1. Write Agile Manifesto:

The Agile Manifesto is a brief document built on 4 values and 12 principles for agile software development. The Agile Manifesto was published in February 2001 and is the work of 17 software development practitioners who observed the increasing need for an alternative to documentation-driven and heavyweight software development processes.

These values and principles prioritize flexibility, adaptability, customer satisfaction, and collaboration, aiming to deliver high-quality software in a more efficient and responsive manner. Agile methodologies, such as Scrum, Kanban, and Extreme Programming (XP), have since emerged as practical implementations of the principles outlined in the Agile Manifesto, guiding teams in iterative development, continuous improvement, and customer-focused delivery.

Four values of the agile manifesto are:

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

12 Principles of agile manifesto are:

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

2. User stories, acceptance criteria- BV and CP

|  |  |  |  |
| --- | --- | --- | --- |
| User story 1 | Tasks – 2 | | Priority: high |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO VIEW MY PURCHASE HISTORY  SO THAT I CAN TRACK MY TRANSACTIONS | | | |
| BV: 200 | | CP : 3 | |
| Acceptance criteria:   * Br-all mandatory * Display list of previous orders * Include order date, id, total amount * Clickable order for detailed view   Alternative flow:   * Sort orders by date or amount   Exceptional flow:   * Display message if no orders found | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 2 | Tasks – 2 | | Priority: High |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO VIEW RESTAURANT RATINGS AND REVIEWS  SO THAT I CAN MAKE INFORMED DECISIONS ABOUT WHERE TO ORDER FROM. | | | |
| BV: 200 | | CP : 4 | |
| Acceptance criteria:   * User can view ratings and reviews for each restaurant. * Reviews are sorted by date or helpfulness and can be filtered by rating. * Users can leave reviews and ratings after order completion. * User scrolls through ratings and reviews.   Alternative flow:   * Allow users to sort reviews based on specific criteria (e.g., taste, delivery speed).   Exceptional flow:   * If restaurant has no reviews, display message indicating lack of feedback | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 3 | Tasks – 2 | | Priority: Medium |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO SUBSCRIBE TO EMAIL NEWSLETTER  SO THAT I CAN RECEIVE LATEST PROMOTIONS AND UPDATES | | | |
| BV: 200 | | CP : 1 | |
| Acceptance criteria:   * Br-all mandatory * Email subscription form * Confirmation message after subscription   Alternative flow:   * Option to select preferred content   Exceptional flow:   * Display error message if email already subscribed | | | |
| User story 4 | Tasks – 2 | | Priority: Medium |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO TRACK MY SHIPMENT  SO THAT I CAN MONITOR DELIVERY STATUS | | | |
| BV : 100 | | CP: 3 | |
| Acceptance criteria:   * Br-all mandatory * Tracking number input field * Display shipment status and etc.   Alternative flow:   * Track shipment via email link   Exceptional flow:   * Provide support contact if tracking is unavailable | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 5 | Tasks – 3 | | Priority: medium |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO ADD PRODUCTS TO MY SHOPPING CART  SO THAT I CAN PURCHASE THEM LATER | | | |
| BV : 100 | | CP : 1 | |
| Acceptance criteria:   * Br-all mandatory * Add to cart button on product pages * Display total items in cart icon   Alternative flow:   * Save cart for later   Exceptional flow:   * Notify customer if item is out of stock | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 6 | Tasks – 3 | | Priority: High |
| VALUE STATEMENT: AS A CUSTOMER,  I WANT TO CUSTOMIZE MY ORDER (E.G., TOPPINGS, SPICE LEVEL)  SO THAT I CAN TAILOR IT TO MY PREFERENCES. | | | |
| BV : 200 | | CP : 5 | |
| Acceptance criteria:   * User selects menu item. * User chooses desired customizations. * Customizations are applied to order. * User proceeds to checkout.   Alternative flow:   * Allow users to save favorite customizations for future orders.   Exceptional flow:   * If customizations are not transmitted correctly, prompt user to review and confirm. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 7 | Tasks – 3 | | Priority: High |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO VIEW NUTRITIONAL INFORMATION FOR MENU ITEMS  SO THAT I CAN MAKE HEALTHIER FOOD CHOICES. | | | |
| BV : 200 | | CP : 4 | |
| Acceptance criteria:   * User can access nutritional information for each menu item (calories, ingredients, allergens). * Nutritional labels are displayed clearly and prominently in app interface. * Information is sourced from reliable sources and regularly updated.   Alternative flow:   * Allow users to filter menu items based on dietary preferences (e.g., vegetarian, low-calorie).   Exceptional flow:   * If nutritional information is unavailable for certain items, display message indicating lack of data. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 8 | Tasks – 3 | | Priority: Medium |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO SAVE MULTIPLE DELIVERY ADDRESSES (E.G., HOME, WORK)  SO THAT I CAN CHOOSE THE APPROPRIATE LOCATION FOR EACH ORDER. | | | |
| BV : 100 | | CP : 4 | |
| Acceptance criteria:   * User can add, edit, and delete delivery addresses in app settings. * Address entry form includes fields for name, address, and optional notes (e.g., delivery instructions). * User can select delivery address during checkout from saved addresses. * User adds new address or edits existing one.   Alternative flow:   * Allow users to set default delivery address for faster checkout.   Exceptional flow:   * If address validation fails, prompt user to review and correct input. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 9 | Tasks – 2 | | Priority: High |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO RECEIVE REAL-TIME NOTIFICATIONS ABOUT ORDER STATUS CHANGES  SO THAT I CAN STAY UPDATED ON MY DELIVERY. | | | |
| BV : 200 | | CP : 6 | |
| Acceptance criteria:   * User receives push notifications for order confirmation, preparation, out for delivery, and delivered. * Notifications include relevant order details (items, estimated delivery time). * User can customize notification preferences in app settings. * User receives status updates as order progresses.   Alternative flow:   * Allow users to track order status manually within the app.   Exceptional flow:   * If notification delivery fails, provide alternative communication channels (email, SMS). | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 10 | Tasks – 2 | | Priority: High |
| VALUE STATEMENT: AS A DELIVERY DRIVER  I WANT TO UPDATE MY AVAILABILITY STATUS (E.G., ONLINE, OFFLINE)  SO THAT I CAN MANAGE MY WORK SCHEDULE EFFECTIVELY. | | | |
| BV : 200 | | CP : 5 | |
| Acceptance criteria:   * Delivery driver can toggle availability status between online and offline. * Status change is reflected instantly in app and backend systems. * Driver receives delivery requests only when online and available. * Driver opens app and toggles availability to online.   Alternative flow:   * Allow drivers to set predefined availability schedules for recurring shifts.   Exceptional flow:   * If driver experiences technical issues with status update, provide manual override option. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 11 | Tasks – 2 | | Priority: High |
| VALUE STATEMENT: AS A REGIONAL ADMIN,  I WANT TO ANALYSE DELIVERY PERFORMANCE BY ZONE  SO THAT I CAN IDENTIFY AREAS FOR OPTIMIZATION AND RESOURCE ALLOCATION. | | | |
| BV : 200 | | CP : 8 | |
| Acceptance criteria:   * Regional admin can access zone-wise performance metrics in analytics dashboard. * Metrics include average delivery time, on-time delivery rate, and customer satisfaction scores. * Admin can compare performance across different zones and identify trends or outliers.   Alternative flow:   * Allow admins to drill down into specific time periods or delivery types for deeper analysis.   Exceptional flow:   * If data visualization tools encounter errors, provide error handling and troubleshooting guidance. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 12 | Tasks – 3 | | Priority: High |
| VALUE STATEMENT: AS A BUSINESS OWNER  I WANT TO REWARD LOYAL CUSTOMERS WITH A LOYALTY PROGRAM  SO THAT I CAN INCENTIVIZE REPEAT ORDERS AND FOSTER CUSTOMER RETENTION. | | | |
| BV : 200 | | CP : 7 | |
| Acceptance criteria:   * User earns loyalty points for each completed order. * Points can be redeemed for discounts, free items, or exclusive perks. * Loyalty program status and points balance are displayed in user profile.   Alternative flow:   * Offer bonus points for referrals or special promotions/events.   Exceptional flow:   * If point’s redemption fails, notify user and provide assistance for troubleshooting. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 13 | Tasks – 3 | | Priority: Medium |
| VALUE STATEMENT: AS A DELIVERY DRIVER  I WANT TO PROVIDE FEEDBACK ON CUSTOMER BEHAVIOUR AND DELIVERY CONDITIONS  SO THAT I CAN REPORT ISSUES OR PRAISE EXCEPTIONAL SERVICE. | | | |
| BV : 100 | | CP :5 | |
| Acceptance criteria:   * Delivery driver can submit feedback on customer behavior, delivery experience, and overall service. * Feedback form includes fields for description, category selection, and optional attachment. * Submitted feedback is reviewed and acted upon by support team. * Driver fills out feedback form.   Alternative flow:   * Allow drivers to provide anonymous feedback for sensitive issues.   Exceptional flow:   * If feedback submission encounters errors, provide option to save draft and retry. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 14 | Tasks – 3 | | Priority: High |
| VALUE STATEMENT: AS A BUSINESS OWNER  I WANT TO OFFER PROMOTIONAL CODES OR VOUCHERS TO CUSTOMERS FOR MARKETING CAMPAIGNS OR SPECIAL EVENTS. | | | |
| BV : 200 | | CP : 8 | |
| Acceptance criteria:   * Business owner can generate unique voucher codes with customizable settings (discount percentage, expiry date). * Vouchers are distributed to target audience via email, SMS, or app notifications. * Users can redeem vouchers during checkout for applicable discounts or rewards. * Code is distributed to customers.   Alternative flow:   * Offer tiered discounts or rewards based on voucher usage or customer loyalty.   Exceptional flow:   * If voucher validation fails, provide error message and troubleshooting steps. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 15 | Tasks – 3 | | Priority: Low |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO SCHEDULE FUTURE ORDERS  SO THAT I CAN PLAN MEALS IN ADVANCE AND AVOID LAST-MINUTE HASSLES. | | | |
| BV : 50 | | CP : 7 | |
| Acceptance criteria:   * User can choose future delivery date and time during checkout. * App displays scheduled orders separately from immediate orders. * Orders are processed and delivered as per scheduled time slot. * User chooses future delivery date and time.   Alternative flow:   * Allow users to edit or cancel scheduled orders within a specified timeframe.   Exceptional flow:   * If scheduled delivery slot becomes unavailable, prompt user to select alternative time. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 16 | Tasks – 3 | | Priority: High |
