**Nurturing Process - Capstone Project2 –Agile-Scrum V2D2 Aug2024**

**Scrum Project Implementation**

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

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 main Values**

* Individuals and interactions over processes and tools
* Working software over comprehensive documentation
* Customer collaboration over contract negotiation
* Responding to change over following a plan

**Twelve Principles of Agile Software**

1.Satisfy the customer through early and continuous delivery of valuable software.

2.Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3.Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4.Business people and developers must work together daily throughout the project.

5.Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity--the art of maximizing the amount of work not done--is essential.

11.The best architectures, requirements, and designs emerge from self-organizing teams.

12.At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

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

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 customer registration portal** | **Priority:** High |
| **Value Statement** | As a customer I want to do the registration on the scrum food app | |
| **BV** | 500 | **CP:**8 |
| **ACCEPTANCE CRITERIA:** | 1. Should have valid email id 2. Send Successful Notification to the user | |

**User Story: 2**

|  |  |  |
| --- | --- | --- |
| **User Story No: 2** | **Tasks: Browse Restaurants by Cuisine, Rating, and Location** | Priority: High |
| **Value Statement** | As a customer I want to **browse restaurants by cuisine, rating, and location** so that I can find my preferred food easily. | |
| **BV** | 500 | CP:5 |
| **ACCEPTANCE CRITERIA:** | 1. Restaurants should be filtered based on cuisine, rating, and location. 2. Users should be able to sort by relevance or delivery time. 3. Clicking a restaurant should navigate to its menu page. | |

**User Story: 3**

|  |  |  |
| --- | --- | --- |
| **User Story No: 1** | **Tasks: Search for a Specific Restaurant or Dish** | **Priority:** High |
| **Value Statement** | As a customer,I want to **search for a restaurant or dish** so that I can quickly find what I want. | |
| **BV** | **500** | **CP:3** |
| **ACCEPTANCE CRITERIA:** | 1. The search bar should allow input of restaurant names and dish names. 2. Results should update dynamically as the user types. 3. Clicking a search result should navigate to the restaurant or dish page. | |

**User Story: 4**

|  |  |  |
| --- | --- | --- |
| **User Story No: 4** | **Tasks: Add Food Items to Cart** | Priority: High |
| **Value Statement** | As a customer,I want to **add food items to my cart** so that I can place my order conveniently. | |
| **BV** | 200 | CP:3 |
| **ACCEPTANCE CRITERIA:** | 1. Cart should update when an item is added. 2. The total price should update dynamically. 3. Users should be able to remove or update quantity before checkout. | |

**User Story: 5**

|  |  |  |
| --- | --- | --- |
| **User Story No: 5** | **Tasks: Track My Order in Real-Time** | Priority: High |
| **Value Statement** | As a customer,I want to **track my order in real-time** so that I know when to expect my food. | |
| **BV** | 500 | CP:8 |
| **ACCEPTANCE CRITERIA:** | 1. The system should update the order status dynamically. 2. A map should display the delivery partner’s location. 3. The estimated delivery time should be shown. | |

**User Story: 6**

|  |  |  |
| --- | --- | --- |
| **User Story No: 6** | **Tasks:** View Restaurant Details | **Priority:** High |
| **Value Statement** | As a customer,As a **customer**, I want to **see restaurant details (menu, ratings, reviews, estimated delivery time)** so that I can make an informed decision. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The restaurant page should display menu items, restaurant images, and delivery time. 2. Customer reviews and ratings should be visible. 3. The system should show if the restaurant is open or closed. | |

**User Story: 7**

|  |  |  |
| --- | --- | --- |
| **User Story No: 7** | Tasks: Customize My Order | **Priority:** High |
| **Value Statement** | As a customer,As a **customer**, I want to **customize my order (extra toppings, remove ingredients, etc.)** so that I get exactly what I want. | |
| **BV** | **200** | **CP:3** |
| **ACCEPTANCE CRITERIA:** | 1. Customization options should be available for each applicable dish. 2. The selected options should be reflected in the final order summary. | |

**User Story: 8**

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 8** | **Tasks: Filter restaurants** | **Priority: High** | **Priority:** High |
| **Value Statement** | As a **customer**, I want to **filter restaurants based on price, delivery time, and offers** so that I can choose the best option for me. |  | |
| **BV** | **500** | **5** | **CP:** |
| **ACCEPTANCE CRITERIA:** | Users should be able to filter restaurants based on cuisine type  sers should be able to filter restaurants based on customer ratings  The system should only display restaurants that can fulfill orders within the selected time range.  The system should display only restaurants with offers when the filter is applied. |  | |

**User Story: 9**

|  |  |  |
| --- | --- | --- |
| **User Story No: 9** | **Tasks:** Apply Discounts & Promo Codes | **Priority:** High |
| **Value Statement** | As a customer,I want to **apply discount codes or promo offers** so that I can save money on my order. | |
| **BV** | **200** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The system should validate and apply eligible promo codes. 2. Invalid or expired promo codes should display an error message. 3. The final price should update accordingly. | |

**User Story: 10**

|  |  |  |
| --- | --- | --- |
| **User Story No: 10** | **Tasks:** Select Payment Method | **Priority:** High |
| **Value Statement** | As a customer,I want to **select my preferred payment method (credit/debit card, wallet, COD, UPI, etc.)** so that I can pay conveniently. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The system should display multiple payment options. 2. Payment should be processed securely. 3. A confirmation message should be displayed after successful payment. | |

**User Story: 11**

|  |  |  |
| --- | --- | --- |
| **User Story No: 11** | **Tasks: Saving delivery address** | **Priority:** High |
| **Value Statement** | As a customer,I want to **save my delivery addresses** so that I can quickly select them while ordering. | |
| **BV** | **100** | **CP:3** |
| **ACCEPTANCE CRITERIA:** | 1.The system should allow users to add a new delivery address from the checkout page or profile settings.  2.Users should be able to edit saved addresses from their profile settings.  3.Users should have an option to delete any saved address.  4.Users should be able to save multiple addresses (e.g., Home, Work, Friend’s House). | |

