Question 1 – write Agile Manifesto – 8 Marks

Four main Values

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. 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

 **Sprint-1 User Stories**

|  |
| --- |
| User Story No: 1 Tasks: 2As a Delivery Boy, I want to register in Scrum Foods so that I can deliver orders. |
| * Priority: Highest
 |
| * BV: 500
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Registration screen with text boxes for User Name, Password, Nation ID, Mobile No, Email, Address, Phone Number
 |
| * + Click on Register Button
 |
| * + Send successful notification to the user
 |

|  |
| --- |
| User Story No: 2 Tasks: 2As a Restaurant Owner, I want to view orders so that I can see the list of orders. |
| * Priority: Highest
 |
| * BV: 500
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + View Order
 |
| * + Display list of orders in tabular form
 |

|  |
| --- |
| User Story No: 3 Tasks: 2As a Customer, I want to add the address so that I can get the order delivered to my address. |
| * Priority: Highest
 |
| * BV: 500
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Text box to enter address
 |
| * Business Rules:
 |
| * + Within the radius of 5 km
 |

|  |
| --- |
| User Story No: 4 Tasks: 2As a Customer, I want to select the payment mode so that I can make the payment of my choice. |
| * Priority: Highest
 |
| * BV: 500
 |
| * CP: 03
 |
| * Acceptance Criteria:
 |
| * + Display payment modes
 |
| * + Radio buttons to select payment modes
 |
| * + Payments button
 |
| * Business Rule:
 |
| * + Can select only one payment mode
 |

|  |
| --- |
| User Story No: 5 Tasks: 1As an Admin, I want to view the restaurants so that I can approve their registration. |
| * Priority: Highest
 |
| * BV: 500
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + List of restaurants
 |
| * + Select restaurant
 |
| * + Verify restaurant details
 |
| * + Approve button
 |
| * + Reject button
 |
| * + Notification to the restaurant
 |

|  |
| --- |
| User Story No: 6 Tasks: 1As a Customer, I want to view the price so that I can order food. |
| * Priority: Low
 |
| * BV: 50
 |
| * CP: 01
 |
| * Acceptance Criteria:
 |
| * + Display price in the list of menu items
 |

|  |
| --- |
| User Story No: 7 Tasks: 2As a Customer, I want to view the contact number of the delivery boy so that I can contact them for order status. |
| * Priority: Low
 |
| * BV: 50
 |
| * CP: 01
 |
| * Acceptance Criteria:
 |
| * + Display delivery boy's mobile number
 |
| * + Display delivery boy's name in the tracking field
 |
| * + Display delivery boy's picture
 |

|  |
| --- |
| User Story No: 8 Tasks: 2 As a Restaurant Owner, I want to provide time slots so that customers can check opening and closing hours. |
| * Priority: Medium
 |
| * BV: 100
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Click on the restaurant dashboard
 |
| * + Add from time to time
 |
| * + Click on submit
 |
| * + Display updated successfully
 |

|  |
| --- |
| User Story No: 9 Tasks: 2As a Business Owner, I want to view the restaurant revenue report so that I can track restaurant revenue. |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 03
 |
| * Acceptance Criteria:
 |
| * + Select Reports
 |
| * + Select Revenue Reports
 |
| * + Select to and from date
 |
| * + Select Region (can select all)
 |
| * + Generate Report
 |
| * + Download Report in Excel
 |

|  |
| --- |
| User Story No: 10 Tasks: 3As a Regional Admin, I want to manage regional restaurants so that I can track their performance. |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 03
 |
| * Acceptance Criteria:
 |
| * + Click on Performance of Restaurants
 |
| * + Select from date to date
 |
| * + Click on Generate Report (includes Restaurant ID, Name, Revenue)
 |
| * + Click on Download Report (should be in Excel)
 |

|  |
| --- |
| User Story No: 11 Tasks: 2As an Admin, I want to see the regional revenue reports so that I can track performance by region. |
| * Priority: Medium
 |
| * BV: 100
 |
| * CP: 03
 |
| * Acceptance Criteria:
 |
| * + Select regional dropdown
 |
| * + View performance of each restaurant in the region (tabular format including Restaurant Name, Revenue)
 |
| * + Download report in Excel or PDF
 |

|  |
| --- |
| User Story No: 12 Tasks: 2As a Customer, I want to chat with the Regional Admin so that I can request a refund. |
| * + Priority: High
 |
| * + BV: 200
 |
| * + CP: 02
 |
| * + Acceptance Criteria:
 |
| * + - All fields mandatory
 |
| * + - Text box fields
 |
| * + - Display Order ID
 |
| * + - Text box for description
 |
| * + - Submit button
 |
| * + - Generate issue ID
 |
| * + - Display successful notification
 |

|  |
| --- |
| User Story No: 13 Tasks: 2AS A HUNGRY USER I WANT TO BROWSE NEARBY RESTAURANTS SO THAT I CAN ORDER THE FOOD |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Each restaurant entry displays its name, cuisine type, and rating
 |
| * + This list can be sorted by distance or rating
 |