| VALUE STATEMENT: AS A CUSTOMER,  I WANT TO VIEW REAL-TIME LOCATION OF DELIVERY DRIVERS ON MAP  SO THAT I CAN TRACK MY ORDER'S PROGRESS. | | | |
| BV : 200 | | CP : 8 | |
| Acceptance criteria:   * User can access map view with real-time location of assigned delivery driver. * Map displays driver's route and estimated time of arrival (ETA). * User receives notifications for significant order status changes (e.g., driver en route, arrived). * User places order and receives confirmation.   Alternative flow:   * Allow users to share tracking link with others (e.g., family members) for joint monitoring.   Exceptional flow:   * If driver location is unavailable, display last known location and ETA based on historical data. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 17 | Tasks – 3 | | Priority: High |
| VALUE STATEMENT: AS A RESTAURANT OWNER  I WANT TO MANAGE MENU ITEMS AND PRICING IN REAL-TIME  SO THAT I CAN UPDATE OFFERINGS AND PROMOTIONS AS NEEDED. | | | |
| BV : 200 | | CP : 9 | |
| Acceptance criteria:   * Restaurant owner can add, edit, or remove menu items from app dashboard. * Changes to menu items (e.g., price, availability) reflect instantly in user-facing app. * Owner receives confirmation and audit trail for menu changes. * Owner selects menu management section.   Alternative flow:   * Allow owners to schedule menu changes for future activation.   Exceptional flow:   * If menu update fails to sync across platforms, provide manual synchronization option. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 18 | Tasks – 3 | | Priority: High |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO SPLIT MY ORDER PAYMENT WITH FRIENDS OR FAMILY MEMBERS  SO THAT WE CAN SHARE THE COST OF THE MEAL. | | | |
| BV : 500 | | CP : 10 | |
| Acceptance criteria:   * User can split order payment with specified number of participants. * App calculates individual payment amounts based on user input. * Participants receive payment request/notification with payment link. * App generates payment links and sends to participants.   Alternative flow:   * Allow users to split payment after order placement via payment history section.   Exceptional flow:   * If payment link expires, allow user to resend or generate new link. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 19 | Tasks – 3 | | Priority: Medium |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO VIEW ESTIMATED DELIVERY TIME BASED ON CURRENT ORDER VOLUME AND TRAFFIC CONDITIONS SO THAT I CAN PLAN MY MEAL ACCORDINGLY. | | | |
| BV : 100 | | CP : 7 | |
| Acceptance criteria:   * User sees estimated delivery time during checkout process. * Delivery time updates dynamically based on changing conditions (e.g., traffic, order queue). * User receives notifications for significant changes in delivery time. * User views estimated delivery time before confirming order.   Alternative flow:   * Allow users to opt for faster delivery with premium fee.   Exceptional flow:   * If delivery time exceeds estimated window, provide compensation or discount to user. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 20 | Tasks – 3 | | Priority: Low |
| VALUE STATEMENT: AS A DELIVERY DRIVER  I WANT TO RECEIVE OPTIMIZED ROUTE DIRECTIONS TO DELIVERY LOCATIONS  SO THAT I CAN MINIMIZE TRAVEL TIME AND FUEL CONSUMPTION. | | | |
| BV : 20 | | CP : 8 | |
| Acceptance criteria:   * Driver receives optimized route directions to delivery locations. * Navigation app recalculates route dynamically based on real-time traffic conditions. * App provides estimated time of arrival (ETA) and alternative routes. * Driver follows navigation instructions to destination.   Alternative flow:   * Allow drivers to manually adjust route or add waypoints for specific stops.   Exceptional flow:   * If navigation app encounters errors, provide alternative navigation options or manual directions. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 21 | Tasks – 2 | | Priority: High |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO BE ABLE TO ADD ITEMS TO MY CART  SO THAT I CAN PROCEED TO CHECKOUT. | | | |
| BV : 200 | | CP : 3 | |
| Acceptance criteria:   * User should be able to browse available items and select desired ones. * Selected items should be added to the cart. * Cart icon should display the total number of items added. * User should be able to view and edit the contents of the cart. * Upon checkout, the user should be directed to the payment page.   Alternative Flow:   * If the user attempts to add an out-of-stock item, display a message indicating the item is unavailable and suggest similar alternatives.   Exceptional flow:   * If there's a system error during item addition, display an error message and prompt the user to try again later. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 22 | Tasks – 2 | | Priority: Medium |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO BE ABLE TO APPLY PROMO CODES DURING CHECKOUT  SO THAT I CAN AVAIL DISCOUNTS. | | | |
| BV : 100 | | CP : 4 | |
| Acceptance criteria:   * User should have an option to enter a promo code on the checkout page. * System should validate the promo code and apply the relevant discount. * Discounted amount should be reflected in the order summary. * Promo code entry field should have validation for correct format.   Alternative Flow:   * If the promo code entered is invalid, display an error message and prompt the user to enter a valid code.   Exceptional flow:   * If the promo code validation fails due to a technical issue, log the error and inform the user that the discount couldn't be applied at the moment. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 23 | Tasks – 2 | | Priority: Medium |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO RECEIVE ORDER CONFIRMATION EMAILS AFTER PLACING AN ORDER  SO THAT I HAVE A RECORD OF MY PURCHASE. | | | |
| BV : 100 | | CP : 3 | |
| Acceptance criteria:   * System should automatically send an order confirmation email to the customer's registered email address. * Email should contain details such as order ID, items purchased, total amount, and estimated delivery time. * Email content should be formatted appropriately for readability. * Order confirmation email should be sent immediately after order placement.   Alternative Flow:   * If the user opts out of receiving email confirmations, a confirmation message should be displayed on the order confirmation page.   Exceptional flow:   * If the email fails to send due to a server issue, log the error and inform the user that the confirmation email couldn't be sent but their order is still placed. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 24 | Tasks – 2 | | Priority: Medium |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO VIEW MY ORDER STATUS IN REAL-TIME  SO THAT I CAN TRACK ITS PROGRESS. | | | |
| BV : 100 | | CP : 3 | |
| Acceptance criteria:   * User should have a dedicated section in the app to view order status. * Order status should be updated in real-time, reflecting stages like 'processing,' 'out for delivery,' and 'delivered.' * User should receive push notifications for significant status updates. * User should be able to refresh the order status manually.   Alternative Flow:   * If the user's internet connection is lost while checking the order status, display a message indicating the connection issue and prompt the user to try again.   Exceptional flow:   * If the order status fails to update due to a technical issue, display an error message and provide a customer support contact for assistance. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 25 | Tasks – 2 | | Priority: Low |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO BE ABLE TO SAVE MULTIPLE DELIVERY ADDRESSES  SO THAT I CAN CHOOSE FROM THEM DURING CHECKOUT. | | | |
| BV : 10 | | CP : 4 | |
| Acceptance criteria:   * User should have an option to save new delivery addresses in the account settings. * Saved addresses should be editable and deletable. * During checkout, user should be able to select from saved addresses or enter a new one.   Alternative Flow:   * If the user attempts to save an address with incomplete information, display an error message and prompt the user to provide all required details.   Exceptional flow:   * If the address saving process fails due to a database error, log the issue and inform the user that their address couldn't be saved at the moment. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 26 | Tasks – 2 | | Priority: Medium |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO BE ABLE TO LEAVE SPECIAL INSTRUCTIONS FOR MY ORDER  SO THAT IT MEETS MY PREFERENCES. | | | |
| BV : 100 | | CP : 2 | |
| Acceptance criteria:   * User should find a text box for special instructions during checkout. * Instructions should be limited to a certain character count for clarity. * Special instructions should be displayed along with the order details for the restaurant to see.   Alternative Flow:   * If the user submits long or invalid instructions, display a warning message indicating the character limit and prompt the user to revise their instructions.   Exceptional flow:   * If the special instructions fail to save due to a server error, inform the user that their instructions couldn't be added at the moment and advise them to contact customer support if necessary. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 27 | Tasks – 2 | | Priority: Low |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO RECEIVE SMS NOTIFICATIONS FOR ORDER UPDATES  SO THAT I CAN STAY INFORMED EVEN WITHOUT ACCESSING THE APP. | | | |
| BV : 50 | | CP : 3 | |
| Acceptance criteria:   * User should receive an SMS notification for order confirmation. * SMS notifications should be sent for order status updates like 'out for delivery' and 'delivered.' * SMS notifications should include relevant order details and a link to track the order.   Alternative flow:   * If the user's phone number is not verified, prompt the user to verify their phone number before enabling SMS notifications.   Exceptional flow:   * If the SMS notification service experiences downtime, display an error message indicating that SMS updates are temporarily unavailable and advise the user to check the app for order status. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 28 | Tasks – 2 | | Priority: High |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO VIEW DETAILED ITEM DESCRIPTIONS AND IMAGES BEFORE ADDING THEM TO MY CART  SO THAT I CAN MAKE INFORMED CHOICES | | | |
| BV : 200 | | CP : 4 | |
