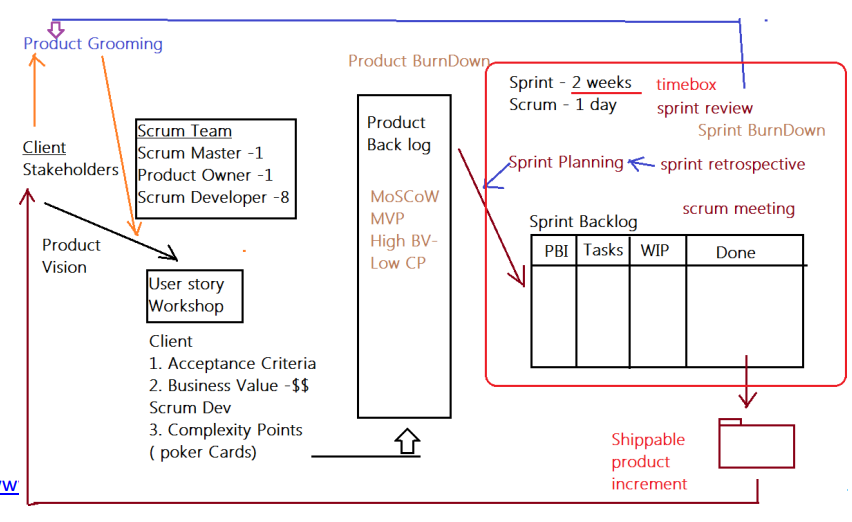
**CAPSTONE PROJECT 2- Agile-Scrum**

**Scrum Project Name:** Scrum Foods (Foods Delivery Applications) **Scrum Project Description:** Scrum Foods provides fast, reliable online food delivery application targeting customers of all age group offering 24/7 service along with tracking of the delivery real time.



|  |
| --- |
| **Client:** COEPD IT Solutions |
| **Stakeholder List:** |
| 1. Business Owner |
| 1. Administrator |
| 1. Regional Administrator |
| 1. Restaurant |
| 1. Delivery Boy |
| 1. Customer |

**Product Vision – Meeting Discussion Document**

|  |
| --- |
| Scrum Project Name: **Scrum Foods** |
| Venue: |
| Date: Start Time: End Time: Duration: |
| **Client: COEPD IT Solutions** |
| **Stakeholder List:** |
| Business Owner |
| Administrator |
| Regional Administrator |
| Restaurant |
| Delivery Boy |
| Customer |
|  |
| Scrum Team |
| Scrum Master: Satya Rathnakar |
| Product Owner: YOU |
| Scrum Developer 1: Linesh Vegad |
| Scrum Developer 2: Yogender |
| Scrum Developer 3: Gowri |
| Scrum Developer 4: A.Lakshmikala |
| Scrum Developer 5: Madhuri |
| Scrum Developer 6: Varun |
| Scrum Developer 7: Rakesh |
| Scrum Developer 8: Rajesh |



|  |  |  |  |
| --- | --- | --- | --- |
| VISION: Customers of any age group who want to have top-rated restaurant’s delicious food in one go at their doorstep with lighting fast delivery, 24/7 availability and reliable services are SCRUM FOODS. Unlike any other food delivery app in market, our utmost priority is customer need and customer satisfaction which makes us stand out of any other food delivery app in this industry. | | | |
| Online food delivery industry is our target segment.  Users/Customers: People who want food deliveries within stipulated time at their required place | Scrum foods provide guaranteed safe food delivery in one-go of top-rated licensed restaurants.  Customers can receive with safe packaged food within expected time to clear their hunger. | Scrum foods will be on mobile, tablets and desktop application.  Real time tracking and providing 24/7 services and customer support Product feasibility can be complex and require attention in every aspect | Open up revenue stream  Be leading online food delivery app in country  Create reputed brand image for other business opportunities. |

**Product Vision – Description Notes**

Customer: Registration, Login, Search and View restaurants, View restaurant's menu, Order food, Payment, Tracking, Cancel order, Feedback & Rating and Logout.

Delivery Boy: Registration, Login, View orders, Select and accept orders, Order pickup and delivery, Status updates, Payment (COD), View Feedback, Raise Issues, View Deliveries report and revenue generated, Logout.

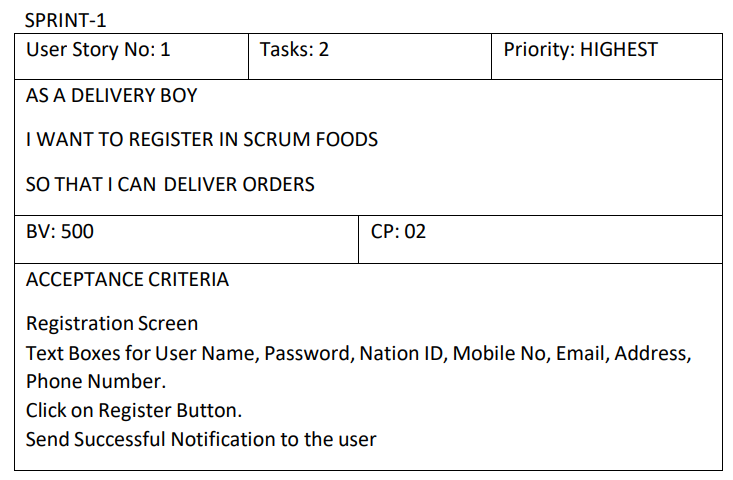
Restaurants: Registration, Login, View Orders, Delivery Boy Verification, Payment, View Feedback, Raise Issues, View revenue generated through Scrum Food app, Logout.

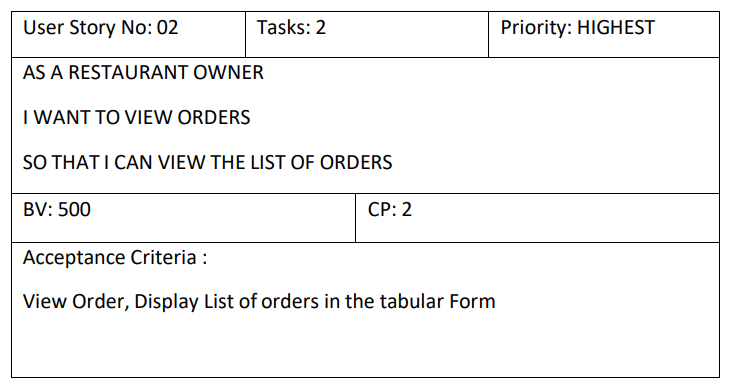
Regional Admin: Admin Login, Tracking/status, Customer feedback, Managing Regional delivery boys and restaurant, View regional revenue, Issues, Refunds, View payment made to regional restaurant and Logout.

