user, I want to add special delivery instructions for added convenience. | |
| **BV:** 50 | **CP: 3** |
| **Acceptance criteria:**   1. Share referral link. 2. Track friend sign-ups through link. 3. Receive referral rewards upon friend's first order. 4. View earned rewards in profile. | |

**User Story 36**

|  |  |
| --- | --- |
| **User Story ID: 036**  **Task: Add in-app chat support for delivery queries.**  **Priority:** High | |
| **Value Statement:** As a user, I want in-app chat support to quickly resolve delivery-related issues | |
| **BV:** 500 | **CP: 5** |
| **Acceptance criteria:**   1. Access chat support during an active order. 2. Send message and receive response. 3. Track issue resolution status. 4. Confirm chat support helps in resolving issue. | |

**User Story 37**

|  |  |
| --- | --- |
| **User Story ID: 037**  **Task: Implement reward points redemption option at checkout**  **Priority:** Medium | |
| **Value Statement:** As a user, I want to redeem my reward points during checkout to get discounts on orders | |
| **BV:** 100 | **CP: 3** |
| **Acceptance criteria:**   1. Select reward points option at checkout. 2. Confirm discount applied. 3. Complete payment with adjusted total. 4. View redeemed points deducted from balance. | |

**User Story 38**

|  |  |
| --- | --- |
| **User Story ID: 038**  **Task: Provide estimated preparation time for orders.**  **Priority:** High | |
| **Value Statement:** As a user, I want an estimated preparation time to plan when to expect my order. | |
| **BV:** 500 | **CP: 3** |
| **Acceptance criteria:**   1. Place order. 2. View estimated preparation time before confirming. 3. Confirm accuracy of preparation time on completion. 4. Receive updated time if preparation exceeds estimate. | |

**User Story 39**

|  |  |
| --- | --- |
| **User Story ID: 039**  **Task: Enable scheduling future orders.**  **Priority:** Medium | |
| **Value Statement:** As a user, I want to schedule my order for a future time for convenience | |
| **BV:** 100 | **CP: 3** |
| **Acceptance criteria:**   1. Choose delivery time while placing order. 2. Confirm selected time. 3. Order scheduled and confirmed. 4. Receive notification before delivery time. | |

**User Story 40**

|  |  |
| --- | --- |
| **User Story ID: 040**  **Task: Implement option to reorder from order history.**  **Priority:** Medium | |
| **Value Statement:** As a user, I want a quick option to reorder previous items for convenience | |
| **BV:** 100 | **CP: 3** |
| **Acceptance criteria:**   1. Access order history. 2. Scroll through it 3. Get which item you have order in past 4. Do order again | |

**Question 3- What is epic? Write 2 epics Business Value and Complexity Points**

An epic is a high level feature, which contains multiple user stories it is large enough that cannot be delivered as defined within a single iteration or is large enough that it can be split into smaller user stories

Login Module

User Profile

Order Management

**Following are the example of Epic:**

**Epic 1. Online Food Ordering and Checkout**

**Value Statement:**

As a user, I want to place an order and check out seamlessly to receive my food from the selected restaurant.

As a user, I want a profile to save my food preferences and view my order history.

**User Stories under this Epic might include:**

**Acceptance Criteria:**

* User can search for restaurants based on location and cuisine.
* User can add items to cart from a restaurant's menu.
* User can apply a promo code to get discounts on order.
* User can choose preferred payment method to complete order.

|  |
| --- |
| **User Story ID**: 001  **Task:** 3  **Priority:** 1 |
| **Value Statement:** As a User/Diner, I want to search for Restaurants based on location and Cuisine so that I can order Food Fastly and in less time |
| **BV:** 1000  **CP:** 5 |
| **Acceptance Criteria:**  **Basic Flow:**   * Open the app in Mobile or Browser * Search menu in “search bar” of App * Add filter to the nearest location * Ensure relevant results shown   **Alternate Flow:**   * User open App but only search for the Menu but didn’t add the filter for nearest location   **Exceptional flow:**   * If App id down due to technical difficulty then user will get prompt message like ‘seek assistance’ or ‘please retry after sometime’ * If there is network issue in user side then, user will be notified with “network is down in your area” |

|  |
| --- |
| **User Story ID**: 002  **Task:** 3  **Priority:** 1 |
| **Value Statement:** As a user, I want a profile to save my food preferences and view my order history |
| **BV: 800**  **CP: 7** |
| **Acceptance Criteria:**  **Basic Flow:**   * As a User, Open the app in Mobile or Browser * Goto Hamberger menu, Select “Your Order History” * It should show user, Highest number of dish that he has ordered or option for particular symbol or notation by which all favourite dishes it could list to user * Ensure relevant results shown   **Alternate Flow:**   * User select “Order History” but there is no list of favourite dish shown to customer * Or only it will show order History date wise only   **Exceptional flow:**   * If App id down due to technical difficulty, then user will get prompt message like ‘seek assistance’ or ‘please retry after sometime’ * If there is network issue in user side then, user will be notified with “network is down in your area” |

**Epic 2: Menu Browsing and Order placement**

**Value Statement:**

As a User, I want to browse the menu so that I can view and select the Food I want to order

As a User, I want to place to my order easily with customizable option (ex: quantity, toppings) so that I can get my meal exactly as I prefer.

**Acceptance Criteria:**

* Users can view the full menu of available food items.
* Users can filter and sort the menu by food categories (e.g., vegetarian, non-vegetarian, desserts, etc.).
* Users can add items to their cart with the option to modify the quantity or select customizations (e.g., toppings, spice levels).
* Users can view the estimated delivery time for their order.
* Users can place orders and choose delivery/pickup options.

|  |
| --- |
| **User Story ID**: 003  **Task:** 4  **Priority:** 2 |
| **Value Statement:** As a User, I want to browse the menu so that I can view and select the Food I want to order |
| **BV: 700**  **CP: 6** |
| **Acceptance Criteria:**  **Basic Flow:**   * As a User, Open the app in Mobile or Browser * Browse through the app to see today’s special menu * Either If user click on Restaurant, then It should depict the hotel menu * Or simply through scrolling in app Automatic menu should Pop up * Ensure relevant results shown   **Alternate Flow:**   * User select “Order History” but there is no list of favourite dish shown to customer * Or only it will show order History date wise only   **Exceptional flow:**   * If App id down due to technical difficulty, then user will get prompt message like ‘seek assistance’ or ‘please retry after sometime’ * If there is network issue in user side then, user will be notified with “network is down in your area” |

**Question 4 –What is the difference between BV and CP**

Business Value is about the benefit, while Complexity points are about the cost or effort. By understanding both, teams can prioritize initiatives that deliver the most value for the least effort.

Business Value and Complexity points represent different aspects of a product or feature initiative. They are often used in prioritization models to evaluate the trade-off between the value an initiative provides and the resources needed to deliver it.

In **Jira**, especially when used for **Agile project management** (Scrum or Kanban), the terms **Complexity Point** and **Business Value** often appear in backlog refinement or sprint planning. Here's what they mean:

**Business Value:**

* **Definition:** A measure of how valuable a task or feature is to the business or customer.
* **Used by:** Product Owner or stakeholders.
* **Purpose:** To prioritize backlog items based on **impact** to the business.
* **Scale:** Typically a numeric value (1–10 or 1–100), or sometimes labels like High/Medium/Low.
* **Example:**
  + A new feature that boosts user engagement may be rated 100.
  + A backend optimization may have a value of 30.

**Complexity Point (also known as Story Points):**

* **Definition:** A relative measure of how complex or time-consuming a task or user story is. This refers to the effort, risk, and uncertainty associated with completing a task or user story.
* **Used by:** Development team.
* **Purpose:** To estimate **effort**, **uncertainty**, and **risk** involved in completing a task.
* **Scale:** Often based on a **Fibonacci sequence** (1, 2, 3, 5, 8, 13...) or T-shirt sizes (S, M, L).
* **Example:**
  + A simple bug fix might be 1 point.
  + A feature with multiple dependencies could be 13 points.