| Acceptance criteria:   * Each item in the menu should have a dedicated page with a description and high-quality images. * User should be able to access item pages by tapping on the item from the menu. * Item pages should display information such as ingredients, portion size, and nutritional information.   Alternative flow:   * If the user taps on an item with incomplete information, display a message indicating that details are being updated and provide an estimated time for availability.   Exceptional flow:   * If the item description fails to load due to a server error, display a generic message indicating a temporary issue and advise the user to try again later. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 29 | Tasks – 2 | | Priority: Medium |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO BE ABLE TO REORDER MY FAVOURITE ITEMS QUICKLY FROM MY ORDER HISTORY SO THAT I CAN SAVE TIME. | | | |
| BV : 100 | | CP : 3 | |
| Acceptance criteria:   * User should find a 'Reorder' button next to each past order in the order history. * Clicking on the 'Reorder' button should add all items from the selected order to the cart. * User should have the option to review and edit the items in the cart before proceeding to checkout.   Alternative flow:   * If the user attempts to reorder an item that is no longer available, display a message indicating the item is out of stock and suggest similar alternatives.   Exceptional flow:   * If the item description fails to load due to a server error, display a generic message indicating a temporary issue and advise the user to try again later. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 30 | Tasks – 2 | | Priority: high |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO BE ABLE TO PROVIDE FEEDBACK ON MY ORDER EXPERIENCE  SO THAT THE SERVICE CAN BE IMPROVED. | | | |
| BV : 500 | | CP : 3 | |
| Acceptance criteria:   * User should receive a prompt to rate and review the order after delivery. * Rating should be on a scale of 1 to 5 stars, with an optional text box for comments. * Feedback should be visible to the restaurant and the delivery service for improvement purposes.   Alternative flow:   * If the user dismisses the feedback prompt initially, display a reminder notification after a certain period or on the next app launch.   Exceptional flow:   * If the feedback submission fails due to a server error, display an error message and inform the user that their feedback couldn't be submitted at the moment. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 31 | Tasks – 2 | | Priority: Medium |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO BE ABLE TO SCHEDULE ORDERS FOR FUTURE DELIVERY  SO THAT I CAN PLAN MEALS IN ADVANCE. | | | |
| BV : 100 | | CP : 3 | |
| Acceptance criteria:   * User should find an option for scheduled delivery during checkout. * User should be able to select date and time for future delivery. * Scheduled orders should be visible in the order history with the delivery date and time.   Alternative flow:   * If the user attempts to schedule a delivery during a time slot that is already full, display a message indicating the unavailability and suggest alternative time slots.   Exceptional flow:   * If the scheduled order fails to save due to a technical issue, display an error message and prompt the user to try again later. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 32 | Tasks – 2 | | Priority: Low |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO BE ABLE TO BROWSE RESTAURANT RATINGS AND REVIEWS  SO THAT I CAN CHOOSE THE BEST OPTIONS. | | | |
| BV : 10 | | CP : 2 | |
| Acceptance criteria:   * Each restaurant in the app should have a rating displayed prominently. * User should be able to access detailed reviews for each restaurant. * Reviews should include feedback on food quality, delivery time, and customer service.   Alternative flow:   * If the user taps on a restaurant with no reviews available, display a message indicating that reviews are pending and encourage the user to check back later.   Exceptional flow:   * If the review section fails to load due to a server error, display a message indicating a temporary issue and advise the user to try again later. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 33 | Tasks – 2 | | Priority: Low |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO RECEIVE NOTIFICATIONS FOR ORDER DELAYS  SO THAT I CAN ADJUST MY PLANS ACCORDINGLY. | | | |
| BV : 50 | | CP : 3 | |
| Acceptance criteria:   * User should receive a push notification if there is a delay in order preparation or delivery. * Notification should include the reason for the delay and an updated estimated time of arrival. * User should have the option to track the order in real-time for more updates.   Alternative flow:   * If the order delay is significant (e.g., more than 30 minutes), display a message offering compensation or alternatives such as order cancellation.   Exceptional flow:   * If the notification service experiences downtime, log the issue and display a generic message indicating that order updates are temporarily unavailable. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 34 | Tasks – 2 | | Priority: High |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO BE ABLE TO FILTER RESTAURANT OPTIONS BASED ON CUISINE TYPE AND DIETARY PREFERENCES  SO THAT I CAN FIND SUITABLE CHOICES EASILY. | | | |
| BV : 200 | | CP : 4 | |
| Acceptance criteria:   * User should find filter options on the restaurant selection page. * Filters should include cuisine types such as Italian, Chinese, and dietary preferences like vegetarian or vegan. * User should be able to apply multiple filters simultaneously for more refined results.   Alternative flow:   * If the user selects multiple filters that yield no results, display a message indicating the lack of available options and suggest broadening the search criteria.   Exceptional flow:   * If the filter options fail to load due to a server error, display a message indicating a temporary issue and advise the user to try again later. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 35 | Tasks – 3 | | Priority: Medium |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO BE ABLE TO VIEW ESTIMATED DELIVERY TIMES FOR DIFFERENT RESTAURANTS  SO THAT I CAN CHOOSE ACCORDINGLY. | | | |
| BV : 100 | | CP : 3 | |
| Acceptance criteria:   * Each restaurant in the app should display an estimated delivery time. * Delivery time should be calculated based on factors like distance and current order volume. * Estimated delivery time should be updated in real-time to reflect any changes.   Alternative flow:   * If the user searches for delivery times during peak hours, display a message indicating potential delays and suggest ordering in advance.   Exceptional flow:   * If the estimated delivery time fails to display due to a technical issue, display a message indicating a temporary issue and advise the user to try again later. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 36 | Tasks – 2 | | Priority: Medium |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO BE ABLE TO VIEW THE DELIVERY ROUTE OF MY ORDER IN REAL-TIME  SO THAT I CAN ANTICIPATE ITS ARRIVAL. | | | |
| BV : 100 | | CP : 2 | |
| Acceptance criteria:   * User should find a map view with the delivery route on the order tracking page. * Map should display the current location of the delivery person along with the route to the delivery address. * User should have the option to zoom in/out and pan the map for better visibility.   Alternative flow:   * If the delivery person deviates significantly from the route, display a warning message and provide an estimated time for route correction.   Exceptional flow:   * If the map view fails to load due to a technical error, display a message indicating a temporary issue and advise the user to try again later. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 37 | Tasks – 2 | | Priority: High |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO RECEIVE PERSONALIZED RECOMMENDATIONS BASED ON MY PAST ORDERS  SO THAT I CAN DISCOVER NEW FAVOURITES. | | | |
| BV : 200 | | CP : 3 | |
| Acceptance criteria:   * User should receive recommendations for similar items or restaurants based on past order history. * Recommendations should be displayed prominently on the home screen or in a dedicated section. * User should have the option to dismiss or hide recommendations if not interested.   Alternative flow:   * If the user dismisses the recommendations initially, display a reminder notification after a certain period or on the next app launch.   Exceptional flow:   * If the recommendation engine fails to generate personalized recommendations due to a technical error, display a generic message indicating a temporary issue and advise the user to check back later. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 38 | Tasks – 2 | | Priority: Low |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO BE ABLE TO SPLIT THE BILL WITH FRIENDS WHEN ORDERING TOGETHER  SO THAT EVERYONE CAN PAY THEIR SHARE. | | | |
| BV : 10 | | CP : 3 | |
| Acceptance criteria:   * User should find an option to split the bill during checkout. * User should be able to select friends from their contact list to split the bill. * System should calculate and display each person's share of the total bill amount.   Alternative flow:   * If the user attempts to split the bill with a friend who hasn't registered in the app, prompt the user to invite them to join or select another payment method.   Exceptional flow:   * If the bill splitting feature fails to load due to a technical error, display a message indicating a temporary issue and advise the user to try again later. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 39 | Tasks – 2 | | Priority: High |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO RECEIVE ALERTS FOR NEW OFFERS AND PROMOTIONS  SO THAT I CAN AVAIL DISCOUNTS ON MY ORDERS. | | | |
| BV : 200 | | CP : 3 | |
| Acceptance criteria:   * User should receive push notifications for new offers and promotions. * Notifications should include details such as discount percentage, validity period, and any promo codes required. * User should have the option to opt-out of promotional notifications if desired.   Alternative flow:   * If the user dismisses the promotional notification initially, display a reminder notification after a certain period or on the next app launch.   Exceptional flow:   * If the notification service experiences downtime, log the issue and display a generic message indicating that promotional updates are temporarily unavailable. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 40 | Tasks – 3 | | Priority: High |
| VALUE STATEMENT: AS A CUSTOMER  I WANT TO BE ABLE TO VIEW AND DOWNLOAD DIGITAL RECEIPTS FOR MY ORDERS  SO THAT I HAVE A RECORD FOR EXPENSE TRACKING. | | | |
| BV : 200 | | CP : 2 | |
| Acceptance criteria:   * User should find an option to view order history and receipts in the app settings. * Each receipt should include details such as order ID, items purchased, total amount, and payment method. * User should have the option to download receipts in PDF format for offline access.   Alternative flow:   * If the user attempts to download a receipt with no internet connection, display a message indicating the download will be available once the connection is restored.   Exceptional flow:   * If the receipt fails to generate or download due to a technical error, display an error message and prompt the user to try again later. | | | |

