**COEPD - Scrum Project Implementation – 100 Marks – Pass is 60%**

**Question 1 – write Agile Manifesto – 8 Marks**

**Answer:**

The Agile Manifesto is a set of values and principles designed to improve software development by promoting flexibility, collaboration, and responsiveness to change. Created in 2001, it arose 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
   * People and collaboration drive success, while tools are simply aids.
2. **Working Software** over comprehensive documentation
   * The priority is to deliver a functional product rather than excessive documentation.
3. **Customer Collaboration** over contract negotiation
   * Working closely with customers ensures that the product meets their needs.
4. **Responding to Change** over following a plan
   * Adaptability to change is valued more than sticking to a fixed plan, especially when requirements evolve.

**Twelve Principles of Agile:**

1. **Customer satisfaction** through early and continuous delivery of valuable software.
2. **Welcome changing requirements**, even late in development, to keep the product relevant.
3. **Deliver working software frequently**, with a preference for shorter timescales (weeks rather than months).
4. **Business people and developers must work together** daily throughout the project.
5. **Build projects around motivated individuals**, giving them the environment and support they need.
6. **Face-to-face conversation** is the most effective way to communicate within a team.
7. **Working software** is the primary measure of progress.
8. **Sustainable development**, where teams can maintain a steady pace indefinitely.
9. **Technical excellence and good design** enhance agility.
10. **Simplicity**, the art of maximizing the amount of work not done, is essential.
11. **Self-organizing teams** produce the best architectures and designs.
12. **Regular reflection** by the team on how to become more effective and adjust accordingly.

These values and principles shape Agile practices, promoting flexibility, continuous improvement, and customer satisfaction to produce high-quality software.

**Question 2 – User Stories- Acceptance Criteria-BV-CP – 40 Marks**

**Write minimum 40 User stories and their Acceptance Criteria along with their BV and CP**

**Answer:**

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.
   * Negotiable: The details of the user story can be discussed and adjusted.
   * Valuable: Provides value to the user or the business.
   * Estimable: It should be possible to estimate the effort required to complete the story.
   * Small: User stories should be small enough to be completed within a sprint or iteration.
   * 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 behavior 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.

**40 User Stories:**

**User Story: 1**

|  |  |  |
| --- | --- | --- |
| **User Story No: 1** | **Tasks: Create login functionality, validate user credentials, integrate authentication system.** | **Priority: High** |
| **Value Statement** | As a user, I want to log into my account to access personalized features. | |
| **BV** | 500 | **CP**:5 |
| **ACCEPTANCE CRITERIA:** | 1. Enter valid credentials. | |
| 2. Click "Login". | |
| 3. Redirect to personalized homepage on successful login. | |
| 4. Display error on invalid credentials. | |

**User Story: 2**

|  |  |  |
| --- | --- | --- |
| **User Story No: 2** | **Tasks: Develop search feature, integrate filter options for cuisine, price, and ratings.** | **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 No: 3** | **Tasks: Implement user profile management, add options to update personal 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:** | 1. Access profile settings. | |
| 2. Update personal details. | |
| 3. Click "Save". | |
| 4. View confirmation of successful update. | |

**User Story 4**

|  |  |  |
| --- | --- | --- |
| **User Story No: 4** | **Tasks: 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 5**

|  |  |  |
| --- | --- | --- |
| **User Story No: 5** | **Tasks: Implement a live chat feature for user support.** | **Priority: Medium** |
| **Value Statement** | As a user, I want to chat with support representatives for real-time assistance. | |
| **BV** | 100 | **CP**:2 |
| **ACCEPTANCE CRITERIA:** | 1. Click on the live chat icon. | |
| 2. Enter a message in the chat window. | |
| 3. Click on the send button. | |
| 4. Receive prompt responses from the support representative. | |
| 5. View chat history after the conversation ends. | |

**User Story 6**

|  |  |  |
| --- | --- | --- |
| **User Story No: 6** | **Tasks: 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 No: 7** | **Tasks: 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 No: 8** | **Tasks: 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 No: 9** | **Tasks: 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. Select item and add to cart. | |
| 2. Access cart to view items. | |
| 3. Edit quantity or remove items. | |
| 4. Click "Checkout" to proceed with selected items. | |

**User Story 10**

|  |  |  |
| --- | --- | --- |
| **User Story No: 10** | **Tasks: 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 No: 11** | **Tasks: Enable users to track delivery in real-time on a map.** | **Priority: High** |
| **Value Statement** | As a user, I want to track my order in real-time to know the estimated delivery time. | |
| **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 No: 12** | **Tasks: 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**:2 |
| **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 No: 13** | **Tasks: 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 No: 14** | **Tasks: 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 No: 15** | **Tasks: 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 No: 16** | **Tasks: 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**:2 |
| **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 No: 17** | **Tasks: 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 No: 18** | **Tasks: 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 No: 19** | **Tasks: 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 No: 20** | **Tasks: 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 No: 21** | **Tasks: 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**: 3 |
| **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 No: 22** | **Tasks: 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 “Canceled.” | |

**User Story 23**

|  |  |  |
| --- | --- | --- |
| **User Story No: 23** | **Tasks: 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**: 3 |
| **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 No: 24** | **Tasks: 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 No: 25** | **Tasks: 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**:2 |
| **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 No: 26** | **Tasks: 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 No: 27** | **Tasks: 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 No: 28** | **Tasks: 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 No: 29** | **Tasks: 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 No: 30** | **Tasks: Implement feedback collection post-delivery for improvement.** | **Priority: Medium** |
| **Value Statement** | As a user, I want to provide feedback after delivery to help improve service. | |
| **BV** | 100 | **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 No: 31** | **Tasks: 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**:5 |
| **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 No: 32** | **Tasks: Integrate GPS for accurate restaurant and customer location tracking.** | **Priority: High** |
| **Value Statement** | As a user, I want GPS-enabled location tracking to ensure precise delivery locations. | |
| **BV** | 500 | **CP**: 2 |
| **ACCEPTANCE CRITERIA:** | 1. Enable GPS on device. | |
| 2. Confirm restaurant and customer locations on map. | |
| 3. Track delivery location in real time. | |
| 4. Receive accurate ETA based on live location. | |

**User Story 33**

|  |  |  |
| --- | --- | --- |
| **User Story No: 33** | **Tasks: 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 No: 34** | **Tasks: 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 No: 35** | **Tasks: Implement customer referral program.** | **Priority: Low** |
| **Value Statement** | As a user, I want a referral program to earn rewards by inviting friends. | |
| **BV** | 50 | **CP**: 2 |
| **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 No: 36** | **Tasks: 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 No: 37** | **Tasks: 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**: 2 |
| **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 No: 38** | **Tasks: 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 No: 39** | **Tasks: 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**:2 |
| **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 No: 40** | **Tasks: 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. Select “Reorder” on past order. | |
| 3. Confirm order placed with same items. | |
| 4. Complete checkout with order pre-filled. | |

**Question 3– What is epic? Write 2 epics**

**Business Value and Complexity Points**

**Answer:**

An epic is a high-level requirement or a significant feature in agile development and project management that represents a large body of work. It is a way to capture a large user story that cannot be completed in a single iteration or sprint. Epics help teams organize and prioritize their work by breaking down complex features into smaller, manageable components known as user stories. This structured approach allows teams to deliver incremental value to stakeholders while maintaining a focus on broader project goals.