🔑 **Note:** It’s **not directly tied to hours or days**—it’s a *relative* measure used to compare tasks against each other.

|  |  |  |
| --- | --- | --- |
|  | **Business Values** | **Complexity Point** |
| **Aspect** | Business Value is about the benefit to Business | Complexity points are about the cost or effort. |
| **Definition** | Business Value refers to the benefits it delivers to the business, such as revenue, cost savings, or market share. | Complexity points, on the other hand, quantify the effort and resources required to implement or develop that initiative |
| **Purpose** | To align development efforts with strategic business objectives and enhance customer satisfaction. | To assess the challenges and risks associated with implementing a feature or epic. |
| **Measure** | A measure of how valuable a task or feature is to the business or customer.  Can be quantified through metrics like ROI, customer feedback, or business impact analysis. | A relative measure of how complex or time-consuming a task or user story is. This refers to the effort, risk, and uncertainty associated with completing a task or user story. |
| **Stakeholder Perspective** | Used by Product Owner or stakeholders. | Used by Development team. |
| **Prioritization** | To prioritize backlog items based on **impact** to the business. | To estimate **effort**, **uncertainty**, and **risk** involved in completing a task. |
| **Measurement** | Measuring scale will be Typically a numeric value (1–10 or 1–100), or sometimes labels like High/Medium/Low. | Measuring scale will be Often based on a **Fibonacci sequence** (1, 2, 3, 5, 8, 13...) or T-shirt sizes (S, M, L). |
| **Techniques** | - **ROI (Return on Investment):** Measures potential financial return  - **Customer Feedback:** Direct input from users on the feature's usefulness  - **Business Impact Analysis**: Evaluates howa feature aligns with strategic goals | -**Story Points:** Relative estimation of effort based on complexity.  -**T-shirt Sizing:** Categorizes features into sizes (XS, S, M, L, XL) based on complexity.  - **Expert Judgment:** Experienced team members provide input on the expected complexity. |
| **Example** | - Adding a new feature to allow customers to track their delivery in real-time, increasing customer satisfaction.  - Offering discounts and promotions that drive more orders and higher revenue. | -Implementing a feature that allows multiple payment methods (credit cards, digital wallets, etc.), requiring complex integrations with payment gateways and extensive testing.  - Developing an algorithm to optimize delivery routes, involving complex calculations and data analysis. |

**Question 5 -Explain about Sprint**

A Sprint is a short, fixed period of time during which a Scrum team works to complete a specific amount of work. It’s a core concept in Agile and Scrum frameworks, mainly used in software development but also in other project types.

**Time-boxed iteration**

A Sprint is a time-boxed iteration (usually 1–4 weeks) in which a Scrum team builds a usable and potentially shippable product increment. In Scrum, a sprint is a timeboxed event, meaning it has a fixed duration, typically between one and four weeks. The sprint itself is not timeboxed in the sense of having a strict maximum duration, but rather it's a container for other timeboxed events within it. Sprint planning, for example, is timeboxed to a maximum of eight hours for a one-month sprint, with shorter sprints having shorter planning timeboxes.

**Here's a breakdown of timeboxing within a sprint and Key Element of Sprint:**

* + 1. **Sprint Planning:**
* Sprint planning is the kickoff meeting for each sprint, where the Scrum team determines the sprint goal (the primary objective for the sprint) and selects the product backlog items (user stories, tasks, or requirements) that they can realistically complete within the sprint's time frame.
* The team also discusses each selected backlog item, clarifies any doubts, and defines acceptance criteria to ensure everyone understands what is needed. o The team then estimates the work and commits to a manageable number of backlog items for the sprint.
  + 1. **Sprint Backlog:**
* The sprint backlog is the set of tasks, user stories, and requirements selected during sprint planning for the current sprint.
* It includes all work that needs to be completed to achieve the sprint goal, and it is maintained by the team to keep track of progress.
* The sprint backlog is a dynamic document that can be updated during the sprint, but the scope of the sprint should remain fixed to avoid disruption.
  + 1. **Daily Stand-up (Daily Scrum):**
* The daily stand-up is a short, time-boxed meeting (usually 15 minutes) held at the same time every day during the sprint.
* During the stand-up, team members discuss three key points:
* what they accomplished the previous day,
* what they plan to do today, and any obstacles they are facing.
* This meeting helps the team stay aligned, identify challenges early, and keep track of progress toward the sprint goal.
  + 1. **Development and Testing:**
* Throughout the sprint, the team actively works on the tasks and stories defined in the sprint backlog.
* Testing and quality assurance are integral to the process; as new features are developed; they are also tested to ensure they meet the acceptance criteria and are free of critical defects.
* The goal is to create a potentially shippable product increment by the end of the sprint, meaning all stories are complete and tested.
  + 1. **Sprint Review:**
* The sprint review is held at the end of each sprint to inspect and evaluate the work completed during the sprint.
* The team demonstrates the completed product increment to stakeholders, who provide feedback and discuss potential improvements or new requirements.
* The sprint review is an opportunity to gather valuable feedback from users and stakeholders, which can be used to refine the product backlog and prioritize future work.
  + 1. **Sprint Retrospective:**
* After the sprint review, the team conducts a sprint retrospective, a meeting focused on self-reflection and improvement.
* During the retrospective, team members discuss what went well, what didn't go well, and how they can improve in future sprints.
* Actionable steps are identified to enhance team collaboration, address issues, and improve the overall sprint process, making it more efficient and effective.

**Sprint Goal and Increment**

* **Sprint Goal:** Each sprint has a clear, overarching objective called the sprint goal, which is a specific outcome the team aims to achieve during the sprint. It provides the team with a focus and direction for the work they are completing.
* **Product Increment:** At the end of each sprint, the team delivers a product increment — a potentially shippable version of the product with new or enhanced functionality. This increment should meet the "Definition of Done" (DoD), meaning it is complete, tested, and ready to be used or reviewed.

**Benefits of Sprints**

1. **Iterative Development:** Sprints enable the team to develop in short, manageable cycles, allowing for iterative progress. This helps the team refine the product continuously based on feedback and changing requirements.
2. **Improved Focus:** With a defined sprint goal and a fixed sprint duration, the team can focus on completing a specific set of tasks without distractions, ensuring faster delivery of results.
3. **Frequent Feedback:** Sprints end with a review that includes stakeholders and product owners, providing opportunities for feedback that can directly influence future work and improve the product.
4. **Transparency:** The daily stand-ups, sprint reviews, and retrospectives increase visibility into the team's progress, challenges, and successes, fostering collaboration and accountability within the team.
5. **Flexibility and Adaptability:** While the sprint itself is not flexible (the scope is fixed once the sprint starts), the iterative nature of sprints allows the product and priorities to adapt between sprints based on feedback and market changes.

**Typical Sprint Timeline**

1. Day 1: Sprint Planning - Define sprint goal and select backlog items.
2. Daily: Stand-up Meetings - Team syncs on daily progress.
3. Last Day: Sprint Review - Demonstrate completed increment and gather feedback.
4. Last Day (After Review): Sprint Retrospective - Reflect on the sprint and identify improvement actions for the next sprint.

**Sprint Workflow Example**

Let's say the Sprint Goal is "Enable users to reset their password via email."

**Product Backlog Item (PBI):** As a user, I want to reset my password via email, so I can access my account if I forget it**.**

**Workflow Breakdown:**

|  |  |
| --- | --- |
| Status | Description |
| To Do | Tasks planned but not started yet |
| WIP (Work in Progress) | Tasks currently being worked on |
| Done | Tasks fully completed and meet the Definition of Done |

Visual Example (Simplified Scrum Board):

PBI Tasks WIP Done

|  |  |  |  |
| --- | --- | --- | --- |
| PBI | Tasks(To-Do) | WIP | Done |
| As a user, I want to reset my password via email, so I can access my account if I forget it. | API endpoint | Front-end form | UI Design |
| Email service | Unit tests |  |
| QA testing |  |  |

**Question 6 - Explain Product backlog and sprint back log**

**Product Backlog**

The Product Backlog is a prioritized list of all features, enhancements, bug fixes, technical work, and knowledge acquisition tasks (like research or experimentation) needed to improve the product. It is maintained by the Product Owner and represents everything the team could possibly work on for the project. The Product Backlog is dynamic; it evolves as new requirements are discovered and priorities shift, based on feedback, business needs, or market demands.

**Key Features of the Product Backlog**:

1. Prioritized: Items are ranked in order of importance, with the most valuable or urgent tasks at the top. This prioritization ensures the team works on the highest-value features first.
2. Detailed: The top items in the backlog are typically more detailed than those further down. Items expected to be completed soon are broken down and fully defined, while lower-priority items may be less detailed.
3. Dynamic and Evolving: The Product Backlog is constantly refined as new ideas are added, requirements change, or priorities shift. Regular refinement ensures that the backlog remains current and relevant.
4. Owned by the Product Owner: While the entire team can contribute to the backlog, the Product Owner is responsible for maintaining it and ensuring items are clear, prioritized, and aligned with the product vision.

**Components of a Product Backlog Item (PBI):**

* **User Stories or Feature Requests**: Descriptions of desired functionalities, often in user story format ("As a [user], I want to [function], so that [benefit].").
* **Acceptance Criteria:** Conditions that must be met for the story to be considered complete.
* **Effort Estimates:** A measure of the effort required to complete the item, often given in story points.

**Example of Product Backlog Items:**

1. As a user, I want to create an account to save my order history. (High Priority)
2. As an admin, I want to generate sales reports to track revenue. (Medium Priority)
3. As a user, I want to reset my password in case I forget it. (Low Priority)

**Sprint Backlog**:

The Sprint Backlog is a subset of the Product Backlog that the Scrum team commits to completing during a single sprint. It is created during the Sprint Planning meeting, where the team selects items from the Product Backlog and plans how to achieve the sprint goal. Unlike the Product Backlog, which evolves continuously, the Sprint Backlog is fixed for the duration of the sprint, providing the team with a focused set of work to complete.