3. **What is Epic and 2 types of epic**

In Agile project management, an Epic is a large body of work that can be broken down into smaller tasks or stories. It typically encompasses a significant business objective and requires multiple sprints to complete. Epics are often used to organize and prioritize work, especially in larger projects, by providing a high-level view of the project's goals and requirements.

Feature: Meal Customization

Description: Allow users to customize their orders by selecting specific ingredients, portion sizes, or dietary preferences.

Tasks:

* Implement user interface for customizing meals.
* Integrate backend logic to handle customizations and update order details.
* Ensure compatibility with various menu items and restaurant offerings.

Journey: Seamless Ordering Experience

Description: Streamline the ordering process to provide a seamless experience from menu browsing to order completion.

Tasks:

* Optimize menu navigation for easy browsing and selection.
* Implement one-click reorder functionality for frequently ordered items.
* Enhance checkout process with saved payment methods and delivery addresses.

Business value:

* BV represents the overall benefit or value that an epic brings to the stakeholders or users of the product.
* It encompasses factors such as revenue generation, customer satisfaction, competitive advantage, and strategic alignment with business objectives.
* BV helps prioritize epics based on their impact on the business or user experience.
* In the context of a food delivery app:
* Epics with higher BV might include those that significantly enhance the user experience, increase customer engagement, or drive revenue growth.
* Examples of high BV epics: Implementing a feature for personalized recommendations to improve user engagement and increase order frequency.
* Enhancing the app's performance and reliability to ensure a seamless ordering experience, leading to higher customer satisfaction and retention.

Complexity Points:

* CP measures the level of effort, complexity, and technical challenge associated with implementing an epic.
* It helps teams estimate the resources, time, and expertise required to complete an epic.
* CP is relative and is used for comparative estimation among epics rather than absolute time estimation.
* In the context of a food delivery app: Epics with higher CP might involve complex technical implementations, architectural changes, or integration with external systems.
* Examples of high CP epics: Overhauling the app's architecture to support scalability and accommodate future growth, which may involve significant development effort and coordination.
* Integrating with third-party services or APIs to enhance functionality, such as integrating with a payment gateway, mapping service for real-time order tracking, or inventory management system for restaurant partners.