|  |
| --- |
| User Story No: 14 Tasks: 2AS A CUSTOMER I WANT TO BROWSE DIFFERENT RESTAURANTS AND MENUS SO THAT I CAN FIND A PLACE TO ORDER FOOD. |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + The menu includes dishes , prices and descriptions
 |
| * + Show the restaurant is open or closed
 |

|  |
| --- |
| User Story No: 15 Tasks: 1AS A CUSTOMER I WANT TO BROWSE FOR SPECIFIC DISHES AND CUISINES SO THAT I CAN FIND A PLACE TO ORDER FOOD |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + App displays relevant restaurant and dishes matching the query
 |

|  |
| --- |
| User Story No: 16 Tasks: 1AS A CUSTOMER I WANT TO FILTER RESTAURANTS SO THAT I CAN FIND A PLACE TO ORDER FOOD |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Filter restaurants by cuisine type and dietary options(vegan, veg,nonveg,egg)
 |

|  |
| --- |
| User Story No: 17 Task:2AS A CUSTOMER I WANT TO TRACK MY ORDER SO THAT I KNOW THE TIME OF DELIVERY |
| * Priority: Medium
 |
| * BV: 200
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + App shows real time update on the order status
 |
| * + Display estimated delivery time
 |

|  |
| --- |
| User Story No: 18 Task:1AS A USER I WANT TO RATE AND REVIEW RESTAURANTS SO THAT I CAN RATE AND REVIEW THE RESTAURANTS I HAVE VISITED. |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Can see reviews from other users to help me make dining decisions
 |

|  |
| --- |
| User Story No: 19 Tasl:1AS A USER I WANT TO VIEW PAST ORDER HISTORY SO THAT I CAN ORDER AGAIN |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Can see the details such as order items, total cost and order date
 |

|  |
| --- |
| User Story No: 20 Task:3AS A USER I WANT TO RECEIVE NOTIFICATIONS SO THAT I CAN RECEIVE UPDATES. |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Notifications for order confirmation
 |
| * + Notification for dispatch
 |
| * + Notification for delivery
 |

|  |
| --- |
| User Story No: 21 Task:1AS A CUSTOMER I WANT TO CONTACT CUSTOMER SUPPORT SO THAT I CAN SUBMIT QUERIES OR ISSUES. |
| * Priority: Medium
 |
| * BV: 200
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Customer support section with contact information
 |

|  |
| --- |
|  User Story No: 22 Task:2AS A RESTAURANT OWNER I WANT TO RECEIVE AND MANAGE ORDERS SO THAT I CAN UPDATE ORDER STATUS |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Manage order status
 |
| * + Notify restaurants about incoming orders
 |

|  |
| --- |
|  User Story No: 23 Task:2AS A RESTAURANT OWNER I WANT TO ACCESS TO CUSTOMER REVIEWS SO THAT I CAN VIEW AND RESPOND TO CUSTOMER REVIEWS |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Owners can address feedback
 |
| * + Owners can improve their services
 |

|  |
| --- |
| User Story No: 24 Task:1 AS A CUSTOMER I WANT TO APPLY PROMOCODES AND DISCOUNTS SO THAT I CAN ORDER AT LOWER PRICE |
| * Priority: Medium
 |
| * BV: 100
 |
| * CP: 04
 |
| * Acceptance Criteria:
 |
| * + Active Promocodes
 |

|  |
| --- |
| User Story No: 25 Task:1AS A USER I WANT TO SAVE FAVOURITE RESTAURANTS AND DISHES SO THAT I CAN ORDER FROM MY FAVOURITES |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Access my list of favourites easily for future orders
 |

|  |
| --- |
| User Story No: 26 Task:7AS A DELIVERY BOY I WANT TO VIEW THE ORDERS SO THAT I ACCEPT THE ORDER  |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 04
 |
| * Acceptance Criteria:
 |
| * + Order visibility
 |
| * + Real-time updates
 |
| * + Order details
 |
| * + Order filtering and sorting
 |
| * + Order map view
 |
| * + Order navigation
 |
| * + Order completion and confirmation
 |

|  |
| --- |
| User Story No: 27 Tasks: 5 AS A DELIVERY BOY I WANT TO LOGIN SO THAT I CAN ACCEPT THE ORDER |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 04
 |
| * Acceptance Criteria:
 |
| * + User Authentication
 |
| * + Error Handling
 |
| * + Password security
 |
| * + Multi-factor Authentication
 |
| * + Compatibility and Usability
 |

|  |
| --- |
| User Story No: 28 Tasks: 5AS A DELIVERY BOY I WANT TO VIEW FEEDBACK SO THAT I CAN KNOW THE CUSTOMERS FEEDBACK. |
| * Priority: Medium
 |
| * BV: 200
 |
| * CP: 04
 |
| * Acceptance Criteria:
 |
| * + Access to feed back system
 |
| * + Feedback Visibility
 |
| * + Feedback sorting and filtering
 |
| * + Response Mechanism
 |
| * + User Support
 |