Epics are typically defined during the initial stages of a project and serve as a guide for the product backlog. They help stakeholders understand the major functionalities and initiatives of a project, allowing for better planning, estimation, and prioritization.

**Key Points About Epics**

1. **Scope and Size**: Epics are broad in scope, capturing a wide range of functionalities. They might describe a single feature or several related features within an application.
2. **Incremental Progress**: Epics are broken down into smaller user stories or tasks that can be tackled individually, making the development process iterative and focused on delivering value regularly.
3. **Alignment with Business Goals**: Since epics address high-level functionalities, they directly align with business goals or user needs, ensuring that they add significant value to the product.
4. **Flexible**: Epics can evolve as the project progresses. Initial ideas can be refined as development unfolds and feedback is gathered, allowing adaptability to changing requirements.
5. **Tracking Progress**: Progress on epics is monitored as individual user stories are completed, helping teams track and manage the delivery of complex functionality.

**Epic 1: User Profile Management**

**Value Statement:**

As a user, I want to manage my profile to customize my account settings and save my preferences.

As a user, I want to log into my account to access personalized features.

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

**Acceptance Criteria:**

* User can edit personal information, such as name, email, and phone number.
* User can change password for security purposes.
* User can add and save multiple delivery addresses to my profile.
* User can view order history to keep track of past purchases.

**Epic 2: 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.

**Epic 3: Menu Browsing and Order Placement**

**Value Statements:**

* 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 my order easily with customizable options (e.g., 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.

**Epic 4: Payment Gateway Integration**

**Value Statements:**

* As a user, I want to pay for my orders securely using different payment methods so that I can complete my purchase conveniently.
* As an admin, I want to integrate a reliable payment gateway to ensure smooth and secure financial transactions.

**Acceptance Criteria:**

* Users can select from multiple payment options (credit card, debit card, UPI, wallet, etc.).
* Payments are processed securely, with end-to-end encryption.
* Users receive a confirmation email or SMS after a successful payment.
* In the event of a failed payment, users are notified with an error message and given an option to retry.
* Users can track their payment history and get receipts for each transaction.

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

**Answer:**

**Business Value:**

1. **Alignment with Business Goals**:
   * **Strategic Focus**: Epics ensure that the development team focuses on initiatives that align with the organization’s strategic objectives. By prioritizing work based on business value, teams can concentrate their efforts on features that deliver the highest impact to customers and stakeholders. This alignment helps organizations meet their business goals more effectively.
   * **Enhanced Customer Satisfaction**: By delivering features that are closely aligned with customer needs and business objectives, epics contribute to higher levels of customer satisfaction. When teams work on what matters most to users, they increase the chances of delivering a product that meets or exceeds expectations.
2. **Improved Stakeholder Engagement**:
   * **Clear Communication**: Epics provide a framework for discussing high-level goals and outcomes with stakeholders. By presenting work in terms of epics, teams can facilitate better conversations about what the project aims to achieve. Stakeholders can provide input on priorities and features, leading to a more collaborative development process.
   * **Progress Tracking**: With epics, stakeholders can easily understand the progress of large features without getting bogged down in the details of individual user stories. This transparency fosters trust and helps keep stakeholders informed about the project’s status, ultimately enhancing stakeholder satisfaction and engagement.

**Complexity Points:**

1. **Cross-Functional Dependencies**:
   * **Coordination Challenges**: Epics often span multiple teams, functions, or departments, leading to increased complexity in coordination. Managing dependencies between different teams can become challenging, particularly when each team has its own priorities and timelines. This can result in delays or misalignments if not effectively managed.
   * **Resource Allocation**: The involvement of various teams may complicate resource allocation, as teams need to collaborate on delivering different components of the epic. Ensuring that the right resources are available when needed can be a logistical challenge, requiring careful planning and communication.
2. **Scalability and Scope Creep**:
   * **Managing Scope Changes**: The high-level nature of epics can make them susceptible to scope creep, where the boundaries of the epic expand over time as new requirements or insights emerge. This can dilute the focus of the development team and lead to difficulties in meeting deadlines or delivering on the original vision.
   * **Estimation Challenges**: Estimating the time and resources needed to complete an epic can be complex, especially when it involves multiple teams or components. As work progresses and additional details are revealed, initial estimates may need to be adjusted, complicating project planning and execution.

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Business Value (BV)** | **Complexity Points (CP)** |
| **Definition** | Represents the value that a feature or epic brings to the business or stakeholders. | Measures the complexity and effort involved in delivering a feature or epic. |
| **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. |
| **Focus** | Focuses on the benefits and outcomes of delivering a feature, such as increased revenue or customer satisfaction. | Focuses on the technical and organizational challenges, such as resource allocation, dependencies, and potential risks. |
| **Measurement** | Can be quantified through metrics like ROI, customer feedback, or business impact analysis. | Often estimated using techniques like story points, T-shirt sizing, or expert judgment based on team experience. |
| **Techniques** | - **ROI (Return on Investment)**: Measures potential financial return. | - **Story Points**: Relative estimation of effort based on complexity. |
| - **Customer Feedback**: Direct input from users on the feature's usefulness. | - **T-shirt Sizing**: Categorizes features into sizes (XS, S, M, L, XL) based on complexity. |
| - **Business Impact Analysis**: Evaluates how a feature aligns with strategic goals. | - **Expert Judgment**: Experienced team members provide input on the expected complexity. |
| **Impact on Prioritization** | High business value features are prioritized to ensure that development efforts yield the maximum return. | Complexity points influence the feasibility of delivering a feature within a certain timeframe, impacting prioritization and resource allocation. |
| **Stakeholder Perspective** | Important for stakeholders to understand what features deliver value to users and the business. | Important for the development team to assess the workload and challenges that need to be managed during implementation. |
| **Timeframe Consideration** | Business value may change over time based on market conditions or strategic shifts, requiring reassessment. | Complexity points may vary depending on team dynamics, technology changes, or new insights gained during development. |
| **Example** | - Adding a new feature to allow customers to track their delivery in real-time, increasing customer satisfaction. | - Implementing a feature that allows multiple payment methods (credit cards, digital wallets, etc.), requiring complex integrations with payment gateways and extensive testing. |
| - Offering discounts and promotions that drive more orders and higher revenue. | - Developing an algorithm to optimize delivery routes, involving complex calculations and data analysis. |

**Question 5 –Explain about Sprint– 5 Marks**

**Answer:**

A Sprint is a time-boxed period, usually lasting 1-4 weeks, during which a Scrum team works to complete a set of prioritized tasks or user stories and deliver a potentially shippable product increment. Sprints are a core aspect of the Scrum framework, enabling teams to work in short, focused cycles that drive consistent progress, encourage frequent feedback, and allow for rapid iteration and improvement.

Each sprint has a clear goal and scope, which is defined and agreed upon by the team and stakeholders during sprint planning. The sprint cycle includes planning, daily stand-ups, development work, testing, and a sprint review and retrospective. The goal of each sprint is to produce a working version of the product with added features, known as a “product increment,” which can be reviewed, evaluated, and potentially delivered to end-users or stakeholders.

**Key Components of a 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.
   * The team then estimates the work and commits to a manageable number of backlog items for the sprint.
2. **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.
3. **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.
4. **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.
5. **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.
6. **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.