**4. Difference between BV and CP**

|  |  |
| --- | --- |
| Business Value | Complexity Points |
| BV represents the perceived worth or benefit of a user story, feature, or epic from the perspective of stakeholders or users of the product. | CP measures the level of effort, complexity, or technical challenge associated with implementing a user story, feature, or epic. |
| It focuses on the impact that a particular piece of work will have on the business, customers, or users. | It is a relative estimation technique used by Agile teams to compare the relative effort required to complete different work items. |
| BV is typically assessed based on factors such as revenue generation, customer satisfaction, market competitiveness, strategic alignment with business goals, and regulatory compliance. | CP takes into account factors such as technical dependencies, unfamiliar technology, integration requirements, extent of changes needed in the existing codebase, and overall difficulty of implementation. |
| BV helps prioritize work items by ensuring that efforts are directed towards tasks that deliver the most value to the business or end-users. | CP assists in estimating the resources, time, and expertise needed to complete a particular piece of work accurately. |
| It is often expressed in qualitative terms, such as high, medium, or low, or using specific metrics relevant to the project's objectives. | It is often expressed in numerical values, such as story points, which are assigned based on the team's consensus during estimation sessions like planning poker. |

**Techniques used in Business value:**

Moscow Prioritization:

This technique categorizes requirements or features into four priority levels: Must have, Should have, Could have, and won’t have. It helps stakeholders prioritize work based on its importance to achieving project goals.

Kano Model:

The Kano Model categorizes features based on their impact on customer satisfaction. It distinguishes between basic, performance and delighter features, helping teams prioritize work that provides the most significant value to users.

Impact Mapping:

Impact Mapping is a collaborative technique that aligns project objectives with user needs and business goals. It helps identify high-impact features or epics by mapping out how each contributes to desired outcomes.

Cost of Delay (CoD):

Cost of Delay calculates the financial impact of delaying the implementation of a feature or epic. It helps prioritize work based on the potential revenue gain, cost reduction, or opportunity capture associated with each item.

**Techniques used in Complexity points:**

Planning Poker:

Planning Poker is a consensus-based estimation technique where Agile teams assign story points to user stories, features, or epics. Team members discuss the complexity and effort required for each item and converges on a consensus estimate.

Relative Sizing:

Relative sizing compares the complexity of new work items to previously completed ones. Teams use reference stories of known complexity to estimate the effort required for new items, maintaining consistency in their estimation approach.

T-shirt Sizing:

T-shirt sizing categorizes work items into sizes such as Small, Medium, Large, and Extra-Large based on their complexity. It provides a quick and intuitive way to estimate relative effort without getting into detailed numerical estimations.

Bucket System:

The Bucket System groups work items into buckets representing different levels of complexity. For example, buckets may range from "Very Simple" to "Very Complex," allowing teams to quickly categorize items based on their perceived complexity.

**5. Explain about Sprint**

A sprint is a time-boxed iteration in Agile development during which a cross-functional team works to deliver a potentially shippable increment of product functionality. Sprints are a fundamental aspect of Agile methodology, particularly in Scrum, where they are the core framework for organizing work.

Stakeholders of sprint:

* Product owner
* Development team
* Scrum master
* Customers
* Delivery drivers
* Restaurant partners
* Shareholders
* Marketing teams
* Regulatory bodies

Time-Boxed Duration: Sprints have a fixed duration, typically ranging from one to four weeks. The length of the sprint is determined based on factors like the team's capacity, the nature of the project and organizational preferences. The most common sprint duration in Scrum is two weeks.

Sprint goal: A sprint goal is a concise statement that describes the objective or focus of a sprint in agile development. It provides clarity and direction to the development team regarding what they aim to achieve during the sprint. The sprint goal is set collaboratively by the Product Owner and the development team during the sprint planning meeting. It serves as a guiding principle for prioritizing work and making decisions throughout the sprint.

Examples of sprint goals for the online food delivery app project could include:

* "Improve Checkout Process and Payment Flow for Enhanced User Experience"
* "Implement Real-time Order Tracking and Notifications for Improved Customer Satisfaction"
* "Optimize Delivery Routing Algorithm to Enhance Efficiency and Reduce Delivery Times"
* "Enhance Menu Navigation and Search Functionality to Facilitate Faster Ordering".

Daily stand-up meetings:

Daily stand-up meetings, also known as daily scrum meetings, are a core practice within the Scrum framework and agile methodology. These meetings are held daily during a sprint and serve as a brief, focused gathering for the development team to synchronize their activities, discuss progress, and identify any impediments or challenges they may be facing.

Purpose:

* Synchronize Activities: Ensure that all team members are aware of what others are working on and any changes to the plan since the last meeting.
* Identify Impediments: Surface any obstacles or roadblocks that may be hindering progress so that they can be addressed promptly.
* Daily stand-up meetings are typically time-boxed to 15 minutes or less to keep them focused and efficient.
* Longer discussions or deeper problem-solving should be taken offline to avoid derailing the meeting.

Sprint Planning:

Sprint planning is a crucial ceremony in Agile and Scrum methodologies where the development team, Scrum Master, and Product Owner come together to plan the work to be done in the upcoming sprint. It's a collaborative meeting that sets the stage for the sprint and ensures that everyone understands what needs to be accomplished and how it will be achieved.

Sprint review and retrospective:

Sprint Review and Sprint Retrospective are two essential ceremonies within the Scrum framework that occur at the end of each sprint. They provide opportunities for the Scrum Team to reflect on their work, gather feedback, and continuously improve their processes.

The purpose of sprint review is also known as the Sprint Demo, is a collaborative meeting where the Scrum Team presents the work completed during the sprint to stakeholders and gathers feedback, the participants of sprint review will be scrum team and stakeholders.

Output:

* Updated Product Backlog reflecting any changes or new items identified during the review.
* Action items or follow-up tasks identified based on stakeholder feedback.

The Sprint Retrospective is a reflective meeting where the Scrum Team reflects on the previous sprint's processes, identifies areas for improvement, and plans actionable steps for implementing those improvements, the participants are scrum team.

Output:

* Action items or improvement backlog containing specific tasks to be addressed in the next sprint.
* Reflections or insights captured from the retrospective meeting for future reference.

Scrum Master: The Scrum Master plays a pivotal role in supporting the Scrum Team and ensuring the successful execution of the sprint. They act as a facilitator, coach, mentor, and servant-leader, enabling the team to maximize its potential and deliver value consistently.

Development and testing:

* Development and testing activities are closely intertwined, with developers and testers collaborating throughout the sprint to ensure the quality and functionality of the product increment.
* Continuous communication, feedback exchange, and shared ownership of quality are essential for effective collaboration between development and testing teams.
* Both development and testing activities contribute to the successful completion of user stories and the delivery of a potentially shippable increment of product functionality by the end of the sprint.

Incremental delivery:

Incremental delivery in the context of a sprint refers to the iterative and incremental approach to delivering value to stakeholders throughout the duration of the sprint. Instead of waiting until the end of the sprint to deliver a complete set of features, the development team aims to deliver a usable and potentially shippable increment of the product incrementally, with each increment building upon the previous one.

Backlog refinement:

Backlog refinement, also known as backlog grooming or backlog refinement, is a recurring activity in Scrum that occurs throughout the sprint to ensure that the product backlog is well-prepared and ready for sprint planning. It involves reviewing, prioritizing, and refining items in the product backlog to ensure they are sufficiently detailed, understood, and ready for implementation by the development team.

6. Explain about product backlog and sprint backlog:

Product backlog:

Definition: The Product Backlog is a prioritized list of all the features, enhancements, bug fixes, and other work items that need to be implemented in the product. It serves as the single source of truth for all the requirements of the product.

Ownership: The Product Backlog is owned and maintained by the Product Owner, who is responsible for ensuring that it reflects the current and evolving needs of the stakeholders.

Contents: Product Backlog items (PBIs) are often expressed as user stories, describing the desired functionality from an end-user perspective. PBIs may also include technical tasks, spikes, research items, or any other work required to deliver value to the product.