|  |
| --- |
| User Story No: 29 Tasks: 5 AS A ADMIN I WANT TO VIEW FEEDBACK SO THAT I CAN KNOW THE CUSTOMERS FEEDBACK |
| * Priority: Medium
 |
| * BV: 200
 |
| * CP: 04
 |
| * Acceptance Criteria:
 |
| * + Access to feedback system
 |
| * + Feedback Visibility
 |
| * + Feedback sorting and filtering
 |
| * + Response Mechanism
 |
| * + User Support
 |

|  |
| --- |
| User Story No: 30 Tasks: 5 AS A RESTAURANT OWNER I WANT TO VIEW FEEDBACK SO THAT I CAN KNOW THE CUSTOMERS FEEDBACK. |
| * Priority: Medium
 |
| * BV: 200
 |
| * CP: 04
 |
| * Acceptance Criteria:
 |
| * + Access to feedback system
 |
| * + Feedback Visibility
 |
| * + Feedback sorting and filtering
 |
| * + Response Mechanism
 |
| * + User Support
 |

|  |
| --- |
| User Story No: 31 Tasks: 3AS A ADMIN I WANT TO KNOW THE ISSUES SO THAT I CAN RESOLVE THEM. |
| * Priority: High
 |
| * BV: 100
 |
| * CP: 03
 |
| * Acceptance Criteria:
 |
| * + Display issue section
 |
| * + Sorting and filtering of issues list
 |
| * + Editing and modifying the issues
 |

|  |
| --- |
| User Story No: 32 Task:3AS A REGIONAL ADMIN I WANT TO KNOW THE ISSUES SO THAT I CAN RESOLVE THEM |
| * Priority: Medium
 |
| * BV: 200
 |
| * CP: 04
 |
| * Acceptance Criteria:
 |
| * + Display issue section
 |
| * + Sorting and filtering of issues list
 |
| * + Editing and modifying the issues
 |

|  |
| --- |
| User Story No: 33 Task:6AS A RESTAURANT OWNER I WANT TO VIEW REVENUE GENERATED SO THAT I VIEW RESTAURANTS REVENUE |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 04
 |
| * Acceptance Criteria:
 |
| * + Select Reports
 |
| * + Select Revenue Reports
 |
| * + Select to and from date
 |
| * + Select Region (can select all)
 |
| * + Generate Report
 |

|  |
| --- |
| User Story No: 34 Tasks: 2AS A RESTAURANT OWNER I WANT TO KNOW DELIVERY BOY SO THAT I VERIFY THE DELIVERY BOY. |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 04
 |
| * Acceptance Criteria:
 |
| * + ID proof
 |
| * + Punctuality and reliability
 |

|  |
| --- |
| User Story No: 35 Tasks: 2AS A CUSTOMER I WANT TO VIEW THE CONTACT NUMBER OF DELIVERY BOY SO THAT I CAN CONTACT DELIVERY BOY FOR THE STATUS. |
| * Priority: Low
 |
| * BV: 50
 |
| * CP: 01
 |
| * Acceptance Criteria:
 |
| * + Display delivery boy mobile number
 |
| * + Display delivery boy name in tracking field)
 |
| * + Display delivery boy picture
 |

|  |
| --- |
| User Story No: 36 Tasks: 2AS A RESTAURANT OWNER I WANT TO PROVIDE TIME SLOTS SO THAT CUSTOMER CAN CHECK OPENING AND CLOSING HOURS |
| * Priority: Medium
 |
| * BV: 100
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Click on restaurant dashboard
 |
| * + Add from time to time
 |
| * + Click on submit
 |
| * + Display updated successfully
 |

|  |
| --- |
| User Story No: 37 Tasks: 3AS A USER I WANT TO RECEIVE NOTIFICATIONS SO THAT I CAN RECEIVE UPDATES |
| * Priority: High
 |
| * BV: 200
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Notifications for order confirmation
 |
| * + Notification for dispatch
 |
| * + Notification for delivery
 |

|  |
| --- |
| User Story No: 38 Tasks: 1AS A CUSTOMER I WANT TO CONTACT CUSTOMER SUPPORT SO THAT I CAN SUBMIT QUERIES OR ISSUES |
| * Priority: Medium
 |
| * BV: 200
 |
| * CP: 02
 |
| * Acceptance Criteria:
 |
| * + Customer support section with contact information
 |

|  |
| --- |
| User Story No: 39 Tasks: 4 AS A CUSTOMER I WANT TO VIEW THE ORDER SO THAT I CAN CANCEL IT. |
| * Priority: Medium
 |
| * BV: 100
 |
| * CP: 03
 |
| * Acceptance Criteria:
 |
| * + Order status
 |
| * + Method of cancellation
 |
| * + Refund policy
 |
| * + Time frame
 |

|  |
| --- |
| User Story No: 40 Tasks: 4AS A REGIONAL ADMIN I WANT TO TRACK THE DELIVERY SO THAT I CAN VIEW THE STATUS OF THE DELIVERY. |
| * Priority: High
 |
| * BV: 100
 |
| * CP: 03
 |
| * Acceptance Criteria:
 |
| * + Real time tracking
 |
| * + Security and data privacy
 |
| * + User friendly Interface
 |
|  |