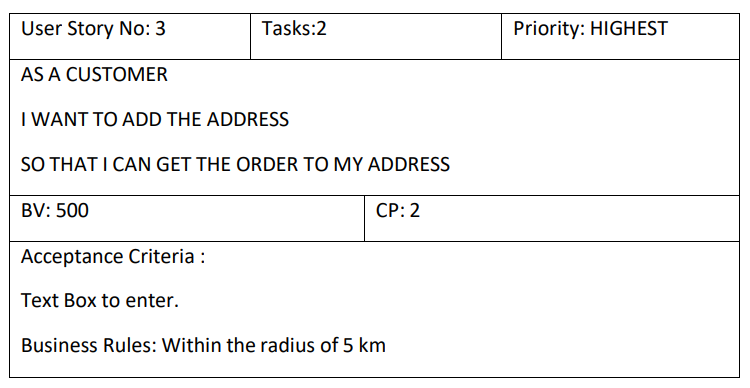
Admin: Login, Managing Regional Admin, Issues, Customer Feedback, Approval/Rejections privileges on restaurants, Delivery boy, Restaurants and Regional Admin requests, Resolve Issues and Logout

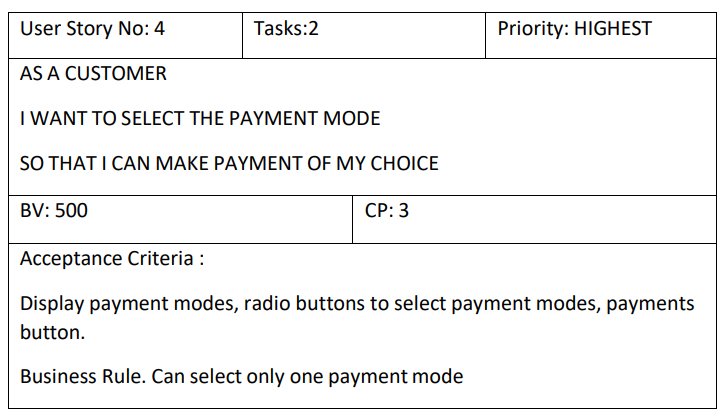
Business Owner: Login, Issues, Reports, Update payments for restaurants and delivery boys and Logout