Prioritization: Items in the Product Backlog are prioritized based on their relative value to the product and its stakeholders. The Product Owner is responsible for continuously prioritizing and re-prioritizing backlog items based on changing requirements, market conditions, and stakeholder feedback.

Details and Estimates: Product Backlog items are typically high-level and may not contain detailed specifications or acceptance criteria. Items are often estimated in terms of their relative effort or complexity using techniques such as story points or t-shirt sizes.

Sprint backlog:

Definition: The Sprint Backlog is a subset of the Product Backlog that contains the list of tasks and user stories that the development team commits to completing during a sprint. It represents the work selected by the team for implementation in the current sprint.

Ownership: The Sprint Backlog is owned and managed by the development team, who are responsible for organizing and executing the work items within the sprint.

Scope: The Sprint Backlog is scoped to the duration of a single sprint, typically lasting one to four weeks. It represents the team's plan for achieving the sprint goal and delivering a potentially shippable increment of the product by the end of the sprint.

Dynamic Nature: The Sprint Backlog is dynamic and may evolve throughout the sprint as the team gains new insights, encounters challenges or adjusts their plan in response to changing circumstances. It is not set in stone and can be adjusted based on the team's progress and feedback received during the sprint.

Task Decomposition: The Sprint Backlog contains detailed tasks or sub-tasks that break down the selected user stories into actionable items. Tasks are estimated, assigned to team members, and tracked throughout the sprint to ensure progress towards the sprint goal.

Relationship between Product Backlog and Sprint Backlog:

The Product Backlog provides the long-term view of all the work that needs to be done to fulfil the product vision, while the Sprint Backlog focuses on the immediate work to be completed within a single sprint.

Items from the Product Backlog are selected and pulled into the Sprint Backlog during sprint planning based on their priority and the team's capacity.

The Sprint Backlog is a reflection of the team's commitment to delivering specific items from the Product Backlog within the sprint timeframe.

Both backlogs are living documents that evolve over time as new requirements emerge, priorities change, and work is completed. They serve as essential tools for managing and delivering value in agile development.

**7. What is Impediments log and write 2 impediments**

Impediments log:

An Impediment Log, also known as an Impediment List or Issue Log, is a document or tool used in Agile and Scrum methodologies to track and manage obstacles, challenges, or blockers that hinder the progress of the development team. It serves as a centralized repository for identifying, documenting, and resolving impediments that may arise during the course of a project or sprint.

|  |  |
| --- | --- |
| Log ID | 1 |
| Description | Payment Gateway Integration Failure: The team encounters difficulties integrating the payment gateway with the app, resulting in issues with processing transactions. |
| Impact | Customers are unable to complete their orders, leading to a loss of revenue and potential dissatisfaction. |
| Priority | High |
| Assigned to | Development Team |
| Status | In progress |
| Action taken | The development team immediately reaches out to the payment gateway provider's support team to troubleshoot the integration issues. |
| Resolution | Engage with the payment gateway provider's support team to troubleshoot integration issues, implement necessary fixes, and conduct thorough testing to ensure smooth transaction processing. |

|  |  |
| --- | --- |
| Log ID | 2 |
| Description | Unavailability of Delivery Partners: Due to unforeseen circumstances, there is a shortage of available delivery partners, leading to delays in order deliveries. |
| Impact | Customers experience longer-than-expected wait times for their orders, resulting in dissatisfaction and potential loss of business. |
| Priority | Medium |
| Assigned to | Scrum master |
| Status | Identified |
| Action taken | Evaluate alternative solutions to increase delivery partner availability, such as hiring additional drivers, partnering with third-party delivery services, or adjusting delivery schedules. |
| Resolution | Investigate alternative options for increasing delivery partner availability, such as hiring additional drivers or partnering with third-party delivery services. Communicate transparently with customers about potential delays and explore incentives to encourage faster delivery times. |

**8. Explain the Velocity of the team:**

Velocity in Agile and Scrum refers to the amount of work a development team can complete within a single sprint, typically measured in terms of story points or other relative units of effort. It provides a means for the team to forecast how much work they can accomplish in future sprints and track their progress over time.

Calculation:

Velocity is typically calculated at the end of each sprint by summing up the story points of all completed user stories and tasks.

It represents the average amount of work completed by the team in previous sprints and is used as a baseline for planning future sprints.

Velocity is a valuable metric in Agile and Scrum that helps teams forecast, plan, and track their progress effectively, ultimately enabling them to deliver value to customers more efficiently and predictably.

**9. Draw Sprint burn down chart and product burn down chart**

Sprint burn down chart



Product burn down chart

**10. Explain about product grooming:**

Product grooming, also known as backlog grooming or refinement, is an essential on-going activity in Agile and Scrum methodologies. It involves reviewing, refining, and prioritizing items in the product backlog to ensure they are well-defined, understood, and ready for implementation by the development team.

* The primary goal of product grooming is to clarify and elaborate on product backlog items to ensure a shared understanding between the product owner, development team, and stakeholders.
* Grooming helps prioritize backlog items based on their value, dependencies, and urgency, ensuring that the most important work is addressed first.
* Product grooming is an on-going activity that occurs throughout the project or product lifecycle, rather than being confined to a single meeting or event.
* The frequency and timing of grooming sessions may vary depending on the size and complexity of the product backlog, as well as the needs of the team and stakeholders.
* Ideally, grooming sessions are held regularly to ensure that the product backlog remains up-to-date, prioritized, and ready for sprint planning.

Activities in Product Grooming:

Review Backlog Items:

The product owner presents backlog items (user stories, bugs, enhancements) to the development team during grooming sessions. The team asks questions, seeks clarification, and discusses the requirements to ensure a shared understanding.

Refine and Break Down:

Backlog items are refined and broken down into smaller, more manageable pieces, if necessary, to make them actionable for implementation within a single sprint. This may involve splitting large user stories, defining sub-tasks or identifying dependencies.

Define Acceptance Criteria:

Acceptance criteria are defined for each backlog item to specify the conditions that must be met for the item to be considered complete. Acceptances criteria help ensure that the team and stakeholders have a clear understanding of what constitutes a successful outcome.

Estimate Effort:

The development team estimates the effort or complexity of backlog items using techniques such as story points, t-shirt sizing, or ideal days. Effort estimation helps inform prioritization and capacity planning for future sprints.

Update Priorities:

Based on the discussion and refinement during grooming sessions, the product owner updates the priorities of backlog items to reflect changing business needs or stakeholder feedback.

**11. Role of scrum master and product owner**

|  |  |  |
| --- | --- | --- |
| Criteria | Product owner | Scrum master |
| Nature of work | Collaborates with all the stakeholders and brings the vision of a product into the product backlog | Acts as a team coach and is responsible for maintaining the quality of the product. |
| Responsibilities | Responsible for completing the project on a time. Acts as an intermediary between the development team and the customers. | Ensures the scrum framework is followed and helps the development team create a quality product. |
| Accountability | Responsible for the project backlog, the timely completion of the product and for providing updates to the clients and stakeholders. | Accountable for the quality of the entire project and for giving updates to management about the completion of the product. |
| Reporting | Reports to top management and the clients. | Reports to top management about the efficiency of the team and the quality of the product. |
| Qualities | Communication and leadership skills, creativity, critical thinking, and a sharp mind are key assets for any product owner. | Thorough knowledge of scrum theory and practices. Being able to lead a team but without a sense of authority. |
| Communication | Product Owners must communicate with stakeholders, development teams, and other project members to gather requirements, provide updates, and make decisions about the product. | Effective communication is essential for a Scrum Master to facilitate collaboration, alignment, and understanding among team members and stakeholders. |
| Leadership skills | Product Owners also require leadership skills to drive the vision, strategy, and direction of the product. | A Scrum Master needs leadership skills to guide the development team and stakeholders through the Scrum process. |
| Problem solving skills | Product Owners encounter various challenges related to product requirements, stakeholder expectations, and market dynamics. | Scrum Masters need strong problem-solving skills to identify impediments, address conflicts and find solutions to challenges that arise during the project. |

**12. Explain all meetings conducted in Scrum project:**

Sprint Planning: Sprint planning is a collaborative meeting that kicks off each sprint in Scrum. Its primary purpose is for the Scrum Team (including the Product Owner, Scrum Master, and Development Team) to plan and agree on the work to be accomplished during the upcoming sprint.