**Key Features of the Sprint Backlog:**

1. **Sprint Goal:** The Sprint Backlog is tied to a specific goal that defines the outcome of the sprint. This goal is agreed upon during sprint planning and gives the team a clear purpose.
2. **Detailed Tasks:** Each Product Backlog Item selected for the sprint is broken down into specific tasks that the team will work on. These tasks are highly detailed, defining the exact steps needed to complete the item.
3. **Commitment by the Team**: The team commits to completing all items in the Sprint Backlog by the end of the sprint. Therefore, the Sprint Backlog is a realistic, achievable subset of the Product Backlog.
4. **Ownership and Self-Management**: The Sprint Backlog is managed by the development team, who update it daily to reflect the work completed and remaining. This helps maintain visibility into progress and adjust daily tasks as needed.

**Example of Sprint Backlog Items:**

If the Sprint Goal is to implement a user account feature, the Sprint Backlog might include tasks such as:

* Set up the user registration page.
* Validate input fields for registration.
* Create API endpoints for user authentication.
* Implement email verification process.
* Conduct testing and QA for the registration process.

**Differences Between Product Backlog and Sprint Backlog:**

|  |  |  |
| --- | --- | --- |
| Aspect | Product Backlog | Sprint Backlog |
| Definition | A master list of all desired features, enhancements, bug fixes, and requirements for the product. | A subset of the product backlog selected for a specific sprint. |
| Owned by | Product Owner | Development Team |
| Scope | Covers the entire product roadmap | Focused on the current sprint (iteration) |
| Time Frame | Long-term (updated regularly) | Short-term (specific to the sprint) |
| Content Type | All user stories, epics, features, technical work | Only stories/tasks committed for that sprint |
| Changes During Sprint | Can be updated at any time | Changes are discouraged once sprint starts |
| Purpose | Acts as a product to-do list | Acts as a sprint to-do list |
| Prioritization | Continuously prioritized by Product Owner | Tasks are broken down and managed by the team |

**Question 7 – What is impediments log? write 2 impediments**

An **Impediments Log** is a **document or tool used to track obstacles** that block or slow down a Scrum team’s progress during a Sprint. It is usually **maintained by the Scrum Master** and regularly reviewed to remove or reduce these blockers as quickly as possible.

An Impediment is also known as an issue log or obstacles log is a document or toll in Agile software development to track and manage the obstacles, bottlenecks or any factors that impede the progress of project teams

**Purpose:**

* **Identify and document:** The log serves as a central repository for identifying and documenting all impediments, including those that might affect team productivity, sprint goals, or project goals.
* **Transparency and accountability:** It promotes transparency by making all team members aware of the issues and how they're being addressed. The Scrum Master or a designated person usually takes responsibility for managing and prioritizing these impediments.
* **Prioritization and resolution:**Impediments are prioritized based on their impact on the team's progress, and the log facilitates the process of removing them to keep the team moving forward.

**2 impediments:**

1. Delivery partner shortage in a specific region
2. Technical issue causing intermittent order processing failure

**Impediment 1: Delivery partner shortage in a specific region**

|  |  |
| --- | --- |
| Log Id | 1 |
| Description | Delivery partner shortage in specific region |
| Impact | Delays in order deliveries and increased customer dissatisfaction |
| Priority | High (due to its impact on customer experience) |
| Assigned to | Operation team and HR team |
| Status | Open |
| Action taken | The operations team is actively recruiting new delivery partners in the region the HR team is working on fast tracking the onboarding process |
| Resolution | Delivery partner recruitment efforts are ongoing and the hr team is streamlining the on boarding process to expedite new hires. Regular updates are being provided in team meetings |

**Impediment 2:** Technical issue causing intermittent order processing failure

|  |  |
| --- | --- |
| Log Id | 2 |
| Description | Technical issue causing intermittent order processing failures |
| Impact |  |
| Priority | Delays in order processing and potential revenue loss Hi due to its impact on revenue and customer experience |
| Assigned to | Tech team and QAT |
| Status | In progress |
| Action taken | The tech team has identified the Root Causeway and is working on a fix the QA team is conducting extensive testing to ensure the issue is resolved |
| Resolution | The tech team has implemented a fix and conducted throughout testing the issue has been resolved and orders are now processing smoothly the team has also implemented |

**Question 8 – Explain Velocity of the Team**

In Agile, velocity is a metric that measures the rate at which a team can complete work within a given timeframe, typically a sprint. It's used to predict future delivery timelines and assess team performance. Velocity is calculated by adding up the completed work items, such as story points or tasks, from a sprint. **Velocity** is a metric used in **Agile** (especially in **Scrum**) to measure the **amount of work** a team completes during a single **Sprint**.

**Definition**:

Velocity is the total number of **story points** (or other units like hours or work items) completed by the team in a Sprint.

**Key aspects of velocity in Agile:**

* **Measurement**: Velocity is typically measured in story points, which are relative units that reflect the complexity and effort of a task.
* **Purpose:** Velocity helps teams plan future sprints, estimate project timelines, and make realistic commitments.
* **Calculation:** It's calculated by summing the completed story points (or other units of work) from a sprint.
* **Not a goal:** Velocity should not be a target for teams to strive for but rather a learning metric to understand team performance and identify areas for improvement.
* **Stabilization:** Teams should strive for stable velocity, which indicates a consistent and predictable workflow.
* **Continuous improvement:** Analysing velocity trends helps teams identify bottlenecks and optimize processes.

**How to use velocity effectively:**

* **Track performance:** Regularly monitor velocity to understand how the team is performing and identify areas for improvement.
* **Improve planning:** Use velocity to predict the amount of work that can be completed in future sprints.
* **Set realistic goals:** Understand the team's capacity based on velocity and set achievable goals.
* **Focus on continuous improvement:** Analyse velocity trends and make adjustments to the process based on the insights gained.
* **Don't overemphasize velocity:** Avoid placing undue pressure on the team to increase velocity constantly, as it can lead to burnout and prioritize quantity over quality.

**Key Characteristics of Velocity:**

**1. Team-Specific**

* Velocity is **unique to each team** because every team estimates story points differently based on their experience and skill set.

**2. Based on Completed Work Only**

* Only items that are **100% Done** (meeting the **Definition of Done**) at the end of the Sprint are included in the velocity count.

**3. Measured in Story Points (Typically)**

* Story points are a relative measure of **complexity**, **effort**, and **risk** associated with a user story.

**How to Calculate Velocity:**

Suppose a team works in 2-week Sprints. At the end of the Sprint, they have completed

| **User Story** | **Story Points** | **Status** |
| --- | --- | --- |
| User Login Feature | 5 | Done |
| Password Reset | 3 | Done |
| Profile Page Redesign | 8 | Not Done |

* Only **Done** stories count: **Velocity = 5 + 3 = 8 story points**

**How Velocity Is Used**

* 1. **Sprint Planning:** Helps determine how much work the team can realistically commit to in future Sprints.

**Example:**  
If the team's average velocity is **20 points per Sprint**, they should select ~20 points worth of stories for the next Sprint.

* 1. **Release Forecasting:** Helps predict how many Sprints are needed to complete the Product Backlog.

**Example:**  
Backlog has 100 points.  
If average velocity is 20 → ~5 Sprints needed to complete it.

**Factors Affecting Velocity**

* Team size (e.g., vacations, new members)
* Technical debt
* Interruptions or unplanned work
* Team maturity and experience
* Tooling or process changes

**Best Practices**

* Track **velocity over several Sprints** to find a reliable average.
* Use it for **planning**, not **performance evaluation**.
* Re-estimate large or unclear stories to keep velocity meaningful.
* Don’t compare velocity between teams – it’s relative!

**Bonus: Sample Velocity Chart**

| **Sprint** | **Velocity (Story Points)** |
| --- | --- |
| Sprint 1 | 15 |
| Sprint 2 | 18 |
| Sprint 3 | 20 |
| Sprint 4 | 17 |

**Average Velocity = (15 + 18 + 20 + 17) / 4 = 17.5**

**Question 9 – Draw Sprint Burn Charts n Product Burn Down Charts**

**Sprint Burn Down Chart**

* **Scope:** Tracks work remaining in the current Sprint.
* **Purpose:** Shows daily progress toward completing the Sprint Goal.
* **Used by:** Development team and Scrum Master.
* **Timeframe:** One Sprint (e.g., 2 weeks).
* **What it shows:**
  + **X-axis:** Days of the Sprint.
  + **Y-axis:** Remaining work (e.g., Story Points, tasks, or hours).
  + An ideal burn line vs actual progress.
* **Goal:** Reach zero work remaining by the end of the Sprint.

**Example:** If the team starts with 40 story points, and they complete 10 per day, the chart should slope downward toward 0.