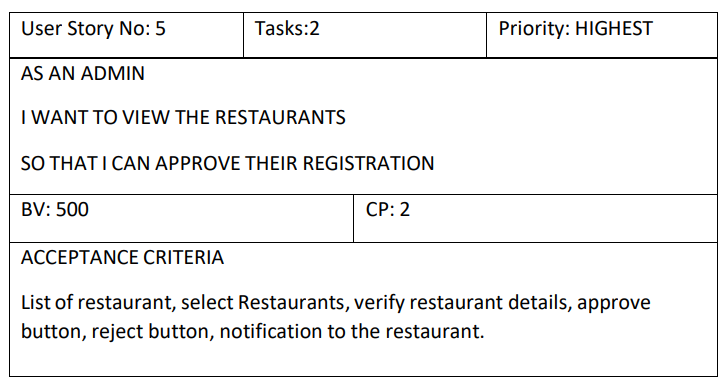
**User Stories:**

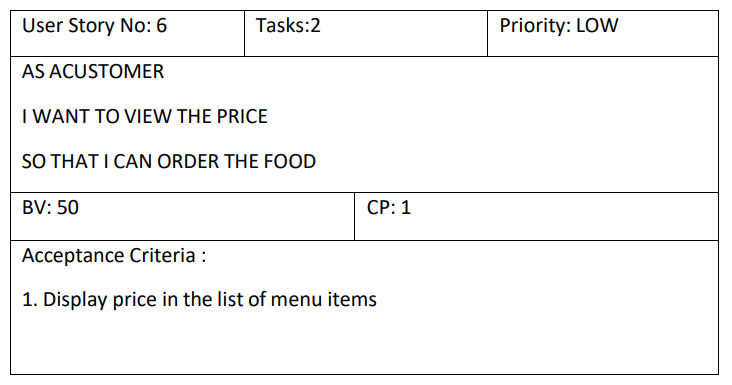
****

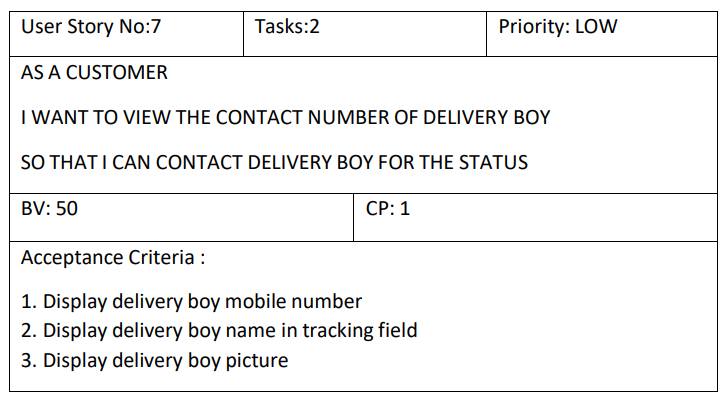
****

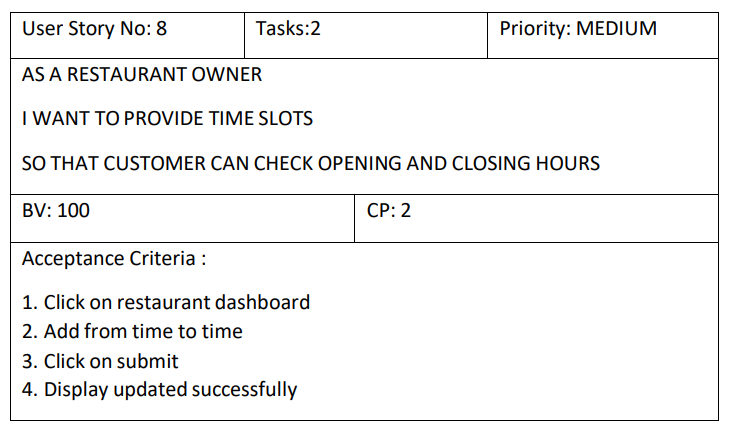
****

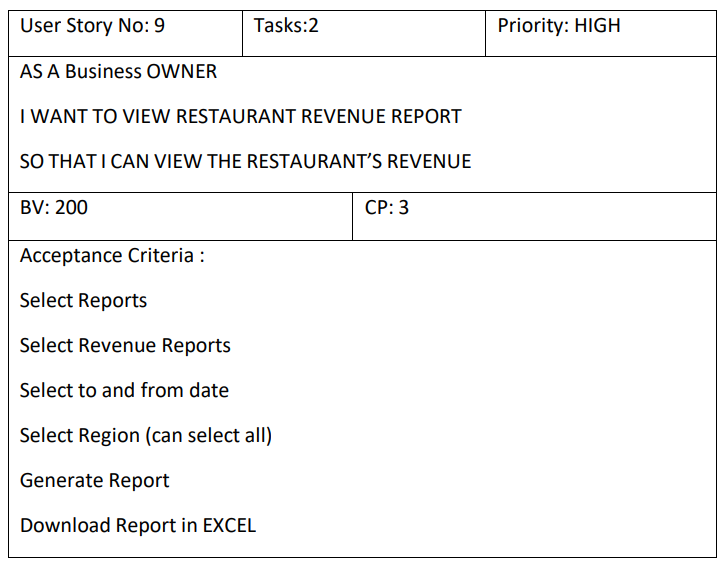
****

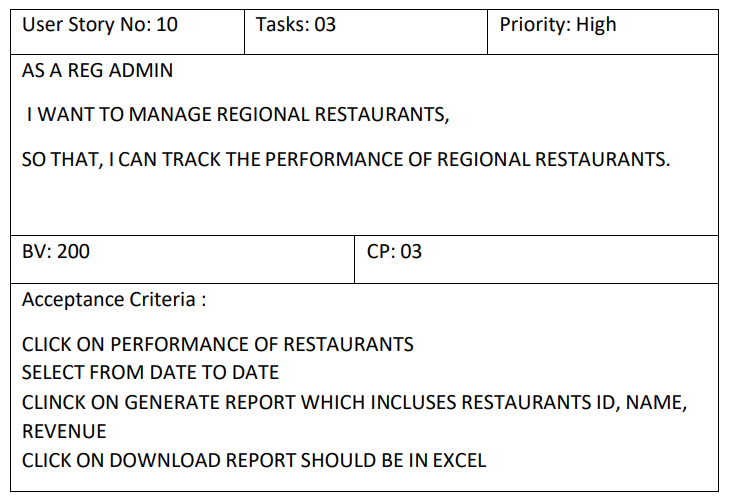
****

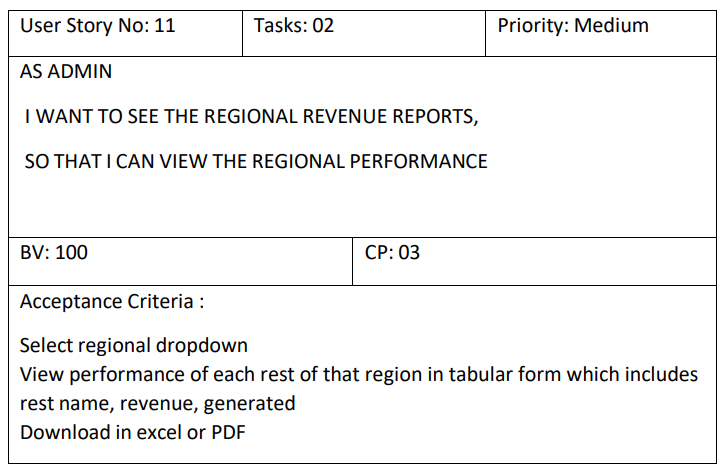
****

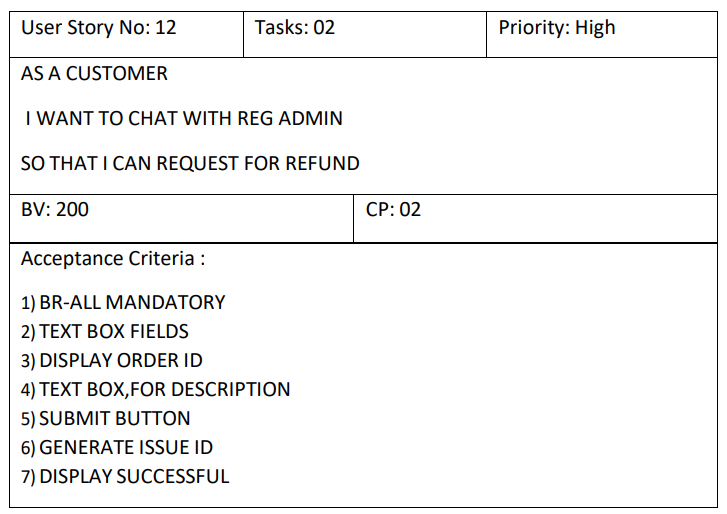
****

****

****

****

****

****

**Adding BV and CP to User stories**

**BV – Business Value:** This is not the cost of Development or the complexity of the feature. Business Value is how important is this feature (user Story) to the Business. This is estimated by Scrum Currency Notes. We provide Rs 1000. Rs 500, Rs 100, Rs 50, Rs 20 and Rs 10 Denominations. These estimations are done by the Stakeholders (Clients). If different values are selected by the stakeholders, then discussions will happen, and they agree to one BV value to that user story.

**CP – Complexity Points:** CP is also known as Story Points (SP). CP is the effort required by the Scrum Developers to develop this feature (user story) using technology. Efforts include time taken to solve the complexity and write the code. CP is estimated by the Scrum Developers by using Poker cards. We provide pokers with values “?”, 1, 2, 3, 5, 8, 13, 20, 40, 100 and BIG. If the entire Project development takes 200 points, then this user story coding effort will be… how many points? … Thinking in this way, Scrum Developers will give CP to the User story.). If different values are selected by the Scrum Developers, then discussions will happen, and they agree to one CP value to that user story.

**Question 1:** Write Agile Manifesto **8 marks**

**Answer:** The **Agile Manifesto** is a set of values and principles that guide agile software development. It was created in **2001** by a group of 17 software developers who wanted a more flexible, customer-focused approach to building software compared to traditional, rigid methodologies like Waterfall.

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

* **Individuals and interactions** over processes and tools
* People and collaboration matter more than rigid procedures.
* **Working software** over comprehensive documentation
* Delivering functional software is more important than excessive paperwork.
* **Customer collaboration** over contract negotiation
* Continuous involvement of customers ensures the right product is built.
* **Responding to change** over following a plan
* Flexibility is key in a fast-changing environment.

**The 12 Agile Principles**:

* **Customer satisfaction** through early and continuous software delivery.
* **Welcome changing requirements**, even late in development.
* Deliver working software **frequently** (weeks rather than months).
* Business people and developers must **work together daily**.
* Build projects around **motivated individuals**, trusting them to get the job done.
* Use **face-to-face communication** whenever possible.
* **Working software** is the primary measure of progress.
* **Sustainable development**—teams should maintain a constant, steady pace.
* Continuous attention to **technical excellence** and good design.
* Simplicity—the **art of maximizing work not done**—is essential.
* The best architectures, requirements, and designs **emerge from self-organizing teams**.
* Regularly reflect on how to become **more effective**, then adjust accordingly.

Agile has influenced many frameworks like **Scrum, Kanban, and SAFe**, which organizations use to implement Agile principles effectively.

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

**Answer:**

|  |
| --- |
| User Story ID: 001  Tasks:4  Priority:1 |
| Value Statement: As a hungry customer, I want to order food using Scrum Foods, so that I can enjoy a delicious meal without any hassle. |
| BV: 1000  CP:5 |
| Acceptance Criteria:   * Customer installs the application and registers. * Home screen displays restaurants and food options. * Customer selects a restaurant, explores the menu, and adds items to the cart. * Proceeds to checkout and confirms the order. * If a restaurant/item is unavailable, alternative suggestions are provided. * Notifications on promotions and discounts appear during checkout. * In case of network issues, an appropriate error message appears. |

|  |
| --- |
| User Story ID: 002  Tasks:3  Priority:1 |
| Value Statement: As a customer, I want to track my food in real-time so that I know exactly when it will arrive. |
| BV: 900  CP:8 |
| Acceptance Criteria:   * Order tracking screen displays real-time updates. * ETA is updated dynamically based on delivery progress. * Alerts notify users of delays due to traffic or other issues. |

|  |
| --- |
| User Story ID: 003  Tasks:3  Priority:1 |
| Value Statement: As a customer, I want multiple payment options so that I can pay conveniently. |
| BV: 800  CP:8 |
| Acceptance Criteria:   * Users can choose between Credit Card, Debit Card, UPI, Wallets, and COD. * Secure payment processing through a trusted gateway. * Payment success/failure notification is shown immediately. |

|  |
| --- |
| User Story ID: 004  Tasks:3  Priority:2 |
| Value Statement: As a delivery boy, I want to view available orders so that I can accept and complete deliveries. |
| BV: 850  CP:6 |
| Acceptance Criteria:   * Login and access assigned deliveries. * Option to accept/reject an order. * Real-time route mapping to customer location. * Order completion status updates. |

|  |
| --- |
| User Story ID: 005  Tasks:2  Priority:2 |
| Value Statement: As a delivery boy, I want a payment summary feature so that I can track my earnings. |
| BV: 700  CP:5 |
| Acceptance Criteria:   * Total earnings displayed based on completed orders. * Option to view payment details and pending settlements. |

|  |
| --- |
| User Story ID: 006  Tasks:5  Priority:1 |
| Value Statement: As a restaurant owner, I want to list my restaurant on Scrum Foods so that customers can order food. |
| BV: 1000  CP:10 |
| Acceptance Criteria:   * Restaurant registration and verification. * Dashboard to manage menu, pricing, and availability. |

|  |
| --- |
| User Story ID: 007  Tasks:3  Priority:2 |
| Value Statement: As a restaurant, I want to receive real-time order notifications so that I can prepare food promptly. |
| BV: 950  CP:6 |
| Acceptance Criteria:   * Orders are displayed with preparation time estimates. * Alerts for new and pending orders. |

|  |
| --- |
| User Story ID: 008  Tasks:5  Priority:1 |
| Value Statement: As an admin, I want to manage restaurants and delivery boys so that only verified users operate on the platform. |
| BV: 1000  CP:10 |
| Acceptance Criteria:   * Admin can approve/reject registration requests. * System flags suspicious activities for review. |

|  |
| --- |
| User Story ID: 009  Tasks:4  Priority:2 |
| Value Statement: As a business owner, I want to generate reports so that I can track revenue and business performance. |
| BV: 900  CP:8 |
| Acceptance Criteria:   * Dashboard displays revenue, order statistics, and trends. * Filters for viewing data by time, location, or vendor. |

|  |
| --- |
| User Story ID: 010  Tasks:4  Priority:2 |
| Value Statement: As a customer, I want to receive push notifications about my order status so that I stay updated. |
| BV: 850  CP:6 |
| Acceptance Criteria:   * Notifications for order confirmation, preparation, dispatch, and delivery. * Alerts for delays or other updates. |

**Question 3:** Epics- What is epic? Write 2 epics with Business Value and Complexity Points **5 marks**

**Answer:** In Agile, an epic is a large body of work that can be broken down into smaller, manageable user stories. Epics typically represent high-level features, goals, or functionalities that span multiple sprints and require significant effort.