**User Story: 12**

|  |  |  |
| --- | --- | --- |
| **User Story No: 12** | **Tasks:** Provide Customer Reviews & Ratings | **Priority:Medium** |
| **Value Statement** | As a customer,I want to **rate and review restaurants and delivery partners** so that I can share my experience with others. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | app should be able to show the feedback and rating option where user  can add review | |

**User Story: 13**

|  |  |  |
| --- | --- | --- |
| **User Story No: 13** | **Tasks: Reorder from order history** | **Priority:Medium** |
| **Value Statement** | As a customer,I want to **reorder my previous orders** so that I can quickly place my favorite orders again. | |
| **BV** | **200** | **CP:3** |
| **ACCEPTANCE CRITERIA:** | 1. The system should allow customers to view their past orders in the "Order History" section. 2. Each past order should have a "Reorder" button to add all items to the cart. 3. Customers should be able to modify items or quantities before checkout. 4. If an item is unavailable, the system should notify the user and suggest alternatives. | |

**User Story: 14**

|  |  |  |
| --- | --- | --- |
| **User Story No: 14** | **Tasks:Get notifications,order updates,promotions and offers.** | **Priority:** High |
| **Value Statement** | As a customer,I want to **get notifications about order updates, promotions, and offers** so that I stay informed. | |
| **BV** | **20** | **CP:3** |
| **ACCEPTANCE CRITERIA:** | 1. The system should send real-time push notifications for order status updates (Order Placed, Preparing, Out for Delivery, Delivered). 2. Customers should receive promotional notifications for discounts, offers, and new restaurants. 3. The system should allow users to opt-in or opt-out of promotional notifications from settings. 4. Customers should receive notifications for order-related issues (e.g., order delayed, item unavailable). 5. Notifications should be accessible from the app’s notification center. | |

**User Story: 15**

|  |  |  |
| --- | --- | --- |
| **User Story No: 15** | **Tasks: Cancel order** | **Priority:** High |
| **Value Statement** | As a customer,I want to **cancel an order within a given time frame** so that I can change my mind if needed. | |
| **BV** | **100** | **CP:8** |
| **ACCEPTANCE CRITERIA:** | App should be able to cancel the order within the 10 min of the given  order | |

**User Story: 16**

|  |  |  |
| --- | --- | --- |
| **User Story No: 16** | **Tasks: Live Chat with Support** | **Priority:Medium** |
| **Value Statement** | As a customer,I want to **contact customer support via chat or call** so that I can resolve any issues with my order. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The system should display real-time order status updates (Order Placed → Preparing → Out for Delivery → Delivered). 2. Customers should see an estimated delivery time that updates dynamically based on driver location. 3. The system should show the assigned delivery partner's name, contact info, and live tracking on a map. 4. If the order is delayed, the system should notify the customer with the reason and a revised ETA. 5. Customers should receive a confirmation once the order is delivered. | |

**User Story: 17**

|  |  |  |
| --- | --- | --- |
| **User Story No: 17** | **Tasks:** Register Restaurant and Upload Menu | **Priority:** High |
| **Value Statement** | As a **restaurant owner**, I want to **register my restaurant and upload my menu** so that I can start receiving orders. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The system should allow restaurant owners to register with valid credentials. 2. The menu should support multiple categories, pricing, and images. | |

**User Story: 18**

|  |  |  |
| --- | --- | --- |
| **User Story No: 18** | **Tasks:** Update Menu Items | **Priority:** High |
| **Value Statement** | As a **restaurant owner**, I want to **edit menu items (add/remove items, update prices, etc.)** so that my menu is always up to date. | |
| **BV** | **100** | **CP:1** |
| **ACCEPTANCE CRITERIA:** | 1. Menu updates should reflect in real-time. 2. Out-of-stock items should be marked as unavailable. | |

**User Story: 19**

|  |  |  |
| --- | --- | --- |
| **User Story No: 19** | **Tasks:** Set Restaurant Business Hours | **Priority:** High |
| **Value Statement** | As a **restaurant owner**, I want to **set available hours for my restaurant** so that customers can order only when I am open. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The system should allow restaurant owners to set opening and closing hours for each day. 2. Customers should not be able to place an order outside of the business hours. 3. A banner should indicate when the restaurant is closed. | |

**User Story: 20**

|  |  |  |
| --- | --- | --- |
| **User Story No: 20** | **Tasks:** Manage Multiple Restaurant Locations | **Priority:** High |
| **Value Statement** | As a **restaurant owner,**I want to **manage multiple locations from a single account** so that I can track all my branches. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. Owners should be able to register multiple locations under one account. 2. Each location should have its own menu, orders, and earnings reports. 3. Users should see different branches when searching for the restaurant. | |

**User Story: 21**

|  |  |  |
| --- | --- | --- |
| **User Story No: 21** | **Tasks:** Update Order Status | **Priority:** High |
| **Value Statement** | As a **restaurant owner**, I want to **update order status (Preparing, Ready for Pickup, Completed, etc.)** so that customers can track their food. | |
| **BV** | **200** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The system should display a list of new orders with details (customer name, items, special instructions, payment status). 2. Each status change should be timestamped and logged. 3. When the restaurant updates the order status, customers should receive a real-time notification (Push, SMS, or Email). 4. Once the order is marked as "Ready for Pickup," an available delivery partner should be notified. | |