Purpose: To plan the work to be completed during the upcoming sprint.

Participants: Scrum Team (Development Team, Scrum Master, Product Owner).

Agenda:

* Review and select user stories from the product backlog for the sprint.
* Define the sprint goal.
* Break down selected user stories into tasks.
* Estimate effort for each task.

Outcome: Sprint backlog, sprint goal, and commitment to deliver a potentially shippable product increment.

Daily stand-up meetings (Daily scrum): The daily stand-up meeting, also known as the daily scrum, is a short, time-boxed meeting that occurs every day during the sprint. Its purpose is to provide a brief opportunity for the Development Team to synchronize their activities, plan their day and identify any impediments that may hinder their progress.

Purpose: To provide a daily synchronization and alignment opportunity for the development team.

Participants: Development Team, Scrum Master (optional), Product Owner (optional).

Agenda:

* Each team member shares
* What they did yesterday.
* What they plan to do today.
* Any impediments or blockers they are facing.

Outcome: Increased visibility, alignment, and identification of impediments.

Duration: Typically 15 minutes or less.

Sprint review: The sprint review, also known as the sprint demo, is held at the end of each sprint to showcase the completed work to stakeholders. Its purpose is to gather feedback, review the product increment and adapt the product backlog based on stakeholder input.

Purpose: To review and demonstrate the completed work from the sprint to stakeholders.

Participants: Scrum Team, stakeholders (including customers, users, and sponsors).

Agenda:

* Demo of completed user stories and features.
* Review of the sprint goal and product increment.
* Feedback and discussion with stakeholders.

Outcome: Feedback from stakeholders, potential updates to the product backlog and confirmation of completed work.

Sprint retrospective: The sprint retrospective is a meeting held at the end of each sprint to reflect on the team's performance and identify opportunities for improvement. It provides a structured opportunity for the Scrum Team to inspect and adapt their processes and practices.

Purpose: To reflect on the previous sprint and identify areas for improvement.

Participants: Scrum Team (Development Team, Scrum Master, Product Owner).

Agenda:

* Reflect on what went well and what could be improved during the sprint.
* Identify action items to address improvement opportunities.
* Discuss any changes to team processes or practices.

Outcome: Action items for process improvement, adjustments to team practices, and increased team effectiveness.

Additional meetings:

Backlog refinement (grooming): Backlog refinement, also known as backlog grooming or backlog refinement is a recurring activity in the Scrum framework that involves reviewing, refining, and prioritizing items in the product backlog. The primary purpose of backlog refinement is to ensure that the product backlog is well-prepared, prioritized and ready for implementation by the development team. To refine and prioritize items in the product backlog.

Ad-hoc meetings: Ad-hoc meetings are unplanned or spontaneous meetings that are called as the need arises to address specific issues, challenges or urgent matters that cannot wait until the next scheduled meeting. To address specific issues, dependencies, or urgent matters that arise during the project.

Release planning: Release planning is the process of defining and organizing the features, enhancements or updates that will be included in a product release. It involves determining the scope, timeline, and priorities for the release based on business objectives, stakeholder requirements, and development capacity. To plan and prioritize features for upcoming releases based on the product roadmap and stakeholder input.

**13. Sprint Size and Scrum Size:**

Sprint Size: The sprint size refers to the length or duration of a sprint in scrum. A sprint is a time boxed period during which the development team works to deliver a potentially shippable product increment. The sprint size is determined during the project planning phase and typically ranges from one to four weeks. The most common sprint duration is two weeks but it can vary depending on the project’s needs, complexity, and team dynamics.

Importance: The size of a sprint impacts various aspects of the project, including planning, delivery cadence and adaptability. Shorter sprints may provide faster feedback and allow for more frequent course corrections, while longer sprints may offer more time for development and testing.

Scrum size: The scrum team size refers to the number of individuals who collaborate together to deliver the product increment in scrum. The scrum team is self-organizing and cross-functional, typically consisting of a product owner, a scrum master, and the development team. The recommended scrum team size is small, ideally between five to nine members, to enable effective communication, collaboration, and flexibility, however, there are no strict rules regarding team size, and it may vary depending on the specific project requirements and organization.

Importance: The size of the Scrum implementation can vary widely, from small teams working on single projects to large enterprises with multiple teams and complex product portfolios. The larger the Scrum implementation, the more coordination, collaboration, and alignment are required among teams and stakeholders.

**14. Explain DOR and DOD:**

Definition of Ready(DOR): The Definition of Ready (DoR) is a set of criteria or conditions that a product backlog item must meet before it can be considered ready to be worked on by the development team during sprint planning in Agile and Scrum projects. The purpose of the DoR is to ensure that backlog items are well-defined, understood, and prepared for implementation, thereby minimizing misunderstandings, delays and rework during the sprint.

Clear Description: Each backlog item should have a clear and concise description that outlines its purpose, functionality and user value. This description should be understood by all members of the Scrum Team, including the product owner, development team and stakeholders.

Acceptance Criteria: The backlog item should include specific, measurable criteria that define when the item is considered complete and meets the expectations of the stakeholders. Acceptance criteria help ensure that the development team understands what needs to be delivered and can verify the item's functionality.

Dependencies Identified: Any dependencies or prerequisites required for the backlog item to be implemented successfully should be identified and addressed before the item is considered ready. This may include dependencies on other backlog items, external systems, or resources.

Estimation Feasibility: The backlog item should be of a size and complexity that makes it feasible to estimate the effort required for implementation. Items that are too large or unclear may need to be further refined or broken down into smaller, more manageable pieces.

Ready for Sprint Planning: Once a backlog item meets the criteria outlined in the Definition of Ready, it is considered ready to be brought into sprint planning for consideration by the development team. At this point, the team can assess the item's priority, estimate its effort, and commit to delivering it within the upcoming sprint.

Definition of Done (DOD): The Definition of Done (DOD) is a set of criteria or conditions that a product increment must meet before it can be considered complete and potentially shippable in Agile and Scrum projects. It serves as a quality standard and ensures that the development team and stakeholders have a shared understanding of what constitutes a finished product increment.

Quality Assurance: The Definition of Done outlines the minimum standards of quality that must be met for a product increment to be considered complete. It includes criteria related to functionality, performance, usability, security, and other quality attributes.

Clear Criteria: The DOD provides clear and specific criteria that define when a product increment is considered done. This helps prevent misunderstandings and ensures that all stakeholders have a shared understanding of the completion criteria.

Incremental Delivery: The Definition of Done supports the principle of incremental delivery by specifying that each product increment must be potentially shippable. This means that the increment should be in a usable and releasable state, even if it's not actually deployed to production at the end of the sprint.

Continuous Improvement: The DOD is not static; it should evolve over time based on feedback, lessons learned, and changes in project requirements. Regularly reviewing and refining the DOD helps the team identify areas for improvement and raise the quality bar over time.

Cross-functional Collaboration: Developing and refining the Definition of Done is a collaborative process that involves the entire Scrum Team, including the product owner, development team and Scrum Master. It encourages cross-functional collaboration and ensures that all team members have input into the definition of done.

Quality Assurance: By adhering to the Definition of Done, the development team can ensure that each product increment meets the required quality standards and delivers value to stakeholders. It provides confidence that the product increment is complete, functional, and ready for review by stakeholders.