**Epic 1: Restaurant Ratings and Reviews**

As a user, I want to view ratings and reviews for the restaurants on Scrum Foods, so that I can make informed decisions about where to order food from.

As a user, I want to provide ratings and reviews for the restaurants on Scrum Foods, so that I can share my experiences with other users and contribute to the community.

Acceptance Criteria:

* Users can view average ratings and reviews for each restaurant on the restaurant details page.
* User can read detailed reviews and comments left by other customers.
* Users can sort and filter reviews based on the criteria such as ratings, recency and relevance.
* User can rate a restaurant and leave a review after placing an order.
* User can edit or delete their own reviews within a specified timeframe.
* Reviews are displayed in a way that provides helpful insights to other users.
* The rating and review system maintains the integrity and authenticity of user feedback.

**Epic 2: Scheduled Orders**

As a user, I want to schedule food orders in advance on Scrum Foods, so that I can plan meals ahead of time and avoid last-minute hassles.

As a user, I want to have the flexibility to choose specific delivery time slots for scheduled orders on Scrum Foods.

Acceptance Criteria:

* Users can select a future date and time for placing a scheduled order.
* Users can choose from available delivery time slots during the day, evening and night.
* User can browse and order from all available restaurants during the scheduled order placement.
* User can modify or cancel scheduled order within a specified timeframe.
* User can receive remainders and notifications regarding their scheduled order, including order confirmation and upcoming delivery alerts.
* Delivery executives receive clear instructions about scheduled orders and prioritize them accordingly.
* The system handles scheduled orders seamlessly, ensuring timely delivery and accurate order fulfilment.

**Question 4:** BV-CP – What is the difference between BV and CP **2 marks**

**Answer:**

|  |  |  |
| --- | --- | --- |
| Aspect | Business Value (BV) | Complexity Points (CP) |
| Definition | Represents the importance of a feature to the business. | Represents the effort required to develop the feature. |
| Who Estimates | Stakeholders/Clients decide BV. | Scrum Developers estimate CP using Planning Poker. |
| Measurement Unit | Uses monetary denominations (e.g., Rs 1000, Rs 500, Rs 100). | Uses Fibonacci sequence values (1, 2, 3, 5, 8, etc.). |
| Focus Area | Focuses on business impact and value to users. | Focuses on the effort, complexity, and development time. |
| Purpose | Helps prioritize features based on business goals. | Helps estimate development effort and sprint capacity. |
| Example | A "Secure Payment System" may have **BV = Rs 2000** due to high importance. | The same feature may have **CP = 8** due to technical complexity. |

**Key Takeaways:**

* BV helps decide what to build first (business priority).
* CP helps estimate the effort required to build it (development workload).

**Question 5:** Explain about Sprint– Sprint Understanding: **5 marks**

**Answer:** A Sprint is a time-boxed iteration in the Scrum framework where a team works on a set of predefined tasks to create a potentially shippable product increment. Sprints are typically 1 to 4 weeks long and follow a structured workflow, including planning, execution, review, and retrospective.

**Key Aspects of a Sprint:**

* **Time-boxed Duration:** A Sprint is time-boxed, meaning it has a fixed duration, usually between 1 to 4 weeks. The time frame is chosen based on team capacity and project needs. Once set, it does not change to ensure predictability and focus.
* **Define Goal:** Each Sprint has a Sprint Goal, which is a clear, concise objective that guides the development work. It ensures alignment between the team and stakeholders, helping to measure success at the end of the Sprint.
* **Sprint Planning:** Sprint Planning is a meeting held at the beginning of each Sprint where the team:
* Reviews the **Product Backlog** (list of prioritized features).
* Selects items to work on during the Sprint (Sprint Backlog).
* Breaks down tasks and estimates effort.
* Aligns on the **Sprint Goal**.
* **Daily Standup Meetings:** A **Daily Standup (Daily Scrum)** is a **15-minute time-boxed meeting** where team members share updates:
* What they worked on yesterday.
* What they plan to do today.
* Any blockers or issues they face.

This keeps everyone aligned and helps remove roadblocks quickly.

* **Development and Testing:** During the Sprint, the team works on:
* **Development** – Writing code and implementing features.
* **Testing** – Performing unit testing, integration testing, and automated testing.
* **Bug Fixing** – Resolving defects to maintain quality.

Testing happens **continuously** to ensure the product remains stable.

* **Incremental Delivery:** At the end of the Sprint, the team delivers a **potentially shippable increment**—a working product that adds value. This aligns with **Agile principles** of iterative development and quick feedback loops.
* **Sprint Review:** A **Sprint Review** is held at the end of the Sprint where:
* The team demonstrates the completed work to stakeholders.
* Feedback is collected to improve future iterations.
* Adjustments are made to the Product Backlog.
* **Sprint Retrospective:** A **Sprint Retrospective** is a meeting where the team reflects on the Sprint to:
* Identify **what went well**.
* Discuss **challenges and areas for improvement**.
* Define **action items** for the next Sprint.

The goal is **continuous improvement** in processes and teamwork.

* Backlog Refinement: Backlog Refinement (or Grooming) is an ongoing process where:
* The **Product Owner and team** review and prioritize backlog items.
* Items are clarified, broken down, and estimated.
* Dependencies and risks are identified.

This ensures the backlog is **ready for the next Sprint Planning session**.

A **Sprint** is the backbone of Agile development, ensuring fast feedback, continuous delivery, and adaptability. It follows a structured approach with planning, execution, review, and improvement, helping teams deliver high-quality software efficiently.

**Vision Statement:**

Customers of any age group who want top-rated restaurant food delivered quickly, with **lightning-fast delivery, 24/7 availability, and reliable service,** will choose **Scrum Foods**. Unlike other food delivery apps, our priority is **customer needs and satisfaction**, setting us apart in the industry.

* **Target Segment:** Online food delivery industry.
* **Users/Customers:** People who require food deliveries within a stipulated time at their chosen location. Scrum Foods guarantees **safe, single-go delivery** from top-rated, licensed restaurants.
* **Platform Availability:** Mobile, tablets, and desktop application.
* **Key Features:**
* **Real-time tracking** and 24/7 customer support.
* **Product feasibility:** Requires attention to various aspects of service.
* **Revenue generation opportunities.**
* **Aspiration:** Become the **leading** online food delivery app in the country.
* **Brand reputation:** Establish a recognized name for business opportunities.
* **Stakeholder List:**
* Business Owner
* Administrator
* Regional Administrator
* Restaurant
* Delivery Boy
* Customer
* **Scrum Team:**
* **Scrum Master:** Satya Rathnakar
* **Product Owner:** Me
* **Scrum Developers:** Linesh Vegad, Yogender, Gowri, A. Lakshmikala, Madhuri, Varun, Rakesh, Rajesh
* **Product Vision – Description Notes:**
* **Customer Functionalities:**
  + - Registration, Login, Search & View restaurants.
    - View menu, Order food, Payment processing.
    - Tracking, Cancel order, Feedback & Rating.
    - Logout.
* **Delivery Boy Functionalities:**
* Registration, Login, View orders, Accept orders.
* Order pickup & delivery, Status updates, COD payments.
* View Feedback, Raise Issues, View revenue & deliveries.
* Logout.
* **Restaurant Functionalities:**
* Registration, Login, View Orders
* Delivery Boy verification, Payment processing.
* View Feedback, Raise Issues, Revenue tracking.
* Logout.
* **Regional Admin Functionalities:**
* Admin Login, Order tracking/status.
* Customer feedback, Managing regional deliveries & restaurants.
* View regional revenue, Handle issues, Refunds.
* View payments made to regional restaurants.
* Logout.
* **Admin Functionalities:**
* Login, Managing Regional Admins.
* Handling issues, Customer feedback.
* Approval/Rejections of restaurant & delivery boy requests.
* Resolving issues.
* Logout.
* **Business Owner Functionalities:**
* Login, Handle issues, Generate Reports.
* Update payments for restaurants & delivery boys.
* Logout.
* **Sprint Backlog:**