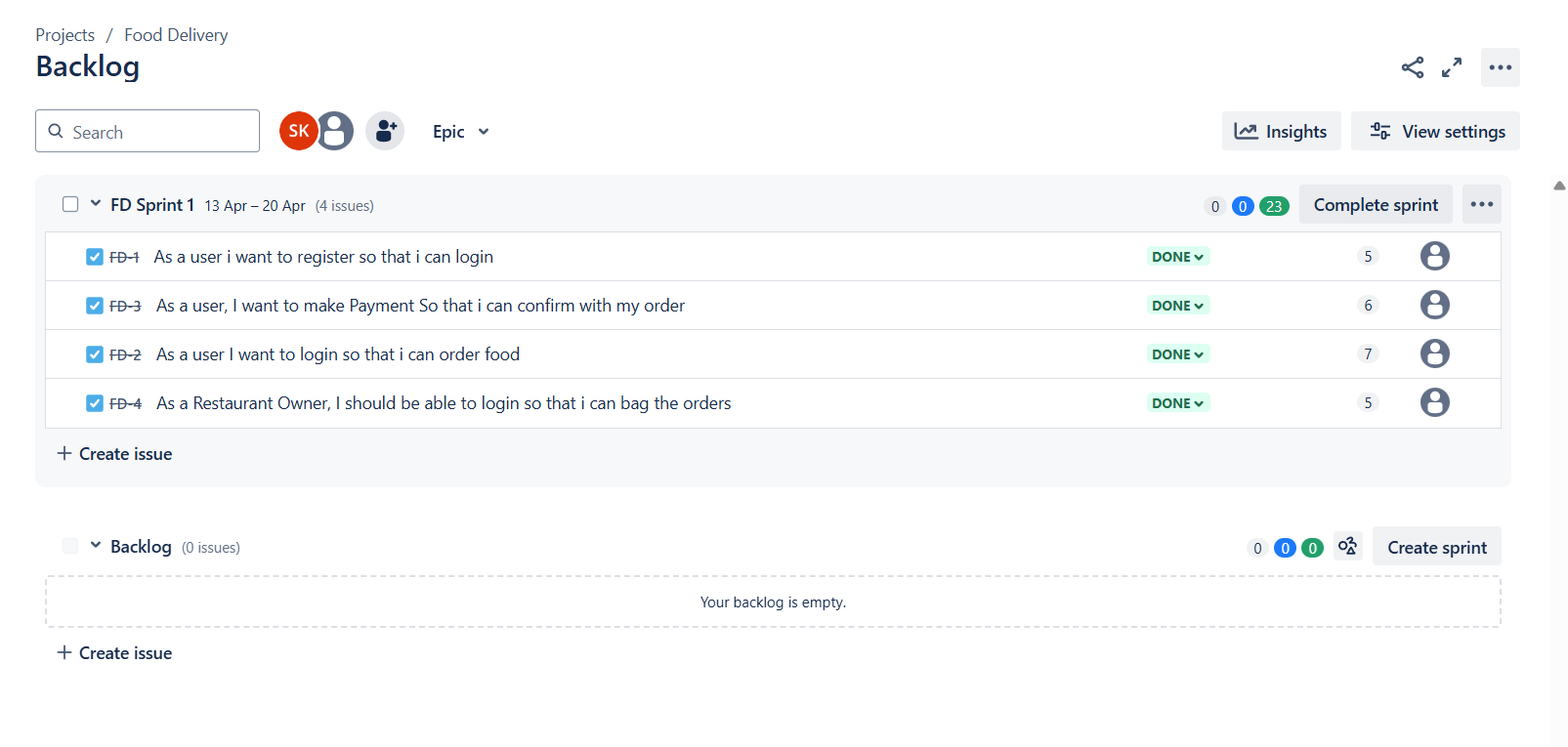
**Product Burn Down Chart**

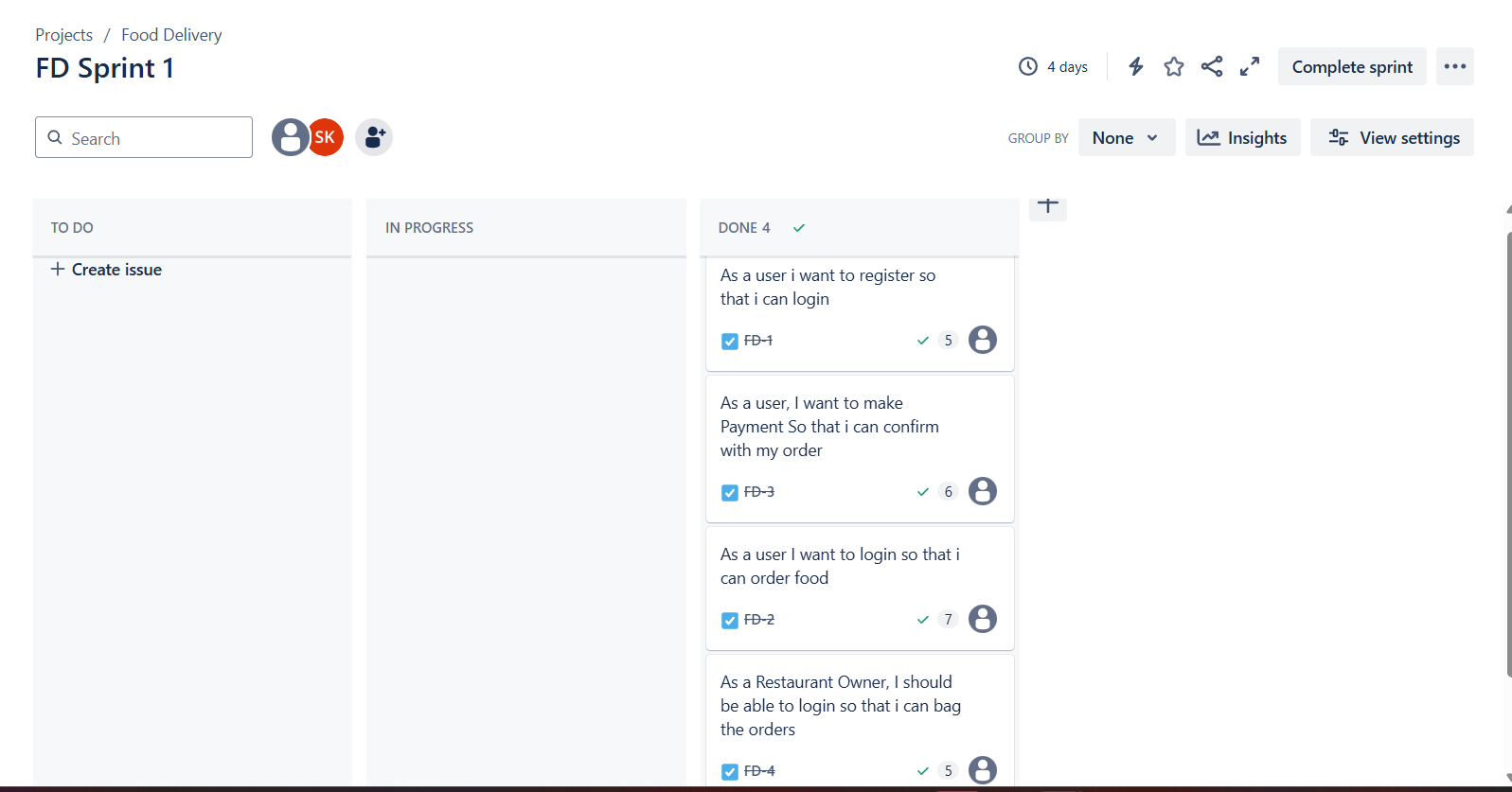
* **Scope:** Tracks work remaining for the entire product or project (Product Backlog).
* **Purpose:** Visualizes overall progress toward product completion or a major release.
* **Used by:** Product Owner, stakeholders, team.
* **Timeframe:** Across multiple sprints (release or project level).
* **What it shows:**
  + **X-axis:** Sprints or time periods.
  + **Y-axis:** Total remaining story points or backlog items.
  + Helps forecast delivery date based on velocity.