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

**An Epic is a large user story that cannot be completed within a single sprint and needs to be broken down into smaller, manageable user stories. It represents a significant feature or business requirement that delivers substantial value to the user. Epics typically span multiple sprints and provide a high-level understanding of a feature before it is broken down into smaller user stories.**

**Epic 1: Order Management System**

**Description:
The Order Management System allows customers to place, track, and manage their food orders efficiently. It also enables restaurant owners and delivery personnel to process and fulfill orders smoothly.**

**User Stories (Examples):**

1. **As a Customer, I want to place an order so that I can receive food at my location.**
2. **As a Customer, I want to track my order status so that I know when my food will arrive.**
3. **As a Restaurant Owner, I want to manage incoming orders so that I can prepare them accordingly.**
4. **As a Delivery Boy, I want to update the delivery status so that the customer knows when their order is out for delivery.**
5. **As an Admin, I want to analyze order trends so that I can optimize platform performance.**

**Epic 2: Restaurant Registration & Management**

**Description:
This feature allows restaurant owners to register, manage their menus, update their business details, and monitor revenue reports through an admin dashboard.**

**User Stories (Examples):**

1. **As a Restaurant Owner, I want to register my restaurant so that I can receive orders from customers.**
2. **As a Restaurant Owner, I want to add and update menu items so that customers can order available dishes.**
3. **As an Admin, I want to verify restaurant details so that only legitimate businesses are onboarded.**
4. **As a Restaurant Owner, I want to set operating hours so that customers know when the restaurant is open.**
5. **As a Restaurant Owner, I want to generate sales reports so that I can analyze my restaurant’s performance.**

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

|  |  |
| --- | --- |
| **Business Value (BV)** | **Complexity Points (CP) / Story Points** |
| Measures the **strategic importance** of work to the business. | Measures the **effort, complexity, and risk** of completing work. |
| **Why** should this be done? (ROI, customer impact) | **How hard** is it to complete? (time, difficulty) |
| Product Owner / Business Stakeholders | Development Team (Engineers, Testers, etc.) |
| Prioritization (what to work on first) | Estimation (how much work fits in a sprint) |
| Often 1-10, $$ value, or High/Medium/Low | Fibonacci-like (1, 2, 3, 5, 8, 13, etc.) |
| A feature that increases revenue = **High BV (10)** | A complex API integration = **8 Story Points** |
| Used in backlog prioritization (WSJF in SAFe) | Used in sprint planning & velocity tracking |
| Decides **what** gets built next | Decides **how much** can be built in a sprint |

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

**What is sprint Duration: 2 Weeks - Your sprint Value \_\_\_\_2 weeks \_\_\_\_\_\_\_(Sprint are limited to one calendar month / do days or less)**

**Scrum is a subunit of Sprint. What is scrum Duration: 1 day – Your scrum Value\_\_\_\_\_\_\_\_1 days \_\_\_\_\_\_\_\_**

**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 commited user stories by dev and QA team for that particular spint will be added in the sprint backlog**

|  |  |  |  |
| --- | --- | --- | --- |
| **PBI**  | **Tasks**  |  **WIP**  |  **Done**  |
| **Log In** | **Login user with email and password. A temporary password will be sent to the admin's 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** | **DOD** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

**1. Product Backlog**

**Definition:**

**The Product Backlog is a prioritized list of all the features, enhancements, bug fixes, and technical work needed to improve the product. It acts as the single source of truth for all future work.**

**Key Characteristics:**

* **Managed by the Product Owner.**
* **Contains all product requirements (features, technical work, bugs).**
* **Continuously updated and refined (Backlog Grooming).**
* **Items are ranked by priority (High-value features come first).**

**2. Sprint Backlog**

**Definition:**

**The Sprint Backlog is a subset of the Product Backlog containing items the team commits to completing during a single Sprint (usually 1-4 weeks).**

**Key Characteristics:**

* **Selected from the Product Backlog during Sprint Planning.**
* **Owned and managed by the Development Team.**
* **Contains tasks needed to complete backlog items.**
* **No changes mid-sprint (unless critical).**

**Product Backlog vs. Sprint Backlog**

|  |  |  |
| --- | --- | --- |
| ****Aspect**** | ****Product Backlog**** | ****Sprint Backlog**** |
| ****Definition**** | A **prioritized list** of all desired work (features, bugs, improvements) for the product. | A **subset of the Product Backlog** selected for a specific sprint, broken into actionable tasks. |
| ****Ownership**** | Managed by the **Product Owner (PO)**. | Owned by the **Development Team**. |
| ****Content**** | Contains **user stories, epics, bugs, and technical debt** (not time-bound). | Contains **tasks** needed to complete selected user stories (sprint-specific). |
| ****Flexibility**** | **Dynamic**—constantly refined, reprioritized, and updated. | **Fixed** during the sprint (scope changes require team agreement). |
| ****Purpose**** | Ensures **long-term product vision** and roadmap alignment. | Defines **what the team commits to deliver** in the current sprint. |
| ****Estimation**** | Estimated in **Story Points (complexity)** or **Business Value (priority)**. | Tasks are often estimated in **hours** or **effort**. |
| ****Example**** | - "As a user, I want to reset my password via email."- "Improve API response time." | - "Design password reset UI."- "Implement email service for temp passwords." |