|  |  |  |  |
| --- | --- | --- | --- |
| **PBI** | **Tasks** | **WIP** | **Done** |
| Customer Registration | Develop registration form, Validate inputs | UI completed, Backend pending | Completed & Tested |
| Order Placement | Build order system, Integrate payment gateway | API integration in progress | Completed & Tested |
| Real-time Order Tracking | Implement tracking system, Map integration | Design finalized | Completed & Tested |
| Delivery Boy Assignment | Auto-assign delivery, Update order status | Logic implementation ongoing | Completed & Tested |
| Customer Feedback Mechanism | Build feedback module, Rating system | UI ready, Backend pending | Completed & Tested |

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

Sprint Planning Meeting: All 8 Scrum Developers will gather before the sprint starts and understand how many user stories, they can develop in 1 sprint (2 weeks), and move them from the product Backlog to the sprint Backlog. They take inputs from Sprint Retrospective meeting. Scrum Meeting – Daily Stand-up Meeting End of every Scrum, Scrum Developers will participate in Scrum meeting. Here they must answer 3 Questions.

a. What task did you work in this scrum?

b. What task will you work on next scrum?

c. Any Challenges/impediments? When you will complete the user story? **5 Marks**

**Answer:** In Scrum, the Product Backlog and Sprint Backlog are essential components of managing work efficiently.

The **Product Backlog** is a **prioritized list** of all the features, enhancements, bug fixes, and technical tasks required for the product. It is **owned by the Product Owner** and continuously refined based on business needs and feedback. Whereas, the **Sprint Backlog** is a subset of the **Product Backlog**, containing items selected for a specific Sprint. The **Development Team owns it** and commits to completing these items within the Sprint (usually 1-4 weeks).

|  |  |  |
| --- | --- | --- |
| **Feature** | **Product Backlog** | **Sprint Backlog** |
| **Definition** | A prioritized list of all features, enhancements, and fixes required for the product. | A subset of the Product Backlog, containing items selected for a specific Sprint. |
| **Ownership** | Product Owner | Scrum Development Team |
| **Scope** | Covers the **entire product** lifecycle. | Focuses on **one Sprint** (1-4 weeks). |
| **Contents** | High-level user stories, technical tasks, and business requirements. | Detailed tasks required to complete selected user stories. |
| **Burndown** | Release burndown metric is used. | Sprint burndown metric is used. |
| **Prioritization** | Items are prioritized based on **Business Value (BV)** and stakeholder needs. | Items are selected based on **Sprint Goal** and team capacity. |
| **Changes** | Continuously updated and refined. | Fixed during the Sprint (changes are discouraged). |
| **Estimations** | Business Value (BV) is estimated by stakeholders. Complexity Points (CP) are estimated by developers. | Developers break down tasks and estimate efforts using Story Points (CP). |
| **Meetings** | Daily Scrum Meetings:   * At the end of every Scrum (daily stand-up), all Scrum Developers participate in a **brief** meeting to discuss progress. They answer these three key questions:  1. **What task did you work on in this scrum?** 2. **What task will you work on next scrum?** 3. **Any challenges/impediments? When do you expect to complete the user story?** | Sprint Planning Meetings:   * Before a new sprint begins, all **8 Scrum Developers** gather to determine how many **user stories** they can complete in a **2-week sprint**. * They analyse the **effort required (Complexity Points)** and **business importance (Business Value)** of each task. * Based on these factors, they move selected tasks from the **Product Backlog to the Sprint Backlog** for development in the upcoming sprint. * The team also considers inputs from the **Sprint Retrospective** to improve their planning process. |
| **Example Items** | - User Registration  - Payment Integration  - Order Tracking  - Feedback System | - Develop User Registration (Linesh - CP: 5)  - Set up Payment Gateway (Yogender - CP: 20)  - Implement Order Tracking (Varun - CP: 13) |

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

**Answer:** An **Impediments Log** is a documented list of obstacles, blockers, or issues that hinder the progress of the Scrum team. It is also known as Issue Log or Obstacle Log. The **Scrum Master** is responsible for identifying and resolving these impediments to ensure smooth workflow and project execution.

**Features of Impediments in Scrum:**

Impediments in Scrum are obstacles that slow down or completely block the team's progress. They can be technical, organizational, or resource-related. Below are the key features of impediments:

* **Blocks Team Progress:**
* An impediment prevents or slows down the development team from achieving sprint goals.
* Example: A critical API is not responding, delaying feature implementation.
* **Can Be Internal or External:**
* I**nternal:** Issues within the team, such as a lack of technical expertise.
* **External:** Dependencies on third-party vendors or approvals from management.
* **Requires Scrum Master’s Intervention:**
* The **Scrum Master** is responsible for identifying, tracking, and resolving impediments.
* If an impediment is beyond the Scrum Master’s control, it is escalated to the appropriate stakeholder.
* **Can Be Technical, Process-Related, or People-Oriented:**
* **Technical:** Bugs, software crashes, or infrastructure issues.
* **Process-Related:** Bureaucratic approval delays, unclear requirements.
* **People-Oriented:** Team conflicts, lack of resources, unavailability of key members.
* **Must Be Logged and Tracked:**
* Impediments should be **documented in an Impediments Log** for visibility.
* Example: Using a **Scrum Board** or project management tool like Jira or Trello.
* **Should Be Resolved Quickly:**
* The goal is to **remove impediments as soon as possible** to maintain sprint velocity.
* Example: If a developer is stuck, pairing them with an experienced teammate may help.

**Two Examples of Impediments in Scrum Foods Project:**

* **Resource Unavailability:**
* Example: One of the **Scrum Developers (Yogender)** is on unplanned leave, causing delays in implementing the **payment gateway feature**.
* Impact: The **sprint goal** may not be achieved on time, affecting overall project deadlines.
* **Technical Issues:**
* Example: The **real-time tracking feature** is not working correctly due to integration issues with the GPS API.
* Impact: Delays in development, affecting the **customer experience and delivery efficiency**.

**Impediments Log Template:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Date Reported** | **Impediments Description** | **Impact on Team** | **Priority** | **Owner** | **Resolution Plan** | **Status** |
| 001 | 25/02/2025 | Payment gateway API approval is pending. | Delay in order payment feature. | High | Scrum Master | Follow up with API provider and escalate if needed. | In Progress |
| 002 | 25/02/2025 | Server downtime affecting real-time tracking. | Delivery tracking not working. | High | Dev Team | Investigate server logs, apply fixes. | Open |
| 003 | 24/02/2025 | Yogender (Scrum Developer) on unplanned leave. | Delayed coding of feedback module. | Medium | Scrum  Master | Reassign task to another developer. | Resolved |
| 004 | 23/02/2025 | Confusion in user story acceptance criteria. | Development rework needed. | Low | Product Owner | Clarify requirements in the next stand-up | Resolved |

**Question 8 –** Explain Velocity of the Team. Velocity – How many CP is covered in this sprint **1 Mark**

**Answer:** Velocity refers to the measure of the amount of work, a development team can complete during the sprint. **Velocity** is a key metric in Scrum that measures the amount of work a team can complete in a sprint. It is calculated based on the total **Complexity Points (CP) / Story Points (SP)** completed in a sprint.