**User Story: 22**

|  |  |  |
| --- | --- | --- |
| **User Story No: 22** | **Tasks:** View Earnings & Financial Reports | **Priority:** High |
| **Value Statement** | As a **restaurant owner**,I want to **view my earnings and financial reports** so that I can track my revenue. | |
| **BV** | **100** | **CP:3** |
| **ACCEPTANCE CRITERIA:** | 1. The system should display daily, weekly, and monthly sales reports. 2. The earnings should be categorized by order type (delivery, pickup). 3. A breakdown of commission and fees should be visible. | |

**User Story: 23**

|  |  |  |
| --- | --- | --- |
| **User Story No: 23** | **Tasks:** Manage Restaurant Promotions & Discounts | **Priority:** High |
| **Value Statement** | As a **restaurant owner**, I want to **offer discounts or special promotions** so that I can attract more customers. | |
| **BV** | **200** | **CP:2** |
| **ACCEPTANCE CRITERIA:** | 1. Owners should be able to create discount codes with conditions (percentage off, minimum order value). 2. Promotions should have start and end dates. 3. Customers should be able to apply valid discount codes at checkout. | |

**User Story: 24**

|  |  |  |
| --- | --- | --- |
| **User Story No: 24** | **Tasks:** Reply to Customer Reviews | **Priority:** High |
| **Value Statement** | As a **restaurant owner**, I want to **respond to customer reviews and feedback** so that I can maintain a good reputation. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The system should display customer reviews on the restaurant’s profile. 2. Owners should be able to respond to reviews publicly. 3. Responses should be moderated for abusive language | |

**User Story: 25**

|  |  |  |
| --- | --- | --- |
| **User Story No: 25** | **Tasks:** Generate Sales Insights & Popular Dishes Report | **Priority:** High |
| **Value Statement** | As a **restaurant owner**, I want to **view sales reports and earnings analytics** so that I can track my business performance. | |
| **BV** | **200** | **CP:3** |
| **ACCEPTANCE CRITERIA:** | 1. The system should provide a report of top-selling dishes. 2. Insights should include total revenue and order count per item. 3. Owners should be able to export reports for further analysis. | |

**User Story: 26**

|  |  |  |
| --- | --- | --- |
| **User Story No: 26** | **Tasks:** Assign Delivery Partners | **Priority:** High |
| **Value Statement** | As a **restaurant owner**, I want to **integrate my restaurant with third-party delivery partners** so that I can ensure smooth deliveries. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The restaurant should be able to assign an available delivery person. 2. The system should send order details to the assigned delivery partner. 3. The customer should be able to track the assigned delivery person | |

**User Story: 27**

|  |  |  |
| --- | --- | --- |
| **User Story No: 27** | **Tasks:** Receive and Manage Orders | **Priority:** High |
| **Value Statement** | As a **restaurant owner**, I want to **receive and manage orders** so that I can process them efficiently. | |
| **BV** | **100** | **CP:1** |
| **ACCEPTANCE CRITERIA:** | 1. Orders should appear in the restaurant dashboard in real-time. 2. Each order should display details (customer name, items, total amount, delivery method). 3. The restaurant should be able to change the order status (e.g., "Preparing," "Ready for Pickup"). 4. A notification should be sent to the delivery partner or customer when the order is ready. | |

**User Story: 28**

|  |  |  |
| --- | --- | --- |
| **User Story No: 28** | **Tasks:** Accept or Reject Orders | **Priority:** High |
| **Value Statement** | As a **restaurant owner**, I want to **accept or reject incoming orders** based on availability. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. New orders should appear in the dashboard with an "Accept" or "Reject" button. 2. If an order is rejected, the customer should receive a notification. 3. Accepted orders should move to the "In Progress" section. | |

**User Story: 29**

|  |  |  |
| --- | --- | --- |
| **User Story No: 29** | **Tasks: R**egister as a Delivery Partner | **Priority:** High |
| **Value Statement** | As a **delivery partner**, I want to **register on the platform** so that I can start delivering orders. | |
| **BV** | **200** | **CP:2** |
| **ACCEPTANCE CRITERIA:** | 1. The system should allow delivery partners to sign up using email, phone number, and password. 2. Required documents (driver’s license, ID, vehicle details) should be uploaded. 3. The system should verify and approve/reject applications. 4. Upon approval, the partner should receive a confirmation notification. | |

**User Story:30**

|  |  |  |
| --- | --- | --- |
| **User Story No: 30** | **Tasks:** Set Availability Status | **Priority:** High |
| **Value Statement** | As a **delivery partner**, I want to **toggle my availability** so that I can choose when to accept orders. | |
| **BV** | **300** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The system should allow partners to mark themselves as "Available" or "Unavailable." 2. When unavailable, the system should not assign new orders. 3. Availability status should update in real time. | |

**User Story: 31**

|  |  |  |
| --- | --- | --- |
| **User Story No:31** | **Tasks:** Receive Delivery Requests | **Priority:** High |
| **Value Statement** | As a **delivery partner**, I want to **receive delivery requests** so that I can choose to accept or reject them. | |
| **BV** | **200** | **CP:1** |
| **ACCEPTANCE CRITERIA:** | 1. New delivery requests should appear in the app with restaurant and customer details. 2. A countdown timer should be displayed to accept or reject the request. 3. If rejected or ignored, the request should be reassigned to another driver. | |

**User Story: 32**

|  |  |  |
| --- | --- | --- |
| **User Story No: 32** | **Tasks:** View Order Details | **Priority:** High |
| **Value Statement** | As a **delivery partner**, I want to **view order details** so that I know what to pick up and deliver. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The order screen should display restaurant name, customer details, and delivery address. 2. The estimated pickup and delivery time should be visible. 3. The total distance and earnings should be displayed. | |

**User Story: 33**

|  |  |  |
| --- | --- | --- |
| **User Story No: 33** | **Tasks:** Navigate to Restaurant for Pickup | **Priority:** High |
| **Value Statement** | As a **delivery partner**, I want to **get navigation to the restaurant** so that I can pick up the order easily. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The app should provide a button to open Google Maps or another GPS service. 2. The route should update based on real-time traffic. 3. The estimated time to the restaurant should be displayed. | |