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

**Answer:**

**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:**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Product Backlog** | **Sprint Backlog** |
| **Purpose** | Broad list of everything to build for the product | Subset of tasks committed to for a specific sprint |
| **Scope** | Entire product | Current sprint only |
| **Ownership** | Product Owner | Development team |
| **Flexibility** | Continuously evolving | Fixed for the sprint duration |
| **Prioritization** | Items prioritized by value | No further prioritization; all items are committed |
| **Granularity** | High-level items prioritized; lower priority may be vague | Items are broken down into actionable tasks |

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

**Answer:**

An Impediments Log is a tool used in Agile Scrum to track and manage issues or obstacles that hinder a Scrum team's progress. Impediments can be anything that slows down or blocks the team's ability to complete their work during a sprint. These issues are often identified during daily stand-ups or sprint retrospectives, and the Scrum Master is responsible for addressing and resolving them to maintain team productivity.

**Purpose of the Impediments Log:**

* Transparency: It provides a visible record of all issues affecting the team.
* Prioritization: The log helps prioritize issues so that critical impediments can be addressed first.
* Continuous Improvement: By tracking and resolving impediments, the team can improve its workflow and efficiency over time.

**Common Types of Impediments:**

Impediments can range from technical issues to resource shortages or dependencies on external teams. Here are two examples:

**Technical Issue Impediment**

* + **Description**: The development environment is down due to a server issue, preventing the team from testing new features.
  + **Impact**: Blocks the team from completing any testing tasks, delaying feature delivery.
  + **Resolution**: The Scrum Master coordinates with the IT team to resolve the server issue as soon as possible.

**Dependency Impediment**

* **Description**: The team is waiting on API documentation from an external vendor, which is required to integrate a new feature.
* **Impact**: The feature development is blocked until the documentation is provided, delaying progress.
* **Resolution**: The Scrum Master follows up with the vendor to expedite the documentation or finds alternative resources.

**Impediment Log 1:**

|  |  |
| --- | --- |
| **Description** | There was a delay in receiving the required software from the vendor, which was critical for proceeding with the testing phase of the project. This software was a key component for the next milestone, and without it, the testing team couldn't begin their work. |
| **Impact** | Due to the delay in receiving the software, the testing phase, which was scheduled to start, couldn't commence. This resulted in a significant delay to the overall project timeline, affecting all downstream activities that were dependent on the testing phase. As a result, the project delivery date was at risk of being pushed further back, which could also impact the budget and resource allocation. |
| **Priority** | High – The delay in receiving the software was a high-priority issue as it directly impacted the timeline and project milestones. It required immediate attention to ensure the testing phase could proceed without further delays. |
| **Assigned To** | Vendor Manager – The Vendor Manager was responsible for coordinating with the software vendor and ensuring that the necessary software was delivered on time. They needed to track the vendor's progress and escalate the issue if necessary to prevent further delays. |
| **Status** | Open – This issue was still unresolved, with the vendor being the primary point of contact for the resolution. The team was waiting on the software to proceed with the testing. |
| **Action Taken** | The Vendor Manager proactively followed up with the vendor to expedite the delivery of the software. Regular check-ins were established with the vendor to track progress and ensure no further delays occurred. Additionally, an alternative plan was prepared in case the vendor could not meet the revised delivery date. |
| **Resolution** | After 3 days of delay, the vendor was able to deliver the software. The testing phase resumed, but the project timeline had to be adjusted to accommodate the delay. The testing team was informed and began their work immediately, although the project completion date was extended by a few days. |

**Impediment Log 2**

|  |  |
| --- | --- |
| **Description** | There was a shortage of delivery partners in a specific region, which affected the timely delivery of orders to customers. This shortage created a backlog of deliveries and impacted the customer satisfaction in that region. Due to limited availability of delivery partners, some orders were delayed, and the logistics team was unable to meet the promised delivery timelines. |
| **Impact** | The delivery delays led to a backlog, which affected the overall customer experience. Customers began reporting delays in receiving their orders, and this impacted customer retention and brand reputation. The delay also increased customer support inquiries, leading to resource strain on the customer support team. Additionally, the region's sales targets were at risk due to delayed deliveries, as customers were less likely to place repeat orders if their expectations weren’t met. |
| **Priority** | High – This issue was of high priority because it affected customer satisfaction and the company’s ability to fulfill orders on time. It was crucial to resolve this issue quickly to prevent further customer dissatisfaction and operational disruptions. |
| **Assigned To** | Logistics Manager – The Logistics Manager was responsible for coordinating with the delivery partner network to address the shortage. They had to assess whether the shortage was due to regional resource constraints or operational issues and find alternative ways to ensure deliveries were completed on time. |
| **Status** | Open – The issue was still ongoing, as the shortage of delivery partners was not yet fully resolved. Immediate action was being taken to hire additional resources or find other delivery partners to cover the region. |
| **Action Taken** | The Logistics Manager began negotiations with existing delivery partners to increase the workforce in the specific region. They also explored alternative delivery partners and logistics companies to temporarily cover the region. Additionally, a communication plan was established to inform customers of potential delays, and priority was given to urgent orders. To manage customer expectations, a revised delivery window was provided. |
| **Resolution** | The shortage was partially resolved by bringing in new delivery partners, which helped reduce the backlog. While some delays still occurred, the region was able to stabilize the delivery process within a few days. To prevent future shortages, the company expanded its partnership network and improved forecasting to better allocate resources to high-demand regions. |

**Question 8 – Explain Velocity of the Team – 1 Marks**

**Answer:**

Velocity is a key metric in Agile Scrum that represents the amount of work a team completes during a sprint, often measured in terms of story points, hours, or other units of completed work. Velocity is specific to each team and is used primarily for planning and forecasting future sprints. By tracking velocity over time, teams can better estimate how much work they can take on in upcoming sprints and make data-driven decisions to improve performance and productivity.

**How Velocity Works:**

1. **Calculation**: Velocity is calculated at the end of each sprint by summing up the story points (or other units) of all fully completed user stories or tasks. Only items meeting the **Definition of Done** (DoD) are counted, ensuring they are truly complete and deliverable.
2. **Consistency**: Since teams often take a few sprints to find a sustainable rhythm, their initial velocity may vary. Over time, the team’s velocity stabilizes as they find their pace, leading to more consistent sprint planning and estimations.
3. **Forecasting**: Velocity enables teams to predict how many story points they can handle in future sprints. For example, if a team has a stable velocity of 30 story points, they can use this to plan how many user stories or tasks to include in the next sprint.

**Example:**

Let’s say a team completes 25 story points in the first sprint, 28 in the second, and 30 in the third. The team’s average velocity over these sprints is roughly 27.6 story points. This average gives the team a baseline for planning future sprints, helping ensure they commit to a manageable workload that aligns with their proven capacity.

**Limitations:**

While velocity is useful, it’s essential not to overemphasize it as a strict measure of productivity. Team velocity can fluctuate due to factors like sprint goals, technical complexities, team size, and other dynamics. Therefore, it should be used as a guide rather than a rigid target.