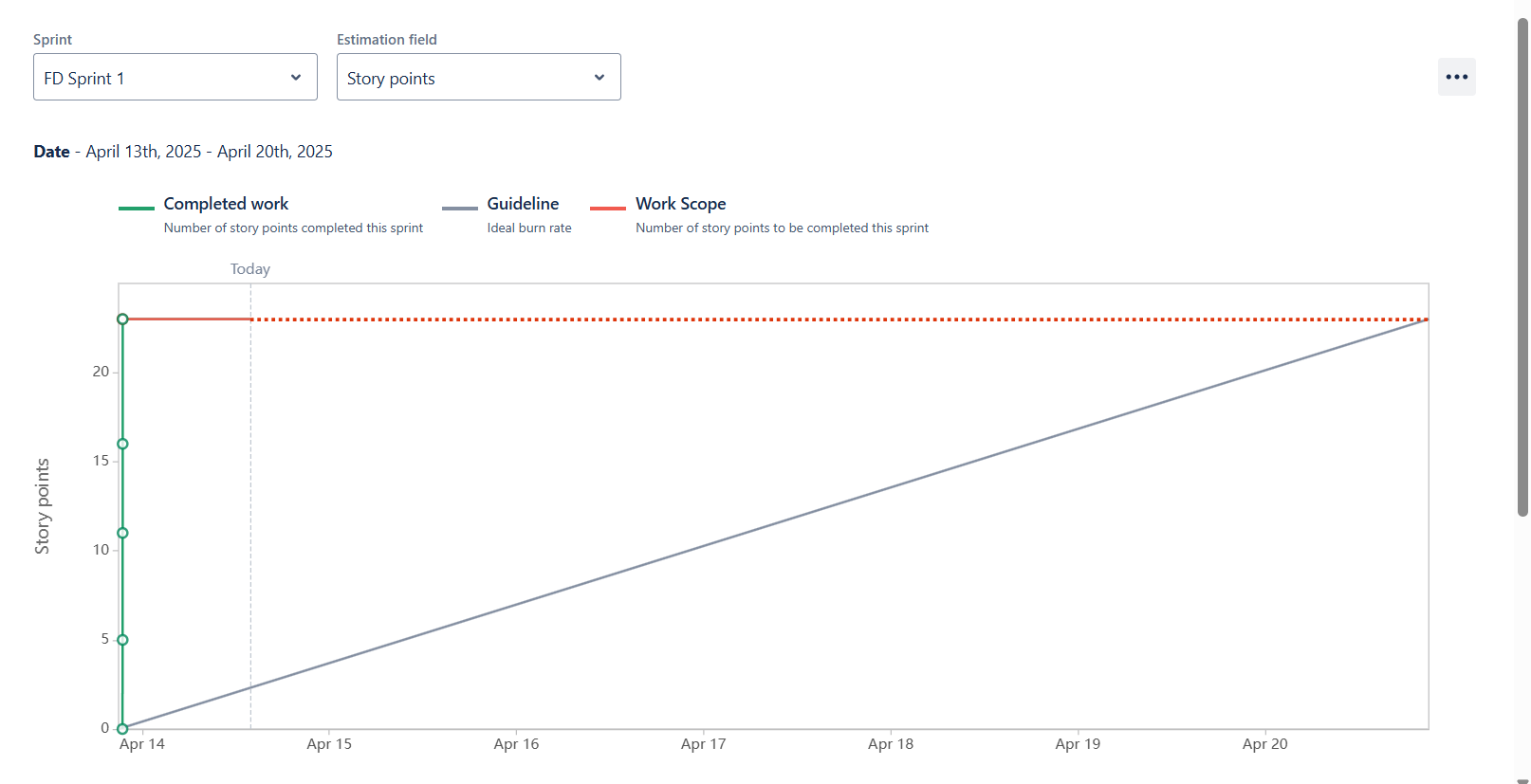
**Example:** You start with 300 story points in the product backlog. Each sprint you burn down ~50 points. The chart helps estimate when you’ll finish

**Key Difference:**

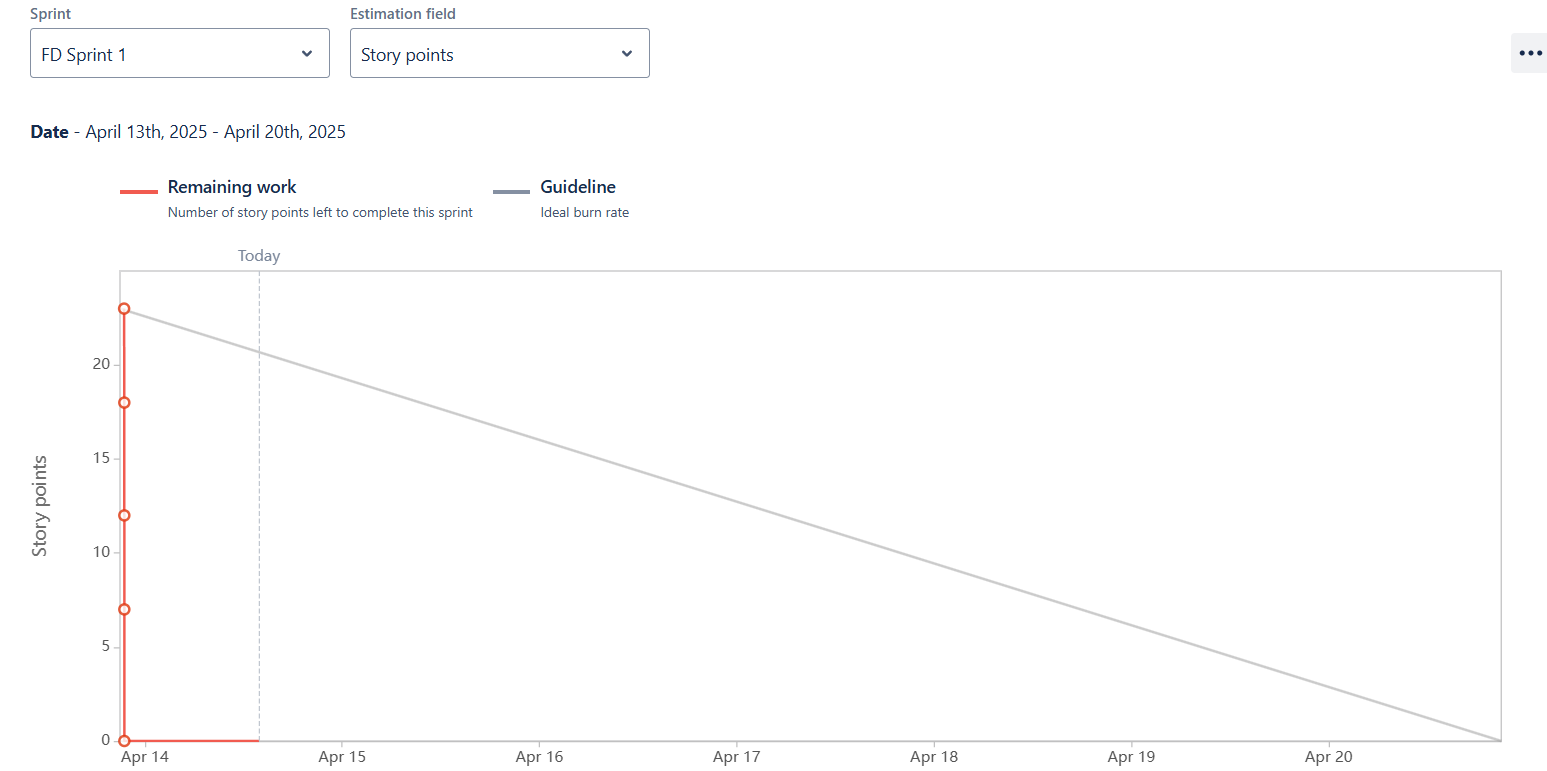
| **Feature** | **Sprint Burn Down** | **Product Burn Down** |
| --- | --- | --- |
| Tracks | Sprint scope | Entire product backlog |
| Timeframe | 1 Sprint | Across multiple Sprints |
| Audience | Team-focused | Stakeholder & team focused |
| Helps with | Daily Sprint progress | Release planning & forecasting |

****

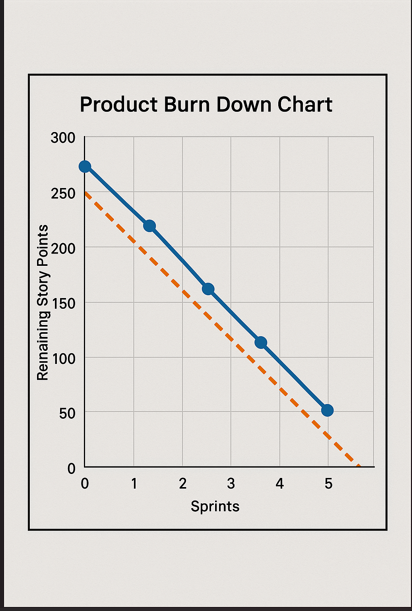
****

**Burn Up chart**

**Burn Down chart:**

****

**Product BurnDown Chart:**

****

**Question 10 – Explain about Product Grooming**

In Agile development, Product Grooming (also known as Backlog Refinement) is an ongoing process where the Product Owner and the Agile team collaborate to review, clarify, prioritize, and update the product backlog. This ensures that the backlog is well-organized, up-to-date, and aligned with the product's goals and priorities. It also helps the team to have a clear understanding of what needs to be worked on next.

**Here are the key aspects of Product Grooming in Agile:**

1. **Clarification of User Stories**

* During grooming sessions, user stories or backlog items are reviewed and clarified. The team ensures that each story is well-defined, understandable, and provides enough detail for the team to work on.
* This might involve breaking down large stories (epics) into smaller, more manageable pieces, or adding acceptance criteria to make the story clearer.

2. **Prioritization of Items**

* The Product Owner ensures that the backlog is prioritized based on the value each item provides to the business and customers. High-value or high-priority items should be at the top of the backlog to ensure that they are worked on first.
* Prioritization may change based on feedback, market shifts, or business needs, so the backlog needs to be flexible.

3. **Estimation of Work**

* The team may provide estimates for how much effort is required to complete each user story. This is typically done using story points or other estimation techniques (e.g., t-shirt sizing).
* Estimating work allows the team to gauge the complexity of tasks and better plan their upcoming sprints.

4. **Adding New Items and Removing Obsolete Ones**

* New user stories or backlog items are added as new requirements emerge. Similarly, obsolete or no longer relevant items are removed from the backlog.
* It's important to continuously refine the backlog so that it represents the current and evolving needs of the project.

5. **Ensuring Ready Items for the Next Sprint**

* The goal of grooming is to ensure that there are enough "ready" items in the backlog for the upcoming sprint. This means the items are well-understood, properly estimated, and prioritized.
* Typically, the backlog is groomed well in advance (e.g., a week or two) before the sprint planning meeting to ensure that there are sufficient high-priority items ready for selection.