**User Story: 34**

|  |  |  |
| --- | --- | --- |
| **User Story No: 34** | **Tasks:** Report an Issue with a Delivery | **Priority:** High |
| **Value Statement** | As a **delivery partner**, I want to **report issues with deliveries** so that I can notify the platform of problems. | |
| **BV** | **200** | **CP:3** |
| **ACCEPTANCE CRITERIA:** | 1. A "Report Issue" button should be available in the order details. 2. Issues should be categorized (e.g., "Customer Unreachable," "Wrong Address," "Payment Issue"). 3. The admin should receive the issue report and provide a resolution. | |

**User Story: 35**

|  |  |  |
| --- | --- | --- |
| **User Story No: 35** | **Tasks:** Manage Restaurant Registrations | **Priority:** High |
| **Value Statement** | As a **admin**, I want to **review and approve/reject restaurant registrations** so that only verified restaurants are listed on the platform. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The system should display a list of pending restaurant registrations. 2. Each registration should include business details (name, address, tax ID, menu, contact info). 3. Admin should have the option to "Approve" or "Reject" with a reason. 4. Approved restaurants should be visible to customers. 5. Rejected restaurants should receive an email notification with the rejection reason. | |

**User Story: 36**

|  |  |  |
| --- | --- | --- |
| **User Story No: 36** | **Tasks:** Monitor and Resolve Order Issues | **Priority:** High |
| **Value Statement** | As a **admin**, I want to **view and resolve reported order issues** so that I can provide customer support. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. Admin should be able to view all reported issues (e.g., wrong order, missing items, late delivery). 2. Each issue should include order details and user comments. 3. Admin should be able to refund, compensate, or escalate issues. 4. Customers should receive an update when their issue is resolved. | |

**User Story: 37**

|  |  |  |
| --- | --- | --- |
| **User Story No: 37** | **Tasks:** Manage Users (Customers, Restaurant Owners, Delivery Partners) | **Priority:** High |
| **Value Statement** | As a A**dmin**, I want to **view and manage all user accounts** so that I can take action on policy violations or account issues. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The system should display a list of all users with their roles. 2. Admin should be able to search users by name, email, or phone number. 3. Admin should have the ability to deactivate or suspend an account. 4. Suspended users should be notified and prevented from logging in | |

**User Story: 38**

|  |  |  |
| --- | --- | --- |
| **User Story No: 38** | **Tasks:** Monitor and Resolve Order Issues | **Priority:** High |
| **Value Statement** | As a **admin**, I want to **view and resolve reported order issues** so that I can provide customer support. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. Admin should be able to view all reported issues (e.g., wrong order, missing items, late delivery). 2. Each issue should include order details and user comments. 3. Admin should be able to refund, compensate, or escalate issues. 4. Customers should receive an update when their issue is resolved. | |

**User Story: 39**

|  |  |  |
| --- | --- | --- |
| **User Story No: 39** | **Tasks:** Manage Payments and Payouts | **Priority:** High |
| **Value Statement** | As a **admin**, I want to **track and process payments for restaurants and delivery partners** so that everyone gets paid correctly. | |
| **BV** | **500** | **CP:5** |
| **ACCEPTANCE CRITERIA:** | 1. The system should display pending, processed, and failed payouts. 2. Restaurants and delivery partners should receive payments on a scheduled basis. 3. The system should notify users when payments are processed. 4. Admin should be able to manually process or hold payments if necessary. | |

**User Story: 40**

|  |  |  |
| --- | --- | --- |
| **User Story No: 40** | **Tasks:** Manage Complaints and Support Requests | **Priority:** High |
| **Value Statement** | As a **admin**, I want to **handle complaints from customers, restaurants, and delivery partners** so that I can maintain platform quality. | |
| **BV** | **200** | **CP:2** |
| **ACCEPTANCE CRITERIA:** | 1. Complaints should be categorized (e.g., "Late Delivery," "Payment Issue," "Account Issue"). 2. Admin should be able to respond and resolve complaints. 3. Users should be notified when their complaint is addressed. | |

**Question 3– What is epic? Write 2 epics – 5 Marks**

Business Value and Complexity Point

**Answer**

* Epics, are larger user stories that can be broken down into smaller, manageable stories before the start of a sprint.
* An epic is a significant piece of work that will strategically add value to your product and business.
* 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.

**1. Login** – Consumer should be able to login with Email ID and Password

**2. Order History** – Customers should be able to locate the Current orders and previous orders

**3. Search and Filter** – Customers should be able to search and filters all the available Food option using key word

**4. Design Delivery Map** – As a part of the order status users would like to view the location and arrival time for their food .

**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.

**Question 4 –What is the difference between BV and CP – 2 Marks**

**Business value**

* Business value is the vision where in product owner uses to order the product backlog. It can be derived by lowering costs, increasing revenue, growing customer satisfaction, reducing risk or enhancing capability.
* Business value is what delivered to end customer and how they received it. Feedback from customer is good measure to know the value delivered.

**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.

**Story Points/Complexity Points**

* Rough Estimation of user stories, will be given by Dev and QA team in the form of Planning Poker or Fibonacci series.
* Story points/velocity help development team to plan their work and to have better delivery predictability
* Story points are an important part of user story mapping, and most agile teams use them when planning their work out each sprint.

**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 misalignment's 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 time frame, 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**

* 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.

**Sprint Goal**

* 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.

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.

**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.

**PBI: Product Backlog Item** - Contain list of all the user stories and Epic created by Product owner.

**Task:** Unit of Work done by 1 Developer in 1 Scrum-

**During sprint palling meeting** - So in every user story will be divided in to sub task and will be allotted / Picked up by the developers in that sprint .