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

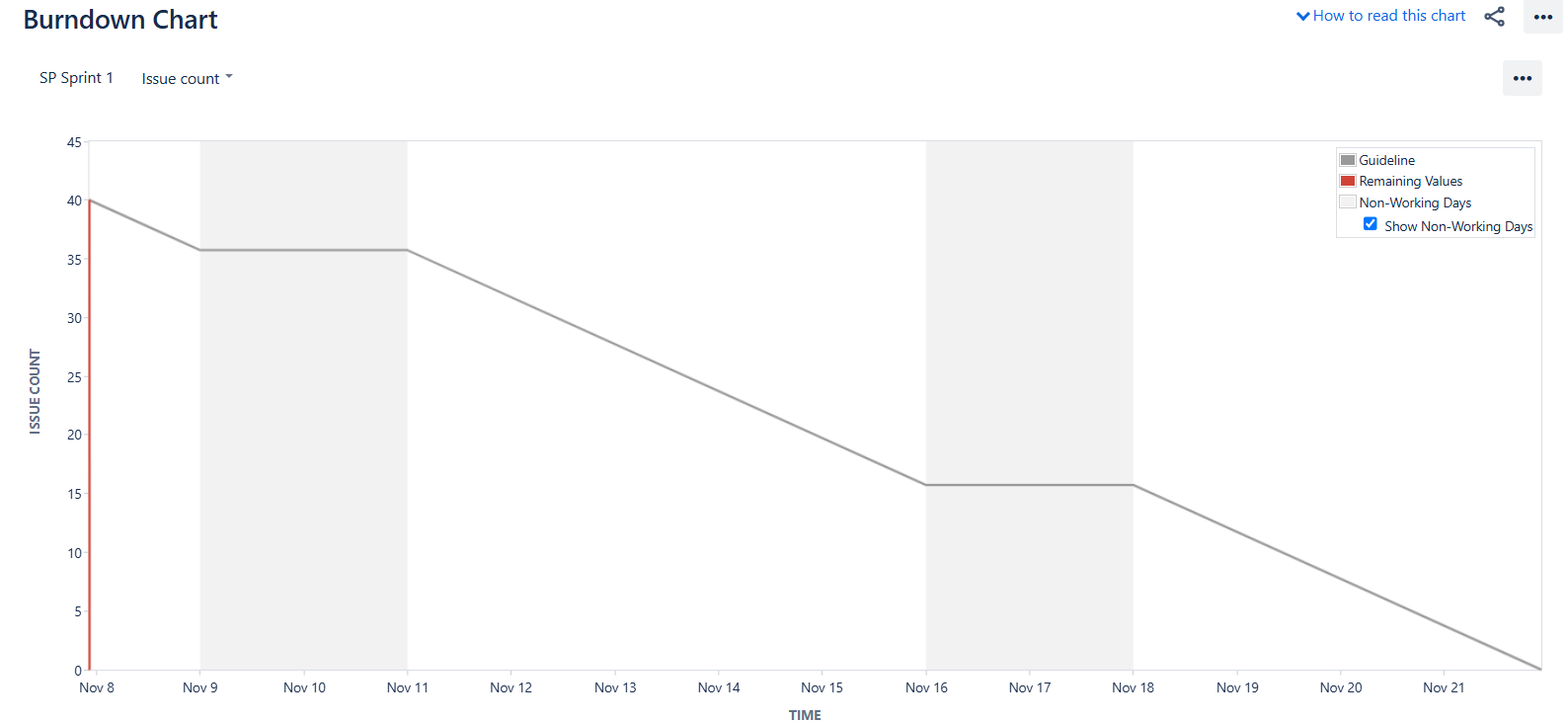
**Answer:**

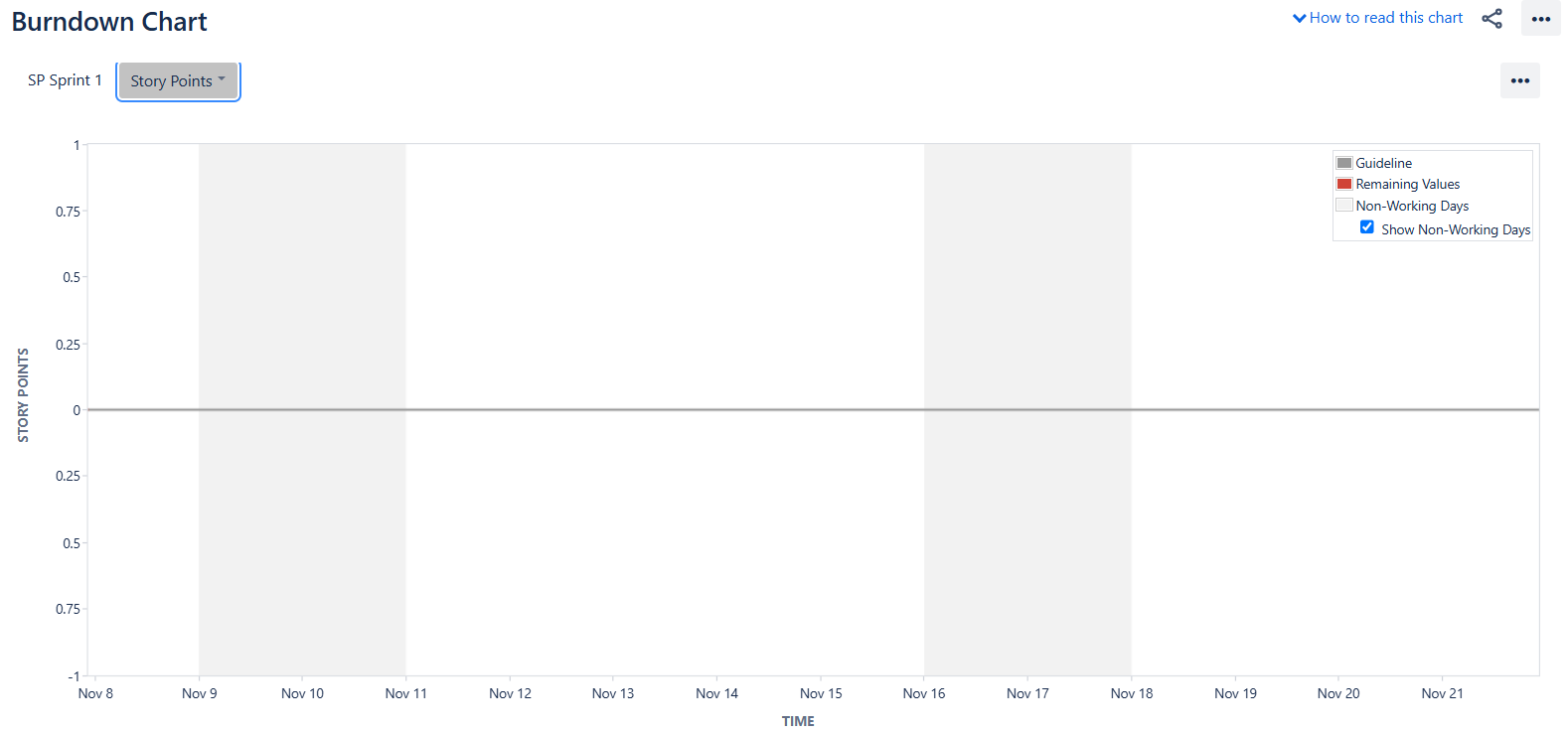
**Sprint Burn-Down Chart:**

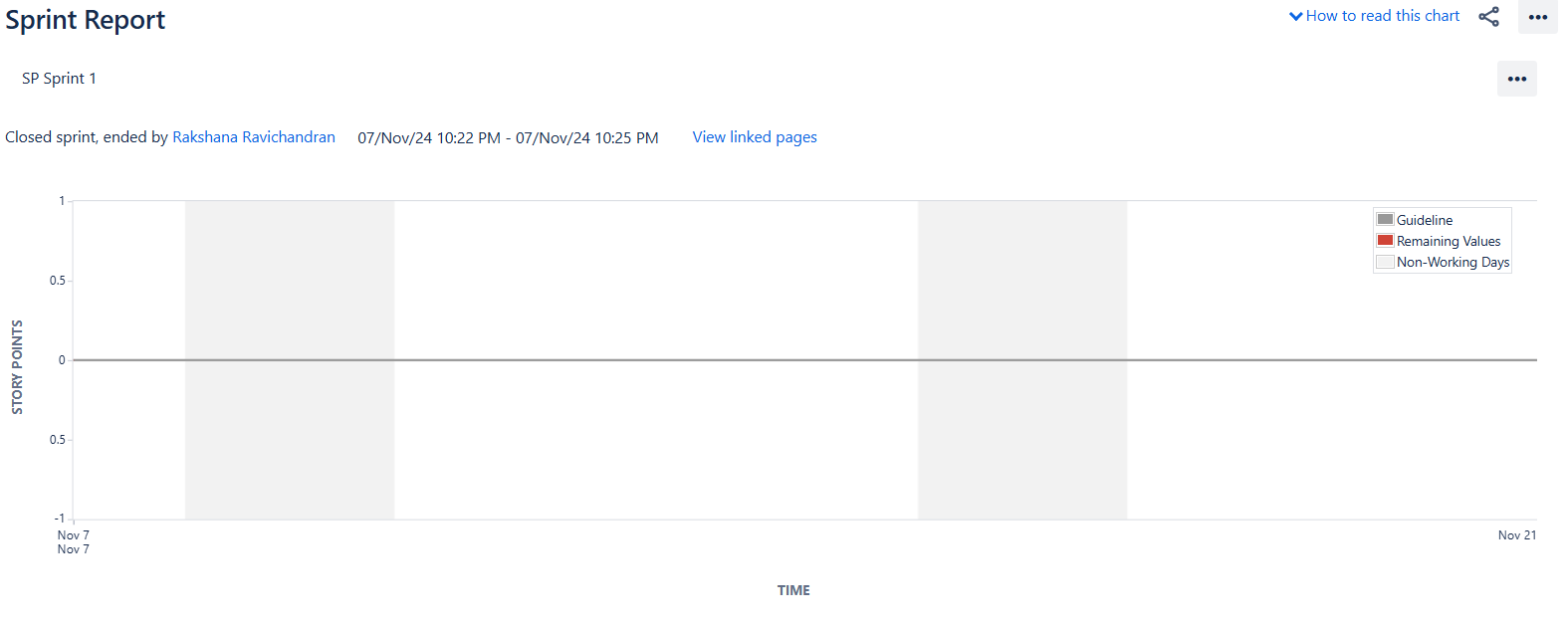
* **Purpose:** Tracks the amount of work remaining in a single sprint (usually two weeks).
* **Focus:** Helps the Scrum team see if they are on track to complete the sprint goals.
* **Data Tracked:** Daily updates on the amount of work left (measured in story points, hours, or tasks).
* **X-Axis:** Days in the sprint.
* **Y-Axis:** Remaining work.
* **Usage:** Updated daily by the Scrum Master, it visually shows how much work is left, allowing the team to identify if they are behind or ahead of schedule and to adjust their efforts.

**Product Burn-Down Chart:**

* **Purpose:** Tracks the work remaining in the entire project or product backlog over multiple sprints.
* **Focus:** Provides a high-level view for the Product Owner and stakeholders on when the entire product might be completed.
* **Data Tracked:** Remaining product backlog items across all sprints, with projections on completion.
* **X-Axis:** Sprints or time intervals (weeks or months).
* **Y-Axis:** Remaining work (total story points or backlog items).
* **Usage:** Updated at the end of each sprint, it helps the Product Owner adjust priorities or scope based on progress, helping manage stakeholder expectations on delivery timelines.







**Question 10 – Explain about Product Grooming**

**Answer:**

Product Grooming (also known as Backlog Grooming or Backlog Refinement) is an Agile practice focused on continuously updating and refining the Product Backlog to ensure that it contains well-defined, prioritized, and achievable items for upcoming sprints. It’s a collaborative process involving the Product Owner, Scrum Master, and Development Team, often supported by stakeholders as needed.

**Goals of Product Grooming**

1. Clarify Requirements: Break down high-level ideas into smaller, more manageable user stories or tasks. This clarification includes discussions around requirements, functionality, and expectations.
2. Prioritize Items: Ensure that the most valuable items are at the top of the backlog. This prioritization helps the team understand which items should be focused on in upcoming sprints.
3. Estimate Effort: The team evaluates each backlog item to provide estimates (often in story points or hours). These estimates guide planning and help forecast the work capacity.