**Question 7 – What is impediments log? write 2 impediments – 5 Marks**

**An impediment log, also known as an issue log or obstacle log, is a document or toolused in Agile software development to track and manage obstacles, bottlenecks, or anyfactors that impede the progress of a project or team**

**2 Impediments:**

* **Delivery partner shortage in a specific region**
* **Technical issue causing intermittent order processing failure**

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

|  |  |
| --- | --- |
| **LOGIN ID** | 1 |
| **Description** | Delivery partner storage in specific region |
| **Impact** | Delays in order deliveries and incmates customer dissatisfaction |
| **Priority** | High (due to its impact on customer experience) |
| **Assigned to** | Operations team and HR team |
| **Status** | Open |
| **Action Taken** | The operations team is actively recrtating new delivery partnem in the region. The HR team is working on fast-tracking the onboarding process. |
| **Resolution** | Delivery partner recruitment efforts are ongoing and the HR team is streamlining the onboarding process to expedite new hires. Regular updates are being provided in team meetings |

**Technical issue causing intermittent order processing failure**

|  |  |
| --- | --- |
| **LOGIN ID** | 2 |
| **Description** | Technical Issue causing intermittent order processing failures |
| **Impact** | Delays in order processing and potential revenue loss |
| **Priority** | High (due to its impact on revenue and customer experience) |
| **Assigned to** | Tech team and QA team |
| **Status** | In progress |
| **Action Taken** | The tech team has identified the root cause and is working on a fix. The QA team is conducting extensive testing to ensure the issue is resolved |
| **Resolution** | The tech team has implemented a fix and conducted through testing. The issue has been resolved, and orders are now processing smoothly |

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

**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**

**Velocity refers to the measure of the amount of work a development team can complete during a sprint.**

**The calculation of velocity is performed by the development team itself, as they are responsible for estimating the effort required to complete each user story or backlog item.**

**Story Point Estimation:**

**Story point estimation is a technique used in Agile software development to estimate the effort required for a specific task or user story. It's a relative measure of complexity rather than a fixed time unit.**

**Team members assign story points based on their understanding of the work involved, considering factors like complexity, effort, and uncertainty. The actual time a story point represents can vary from team to team. For some, it might equate to hours, while for others, it might represent days.**

**It's important to establish a consistent baseline within the team so that story point estimates can be used effectively for planning and prioritization.**

**Tracking Completed Work**

**Tracking completed work in Agile development typically involves calculating the total story points completed by the team over a specific time frame, usually a sprint or iteration.**

**Here's how you can calculate completed work:**

* **Identify Completed Stories: At the end of the sprint or iteration, review the user stories or tasks that were completed and accepted as done.**
* **Sum Story Points: Add up the story points assigned to all the completed user stories. Exclude any story points that were not fully finished or accepted during the sprint.**
* **Calculate Total Completed Work: The sum of story points completed represents the total completed work for that sprint.**

**This completed work can be used to calculate the team's velocity for that specific sprint, as mentioned in the previous response. It provides insights into the team's capacity and helps with future sprint planning and estimation.**

**Summing Story Points**

**Summing story points involves adding up the numerical values assigned to individual user stories or tasks during the estimation process in Agile development. Story points represent the effort, complexity, and size of a piece of work relative to other items on the backlog.**

**Here's how you can sum story points:**

* **List Completed User Stories: Gather a list of user stories or tasks that have been completed during a specific sprint or iteration.**
* **Identify Story Point Values: Each user story or task should have a story point value assigned to it during the estimation process. These values are usually relative, such as 1, 2, 3, 5, 8, 13, etc., representing increasing levels of complexity or effort.**
* **Add Up Story Point Values: Sum up the story point values for all the completed user stories or tasks.**

**Example:
If you completed user stories with story point values of 3, 5, and 8, the sum would be:**

**3+5+8=163 + 5 + 8 = 163+5+8=16**

**The sum of story points provides a quantitative measure of the work completed by the team during a sprint. This sum is often used to calculate the team's velocity, which helps in future sprint planning and estimation.**

**Average Velocity in Agile**

**Average velocity in Agile development refers to the average amount of work, measured in story points, that a team completes during a series of sprints or iterations. It is a key metric used for planning and estimating future work.**

**How to Calculate Average Velocity:**

* **Select a Time Frame: Choose a specific number of past sprints or iterations for which you want to calculate the average velocity. For example, you might choose the last 5 sprints.**
* **Sum Completed Story Points: Add up the total story points completed by the team in each of the selected sprints. This will give you the total completed work for the chosen time frame.**
* **Calculate the Average: Divide the total completed story points by the number of sprints selected. This will give you the average velocity for that period.**