**WIP: Work In Progress- work in process** – the features that are in the production process / Phase but not yet the finished product.

WIP therefore refers to all the task that are at various stages of the production

Process.

**Sprint Backlog** - List of committed user stories by dev and QA team for that particular sprint will be added in the sprint backlog.

|  |  |  |  |
| --- | --- | --- | --- |
| **PBI** | **TASKS** | **WIP** | **DONE** |
| Login | Login user with email and password. A temporary  password will be sent to the admins registered email  address with a link for a password reset. | View, filter previously  added data | DOD |
|  |  | Select reporting period, Indicators |  |
|  |  | Collect Scope 1 data |  |
|  |  | Collect Scope 2 data |  |

**Question 6 – Explain Product backlog and sprint back log– 5 Marks**

**Product backlog**

* A product backlog is a prioritized list of work for the development team that is derived from the road map and its requirements.
* The most important items are shown at the top of the product backlog so the team knows what to deliver first.
* 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 back log**

* Sprint back log is the subset of product backlog
* A sprint backlog is the set of items that a cross-functional product team selects from its product backlog to work on during the upcoming sprint.
* Typically the team will agree on these items during its sprint planning session. In fact, the sprint backlog represents the primary output of sprint planning.

**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.

|  |  |  |
| --- | --- | --- |
| **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 – 5 Marks**

* An Agile impediment log is used to record, track and resolve a list of obstacles, challenges that delay the team's performance.
* An Impediment is anything that keeps the Team from getting work Done and that slows Velocity.

Ex- sick team member, a missing resource, lack of management support Business or customer issues; Unresolved dependencies

**Organizational Impediments** – issues that are dependent on others to solve. These issues include but are not limited to:

* Slow internet
* Issues with obtaining input from other teams or divisions
* Lack of training

### ****Impediment 1: Delays in Real-Time Order Assignment****

**Description:**  
The system takes too long to assign delivery requests to available drivers, causing order fulfillment delays and poor customer experience.

**Impact:**

* Increased customer complaints due to delayed deliveries.
* Decreased efficiency in delivery partner utilization.
* Potential revenue loss for restaurants due to cold food deliveries.

**Possible Resolution:**

* Optimize the order assignment algorithm to consider real-time driver availability, distance, and restaurant preparation time.
* Implement load testing to identify system bottlenecks.
* Introduce fallback mechanisms to manually assign orders if automation fails.

### ****Impediment 2: Restaurant Menu and Pricing Updates Are Slow****

**Description:**  
Restaurants frequently update their menus and pricing, but the approval process by the admin is slow, leading to outdated information being displayed to customers.

**Impact:**

* Customers may place orders with incorrect pricing, causing refund or dispute issues.
* Restaurants may lose sales due to unavailable items being displayed.
* Increased workload for customer support to handle complaints.

**Possible Resolution:**

* Automate menu updates with a real-time sync feature while keeping an audit trail.
* Implement an AI-driven moderation system to flag only suspicious changes for manual review.
* Allow restaurants to mark items as "Temporarily Unavailable" without requiring admin approval.

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

**Velocity – How many CP is covered in this sprint**

Velocity is a measure of the amount of work a Team can tackle during a single Sprint and is the key metric in Scrum. Velocity is calculated at the end of the Sprint by totaling the Points for all fully completed User Stories.

Estimated time for this course: 5 minutes.

Actual velocity is calculated by dividing the total Story Points completed by the team by the number of Sprints. For instance, if the Scrum Team has finished a total of 80 points over 4 Sprints then the actual velocity of the team would be 20 points per Sprint

80 / 4 = 20

**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– 3 Marks**

A product burn down chart shows how much work remains for the entire project, whereas a sprint burn down chart shows how much work remains in a specific iteration.

A sprint burndown chart includes:

**X-axis**- The horizontal axis of the graph represents the remaining amount of time to complete the project usually depicted in days. Sprints

**Y-axis-** The vertical axis of the graph represents the effort needed to complete the project. Story Pt

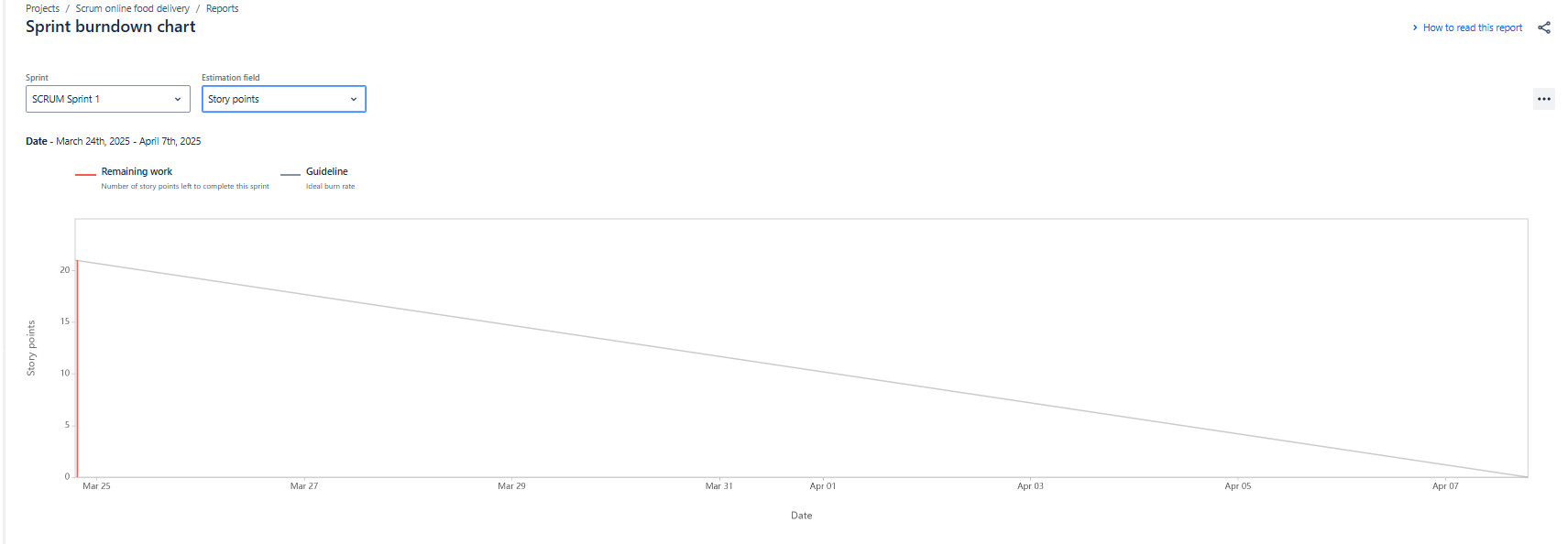
**Actual work line** - This represents the actual number of tasks remaining. It might be straight in some cases; however, it often changes due to unforeseen issues in the project or an increase in the number of tasks.