4. Define Acceptance Criteria: Establish clear acceptance criteria for each item, making it easier for the team to understand when the work is complete.
5. Reduce Ambiguity: Refine each item by resolving any uncertainties or gaps in requirements. This minimizes misunderstandings and ensures that the items are achievable.

**Key Activities During Product Grooming**

1. Decomposition: Large items (epics) are broken down into smaller user stories or tasks, making them easier to estimate and implement.
2. Prioritization: The Product Owner reorders the backlog based on changing business needs, customer feedback, or new priorities.
3. Estimation: The team discusses each item to provide a rough estimate of the time and effort needed.
4. Clarifying Details: Team members ask questions to clear up any uncertainties and to ensure each item is “ready” for development.
5. Setting Acceptance Criteria: Acceptance criteria are defined, outlining what conditions need to be met for the item to be considered complete.

**Benefits of Product Grooming**

* Improved Planning: Groomed backlogs lead to more effective sprint planning since the team has clear, actionable, and prioritized items.
* Enhanced Productivity: The team can focus on development without needing to frequently clarify requirements mid-sprint.
* Better Stakeholder Alignment: Regular grooming aligns the team’s work with business goals, adapting to any changes in priorities or market conditions.
* Higher Quality: Clear requirements and acceptance criteria lead to fewer misunderstandings and better alignment with stakeholder expectations.

**Frequency and Timing**

Product Grooming sessions are typically held once or twice per sprint, depending on the team’s needs and the product’s complexity. These sessions are often shorter than other Scrum ceremonies, lasting between 30 minutes to an hour.

Example

**Imagine a product backlog with high-level ideas like:**

1. Build a customer feedback feature
2. Implement order tracking
3. Optimize search functionality

**During grooming, these ideas would be refined into smaller, detailed user stories, such as:**

* As a user, I want to rate my order so I can provide feedback.
* As a user, I want to track my order in real time so I can see when it will arrive.
* As a user, I want to filter search results by price and rating so I can find the best options.