**15. Explain Prioritization techniques and MVP:**

MOSCOW method:

Must have: Items categorized as "Must have" are critical to the project's success and must be delivered within a specified timeframe. These items represent the core functionalities or features that are essential for the product to be considered viable or complete. Failure to deliver must-have items may jeopardize the project's goals or objectives.

Should have: Items categorized as "Should have" are important but not critical to the project's success. These items are desirable and should be included in the product if resources and time permit. They typically represent features or functionalities that add significant value to the product but are not essential for its basic functionality.

Could have: Items categorized as "Could have" are nice-to-have or optional features that would enhance the product but are not essential for its core functionality. These items are typically considered lower priority compared to must-have and should-have items and may be included in future releases if time and resources allow.

Won't have: Items categorized as "Won't have" are considered out of scope for the current project or release. These items may represent features or functionalities that are not feasible to implement within the project's constraints or are deemed unnecessary or low value. They are explicitly excluded from the project's scope and will not be considered for implementation.

The Moscow method helps product owners and stakeholders prioritize backlog items based on their criticality, enabling the development team to focus their efforts on delivering the most important and valuable features first. It provides clarity and alignment on the project's priorities and ensures that resources are allocated effectively to meet the project's goals and objectives.

Minimum Viable product (MVP): A Minimum Viable Product (MVP) is a version of a product that contains only the core features or functionalities required to meet the needs of early adopters or users. The primary goal of an MVP is to quickly validate assumptions, gather feedback and learn from real-world usage with minimal development effort. Here are some key characteristics of an MVP:

Core Features: An MVP includes only the essential features or functionalities that address the core needs or pain points of the target users. It avoids adding unnecessary or complex features that don't contribute to the primary value proposition.

Quick Development: Developing an MVP focuses on rapid iteration and quick delivery to the market. It emphasizes speed and efficiency in building and releasing the product to gather feedback and validate assumptions as early as possible.

Feedback-driven: An MVP is designed to gather feedback from early adopters or users to inform future development decisions. It allows product teams to iterate and evolve the product based on real-world usage and user feedback.

Risk Reduction: By releasing an MVP, product teams can mitigate the risk of investing resources into building a full-featured product that may not meet the needs or expectations of users. It allows teams to test hypotheses and validate product-market fit before scaling up.

Iterative Development: An MVP is just the starting point; it serves as a foundation for iterative development and continuous improvement. Product teams can use feedback and insights gathered from the MVP to prioritize and plan future iterations of the product.

In summary, prioritization techniques help Agile teams determine the order in which backlog items should be addressed, while an MVP enables teams to quickly validate assumptions and gather feedback by delivering a minimal version of the product to the market. Both concepts are essential components of agile product development and help teams focus on delivering value to users efficiently.

**16. Difference between the Business analyst and product owner**

|  |  |  |
| --- | --- | --- |
| Criteria | Business analyst | Product owner |
| Role | BAs focus on analysing business processes, identifying needs, and defining requirements to solve business problems or improve operations. They act as intermediaries between stakeholders and the development team, ensuring that project requirements are clearly defined and understood. | POs focus on managing the product backlog, setting priorities, and defining the vision and roadmap for the product. They are responsible for maximizing the value of the product and ensuring that it meets the needs of customers and stakeholders. |
| Responsibilities | BAs are responsible for gathering and documenting business requirements, analysing workflows, conducting stakeholder interviews, and defining acceptance criteria. They may also assist in testing and validation activities. | POs are responsible for defining the product vision, maintaining and prioritizing the product backlog, and making decisions about what features and enhancements should be included in each release. They work closely with stakeholders to understand their needs and ensure that the product delivers value. |
| Stakeholder Interaction | BAs primarily interact with stakeholders to gather requirements, elicit feedback, and ensure that project deliverables meet business objectives. They act as liaisons between business users, subject matter experts, and the development team. | POs interact with a broader range of stakeholders, including customers, users, executives, and development team members. They represent the interests of stakeholders and prioritize backlog items based on their input and feedback. |
| Decision making authority | BAs typically do not have decision-making authority over project scope, priorities, or budget. They provide recommendations and analysis to stakeholders and project managers but do not have final say in project decisions. | POs have decision-making authority over the product backlog, including setting priorities, defining release scope, and accepting or rejecting deliverables. They are empowered to make decisions that maximize the value of the product. |
| Ownership for deliverables | BAs are responsible for delivering artifacts such as requirements documents, process flows, use cases, and user stories. These artifacts serve as inputs to the development process and guide the implementation of solutions. | POs own the product backlog and are accountable for the success of the product. They are responsible for ensuring that the product meets customer needs, delivers value, and achieves business objectives. |
| Technical knowledge | BAs may possess technical knowledge related to the industry or domain in which they work, but their primary expertise lies in business analysis techniques, requirements elicitation, and process modelling. | While technical knowledge is beneficial for POs, especially in understanding the capabilities and constraints of the development team, it is not always required. POs focus more on market research, customer feedback, and product strategy. |
| Scope | BAs typically focus on individual projects or initiatives within an organization. Their scope of influence is limited to the specific requirements and deliverables associated with each project. | POs have a broader scope of influence that extends across the entire product lifecycle. They are responsible for defining the overall direction and strategy for the product and ensuring its alignment with business goals. |
| Collaboration with development team | BAs collaborate with the development team to clarify requirements, answer questions, and ensure that deliverables meet business needs. They may work closely with developers, testers, and other team members throughout the project lifecycle. | POs actively collaborate with the development team to prioritize backlog items, provide clarification on requirements, and make decisions about product features. They serve as the primary point of contact for the team regarding product-related matters. |

**17. Sample resume of 3 years’ experience product owner CV**

Karan lingadal

Hsr layout

Bangalore, Karnataka- 560102

E- karanlingadal@gmail.com

Ph. no: +91-9380202329

Objective: To Dedicated and results-oriented Product Owner with 3 years of experience in Agile software development seeking a challenging role to leverage my skills in defining product vision, managing product backlog, and driving product success through effective stakeholder collaboration.

Professional experience:

Amazon company – Product owner – Bengaluru (Mar 10 2022)

* Defined and communicated the product vision, strategy, and roadmap to stakeholders and development teams.
* Managed and prioritized the product backlog, ensuring alignment with business goals and customer needs.
* Collaborated with stakeholders to gather requirements, elicit feedback, and validate product features.
* Worked closely with cross-functional teams to deliver high-quality product increments within sprint cycles.
* Conducted user acceptance testing (UAT) and gathered feedback to iterate and improve product functionality.
* Facilitated sprint planning, backlog grooming, and sprint review meetings to drive project success.

Reliance Digital – Associate product owner- Bengaluru (Jan 2 2021 – Mar 2 2022)

* Assisted in defining the product vision, strategy, and roadmap, aligning with business objectives and market trends.
* Supported the Product Owner in managing and prioritizing the product backlog, ensuring delivery of key features and enhancements.
* Coordinated with development teams to ensure a clear understanding of requirements and acceptance criteria.
* Conducted market research and competitor analysis to identify opportunities for product improvement and differentiation.
* Participated in sprint planning, backlog refinement, and sprint review meetings to ensure successful product delivery.

Educational qualification:

* Master of Business administration (Finance and marketing) – Karnataka university – 72% (Dec 2020)
* Bachelor of Business administration (Marketing) – Karnataka university- 70% - (Sep 2018)

Certifications:

* Certified Scrum Product Owner (CSPO)
* Agile Certified Practitioner (ACP)

Skills:

* Product Management
* Agile Methodologies (Scrum, Kanban)
* Product Road mapping
* Backlog Management
* Stakeholder Collaboration
* Requirements Gathering
* User Story Writing
* Sprint Planning
* Market Analysis
* User Acceptance Testing (UAT)

Languages:

English, Kannada and Hindi