**Formula: Average Velocity=** **Total Completed Story Points​/Number Of Sprints**

**Use for Sprint Planning:**

**✅ The average velocity serves as a guideline for future sprint planning.
✅ Helps the team estimate how much work they can commit to in upcoming iterations based on historical performance.**

**Things to Keep in Mind:**

**Velocity is a rough estimate and can fluctuate based on:**

* **Team capacity (availability, workload changes).**
* **Changes in team composition (new members, departures).**
* **Improvements in estimation accuracy over time.**

**Question 9 – Draw Sprint Burn Charts n Product Burn Down Charts– 3 Marks**

 **SPRINT BURNDOWN CHART**

****

**PRODUCT BURNDOWN CHART**

**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, integrations, 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 sprints.**

**Product grooming, also known as backlog grooming or refinement, is a crucial activity in Agile development that involves preparing and refining items in the product backlog to ensure they are well-understood, prioritized, and ready for development. Let's break down the process step by step:**

**● Setting the Context: At the beginning of the backlog grooming process, the team and relevant stakeholders come together to understand the overall goals and objectives of the project. This helps set the context for the work to be done and aligns everyone's understanding.**

**● Backlog Review: The product owner and the development team review the items in the product backlog. This involves assessing the user stories, tasks, and other items to ensure they are accurate, up-to-date, and still relevant to the project's goals.**

 **● Prioritization: During backlog grooming, the team collaboratively prioritizes the backlog items based on their value to the product and the needs of the users or customers. This helps ensure that the most important and valuable work is addressed first.**

**● Refinement and Estimation: In this step, the backlog items are refined to provide clear and detailed descriptions. The team breaks down user stories into smaller tasks and discusses the technical requirements. Estimation involves assigning story points or other sizing metrics to each item, indicating the relative effort needed for implementation.**

 **● Dependency Analysis: The team examines potential dependencies between backlog items. Identifying and understanding dependencies helps in planning the order of implementation and managing potential bottlenecks.**

 **● Acceptance Criteria: Well-defined acceptance criteria are established for each backlog item. These criteria outline the conditions that must be met for the item to be considered complete and ready for delivery. Clear acceptance criteria help prevent misunderstandings and ensure a shared understanding of what is expected.**

**● Backlog Grooming Meetings: These are recurring meetings where the product owner and the development team come together to perform the activities mentioned above. These meetings often occur before sprint planning sessions to ensure that the upcoming sprint backlog is well-prepared. Backlog grooming is an iterative process that helps maintain a healthy and well-organized product backlog. It ensures that the development team always has a prioritized list of well-defined, estimated, and ready-to-develop items. This, in turn, supports the efficient planning and execution of sprints and helps the team deliver value to customers in a more predictable and effective manner.**

**Question 11 – Explain the roles of Scrum Master and Product Owner – 3 Marks**

**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 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 in order to Product Goal 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 lifecycle.**

 **• Defining the vision**

 **• Prioritizing the product backlog**

**• Taking an overview of development stages**

 **• Handling communications**

**• Knowing what the client needs**

**• Evaluating progress**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Product Owner (PO)** | **Scrum Master (SM)** |
| **Nature of Work** | Collaborates with stakeholders to define the product vision and prioritize the backlog. | Acts as a team coach, ensures Scrum adherence, and focuses on product quality. |
| **Responsibilities** | - Ensures project completion on time.- Bridges gap between dev team and customers. | - Ensures Scrum framework is followed.- Helps team deliver a high-quality product. |
| **Accountability** | - Owns the backlog and timely delivery.- Provides updates to clients/stakeholders. | - Accountable for product quality.- Updates management on progress/team efficiency. |
| **Reporting** | Reports to top management and clients. | Reports to management on team efficiency and product quality. |
| **Key Qualities** | - Strong communication & leadership.- Creativity, critical thinking, strategic mindset. | - Deep Scrum knowledge.- Servant leadership (guides without authority). |

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

**Sprint Planning**: This meeting kicks off each sprint, which is a time-boxed

iteration of work, usually spanning 2-4 weeks. During this meeting, the Scrum

team, including the Product Owner, Scrum Master, and Development Team,

collaborates to determine which backlog items (user stories, features, etc.) will be

worked on in the upcoming sprint. The team also breaks down these items into

tasks and estimates the effort required.

**Daily Stand-up (Daily Scrum)**: Held daily during the sprint, this short meeting

aims to facilitate quick and focused communication among team members. Each

team member answers three key questions: What did I accomplish since the last

stand-up? What will I work on until the next stand-up? Are there any obstacles or

impediments in my way? This meeting helps keep everyone aligned and

informed about the progress and challenges.

**Sprint Review**: At the end of each sprint, the team holds a review meeting to

showcase the work completed during the sprint to stakeholders, customers, and

the Product Owner. The team demonstrates the potentially shippable product

increment and gathers feedback. Based on this feedback, the Product Owner

can update the backlog.