**Each story is then prioritized, estimated, and clarified to ensure it’s actionable.**

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

**Answer:**

In Scrum, the Scrum Master and Product Owner are key roles, each with specific responsibilities that contribute to the effectiveness and success of the team. Here’s an overview of these roles and a look at how they operate, with a reflection on their experiences from Sprint 1.

|  |  |  |
| --- | --- | --- |
| **Role** | **Scrum Master** | **Product Owner** |
| **Primary Responsibility** | Facilitates the Scrum process, ensuring that the team adheres to Scrum principles and practices. | Represents the stakeholders and is responsible for defining and prioritizing the product backlog. |
| **Focus Area** | Focuses on team performance, removing obstacles, and ensuring smooth Scrum events (e.g., sprint planning, daily standups, sprint review). | Focuses on delivering value to the customer by maintaining a clear vision and ensuring that the right product features are being built. |
| **Collaboration** | Works closely with the Scrum team, removes blockers, and ensures that team members are well-supported. | Collaborates with stakeholders, customers, and the Scrum team to define features and priorities. |
| **Decision-Making** | Does not make decisions about the product or its features but supports the team in making decisions. | Has decision-making authority regarding the product backlog and ensures that the most important features are delivered. |
| **Responsibilities** | - Coaches the team on Scrum practices. | - Manages and prioritizes the product backlog. |
| - Removes impediments. | - Defines product vision. |
| - Protects the team from external distractions. | - Makes decisions on feature priority and releases. |
| - Facilitates Scrum ceremonies. | - Ensures the product delivers customer value. |
| **Interaction with Stakeholders** | Limited interaction with stakeholders; mainly focuses on ensuring team health and process. | Directly interacts with stakeholders, customers, and users to gather feedback and align the product with business needs. |
| **Key Skills** | - Facilitation | - Communication |
| - Conflict resolution | - Prioritization |
| - Coaching | - Decision-making |
| - Process management | - Stakeholder management |

**Role of the Scrum Master**

The **Scrum Master** is responsible for facilitating the Scrum process, supporting the team in following Scrum principles, and removing obstacles that may impede progress. They act as a **servant leader** for the Scrum team, ensuring that the team can work efficiently within the Scrum framework.

**Key Responsibilities of the Scrum Master:**

1. **Facilitate Scrum Events**: Ensures that Sprint Planning, Daily Stand-ups, Sprint Review, and Sprint Retrospective are held efficiently and productively.
2. **Remove Impediments**: Identifies and resolves any obstacles that block the team’s progress to maintain momentum and productivity.
3. **Coach the Team**: Educates the team on Scrum principles, encourages collaboration, and promotes continuous improvement.
4. **Protect the Team**: Shields the team from external disruptions, keeping them focused on Sprint goals.
5. **Monitor Progress**: Tracks sprint progress through tools like burndown charts or velocity charts, providing transparency to stakeholders and helping the team stay on track.

**Scrum Master Experience in Sprint 1:** In the first sprint, the Scrum Master often encounters challenges related to team alignment and understanding of Scrum processes. They may spend more time coaching team members on roles, ceremonies, and the importance of sticking to Scrum values. For example, the Scrum Master may have facilitated initial discussions to align the team on Sprint goals and resolved initial issues, such as access to tools or understanding task prioritization.

**Role of the Product Owner**

The **Product Owner** is responsible for maximizing the product's value by managing the **Product Backlog**. They act as the **voice of the customer** and ensure that the development team works on the highest-priority items that deliver the most value to the users.

**Key Responsibilities of the Product Owner:**

1. **Define and Prioritize Product Backlog**: Maintains and refines the Product Backlog, prioritizing items based on business value and customer needs.
2. **Communicate Vision**: Clearly communicates the product vision, goals, and backlog items to the team.
3. **Set Clear Acceptance Criteria**: Defines detailed acceptance criteria for each backlog item to ensure the team understands the requirements.
4. **Engage with Stakeholders**: Gathers feedback from stakeholders, customers, and end-users to inform backlog prioritization and product development direction.
5. **Make Quick Decisions**: Makes timely decisions to clarify requirements, prioritize items, and answer team questions during the sprint.

**Product Owner Experience in Sprint 1**: As the Product Owner for Sprint 1, you focused on defining and prioritizing the initial set of backlog items, breaking down epics into actionable user stories, and setting clear acceptance criteria for each item. Sprint 1 is often a learning experience, involving close interaction with the team to clarify backlog items and prioritize based on initial feedback. You likely engaged with the team frequently to answer questions and adjust priorities based on progress and any blockers encountered.

**Key Takeaways from Sprint 1 as Product Owner**:

* **Importance of Clear Backlog Items**: Clear, actionable backlog items help the team work more efficiently.
* **Frequent Communication**: Regular communication with the Scrum Master and the team helps address issues quickly.
* **Flexibility and Adaptability**: Being open to feedback and ready to adjust priorities as the team progresses ensures alignment with goals.

In essence, Sprint 1 is a foundational experience for both the Scrum Master and Product Owner, setting the tone for future collaboration and continuous improvement.

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

**Answer:**

In a Scrum project, several key meetings, or Scrum ceremonies, are conducted to facilitate effective communication, align the team, and ensure progress toward project goals. Each meeting serves a unique purpose and helps maintain the Scrum framework’s iterative, incremental approach.

**1. Sprint Planning Meeting**

* **Purpose**: Plan the upcoming sprint by defining the sprint goal and selecting Product Backlog items to be completed during the sprint.
* **Attendees**: Scrum Master, Product Owner, Development Team
* **Duration**: Typically 2 hours per week of sprint length (e.g., a 2-week sprint has a 4-hour Sprint Planning)
* **Process**:
  + The **Product Owner** presents high-priority items from the Product Backlog.
  + The team discusses these items, clarifies requirements, and estimates effort.
  + The team selects items they believe they can complete, creating a **Sprint Backlog**.
  + A **Sprint Goal** is defined, providing focus for the sprint.
* **Outcome**: A clear Sprint Backlog and a Sprint Goal that the team commits to achieving.

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

* **Purpose**: Facilitate daily coordination, promote transparency, and identify any impediments.
* **Attendees**: Scrum Master, Development Team (Product Owner may attend but is not required)
* **Duration**: 15 minutes, held at the same time and place daily.
* **Process**:
  + Each team member answers three questions:
    1. What did I work on yesterday?
    2. What will I work on today?
    3. Are there any blockers or impediments?
* **Outcome**: Team members are aligned on progress, have a clear plan for the day, and any obstacles are identified.

**3.Sprint Review (Demo)**

* **Purpose**: Inspect the increment of the product developed during the sprint, gather feedback, and adapt the Product Backlog as needed.
* **Attendees**: Scrum Master, Product Owner, Development Team, Stakeholders
* **Duration**: Typically 1 hour per week of sprint length (e.g., a 2-hour Sprint Review for a 2-week sprint)
* **Process**:
  + The team demonstrates the completed work to stakeholders, showing what has been done according to the **Definition of Done**.
  + Stakeholders and the Product Owner provide feedback on the increment, helping to refine the Product Backlog.
  + The Product Owner may update backlog priorities based on feedback.
* **Outcome**: Stakeholders see progress, provide feedback, and the Product Backlog is updated for future sprints.

**4. Sprint Retrospective**

* **Purpose**: Reflect on the past sprint, discuss what went well, what could be improved, and decide on actionable steps for improvement.
* **Attendees**: Scrum Master, Development Team (Product Owner may attend if the team agrees)
* **Duration**: Typically 45 minutes per week of sprint length (e.g., a 1.5-hour Retrospective for a 2-week sprint)
* **Process**:
  + The team reviews the sprint and discusses successes and challenges.
  + Members share ideas for improvements in processes, tools, or team dynamics.
  + The team decides on specific action items to implement in the next sprint.
* **Outcome**: A set of improvement actions that the team will work on in future sprints, fostering continuous improvement.

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

* **Purpose**: Ensure that Product Backlog items are well-defined, prioritized, and ready for future sprints.
* **Attendees**: Product Owner, Development Team, Scrum Master (optional)
* **Duration**: Generally, 5-10% of the team’s time during a sprint.
* **Process**:
  + The Product Owner presents backlog items to the team for discussion and refinement.
  + The team clarifies requirements, estimates effort, and discusses priorities.
  + The Product Owner may reprioritize based on team feedback or new insights.
* **Outcome**: A refined, prioritized Product Backlog with items that are ready for the next sprint, aligning team understanding and preparation.