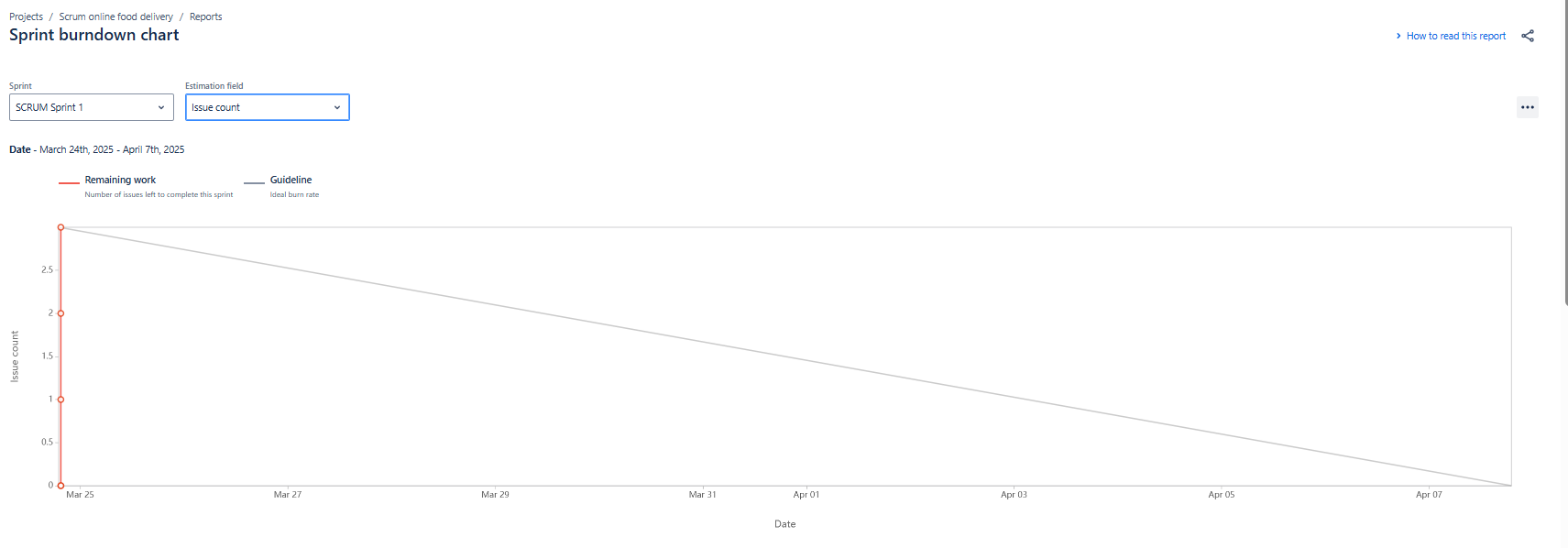
**Sprint Burn-Down Chart:**

* **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 – 2 Marks**

* Grooming is an open discussion between the development team and product owner. The user stories are discussed to help the team gain a better understanding of the functionality that is needed to fulfill a story. This includes design considerations, integration, and expected user interactions.
* Product Backlog grooming is a regular session where backlog items are discussed, reviewed, and prioritized by product managers, product owners, and the rest of the team.
* The primary goal of backlog grooming is to keep the backlog up-to-date and ensure that backlog items are prepared for upcoming Sprint.

**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 – 3 Marks**

**Scrum Master**

* A Scrum Master popularly known as a coach, motivator and leader of an Agile team.
* The role of a Scrum Master is to educate the team on Agile processes and help team members follow Scrum practices religiously. Facilitation scrum event as and when it is required .
* The Scrum Master collaborates both with the Product Owner (PO) who focuses on building the right product, and the development team that focuses on building the product right.
* A Scrum Master’s job is essentially to help everyone understand and imbibe Scrum values, principles, and practices and get the best product out to the customer.

The Scrum Master acts as a **servant-leader** for the Scrum Team, ensuring the team follows Scrum principles and removes obstacles that hinder progress. Their key responsibilities include:

* **Facilitating Scrum Events**: Organizing and leading Daily Stand ups, Sprint Planning, Sprint Review, and Sprint Retrospective.
* **Coaching the Team**: Helping the development team understand and implement Scrum best practices.
* **Removing Impediments**: Identifying and eliminating roadblocks that slow down the team’s work.
* **Ensuring Collaboration**: Fostering communication between the team, Product Owner, and stakeholders.
* **Protecting the Team**: Shielding the team from unnecessary distractions and external pressures.