6. **Dealing with Dependencies**

* During the grooming process, the team identifies and manages dependencies between tasks or external systems that could affect the product's progress.
* Dependencies need to be addressed early to avoid roadblocks during development.

7. **Continuous Feedback Integration**

* Feedback from stakeholders, customers, or previous sprint reviews is integrated into the backlog. This allows the team to adapt quickly to changes in requirements or direction.

**Benefits of Product Grooming in Agile:**

* **Improved Sprint Planning:** With a well-groomed backlog, sprint planning becomes more efficient because the team has a clear understanding of the work to be done.
* **Better Prioritization:** The product owner can make sure that the team is working on the most important features or fixes, providing greater value to customers.
* **Enhanced Collaboration:** Grooming fosters communication and collaboration between the product owner and the development team.
* **Improved Product Quality:** By addressing issues like ambiguity in user stories or dependencies early, the team reduces the likelihood of rework during development.

**Best Practices for Product Grooming:**

1. **Regular, Short Grooming Sessions:** Grooming doesn't need to be a lengthy meeting but should be a regular practice (e.g., weekly) to keep the backlog up to date.
2. **Involve the Right People:** Include the Product Owner, relevant team members (like developers, testers, UX designers), and sometimes stakeholders in the grooming sessions.
3. **Keep the Focus on Value:** Prioritize the highest-value items in the backlog to ensure the team is always focused on what matters most.
4. **Limit the Time:** Avoid making grooming sessions too long—usually 30 to 60 minutes is a good duration. Long sessions can cause burnout and decrease productivity.
5. **Review Regularly:** Don't wait until just before a sprint to groom the backlog—review it continuously to keep things fresh and relevant.

In summary, Product Grooming in Agile is a vital activity to ensure the backlog is well-managed, prioritized, and ready for sprint planning. It allows teams to be more efficient and responsive to change while keeping the product focused on delivering value to customers.

**Product insights used in Product grooming:**

Product insights play a crucial role in product grooming (backlog refinement) because they help guide decision-making about which features to prioritize, what improvements to make, and how to adapt the product to better meet customer needs. These insights are gathered from various sources and provide valuable data for shaping the product’s future direction. Here's how product insights can be used to improve the product grooming process:

1. **Customer Feedback and Reviews**

* Insight: What are users saying about the product? Are there any common pain points or feature requests?
* Application in Grooming: Use customer feedback (through surveys, interviews, or reviews) to identify areas of improvement. If a particular feature is frequently mentioned as confusing or not functioning as expected, this can be prioritized for fixing or redesigning in the grooming session.

2. **Usage Analytics**

* Insight: How are users interacting with the product? Which features are most or least used?
* Application in Grooming: If data shows that certain features are underused, this could suggest that the product team needs to either improve these features or consider removing them. Conversely, heavily used features may need enhancements or scaling.

3**. Performance Metrics**

* Insight: Are there any performance bottlenecks? For instance, slow load times, crashes, or system downtimes.
* Application in Grooming: Performance issues often emerge through product monitoring tools. These issues should be addressed in the grooming process to enhance the user experience and ensure the product is functioning smoothly.

4. **Market Trends and Competitor Analysis**

* Insight: What are competitors offering that your product doesn’t? Are there emerging trends or technologies that the product should integrate?
* Application in Grooming: Competitive analysis and market trends can highlight potential gaps in your product’s features, prompting the team to prioritize certain updates or additions that align with customer demands and industry shifts.

5. **Business Goals and KPIs**

* Insight: What business outcomes are most important? This could be revenue growth, customer retention, or user acquisition.
* Application in Grooming: Align the backlog items with business goals by ensuring that features or improvements that support these outcomes are prioritized. For example, if the goal is to increase customer retention, improving onboarding or adding a loyalty program might be key priorities.

6**. Technical Debt and Code Quality**

* Insight: Are there any areas of the codebase that need refactoring or optimization? Is there any technical debt that could slow future development?
* Application in Grooming: Regularly review the technical debt and ensure that addressing it is part of the grooming process. Ignoring technical debt can lead to problems later in the product’s lifecycle, making it important to prioritize this in the backlog.

7. **User Research and Testing Results**

* Insight: Have any usability tests or user studies been conducted? What feedback have users provided during testing?
* Application in Grooming: Insights from user research can guide the refinement of existing features or the creation of new ones. If usability tests reveal that users struggle with a particular aspect of the product, the grooming session may prioritize improving that experience.

8. **Retention and Churn Analysis**

* Insight: Are there any patterns in user retention or churn? Are users dropping off after using certain features or at specific touchpoints in the product?
* Application in Grooming: Understanding why users leave or stay can help prioritize features that keep users engaged. For example, if churn analysis reveals that users are quitting after the first month, the team might focus on improving engagement during the early stages of use.

9. **Customer Support and Tickets**

* Insight: What issues are customers submitting tickets or support requests about? Are there recurring issues that could be addressed in the product?
* Application in Grooming: Product grooming sessions can prioritize resolving the most common support issues. These insights ensure that the team is addressing the pain points that directly impact users’ experience.

10. **Internal Stakeholder Feedback**

* Insight: What feedback are internal teams (sales, marketing, customer support) providing about the product? Are they facing challenges that could be addressed by changes to the product?
* Application in Grooming: Input from internal stakeholders can highlight additional opportunities for improvement, such as sales teams needing specific features to close deals or support teams needing easier access to user data.

**Examples of How Product Insights Inform Product Grooming:**

* **Customer Pain Point:** Multiple users report that they find the checkout process confusing.

**Insight:** Improve the checkout flow by simplifying it and providing clearer instructions.

**Application in Grooming:** Add a user story to the backlog for redesigning the checkout process, prioritizing it based on the frequency of user complaints.

* **Feature Usage:** Analytics show that 90% of users are not using a particular feature (e.g., advanced filters).

**Insight:** The feature is not adding enough value or is hard to use.

**Application in Grooming:** Either remove the feature or make significant improvements to increase its utility.

* **Customer Retention:** User testing reveals that users are abandoning the app after a few weeks due to poor onboarding.

**Insight**: Revamp the onboarding process to improve early engagement.

**Application in Grooming**: Add high-priority backlog items focused on improving onboarding, such as tutorials, better guidance, and interactive walkthroughs.

**How to Leverage Product Insights for Grooming:**

1. **Gather Data**: Continuously collect customer feedback, usage data, market research, and performance metrics.
2. **Analyse Insights**: Regularly review the collected data and identify recurring patterns or issues that need to be addressed.
3. **Collaborate with Stakeholders**: Include insights from various stakeholders (e.g., customers, developers, designers, sales, and support teams) during the grooming process.
4. **Refine the Backlog**: Use these insights to refine and prioritize the product backlog, ensuring the team works on the most impactful items that align with user needs and business goals.
5. **Iterate and Adjust:** Product grooming is an iterative process. Regularly revisit the insights to adapt to changing circumstances and continuously improve the product.

By incorporating these product insights into the grooming process, teams can ensure they are always focused on delivering the highest value to their users and the business.

**Question 11 – Explain the roles of Scrum Master and Product Owner**

Top of Form

Scrum Master is focused on the process and ensuring that Scrum is working smoothly, while the Product Owner is focused on the product and ensuring that the team is building the right thing. Both roles are crucial in ensuring that the Scrum team delivers value effectively and continuously improves.

In Scrum, both the Scrum Master and Product Owner are key roles that work collaboratively to ensure the success of the Scrum team and the overall project. However, their responsibilities and focus areas differ.

Here’s a breakdown of each role and their key responsibilities:

1. **Scrum Master**

The Scrum Master is a servant-leader for the Scrum team. They are responsible for ensuring that Scrum is understood and enacted properly. The Scrum Master helps the team adhere to Scrum practices, removes obstacles (impediments), and supports the development process. Their main goal is to help the team become more effective and productive.

**Key Responsibilities:**

* **Facilitating Scrum Events**: The Scrum Master facilitates key Scrum ceremonies (such as Sprint Planning, Daily Scrum/Stand-up, Sprint Review, and Sprint Retrospective) to ensure they run smoothly, stay focused, and achieve the desired outcomes.
* **Removing Impediments:** They help remove any obstacles or roadblocks that might prevent the team from making progress. These could be anything from technical challenges to organizational or resource constraints.
* **Coaching the Team:** The Scrum Master coaches the Scrum team, helping them understand and apply Scrum principles and practices. This includes promoting continuous improvement and self-organization within the team.
* **Supporting the Product Owner**: While the Product Owner is responsible for managing the product backlog and prioritizing work, the Scrum Master helps ensure that the Product Owner's needs are met by coaching them in Scrum practices and assisting with backlog refinement.
* **Shielding the Team:** The Scrum Master shields the team from external distractions, interruptions, or interference from outside the team, allowing them to focus on delivering the Sprint goals.
* **Promoting a Culture of Continuous Improvement:** The Scrum Master ensures that the team continuously inspects and adapts their processes to improve efficiency and effectiveness over time.
* **Servant Leadership:** The Scrum Master embodies a servant-leader approach, always putting the team’s needs first and ensuring that they have what they need to succeed.