**Summary of Scrum Meetings**

|  |  |  |  |
| --- | --- | --- | --- |
| **Meeting** | **Purpose** | **Attendees** | **Duration** |
| **Sprint Planning** | Plan the sprint, define goals, and create the Sprint Backlog | Scrum Master, Product Owner, Development Team | 2 hrs per week of sprint |
| **Daily Stand-up** | Daily coordination, transparency, and identify blockers | Development Team, Scrum Master | 15 minutes |
| **Sprint Review** | Inspect product increment, get feedback, update backlog | Scrum Team, Stakeholders | 1 hr per week of sprint |
| **Sprint Retrospective** | Reflect on sprint, identify improvements | Scrum Master, Development Team | 45 min per week of sprint |
| **Backlog Refinement** | Refine backlog, prepare items for upcoming sprints | Product Owner, Development Team, Scrum Master | 5-10% of sprint duration |

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

**Answer:**

Sprint Size and Scrum Size are important concepts in Scrum, affecting how work is planned, executed, and completed by the team.

**Sprint Size**

**Sprint Size** refers to the duration of a sprint and the amount of work planned within that time frame. The size of a sprint is determined based on factors like the team’s capacity, project complexity, and the need for frequent feedback or iterations.

**Key Aspects of Sprint Size:**

1. **Duration**: A sprint is typically between 1 and 4 weeks long, with 2 weeks being the most common. The sprint size is chosen to balance fast feedback with enough time to produce meaningful work.
2. **Workload**: The team commits to completing a specific amount of work (usually measured in story points or complexity points) that they believe can realistically be accomplished within the sprint’s timeframe. The amount of work planned for the sprint is often referred to as the "Sprint Backlog."
3. **Consistency**: Sprint size usually remains consistent throughout a project to create a predictable cadence, helping stakeholders and the team to align expectations.
4. **Goal Alignment**: The sprint is focused on achieving a specific sprint goal, which reflects the value the team aims to deliver by the end of that sprint.

**Choosing the Right Sprint Size:**

The ideal sprint size balances predictability with flexibility. A sprint that’s too short might not allow the team enough time to complete valuable increments, while a sprint that’s too long could lead to less frequent feedback and potentially wasted effort if requirements change. Teams typically experiment initially to find the duration that fits their workflow best.

**Scrum Size**

**Scrum Size** usually refers to the size and composition of the Scrum team. Scrum teams are designed to be small, cross-functional, and self-managing to ensure high productivity and effective collaboration.

**Key Aspects of Scrum Size:**

1. **Team Size**: A Scrum team generally consists of 5 to 9 people, including:
   * **Product Owner**: Responsible for maximizing the product’s value by managing the Product Backlog and setting priorities.
   * **Scrum Master**: Facilitates Scrum processes, removes impediments, and ensures the team follows Scrum practices.
   * **Development Team**: Responsible for delivering the increment. This group is cross-functional, meaning it has all the necessary skills (design, development, testing) to complete the work.
2. **Cross-Functionality**: Scrum teams are typically self-sufficient, meaning they have all the skills needed within the team to deliver increments without relying on external teams.
3. **Self-Management**: Scrum teams are empowered to organize their work, decide how to best accomplish their goals, and identify improvements.
4. **Optimal Size for Efficiency**: Keeping the team size between 5 and 9 members allows for effective communication, collaboration, and agile decision-making. Smaller teams may lack the necessary skills or capacity to deliver significant increments, while larger teams may face communication and alignment challenges.

**Adjusting Scrum Size:**

While a typical Scrum team has 5-9 members, the exact size can be adapted based on the project’s needs, complexity, and available resources. However, if the team gets too large, it may be split into multiple Scrum teams, each working on related goals, with a need for coordination across teams.

**Summary: Sprint Size vs. Scrum Size**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Sprint Size** | **Scrum Size** |
| **Definition** | Duration of a sprint and the work planned for that time. | Number of team members in a Scrum team. |
| **Typical Size** | 1-4 weeks per sprint, with work measured in story points. | 5-9 people per team. |
| **Purpose** | Defines the cadence and amount of work per iteration. | Ensures optimal productivity, cross-functionality, and self-management. |
| **Flexibility** | The length can be adjusted but typically remains consistent. | Can vary slightly but usually within 5-9 members. |

**Question 14 – Explain DOR and DOD**

**Answer:**

In Scrum, Definition of Ready (DoR) and Definition of Done (DoD) are essential concepts that ensure clarity and quality in the development process. They serve as criteria for determining when a task is ready to be worked on (DoR) and when a task is considered complete (DoD).

**Definition of Ready (DoR)**

The Definition of Ready (DoR) is a set of criteria that a backlog item must meet before it can be considered "ready" to start in a sprint. It ensures that all necessary details, requirements, and resources are in place so the team can confidently begin work.

**Purpose of DoR:**

* Prevents delays or confusion due to missing information or requirements.
* Improves the team’s ability to focus on development without constant clarification.
* Helps maintain a smooth workflow by ensuring that items are well-prepared.

**Typical DoR Criteria:**

1. Clear Acceptance Criteria: Acceptance criteria are well-defined and understood by the team.
2. Detailed Requirements: Requirements and specifications are documented and available.
3. Dependencies Resolved: Any dependencies on other teams, resources, or third parties are addressed.
4. Prioritization: The item is prioritized in the Product Backlog.
5. Estimated Effort: The item has been discussed and estimated by the team.
6. Approved by Product Owner: The Product Owner has validated the item and confirmed its readiness.

**Example of DoR: For a "User Registration" feature to be considered ready, it might need:**

* User stories and acceptance criteria defined.
* UI mockups or design specifications.
* Clarification on data storage and validation rules.
* Approval from the Product Owner.

**Definition of Done (DoD)**

The Definition of Done (DoD) is a checklist that defines when a product increment (such as a user story, feature, or task) is considered "done" and ready for delivery. It ensures that the completed work meets the required quality and is fully functional, tested, and deployable.

**Purpose of DoD:**

* Guarantees consistent quality across all deliverables.
* Helps reduce technical debt by ensuring that all items meet standards before they are considered done.
* Provides a clear understanding of what "done" means, aligning expectations across the team and stakeholders.

**Typical DoD Criteria:**

1. Code Completed: The feature is fully developed, with no incomplete functionality.
2. Code Reviewed: Code has been reviewed by other team members.
3. Unit Tests Passed: Automated unit tests have been written and passed.
4. Integration Tested: The feature has been tested in the integrated system environment.
5. Documentation Updated: Any necessary documentation, such as code comments or user manuals, has been updated.
6. Acceptance Criteria Met: All acceptance criteria for the story have been verified.
7. Approved by Product Owner: The Product Owner has accepted the item as complete.

**Example of DoD: For a "Login Functionality" feature to be considered done, it might need:**

* Developed code with all functionalities, such as password validation and error handling.
* Code review completed and feedback addressed.
* Successful execution of unit and integration tests.
* Updated documentation or release notes.
* Approval from the Product Owner.

**Differences Between DoR and DoD:**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Definition of Ready (DoR)** | **Definition of Done (DoD)** |
| **Purpose** | Ensures backlog items are ready for development | Confirms that backlog items meet completion and quality standards |
| **Application Stage** | Before starting the work on a backlog item | After completing work on a backlog item |
| **Focus** | Preparation, clarity, and alignment | Quality, functionality, and completion |
| **Typical Criteria** | Requirements, acceptance criteria, dependencies resolved | Code completion, testing, documentation, approval |

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

**Answer:**

Prioritization Techniques and Minimum Viable Product (MVP) are crucial concepts in product development within Agile and Scrum methodologies. They help teams deliver high-value features efficiently while managing time and resources effectively. Let's explore each in detail.