**Sprint Retrospective**: Also held at the end of each sprint, the retrospective is a

dedicated time for the team to reflect on their processes and practices. The team

discusses what went well, what could be improved, and any potential changes

they'd like to make in the next sprint to enhance their efficiency and

effectiveness.

**Backlog Refinement (Grooming)**: While not officially part of the Scrum events,

backlog refinement is an important ongoing activity. During these sessions, the

team and the Product Owner review and refine backlog items, adding details,

clarifications, and estimates to make them ready for inclusion in future sprints.

**Product Backlog Refinement**: This meeting focuses on refining the product

backlog items. The team and the Product Owner discuss and clarify

requirements, priorities, and any changes needed in the backlog items. This

ensures that the backlog is well-prepared for upcoming sprints.

• Sprint planning meeting. Before your team begins a Scrum sprint, you need to know where you're going. ...

• Daily standup meeting. ...

 • Sprint review meeting. ...

• Sprint retrospective meeting.

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.

**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.

**Sprint Size**: In Scrum, a "sprint" is a time-boxed iteration during which the development team works to deliver a potentially shippable product increment. The length of a sprint is referred to as the "sprint duration" and is usually fixed throughout the project. Common sprint durations are 1 to 4 weeks. The choice of sprint duration depends on factors such as team velocity, project complexity, and business needs. A shorter sprint encourages more frequent opportunities for feedback and adaptation, while a longer sprint provides more time for development.

**Scrum Team Size**: The Scrum team size refers to the number of individuals who collectively contribute to the development of the product. A Scrum team consists of three key roles: the Product Owner, the Scrum Master, and the Development Team. The Development Team, in particular, is responsible for creating the product increment. Scrum recommends that the Development Team size be kept small, typically between 3 to 9 members, to facilitate effective communication, collaboration, and decision-making.

**Question 14 – Explain DOR and DOD – 2 Marks**

**Definition of Ready (DOR): The Definition of Ready outlines the criteria that a product backlog item (user story, feature, task, etc.) should meet before it is considered ready to be taken into a sprint for development. The DOR ensures that the item is well-defined, understood, and prepared for efficient development. The specific criteria in the DOR can vary from team to team, but commonly include elements such as:**

**● Clear description and acceptance criteria: The item's requirements are clearly stated, and the conditions for its successful completion are well-defined.**

**● Dependencies identified: Any dependencies on external factors, teams, or resources are identified and addressed.**

**● Estimable: The team has enough information to provide a reasonable estimate of the effort required.**

**● Testable: It's possible to determine whether the item has been successfully implemented through testing.**

**● Minimal ambiguity: The item's details are clear, and any uncertainties are resolved.**

**Definition of Done (DOD): The Definition of Done outlines the criteria that must be met for a product increment or backlog item to be considered complete and potentially shippable. The DOD ensures that the team maintains a consistent level of quality and completeness in their work. The specific criteria in the DOD can vary based on the team's standards, the nature of the project, and the industry, but commonly include elements such as:**

**● Code complete: All development work is finished, including coding, testing, and integration.**

**● Peer-reviewed: Code has been reviewed by other team members for quality and adherence to coding standards.**

**● Automated tests passed: Automated tests (unit tests, integration tests, etc.) have been successfully executed and passed.**

**● Functional requirements met: The item meets all specified acceptance criteria and functional requirements.**

**● Documentation updated: Any necessary documentation, user guides, or technical documentation has been updated.**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Definition of Ready (DoR)(Must be true BEFORE sprint starts)** | **Definition of Done (DoD)(Must be true BEFORE story is "Done")** |
| **Purpose** | Ensures the story is ready to be worked on. | Ensures the story is shippable. |
| **Clarity** | Story is clear, concise, and understood by the team. | Code fulfills all requirements. |
| **Scope** | Feasible within sprint capacity. | All tasks completed (no half-done work). |
| **Acceptance Criteria** | Explicitly defined (e.g., "User can filter search results"). | Met and verified (e.g., tested by PO). |
| **Dependencies** | Identified and resolved (e.g., API access ready). | Implemented and documented (e.g., DB changes deployed). |
| **Estimations** | Story points or time estimated. | Effort validated (no surprises). |
| **Testing** | Testable (QA can create test cases). | Tested (unit, integration, system, UAT). |
| **Code Quality** | – | Code reviewed, merged, and follows standards. |
| **Deployment** | – | Deployed to test/staging environment. |
| **Documentation** | Mockups/requirements attached. | Updated (code comments, user guides, etc.). |
| **Approval** | Team agrees it’s ready to start. | Stakeholders (PO, QA) sign off. |

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

**Prioritization Technique –**

**MOSCOW” Technique: Must should could would**

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

**“FURPS” Technique This technique is used to validate must requirement. F-Functionality U-Usability R-Reliability P-Performance S-Supportability (Extendable, Testable & Enhance-able)**