**How Velocity Works:**

* Velocity is determined by adding up the **CP of all completed user stories** at the end of a sprint.
* It helps in predicting the **team’s capacity** for future sprints.
* It becomes more reliable after **a few sprints**, as the team establishes a stable working rhythm.

**Velocity Calculation in Scrum:**

Velocity in Scrum is determined based on Story Points (SP) assigned to user stories and completed work during each sprint. It helps in estimating how much work a team can handle in future sprints.

1. Story Point Estimations:

* Before starting the sprint, the Scrum team assigns **Story Points (SP)** to each user story based on:
* **Complexity** (How difficult is the task?)
* **Effort** (How much time/work is required?)
* **Uncertainty** (Are there unknowns or dependencies?)
* Story points are estimated using **Planning Poker** with values like **1, 2, 3, 5, 8, 13, 20, etc.**
* Example:
* User Story 1: "Login Feature" → 3 SP
* User Story 2: "Payment Integration" → 8 SP
* User Story 3: "Order Tracking" → 5 SP

1. Tracking Completed Work:

* At the end of the sprint, the team **reviews completed stories** and only counts those that meet the **Definition of Done (DoD)**.
* If a user story is **partially completed**, it is **not included** in velocity.
* Formula of Velocity: Velocity=∑ (Completed CP per Sprint)
* Example:

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story** | **Estimated SP** | **Status** | **Included in Velocity** |
| Login Feature | 3 SP | Done | Yes |
| Payment Integration | 8 SP | In Progress | No |
| Order Tracking | 5 SP | Done | Yes |

Total Velocity for Sprint1: 3 SP+5 SP= 8 SP

1. Summing Story Points (Sprint-wise Velocity Calculation):

* Velocity is calculated by summing up **all completed** story points at the end of each sprint.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint** | **Total SP Committed** | **Total SP Completed** | **Velocity** |
| Sprint 1 | 15 | 8 | 8 |
| Sprint 2 | 20 | 18 | 18 |
| Sprint 3 | 22 | 20 | 20 |

1. Calculating Average Velocity:

* After multiple sprints, the **average velocity** is calculated to forecast future sprint capacity.
* Average Velocity=Total Sprints∑ (Completed Story Points per Sprint)​/ Total Sprints
* Example: Average Velocity= (8+18+20)/3 = 46/3 = 15.3
* The team can plan approximately 15-16 Story Points per sprint in future sprints.

Why is Velocity Important?

* Helps in **Sprint Planning** to determine how many user stories can be taken.
* Improves **release forecasting** by predicting delivery timelines.
* Tracks **team performance** and identifies potential bottlenecks.

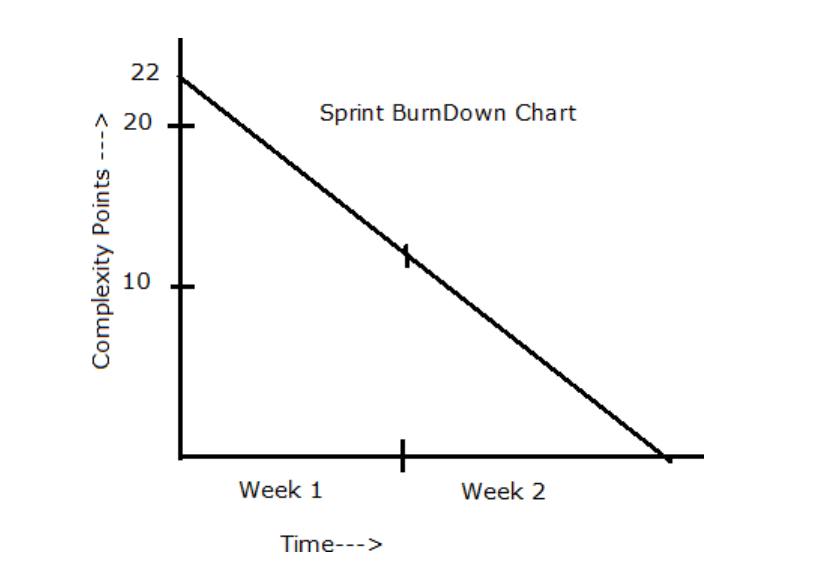
Key Points about Velocity:

* Measured at the **end of each sprint**.
* It **varies** from team to team.
* Only **completed** stories count towards velocity.
* Improves **forecasting** of product release timelines.

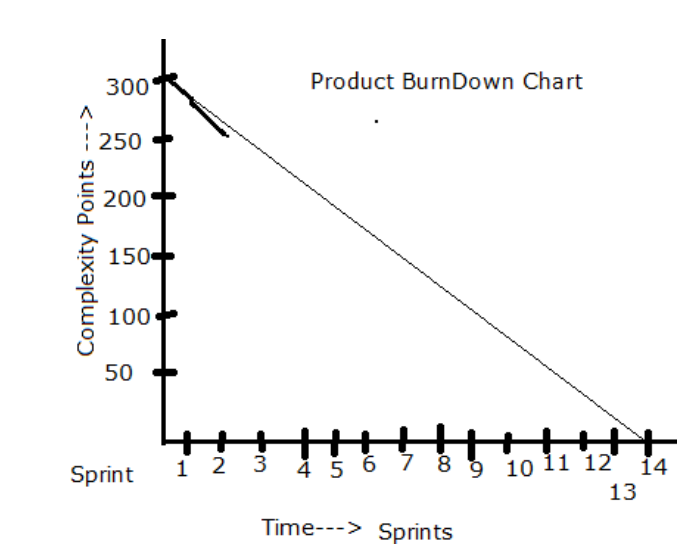
**Question 9 –** Draw Sprint Burn Charts and Product Burn Down Charts **3 Marks**

Answer:

Sprint Burn Down Chart:



Product Burn Down Chart:



**Question 10 –** Explain about Product Grooming **2 Marks**

**Answer:** Product Grooming, also known as Backlog Refinement, is an ongoing process in Scrum where the Product Owner and Scrum Team review, update, and prioritize the Product Backlog items to ensure they are well-defined, estimated, and ready for upcoming sprints.

**Key Activities in Product Grooming:**

* **Adding New User Stories**
* Identifying and adding new user stories based on stakeholder feedback, business needs, or technical improvements.
* **Updating Existing User Stories**
* Refining acceptance criteria, adjusting priorities, and modifying stories to match evolving requirements.
* **Prioritizing the Backlog**
* Arranging stories based on business value, urgency, and dependencies.
* **Estimating Effort (Story Points)**
* Using techniques like Planning Poker to assign Complexity Points (CP) or Story Points (SP) for better sprint planning.
* **Breaking Down Large Stories (Epics to User Stories)**
* Splitting complex epics into smaller, manageable user stories that can be completed within a sprint.
* **Removing Unnecessary Items**
* Eliminating outdated or irrelevant backlog items to keep the backlog clean and relevant.