**1. Prioritization Techniques**

Prioritization techniques guide decision-making, helping teams focus on the most valuable features or tasks in a project. Some common prioritization methods include:

* **MoSCoW Method**: Divides features into Must-have, Should-have, Could-have, and Won't-have, allowing teams to prioritize based on necessity and project scope.
* **Kano Model**: Categorizes features as Basic Needs, Performance Needs, and Excitement Needs, focusing on customer satisfaction. Basic Needs are expected, Performance Needs increase satisfaction with improvement, and Excitement Needs delight users but aren’t expected.
* **RICE Scoring**: Stands for Reach, Impact, Confidence, and Effort. It’s a quantitative method where teams score features based on the number of people affected (Reach), the positive impact (Impact), certainty of success (Confidence), and resources required (Effort). Features with high scores are prioritized.
* **Value vs. Effort Matrix**: Plots features on a 2x2 matrix based on their value and effort levels. Features that deliver high value with low effort are prioritized.
* **ICE Scoring**: Similar to RICE, but with fewer criteria: Impact, Confidence, and Ease. This technique quickly identifies high-value features with the potential for significant positive impact.

**2. Minimum Viable Product (MVP)**

An MVP is the simplest version of a product that includes just enough features to satisfy early customers and provide feedback for future development. The idea is to launch quickly, learn from real users, and iterate based on actual feedback. Key aspects of an MVP include:

* **Core Functionality**: Only essential features are included to solve the primary problem for users. Extra features are deferred to later stages.
* **Quick Validation**: MVPs allow teams to test assumptions about the market, customer needs, and product functionality without heavy investment.
* **Iterative Improvement**: Feedback from the MVP helps teams enhance the product, ensuring future versions meet user expectations and deliver greater value.

Combining prioritization techniques with MVP development helps create a product that aligns with user needs and delivers value efficiently, ultimately improving the likelihood of product success.

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

**Answer:**

**Business Analyst (BA):**

Focuses on understanding business needs, documenting requirements, and ensuring solutions meet those needs. They analyze processes, identify improvements, and work closely with stakeholders to bridge the gap between business and technical teams.

**Product Owner (PO):**

Responsible for maximizing product value by defining, prioritizing, and managing the product backlog. The PO acts as a bridge between the stakeholders and development team, making prioritization decisions to keep the product aligned with the vision and goals.

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Business Analyst (BA)** | **Product Owner (PO)** |
| **1. Focus** | Business needs, detailed requirements | Product value, roadmap |
| **2. Primary Role** | Gathering and analyzing requirements | Defining and prioritizing the product backlog |
| **3. Requirements Approach** | Detailed ("how" and "what") | High-level ("why" and value-based) |
| **4. Key Responsibilities** | Documenting requirements (BRD, FRD), process analysis | Managing product backlog, defining user stories |
| **5. Team Interaction** | Works with business stakeholders, IT, and QA | Works closely with development and product stakeholders |
| **6. Decision Power** | Limited decision-making; provides insights | High, makes final decisions on feature priorities |
| **7. Outputs** | BRD, FRD, process flows, and models | Product backlog, user stories, acceptance criteria |
| **8. Approach to Change** | Analyzes and documents changes in processes or requirements | Evaluates and prioritizes changes to the product based on value |
| **9. Involvement in Agile** | Works with teams in all project phases, often providing requirements | Integral in Agile, attends sprint planning, reviews, and retrospectives |
| **10. Stakeholder Focus** | Focuses on gathering requirements from stakeholders and translating them to the team | Focuses on ensuring stakeholder needs are represented in the product and backlog |

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

**Answer:**

**Sample Resume:**

**Rakshana R**

**Email: rakshana@example.com**

**Phone: +91-XXXXXXXXXX**

**LinkedIn: linkedin.com/in/rakshana**

**Location: Vellore, Tamil Nadu, India**

**SUMMARY**

Dynamic and results-driven Product Owner with 3 years of experience in leading cross-functional teams and delivering high-quality products on time. Expertise in product development, agile methodologies, and continuous process improvement. Adept at gathering and analyzing requirements, defining product roadmaps, and aligning business goals with user needs. Passionate about delivering exceptional user experiences and driving business growth.

**PROFESSIONAL EXPERIENCE**

**Product Owner**

*ABC Technologies, Chennai*

*June 2021 – Present*

* Lead product development teams in building customer-centric software products for e-commerce and fintech industries.
* Collaborated with stakeholders to define product vision, strategy, and roadmap, ensuring alignment with business goals.
* Gathered, analyzed, and prioritized product requirements using Agile and Scrum methodologies.
* Worked closely with designers, engineers, and quality assurance teams to ensure timely delivery of high-quality features.
* Defined user stories, acceptance criteria, and worked on backlog grooming to ensure smooth sprint execution.
* Conducted regular product demos to internal stakeholders, presenting updates, insights, and customer feedback.
* Analyzed product performance metrics, identifying areas of improvement and implementing solutions for optimization.
* Managed product lifecycle from concept to launch, ensuring stakeholder buy-in and meeting user expectations.

**Key Achievements:**

* Successfully launched 5 major features that increased user engagement by 25%.
* Reduced product defects by 15% by improving requirements clarity and collaboration across teams.
* Spearheaded the launch of a new feature that contributed to a 20% increase in revenue for the company.

**Associate Product Owner**

*XYZ Solutions, Chennai*

*May 2019 – May 2021*

* Assisted the Product Owner in managing the product backlog and ensuring timely delivery of product features.
* Collaborated with business analysts to understand customer requirements and translate them into actionable product features.
* Conducted competitive analysis and market research to inform product decisions and feature prioritization.
* Worked with the development team to clarify user stories and ensure clear communication of requirements.
* Supported sprint planning, retrospectives, and backlog refinement sessions.
* Coordinated with the QA team to ensure proper testing of features and timely bug fixes.

**Key Achievements:**

* Contributed to the successful launch of a mobile app feature that improved customer retention by 18%.
* Developed product documentation and user guides, enhancing user onboarding and reducing support queries.

**EDUCATION**

**Bachelor of Technology (B.Tech)**  
*XYZ University, Vellore*  
*Graduated: 2018*  
Specialization: Computer Science Engineering

**SKILLS**

* **Agile Methodologies:** Scrum, Kanban, Sprint Planning
* **Product Management Tools:** Jira
* **Data Analysis Tools:** Google Analytics, Excel
* **Collaboration Tools:** Slack, MS Teams, Zoom
* **Requirements Gathering & Documentation:** User Stories, BRD, FRD, Use Cases
* **Communication:** Stakeholder Management, Cross-functional Team Leadership

**CERTIFICATIONS**

* **Certified Scrum Product Owner (CSPO)** – Scrum Alliance, 2021
* **Google Analytics for Beginners** – Google, 2020

**PROJECTS**

**E-commerce Mobile App Redesign**

* Led the redesign of a major e-commerce mobile app, improving the user interface and experience based on customer feedback.
* Resulted in a 15% increase in conversion rates and a 20% decrease in cart abandonment.

**Fintech Platform Integration**

* Managed the integration of a payment gateway for a fintech platform, streamlining payment processes and reducing transaction failures.
* Increased user satisfaction with smoother payment processing and reduced support tickets.