**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.**

 **Prioritize Requirements**

**Goal: Rank requirements by importance to focus development efforts.**

**Techniques:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |  |  |
| --- | --- | --- |
| Method | How It Works | When to Use |
| $100 Test | **Stakeholders "spend" a fictional $100 on features they value most.** | **Quick, democratic prioritization.** |
| Top 10 Requirements | **Stakeholders list their top 10 must-have features; overlap identifies priorities.** | **Aligning stakeholder expectations.** |
| Numerical Assignment | **Assign weights (e.g., 1-5) to requirements based on importance/cost.** | **Quantitative decision-making.** |
| MoSCoW | **Categorize as: Must-have, Should-have, Could-have, Won’t-have.** | **Agile projects with clear scope phases.** |

 |

**Example (MoSCoW):**

* **Must-have: User authentication (security).**
* **Should-have: Search functionality (core UX).**
* **Could-have: Dark mode (aesthetic).**
* **Won’t-have: AI recommendations (v2).**

 **Validating Requirements**

**Goal: Ensure requirements are clear, feasible, and aligned with business goals.**

**Techniques:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |  |  |
| --- | --- | --- |
| Method | Purpose | Key Questions |
| FURPS | **Evaluate requirements by: Functionality, Usability, Reliability, Performance, Supportability.** | **Does it meet functional needs? Is it user-friendly? Scalable?** |
| CUCV | **Check for Completeness, Unambiguity, Consistency, Verifiability.** | **Are there gaps? Can it be tested?** |
| CAE | **Assess Correctness, Ambiguity, Essentiality.** | **Is it technically accurate? Is it critical to the project?** |
| APVU | **Classify requirements as Ad-hoc, Planned, Varying, Unplanned.** | **How stable/urgent is the requirement?** |
| SMART | **Ensure requirements are Specific, Measurable, Achievable, Relevant, Time-bound.** | **Can success be quantified? Is it realistic?** |

 |

**Minimum Viable Product (MVP): An MVP is the smallest version of a product that includes just enough features to provide value to early adopters and gather feedback. The MVP approach helps validate assumptions, learn from users, and iteratively build upon a product's foundation. It involves:**

**● Core Functionality: An MVP focuses on delivering the core functionalities that address the primary needs or pain points of the target users.**

**● Minimal Features: The MVP omits non-essential features to avoid unnecessary complexity and expedite development.**

**● Testing Hypotheses: The MVP tests assumptions and hypotheses about user behavior, market demand, and product viability.**

**● Iterative Development: Based on user feedback, the product is refined and expanded in subsequent iterations, gradually adding more features.**

**● Early Value: The MVP allows the product to be released faster, gaining valuable insights and attracting early adopters.**

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

**Product Owner role In many organizations, the Product Owner also has a Product Manager title and is responsible for making decisions about the product, for managing the product’s strategic roadmap and for communicating that roadmap. If there are cross-product impacts, then coordination of feature prioritization with other product POs is necessary - because certainly those in the C-suite are going to want to see a consolidated view of all product priorities.**

**Business Analyst role Similar to the PO, the BA plays a critical role in working with the scrum team to execute the product vision by defining needs and recommend solutions that deliver value. The BA goes a bit deeper by breaking down high level product features into user stories, with the appropriate amount of detail. This may result in other BA artifacts –some of the most common examples I’ve experienced are capabilities gap analysis and process flow diagrams.**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Business Analyst (BA)** | **Product Owner (PO)** |
| **Primary Focus** | Analyzes business needs, processes, and requirements. | Owns the product vision, strategy, and value delivery. |
| **Requirement Handling** | Gathers and documents detailed business/functional requirements. | Creates **user stories** and defines product features for development teams. |
| **Problem-Solving** | Identifies inefficiencies and suggests process improvements. | Solves user/customer problems through product features and innovation. |
| **Communication** | Bridges gaps between business stakeholders and IT/development teams. | Collaborates with stakeholders, customers, and dev teams to align on product goals. |
| **Documentation** | Produces business rules, workflows, and requirement specifications. | Maintains the **product backlog** and refines user stories. |
| **Scope Definition** | Defines project scope based on business objectives. | Defines product scope (features, enhancements) and MVP boundaries. |
| **Vision & Strategy** | Supports project-level goals and process optimization. | Drives long-term **product strategy** and market fit. |
| **Backlog Management** | Not responsible for backlogs. | **Owns the backlog**—prioritizes and grooms items for sprints. |
| **Prioritization** | Provides input but doesn’t decide feature priority. | Prioritizes features based on **business value**, ROI, and user needs. |
| **Decision-Making** | Recommends solutions; final decisions are made by others. | **Final authority** on product features and release decisions. |
| **Agile Involvement** | May support projects in waterfall or Agile environments. | Integral to Agile teams (e.g., sprint planning, reviews, retrospectives). |
| **Stakeholder Collaboration** | Works with business units and IT teams. | Engages customers, executives, and dev teams to validate direction. |
| **Acceptance Criteria** | Ensures solutions meet business requirements. | Ensures deliverables meet **user story acceptance criteria** and product vision. |
| **Leadership** | Tactical role focused on analysis and documentation. | **Strategic leadership**—aligns product with company goals and market trends. |
| **Continuous Improvement** | Optimizes business processes and systems. | Iterates on product based on user feedback and metrics. |

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

****