**Product Owner**

* The Product Owner takes the lead in many aspects of a product’s development. As a member of the Scrum Team, the Product Owner provides clarity to the team about a product’s vision and goal.
* All work is derived and prioritized based on the Product Goal in order to deliver value to all stakeholders including those within their organization and all users both inside and out.
* Product Owners identify, measure and maximize value throughout the entire product's life cycle.
* The Product Owner is responsible for **maximizing the value** of the product by managing the **Product Backlog** and ensuring the team is working on the most valuable tasks.

Their key responsibilities include:

* Defining the vision
* Prioritizing the product backlog
* Taking an overview of development stages
* Handling communications
* Knowing what the client needs
* Evaluating progress

**Defining the Product Vision**: Clearly articulating the product goals and road map.

**Managing the Product Backlog**: Prioritizing user stories and refining backlog items based on business needs and stakeholder feedback.

**Collaborating with Stakeholders**: Gathering requirements from customers, business teams, and other stakeholders.

**Ensuring Clear Requirements**: Providing the development team with well-defined and prioritized backlog items.

**Accepting or Rejecting Work**: Reviewing completed work to ensure it meets acceptance criteria and delivers business value.

|  |  |  |
| --- | --- | --- |
| **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 |

**Question 12 – Explain all Meetings Conducted in Scrum Project – 8 Marks**

1. **Sprint Planning Meeting**

This happens at the beginning of each sprint and team decides on what they will be delivering in the sprint.

* **Purpose**: To define the work that will be completed in the upcoming sprint.
* **Participants**: 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)
* **Key Activities**:
* The **Product Owner** presents prioritized backlog items.
* The **Development Team** selects tasks and estimates workload.
* The **Sprint Goal** is set.
* **Outcome**: A Sprint Backlog containing tasks to be completed in the sprint.

**2)Daily Scrum Meeting**

* **Purpose**: A short daily meeting (15 minutes) to synchronize team activities.
* **Participants**: Development Team, Scrum Master (optional), Product Owner (optional).
* **Duration**: 15 minutes, held at the same time and place daily.
* **Key Activities**:

This happens each day where team will just answer 3 questions:

1)What did u do today?

2)what will u do tomorrow?

3)Are there any impediments that is slowing or stopping u?

**Outcome**: Identifies progress and roadblocks for the team.

**3)Sprint Review Meeting**

This happens at the end of the sprint where team will demo the-completed stories to product owner and get it cleared.

* **Purpose**: To demonstrate the completed work to stakeholders and get feedback.
* **Participants**: Scrum Team, Stakeholders, Clients.
* **Duration**: Typically 1 hour per week of sprint length (e.g., a 2-hour Sprint Review for a 2-week sprint)
* **Key Activities**:
* The team presents the completed work.
* Stakeholders provide feedback for future improvements.
* **Outcome**: Potential adjustments to the Product Backlog based on feedback.

1. **Sprint Retrospective Meeting**

* **Purpose**: To reflect on the sprint and identify improvements.
* **Participants**: Scrum Master, Product Owner, Development Team.
* **Duration**: Typically 45 minutes per week of sprint length (e.g., a 1.5-hour Retrospective for a 2-week sprint)
* **Key Activities**:

This happens at the end of the sprint where team will answer these 3 questions:

1)What went well in the sprint?

2)what did not go well?

3)What are the required areas of improvements in next sprint?

4)Create an action plan for process improvements.

* **Outcome**: Continuous improvement strategies for the next sprint.

### ****5)Backlog Refinement (Grooming) Meeting:****

* **Purpose**: To review and refine the Product Backlog.
* **Participants**: Product Owner, Scrum Master, Development Team.
* **Duration**: Generally, 5-10% of the team’s time during a sprint.
* **Key Activities**:
* Prioritizing and breaking down user stories.
* Clarifying requirements and adding acceptance criteria.
* **Outcome**: A well-prepared backlog for future sprints.

**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– 2 Marks**

**Sprint Size**

* Sprints are the soul of Scrum methodology within Agile Project Management.
* A Sprint is a time-boxed event of weeks in which your Scrum team focuses only on a sprint goal.
* The goal is typically a product increment or iteration, often an updated, improved version of your product or software. Normally a sprint happens for two weeks but may extend to 4 weeks.

**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 time frame. 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.

**Scrum Size**

* The optimum size for the scrum team is around 10 members with varying skill sets and large enough to accomplish the tasks comfortably and share, communicate, and collaborate effectively.
* A Scrum team will have 1 Scrum Master, 1Product Owner and 8 to 10 Scrum Developers.
* Each team size will be average 7-8.

**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.

|  |  |  |
| --- | --- | --- |
| **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. |

Q**uestion 14 – Explain DOR and DOD – 2 Marks**

**DOR(Definition Of Ready)**

The **DoR** ensures that a user story or backlog item is fully prepared before development begins. It helps the team avoid ambiguity and ensures smooth execution.

**Key Criteria for DoR:**  
User Story is clearly written and follows the **INVEST** principle (Independent, Negotiable, Valuable, Estimable, Small, Testable).

* Acceptance criteria are well-defined.
* Dependencies are identified and resolved.
* The story is estimated and prioritized in the backlog.
* The team understands the story and has no open questions.

**Example of DoR:**

* "As a user, I should be able to reset my password via email."
* Acceptance Criteria:
* User enters registered email and receives a reset link.
* The reset link expires after 24 hours.

### ****Definition of Done (DoD)**** The **DoD** is a checklist that ensures work is completed and meets the required quality before being considered "done." It applies to user stories, features, or even entire releases.

### ****Key Criteria for DoD:****

* Code is unit tested with a certain coverage percentage.
* Functional testing is completed.
* User acceptance criteria are met.
* The feature is deployed in a staging environment.
* Documentation is updated (if required).
* Code is written, peer-reviewed, and merged.