**When Does Product Grooming Happen?**

* **Before Sprint Planning –** To ensure the backlog is well-prepared for selection in the next sprint.
* Review the Product Backlog – Ensure it is updated with new features, bugs, and improvements.
* Prioritize User Stories – Arrange backlog items based on business value and urgency.
* Check Dependencies – Identify dependencies between user stories that may affect development.
* Set a Clear Agenda – Decide the focus of the session (e.g., new feature refinement, bug triage, prioritization).
* Invite Key Participants – Ensure the Product Owner, Scrum Team, and Scrum Master are available.
* **During the Grooming Session-**
* Clarify User Stories – Ensure each user story follows the **INVEST** criteria:
* **I**ndependent – Can be developed separately
* **N**egotiable – Open for discussion
* **V**aluable – Delivers business value
* **E**stimable – Can be assigned Story Points
* **S**mall – Can be completed within a sprint
* **T**estable – Has clear acceptance criteria
* Define Acceptance Criteria – Add precise success conditions for each user story.
* Estimate Story Points (CP/SP) – Use **Planning Poker** to assign effort levels to stories.
* Break Down Large Stories (Epics → Smaller Stories) – Ensure stories are small enough to be completed in one sprint.
* Remove or Update Old Backlog Items – Delete or modify outdated stories that are no longer relevant.
* Identify Risks & Blockers – Note down any obstacles that might affect development.
* Document Changes – Update the backlog in **JIRA/Trello/Excel** for transparency.
* **After the Grooming Session-**
* Finalize Backlog Priorities – Ensure stories are ordered correctly for the next sprint.
* Confirm Team Alignment – Ensure developers understand upcoming user stories and acceptance criteria.
* Prepare for Sprint Planning – Ensure enough ready-to-develop stories for the next sprint.
* Schedule the Next Grooming Session – Keep backlog refinement continuous (at least once per sprint).

**Who is Involved?**

* Product Owner – Defines and prioritizes backlog items.
* Scrum Master – Facilitates the grooming session.
* Scrum Developers – Provide technical insights and estimate effort.
* Stakeholders (Optional) – Give business input when needed.

**Benefits of Using This Checklist:**

* Keeps backlog clean, well-defined, and ready for development.
* Reduces last-minute confusion in sprint planning.
* Improves team collaboration and efficiency.
* Helps deliver high-quality features for Scrum Foods faster.

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

**Answer:** In Scrum, the Scrum Master and Product Owner have distinct but complementary roles. Both works together to ensure the successful delivery of the product while focusing on different aspects of the development process.

**Scrum Master – The Process Facilitator**

* The Scrum Master ensures the Scrum framework is followed, removes impediments, and helps the team work efficiently.
* Key Responsibilities of a Scrum Master:
* Facilitates Scrum Events – Organizes and moderates Daily Stand-ups, Sprint Planning, Sprint Review, and Retrospective.
* Removes Impediments – Identifies and resolves any roadblocks affecting the development team.
* Coaches the Team – Guides the team on Agile principles, helping them improve efficiency.
* Ensures Scrum Adoption – Educates and enforces best Scrum practices.
* Acts as a Servant Leader – Supports the team instead of managing them, ensuring they are empowered.
* Promotes Continuous Improvement – Encourages feedback and retrospective discussions.
* Example in Scrum Foods:
* If the **delivery tracking feature** is delayed due to a third-party API issue, the Scrum Master will escalate the problem and find a resolution.

**Product Owner – The Value Maximiser**

* The Product Owner (PO) is responsible for defining the product vision, managing the backlog, and ensuring that the development team delivers maximum business value.
* Key Responsibilities of a Product Owner:
* Defines Product Vision – Sets a clear product goal based on customer and business needs.
* Manages the Product Backlog – Creates, prioritizes, and refines user stories.
* Writes Clear User Stories – Ensures stories have clear acceptance criteria.
* Works Closely with Stakeholders – Gathers feedback from business owners, customers, and developers.
* Decides Sprint Priorities – Determines which features should be developed first.
* Ensures Value Delivery – Focuses on building the most valuable features for customers.
* Example in Scrum Foods:
* If customers **complain about long delivery times**, the **Product Owner** prioritizes **route optimization features** in the backlog to improve efficiency.

**Key Differences Between Scrum Master & Product Owner**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Scrum Master** | **Product Owner** |
| Main Focus | Ensuring Scrum processes run smoothly | Maximizing product value |
| Primary Responsibility | Coaching & facilitating | Defining & prioritizing backlog |
| Authority Over Team | No – Acts as a facilitator | Yes – Defines what to build |
| Interaction with Stakeholders | Minimal – More focused on the team | High – Works closely with business owners and customers |
| Decision-Making Power | Guides the team but doesn’t decide on features | Decides the product direction & priorities |
| Handles Impediments | Yes, removes blockers for the team | No, but escalates product-related issues |

**Conclusion:**

* The **Product Owner** ensures the **right product** is built.
* The **Scrum Master** ensures the **product is built the right way** (following Agile principles).

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

**Answer:** In Scrum, several key meetings (also known as Scrum Events) are conducted to ensure smooth project execution. These meetings help in planning, tracking progress, reviewing work, and continuously improving the development process.

1. **Sprint Planning Meeting**

* When: At the beginning of each sprint
* Duration: Time-boxed (Usually 2 hours per week of sprint, e.g., 4 hours for a 2-week sprint)
* Attendees: Product Owner, Scrum Master, Development Team
* Purpose:
* Define the Sprint Goal (What will be achieved in this sprint?)
* Select high-priority Product Backlog Items (PBIs) to be completed
* Break down selected PBIs into smaller tasks
* Assign Story Points (CP/SP) to estimate effort
* Ensure everyone understands their roles and responsibilities
* Outcome: A well-defined Sprint Backlog with clear tasks for the sprint

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

* When: Every day of the sprint
* Duration: Time-boxed (Max 15 minutes)
* Attendees: Development Team, Scrum Master (Product Owner optional)
* Purpose:
* Synchronize team activities and track sprint progress
* Each team member answers three questions:
* What did I do yesterday?
* What will I do today?
* Any blockers or impediments?
* Identify obstacles and resolve them quickly
* Outcome: Everyone is aligned on progress and challenges for the sprint

1. **Sprint Review**

* When: At the end of each sprint
* Duration: Time-boxed (1-2 hours for a 2-week sprint)
* Attendees: Scrum Team, Stakeholders, Clients (optional)
* Purpose:
* Demonstrate completed work to stakeholders
* Gather feedback from stakeholders
* Discuss what went well and any improvements needed
* Update the Product Backlog based on feedback
* Outcome: Stakeholder approval and feedback incorporated for future improvements

1. **Sprint Retrospective**

* When: After Sprint Review, before the next Sprint Planning
* Duration: Time-boxed (1-2 hours)
* Attendees: Scrum Master, Development Team (Product Owner optional)
* Purpose:
* Reflect on the sprint process (What worked? What didn’t?)
* Identify areas for improvement
* Discuss actionable changes for the next sprint
* Improve team collaboration and efficiency
* Outcome: A plan for continuous improvement in future sprints

1. **Backlog Grooming (Product Backlog Refinement)**

* When: Regularly (at least once per sprint)
* Duration: 1-2 hours per session
* Attendees: Product Owner, Scrum Master, Development Team
* Purpose:
* Refine user stories, update descriptions, and add details
* Break down large stories (Epics → User Stories)
* Assign Story Points (CP/SP) for estimation
* Reprioritize the backlog based on business needs
* Remove outdated backlog items
* Outcome: A clean and ready Product Backlog for future sprints

**Summary of Scrum Meetings & Their Purpose**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Meeting Name** | **When** | **Duration** | **Purpose** | **Outcome** |
| Sprint Planning | Beginning of the sprint | 2-4 hours | Define Sprint Goal & select backlog items | Sprint Backlog |
| Daily Scrum | Every day | 15 minutes | Synchronize team progress & remove blockers | Clear daily goals |
| Sprint Review | End of the sprint | 1-2 hours | Showcase work & get stakeholder feedback | Accepted work + feedback |
| Sprint Retrospective | End of the sprint (after Review) | 1-2 hours | Identify areas for improvement | Action plan for improvement |
| Backlog Grooming | Regularly (1-2 times per sprint) | 1-2 hours | Refine backlog & prioritize user stories | Clean & well-defined backlog |

Each Scrum meeting is time-boxed, structured, and has a clear goal to enhance team collaboration, ensure transparency, and deliver high-quality work. By conducting these meetings effectively, the Scrum Foods project can stay on track and continuously improve its processes.