**2. Product Owner**

The Product Owner is the visionary and decision-maker for the product. They represent the stakeholders and customers, ensuring that the Scrum team delivers value that aligns with the product’s goals and user needs. The Product Owner has the final say on prioritizing what the team works on and ensures that the product is moving in the right direction.

**Key Responsibilities:**

* **Managing the Product Backlog:** The Product Owner is responsible for creating, managing, and prioritizing the product backlog—the list of features, bug fixes, technical work, and other deliverables required for the product. They ensure the backlog is transparent, visible, and prioritized based on business needs, value, and customer feedback.
* **Defining Product Vision:** The Product Owner defines and communicates the overall vision of the product, ensuring that the team understands the strategic goals and the purpose behind the work they are doing.
* **Prioritizing Backlog Items:** The Product Owner prioritizes items in the product backlog based on customer needs, business value, and strategic goals. This ensures the team is always working on the highest-priority tasks that deliver the most value.
* **Clarifying Requirements**: The Product Owner provides detailed clarification on product backlog items. They may need to answer questions from the team about the requirements, user stories, or acceptance criteria.
* **Stakeholder Engagement**: The Product Owner acts as the bridge between the Scrum team and external stakeholders, including customers, business leaders, and other teams. They gather feedback, negotiate priorities, and ensure that stakeholders' expectations are aligned with the product's development.
* **Maximizing Value Delivery:** The Product Owner ensures that the Scrum team delivers the maximum value to the business and customers. This includes making tough decisions about trade-offs, balancing technical debt, and ensuring the right features are developed at the right time.
* **Acceptance of Work:** The Product Owner is responsible for accepting or rejecting the work completed by the Scrum team during the Sprint. They ensure that the work meets the acceptance criteria and delivers the value expected.
* **Continuous Backlog Refinement:** The Product Owner continuously refines the product backlog based on feedback, new information, and changing market conditions. This includes breaking down larger items (epics) into smaller, more manageable pieces (user stories).

**Key Differences between Scrum Master and Product OwnerBottom of Form**

|  |  |  |
| --- | --- | --- |
| Aspect | Scrum Master | Product Owner |
| Focus | Ensures Scrum processes are followed and the team is functioning effectively. | Focuses on the product's vision, features, and ensuring maximum value is delivered. |
| Key Responsibilities | Facilitates Scrum ceremonies, removes impediments, coaches the team. | Manages the product backlog, defines product vision, prioritizes work based on value. |
| Interaction with Team | Serves the team by providing guidance and helping them follow Scrum practices. | Serves as the team's point of contact for product-related decisions and priorities. |
| Decision-making Authority | Does not make decisions about what gets built or when. | Has the final say on what the team works on and when, based on priorities. |
| External Interaction | Works with the team and ensures they are protected from outside distractions. | Interacts with stakeholders, customers, and the business to understand needs and priorities. |
| Outcome | Focused on team performance and process improvement. | Focused on delivering the right product features and maximizing business value. |

**Team Experiences with Sprint 1**

* The team collaborated well and began understanding Agile practices.
* Some challenges included unclear user stories and time estimation issues.
* Through the Sprint Retrospective, improvements were identified such as better backlog refinement and clearer communication.
* Overall, Sprint 1 helped the team align and prepare for smoother future sprints.

**Question 12 – Explain all Meetings Conducted in Scrum Project**

Here’s a breakdown of all the meetings (ceremonies) conducted in a Scrum project, including their purpose, participants, and timing:

1. **Sprint Planning Meeting**

* Purpose: To plan the work for the upcoming sprint.
* Participants: Scrum Master, Product Owner, and Development Team.
* What Happens:
  + The team selects items from the Product Backlog to work on.
  + Tasks are discussed, clarified, and estimated.
  + The team defines a Sprint Goal.
* When: At the beginning of each sprint (usually 2–4 hours for a 2-week sprint).

2. **Daily Scrum (Daily Stand-up)**

* Purpose: A short meeting to sync up and plan the day.
* Participants: Development Team (Scrum Master may facilitate).
* What Happens: Each team member answers 3 questions:
  + What did I do yesterday?
  + What will I do today?
  + Are there any blockers?
* When: Daily, usually 15 minutes.

3. **Sprint Review**

* Purpose: To inspect the work completed during the sprint and gather feedback.
* Participants: Scrum Team and Stakeholders.
* What Happens:
  + The team demonstrates the work done.
  + Stakeholders provide feedback.
  + Discuss what to do next based on this feedback.
* When: At the end of the sprint (usually 1–2 hours for a 2-week sprint).

4. **Sprint Retrospective**

* Purpose: To reflect on the sprint and identify ways to improve.
* Participants: Scrum Team (Product Owner, Scrum Master, Development Team).
* What Happens:
  + Team discusses what went well, what didn’t, and how to improve.
  + Action items for improvement are agreed upon.
* When: After the Sprint Review and before the next Sprint Planning (usually 1–1.5 hours).

5. **Backlog Refinement (Grooming Session)**

* Purpose: To keep the product backlog clean, clear, and prioritized.
* Participants: Product Owner, Scrum Master, Development Team.
* What Happens:
  + Review and refine backlog items.
  + Add estimates and break down large items (epics) into user stories.
  + Prioritize work based on business value.
* When: Ongoing (usually once or twice per sprint).

**Summary Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting | Purpose | When | Duration (typical) |
| Sprint Planning | Plan sprint work | Start of sprint | 2–4 hours |
| Daily Scrum | Daily team sync | Every day | 15 minutes |
| Sprint Review | Review and demo sprint work | End of sprint | 1–2 hours |
| Sprint Retrospective | Improve team process | After Sprint Review | 1–1.5 hours |
| Backlog Refinement | Update and prioritize backlog | Mid-sprint or ongoing | ~1 hour/week |

**Question 13 – Explain Sprint Size and Scrum Size**

**Sprint Size**

**Definition**: Sprint size refers to the duration or length of a sprint in a Scrum project.

**Details:**

* Typically ranges from 1 to 4 weeks, most commonly 2 weeks.
* The size is consistent throughout the project to maintain rhythm.
* Chosen based on team capacity, product complexity, and delivery expectations.

**Example:**  
If your team follows a 2-week sprint cycle, then each sprint will last exactly 10 working days.

**Scrum Size**

**Definition:** Scrum size refers to the number of people in a Scrum team.

**Details:**

* Recommended Scrum Team size: 10 or fewer members.
* Ideally includes:
  + 1 Product Owner
  + 1 Scrum Master
  + 3–9 Developers (developers, testers, designers, etc.)
* Small enough to stay agile and large enough to complete meaningful work.

**Why It Matters:**

* Too large: harder to communicate and stay aligned.
* Too small: may lack required skills or struggle with workload.

Summary Table:

|  |  |  |
| --- | --- | --- |
| Term | Meaning | Typical Range |
| Sprint Size | Duration of a sprint | 1 to 4 weeks (commonly 2) |
| Scrum Size | Number of people in the Scrum team | 5 to 10 members recommended |
|  |  |  |

**Question 14 – Explain DOR and DOD**

1. **DOR – Definition of Ready**

**Definition**:  
DOR is a checklist that defines when a **user story or backlog item is ready** to be picked up by the development team in a Sprint.

**Purpose**:  
To ensure that the work item is well-defined, clear, and has enough information for the team to start working on it.

**Common Criteria in DOR**:

* User story is clearly written.
* Acceptance criteria are defined.
* Dependencies are identified and resolved.
* Story is estimated and prioritized.
* Team understands the story.

**Example**:  
A user story is considered "Ready" when it is small, testable, and all questions are clarified.

**2. DOD – Definition of Done**

**Definition**: DOD is a checklist that defines when a **product increment or task is considered complete**.

**Purpose**: To ensure the work meets the quality standards and is potentially shippable.

**Common Criteria in DOD**:

* Code is written and peer-reviewed.
* Unit and integration tests are passed.
* Feature is tested (QA approved).
* Documentation is updated.
* Accepted by the Product Owner.

**Example**:  
A feature is "Done" only when it’s developed, tested, documented, and meets all acceptance criteria.

**Summary Table:**

| **Aspect** | **Definition of Ready (DOR)** | **Definition of Done (DOD)** |
| --- | --- | --- |
| **Purpose** | Story is ready to start | Story is complete and meets quality standards |
| **Used In** | Before development begins | After development is complete |
| **Ensures** | Team has clarity and info to start work | Work is fully completed and ready for release |
| **Helps Prevent** | Unclear or unprepared backlog items | Incomplete or low-quality deliverables |

**Question 15 – Explain Prioritization Techniques and MVP**

**Prioritization Techniques**

* + 1. **MoSCoW method:**

The **MoSCoW method** is a powerful tool for setting clear priorities and ensuring that the most important features are delivered first, while less critical features are addressed later. It’s highly flexible and adaptable to any project, from software development to marketing campaigns, helping to manage stakeholder expectations and focus on what truly matters.

* **Must Have**: Critical, non-negotiable features for success.
* **Should Have**: Important, but not essential; can be postponed.
* **Could Have**: Nice-to-have features, but lower priority.
* **Won’t Have**: Out of scope for this phase or version.