**Example of DoD for a User Story:**

* **C**ode is committed and reviewed.
* Automated tests have passed.
* Feature is tested on different browsers/devices.
* The Product Owner has approved the feature.

|  |  |
| --- | --- |
| **DOR** | **DOD** |
| User story is clear | Code produced all to do items in the code completed |
| User story is testable | Code commented, checked and run against the current version in the source control |
| User story is feasible | Peer reviewed and meeting development  standards |
| User story is defined | Code build without error |
| User story acceptance criteria is  defined | Unit test written and accepted |
| User story dependencies defined | Deployed to system test environment and passed system testes |
| User story team size defined | Passed UAT and signed off as meeting  requirements |
| scrum team accept the user story  artifacts | Any build and deployment or configuration changes are implemented and documented and communicated |
| Performance criteria identified  where appropriate | Relevant documentation, diagrams produced or updated |
| Team has a good idea what it will  mean to demo the user story | “in progress” to “done |

**Question 15 – Explain Prioritization Techniques and MVP – 3 Marks**

Prioritization helps **Product Owners** decide which features or tasks should be developed first.

Common techniques include:

**A)MoSCoW**

* MoSCoW is a prioritization technique used in business analysis and software development to reach a common understanding with stakeholders on the importance they place on the delivery of each requirement .
* Also known as MoSCoW prioritization or MoSCoW analysis.
* **MoSCoW stands for must,should,could and would**

M-Must have this requirement to meet the business needs.

S-Should have this requirement if possible,but project success does not rely on it.

C-Could have this requirement if it does not affect anything else in the project.

W-Would like to have this requirement later,but it won't be delivered this time.

#### ****B) Eisenhower Matrix****

* **Urgent & Important** – Do it now.
* **Important but Not Urgent** – Schedule it.
* **Urgent but Not Important** – Delegate it.
* **Neither Urgent nor Important** – Eliminate it.

#### ****C) Value vs. Effort Matrix****

* Features are plotted on a graph based on **business value** (impact) and **effort** (cost).
* High-value, low-effort tasks are prioritized first.

1. **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.
2. **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.

**F)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.

**FURPS**

* This technique is used to validate must requirement.
* FURPS is an acronym representing a model for classifying software quality attributes (functional &non-functional requirements):

**Functionality** -Feature set,Capabilities,Generality,Security

**Usability** -Human factors,Aesthetics,Consistency,Documentation

**Reliability** -Frequency/severity of failure,Recoverability,Predictability,Accuracy,Mean time to failure

**Performance** -Speed,Efficiency,Resource consumption,Throughput,Response time

**Supportability-**Testability,Extensibility,Adaptability,Maintainability,Compatibility,Configurability,Serviceability,Installability,Localizability,Portability

**100 Dollars Test**

Top 10 requirements = Numerical Assignment-Mandatory, very important, rather important, not important; does not matter.

**Minimum viable product (MVP)**

* A minimum viable product (MVP) is the release of a new product (or a major new feature) that is used to validate customer needs and demands prior to developing a more fully featured product.
* To reduce development time and effort, an MVP includes only the minimum capabilities required to be a viable customer solution.
* A minimum viable product is a version of a product with just enough features to be usable by early customers who can then provide feedback for future product development.
* A focus on releasing an MVP means that developers potentially avoid lengthy and unnecessary work.

**Definition**: The **MVP** is the simplest version of a product that delivers **core functionality** and provides value to users.

**Purpose**: To launch quickly, gather user feedback, and iterate improvements.

**Example**: A **food delivery app MVP** may only include **restaurant selection, order placement, and payment**—leaving advanced features like tracking for later.

**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**: MVP’s 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.

**Question 16 – Difference between Business Analyst n Product Owner – 3 Marks**

**Business Analyst**

A **Business Analyst** focuses on understanding and documenting business needs,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**

* A **Product Owner** makes product-related decisions and ensures development aligns with business goals.
* 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 – 3 Marks**

**SAMPLE RESUME**

**Jaishree Tamilselvan**

**Email address:[jaishreeselvan@gmail.com](mailto:jaishreeselvan@gmail.com)**

**Contact num:8248115038**

**Location:Chennai,Tamilnadu,India.**

**Linked In: linked in.com/in/jaishree**

### ****Professional Summary****

Results-driven **Product Owner** with **3 years of experience** in Agile environments, specializing in **product backlog management, stakeholder collaboration, and user story creation**. Skilled in delivering high-value solutions by bridging the gap between business needs and development teams. Adept at working with cross-functional teams to drive product success.

**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

### ****Work Experience****

### ****Product Owner |ABC Technologies,Chennai | march 2023– Present****

* Defined and prioritized the **product backlog**, ensuring alignment with business goals.
* Collaborated with **stakeholders, developers, and designers** to deliver high-impact features.
* Led **Sprint Planning, Reviews, and Retrospectives**, ensuring smooth Agile execution.
* Created **detailed user stories, acceptance criteria, and wire frames** for development teams.
* Worked with **QA teams** to define test cases and ensure high product quality.
* Increased customer satisfaction by **20%** through continuous feedback loops and enhancements.

**Associate Product Owner|*X*YZ Solutions, Chennai|march 2022-mar 2023**

* Conducted market research and gathered user requirements for product development.
* Assisted the **Product Owner** in backlog refinement and prioritization.
* Created process flows, wire frames, and requirement documents for **cross-functional teams**.

### ****Projects & Achievements****

* Launched a new **e-commerce feature**, leading to a **15% increase in user engagement**.
* Reduced development bottlenecks by **30%** through improved backlog management.

### ****Certifications****

* **Certified Scrum Product Owner (CSPO)** – Scrum Alliance
* **Professional Scrum Product Owner (PSPO)** – Scrum.org

**EDUCATION**

**Bachelor of Engineering(B.E)**  
XYZ University, Vellore  
Graduated: 2017  
Specialization: Computer Science Engineering