**MoSCoW Method in Practice: Example of a New E-Commerce Website**

Let’s apply the MoSCoW technique to an example of an e-commerce website development.

1. **Must Have (M):** These are the core features that are absolutely essential for the website to function as an e-commerce platform:

* **User Login & Registration**: Users must be able to create an account and log in.
* **Product Catalog**: Must display products with prices, descriptions, and images.
* **Shopping Cart**: Users must be able to add products to the cart and proceed to checkout.
* **Payment Gateway Integration**: Must allow users to pay using credit cards or other methods.
* **Order Confirmation**: Users must receive a confirmation after completing the purchase.

1. **Should Have (S)**: These features are important but not necessary for the MVP launch:

* **Product Reviews and Ratings**: Important for user trust but not essential for the launch.
* **Wishlist**: Users can save items for later, which improves user experience.
* **Discount Codes & Coupons**: Important for user retention and marketing but not crucial for the MVP.

1. **Could Have (C):** These are features that can be added later if there’s time or if user feedback suggests they would be beneficial:

* **Live Chat Support**: A customer service feature that is helpful but not critical at launch.
* **Multi-language Support**: Nice to have if the website is targeting international customers, but not necessary initially.
* **Loyalty Program**: A points system or rewards for repeat customers, which can be added in a later phase.

1. **Won’t Have (W):** These are features that will not be part of the current release:

* **Augmented Reality (AR) Shopping Experience**: A futuristic feature that can be added in future iterations but is not necessary for the website’s core functionality.
* **Subscription Model**: A subscription box for regular purchases, which can be explored after the site is running successfully.
  + 1. **Kano Model:** Classifies features by **customer satisfaction.** Basic (expected), Performance (valuable), Exciters (delightful).
    2. **Value vs. Effort Matrix:** Plots features by **business value** vs. **development effort**:

Focus on **High Value + Low Effort** first.

* + 1. **RICE Scoring**  
       🔹 Stands for **Reach, Impact, Confidence, Effort**  
       🔹 RICE Score = (Reach × Impact × Confidence) / Effort  
       🔹 Helps make **data-driven decisions**.
    2. **WSJF (Weighted Shortest Job First)**  
       🔹 Used in **SAFe** Agile  
       🔹 WSJF = Cost of Delay / Job Size  
       🔹 Prioritizes features that offer the **most value in the shortest time**.

**MVP:**

**What is MVP (Minimum Viable Product)?**

An MVP is the most basic version of a product that delivers enough core features to solve a specific problem for early users and collect feedback for future development.

The idea is to build fast, launch early, and improve based on real user interaction — instead of spending months or years building a full product that may not be what the users need.

**Definition**

An MVP is the simplest version of a product that:

* Solves a core user problem
* Includes only essential features
* Is launched quickly to gather user feedback

**Core Concept of MVP**

* **Minimum:** It has the fewest possible features that still make the product usable.
* **Viable**: It is functional enough to be used by real users.
* **Product:** It delivers value and solves a real problem.

**Think of MVP as your first test version — not a prototype, not the final product, but something in between that is usable, valuable, and testable.**

**Why Build an MVP?**

* To test product-market fit early
* To reduce development cost and time
* To minimize risk of building something no one wants
* To gather user feedback for future iterations
* To prioritize features based on real usage

**MVP Development Process**

1. Identify the core problem to solve
2. Define your target audience
3. List must-have features only
4. Build the simplest usable version
5. Launch quickly to early adopters
6. Collect feedback and improve iteratively

**Examples of MVPs**

**🔹 Example 1: E-commerce Website MVP**

**Goal: Allow users to browse and purchase products.**

**MVP Features:**

* User registration/login
* Product catalog
* Add to cart
* Checkout and payment

**Excluded in MVP:**

* Product reviews
* Wishlist
* Loyalty points
* Personalized recommendations

**Benefits of MVP**

| **Benefit** | **Description** |
| --- | --- |
| **Faster Time-to-Market** | Get to users quickly and start learning early |
| **Cost-Effective** | Save time and resources by avoiding unnecessary features |
| **Validates Idea** | Test if users actually want your solution |
| **Customer Feedback Loop** | Build based on real needs, not assumptions |
| **Clearer Product Direction** | Helps you prioritize future features based on actual user data |

**MVP ≠ Prototype**

| **MVP** | **Prototype** |
| --- | --- |
| Usable by real users | Usually internal use |
| Functioning product | Mock-up or demo |
| Designed to learn from the market | Designed to visualize an idea |
| Can grow into final product | May be discarded after testing |

**MVP in Agile/Scrum**

In Agile, MVP is iteratively improved through feedback loops. Product Owners often prioritize features using MoSCoW, RICE, or Value vs Effort techniques to decide what makes it into the MVP.

**Question 16 Difference between Business Analyst n Product Owner**

The roles of Business Analyst (BA) and Product Owner (PO) often overlap, especially in Agile environments, but they have distinct responsibilities and focuses. Here's a breakdown of the key differences:

**Business Analyst (BA)**

**Focus:** Understanding business needs and translating them into detailed requirements.

**Key Responsibilities:**

* Elicit, analyse, and document requirements from stakeholders.
* Create process flows, use cases, and data models.
* Bridge the gap between business and technical teams.
* Ensure the solution meets business needs.
* Often supports the PO in Agile environments.

**Skills:**

* Strong analytical and documentation skills.
* Good communication with both business and tech teams.
* Familiar with tools like UML, BPMN, Excel, etc.

**Product Owner (PO)**

Focus: Defining the product vision and ensuring the team delivers value to the business.

**Key Responsibilities:**

* Own the product backlog and prioritize user stories.
* Act as the voice of the customer/business.
* Define the product roadmap and strategy.
* Work closely with the development team to clarify requirements.
* Make decisions about features, releases, and scope.

**Skills:**

* Strong leadership and decision-making.
* Strategic thinking and stakeholder management.
* Understanding of market/user needs and Agile principles.

**Key Differences**

|  |  |  |
| --- | --- | --- |
| Aspect | Business Analyst (BA) | Product Owner (PO) |
| Main Focus | Requirements gathering & analysis | Product vision, backlog, and prioritization |
| Reports To | Project Manager / Business Side | Product Manager / Business |
| Backlog Role | Helps refine user stories | Owns and prioritizes the backlog |
| Decision Power | Limited | High – makes final decisions on product |
| Customer Focus | Indirect (via requirements) | Direct (focus on delivering value) |
| Typical Tools | Jira, Confluence, Visio, Excel | Jira, Roadmapping tools, Agile boards |

**Can They Coexist?**

Yes! In many Agile teams:

* PO defines *what* needs to be built.
* BA defines *how* it should work based on business needs.

They often collaborate, especially in large or complex projects.

**Question 17 – Prepare a sample Resume of 3yrs exp Product Owner**

**📄 Sample Resume – Product Owner (3 Years Experience)**

**👤 Name:** Priya Sharma  
**📍 Location:** Bangalore, India **📱 Phone:** +91-98765XXXXX **📧 Email:** priyasharma@email.com **🔗 LinkedIn:** linkedin.com/in/priyasharma

**🎯 Career Objective**

**Certified Product Owner with 3 years of experience in Agile environments, skilled in bridging the gap between business needs and technical teams. Proven ability to define product vision, manage backlogs, and deliver high-impact features. Looking to contribute to a dynamic organization that values customer-centric and data-driven solutions.**

**💼 Professional Experience**

**Product Owner  
*XYZ Tech Solutions Pvt. Ltd. – Bangalore*  
Feb 2022 – Present**

* Collaborated with cross-functional Scrum teams to deliver 10+ successful product releases.
* Created and managed Product Backlogs using JIRA and Confluence.
* Conducted Sprint Planning, Review, and Retrospectives with Scrum Master and Developers.
* Defined user stories, acceptance criteria, and prioritized backlog based on business value.
* Worked with stakeholders to align product features with market and customer needs.
* Successfully implemented customer feedback loops leading to a 15% increase in user satisfaction.

**Business Analyst → Associate Product Owner  
*Innovatech Digital Pvt. Ltd. – Pune*  
Jan 2020 – Jan 2022**

* Assisted in gathering and analysing business requirements for digital banking solutions.
* Transitioned into a PO role and led the development of mobile features used by 50K+ users.
* Maintained and refined the product backlog and roadmap in collaboration with leadership.
* Supported QA teams in defining test scenarios and ensuring product quality**.**

**🛠️ Key Skills**

* Agile & Scrum Methodologies
* Product Backlog Management
* Stakeholder Communication
* User Story Mapping
* JIRA, Confluence, Trello
* Wireframing & Prototyping (Figma, Balsamiq)
* Data Analysis & Customer Feedback
* Roadmap Planning

**🎓 Education**

B.E. in Computer Science  
Visvesvaraya Technological University (VTU), Karnataka  
*2015 – 2019*

**📜 Certifications**

* **Certified Scrum Product Owner (CSPO®) – Scrum Alliance**
* **Agile Business Analysis – Simplilearn**

**🏆 Achievements**

* **Recognized as “Emerging Product Leader – 2023” by XYZ Tech Solutions**
* **Reduced feature delivery time by 20% by optimizing backlog grooming sessions**