**Question 13 –** Explain Sprint Size and Scrum Size  **2 Marks**

**Answer:**

Sprint Size

* Definition: Sprint Size refers to the total effort (Story Points or Complexity Points) a team commits to completing in a single sprint.
* Determined by:
* The team's Velocity (average Story Points completed in previous sprints)
* Sprint duration (e.g., 2-week or 4-week sprint)
* Team capacity and availability
* Example: If a Scrum team has a velocity of 40 Story Points per sprint, then the Sprint Size is around 40 Story Points.

Scrum Size

* Definition: Scrum Size refers to the total effort required for the entire project (sum of all Story Points across all sprints).
* Determined by:
* Summing the Story Points of all Product Backlog Items
* Helps estimate total project duration based on team velocity
* Example: If the total estimated effort for the Scrum Foods project is 500 Story Points, and the team's velocity is 50 Story Points per sprint, the project will take around 10 sprints to complete.

**Key Difference Between Sprint Size & Scrum Size**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Sprint Size** | **Scrum Size** |
| Definition | Effort per sprint | Total effort for the entire project |
| Measured In | Story Points per sprint | Total Story Points of all sprints |
| Used For | Sprint planning | Project estimation |
| Example | 40 Story Points per sprint | 500 Story Points for the project |

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

**Answer:** In Scrum, **DoR (Definition of Ready)** and **DoD (Definition of Done)** help ensure that work items are well-prepared before development starts and are properly completed before being considered "done."

1. Definition of Ready (DoR)

* The Definition of Ready (DoR) ensures that a User Story or task is fully prepared before being taken into a sprint. It means the story has clear requirements, acceptance criteria, and no major blockers.
* Criteria for DoR:
* The user story is well-defined and understood by the team.
* Acceptance criteria are clearly written.
* Dependencies are identified and resolved.
* The story is small enough to be completed within a sprint.
* The story has an estimated Story Point (CP/SP).
* Example in Scrum Foods:

For a feature like “Order Food”, before development starts, the DoR ensures:

* User story describes the order flow.
* Payment methods are defined.
* Acceptance criteria: The user should be able to place an order successfully.

1. Definition of Done (DoD)

* The Definition of Done (DoD) ensures that a User Story or feature is fully completed and meets all quality standards before being considered "done."
* Criteria for DoR:
* Code is written, reviewed, and merged.
* Feature is tested (unit tests, integration tests, acceptance tests).
* No critical defects remain.
* Feature is documented (if necessary).
* Deployed to staging or production environment.
* Approved by the Product Owner.
* Example in Scrum Foods:

For the “Real-time Order Tracking” feature, the DoD ensures:

* Code is implemented and peer-reviewed.
* Feature is tested on mobile and desktop.
* Bug-free and meets all acceptance criteria.
* Successfully deployed and approved by stakeholders.

**Key Difference Between DoR & DoD**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Definition of Ready (DoR)** | **Definition of Done (DoD)** |
| Purpose | Ensures a story is ready for development | Ensures a story is fully completed |
| Focus | Before development starts | After development ends |
| Criteria | Well-defined, clear scope, dependencies resolved | Coded, tested, reviewed, deployed |
| Example | User story is written, acceptance criteria defined | Feature is bug-free, deployed, and approved |

**Conclusion:**

* DoR ensures that work is well-prepared before development starts.
* DoD ensures that completed work meets quality standards before release.

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

**Answer:** In Scrum, prioritization techniques help the team decide which features or user stories should be developed first based on business value, urgency, and complexity. The **Minimum Viable Product (MVP)** helps deliver the most essential features first to validate the product idea.

1. **Prioritization Techniques:** Prioritization helps **focus on high-value features** first, ensuring better business impact. Here are the most commonly used prioritization techniques:

* **MoSCoW Method:**
* Cate gories:
* **M** – Must Have (Critical features that are non-negotiable)
* **S** – Should Have (Important but not urgent)
* **C** – Could Have (Nice to have, but not necessary)
* **W** – Won’t Have (Not required for now but can be considered later)
* Examples **in Scrum Foods:**
* **Must Have:** User login, order placement, secure payment
* **Should Have:** Order tracking, restaurant reviews
* **Could Have:** Loyalty points, social media integration
* **Won’t Have:** Augmented Reality menu preview
* **Kano Method:**
* Categories:
* **Basic Needs (Must-Have) –** Features users expect (e.g., order food, make payments)
* **Performance Needs (Should-Have) –** Features that improve experience (e.g., faster delivery, order tracking)
* **Delight Needs (Could-Have) –** Features that surprise users (e.g., AI-based food recommendations)
* Examples **in Scrum Foods:**
* **Basic Need:** Users should be able to place an order.
* **Performance Need:** Food should be delivered within 30 minutes.
* **Delight Need:** AI-based suggestions for food based on user preferences.
* **WSJF (Weighted Shortest Job First):**
* Formula:
* WSJF = [Business Value+ Time Criticality+ Risk Reduction]/ Effort (Story Points)
* Higher WSJF Score → Higher Priority
* Examples **in Scrum Foods:**
* Feature A: Live Order Tracking (High business value, medium effort)
* Feature B: Multiple Payment Methods (Lower business value, high effort) **Feature A will be prioritized first due to better WSJF score.**

1. **Minimum Viable Product (MVP):** The **Minimum Viable Product (MVP)** is the **first version of a product** with just enough features to test the business idea and gather feedback from early users.

* Key Aspects of MVP:
* Smallest set of **Must-Have** features
* Quickly released to the market for testing
* Helps in validating assumptions with real users
* Reduces **time, cost, and risk** before full development
* Example of MVP in Scrum Foods:

1. MVP Version 1:

* User registration & login
* Viewing restaurant menus
* Placing an order
* Secure online payment
* Order confirmation notification

1. Future Versions (After MVP Success):

* Order tracking
* Delivery boy live location
* AI-based food recommendations
* Subscription plans

|  |  |  |
| --- | --- | --- |
| **Concept** | **Purpose** | **Example in Scrum Foods** |
| MoSCoW Method | Categorizes features into Must/Should/Could/Won’t Have | Must: Order Placement, Should: Reviews, Could: Loyalty Points |
| Kano Method | Prioritizes based on user expectations | Basic: Payments, Performance: Fast Delivery, Delight: AI Recommendations |
| WSJF | Prioritizes based on value vs. effort | Feature A (Live Tracking) over Feature B (Multiple Payment) |
| MVP | Minimum features for early release & testing | Basic order system, payments, and restaurant menu |

**Conclusion:**

* Prioritization techniques ensure we build the most valuable features first.
* MVP helps launch the product quickly and refine it based on user feedback.

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

**Answer:**

|  |  |  |
| --- | --- | --- |
| Aspect | Business Analyst (BA) | Product Owner (PO) |
| Role | Gathers, analyzes, and documents business requirements. | Owns the product vision, prioritizes backlog, and ensures value delivery. |
| Primary Focus | Understanding business needs and translating them into functional requirements. | Maximizing product value and ensuring development aligns with business goals. |
| Stakeholder Interaction | Works closely with stakeholders to gather and refine requirements. | Collaborates with stakeholders and developers to set priorities. |
| Backlog Management | May assist in backlog refinement but does not own it. | Responsible for maintaining and prioritizing the Product Backlog. |
| Decision Making | Provides recommendations based on business analysis. | Makes final decisions on features, scope, and priorities. |
| Presence in Agile Team | May or may not be part of the Scrum team. | Always a core part of the Scrum team. |
| Example in Scrum Foods | Analyzes customer needs, defines restaurant registration flow. | Decides whether to prioritize real-time tracking or AI recommendations. |

**Role of the Business Analyst (BA):**

* **Requirement Gathering:**
* Conducts market research to see if AI recommendations improve customer engagement.
* Interviews customers, restaurants, and delivery partners to understand preferences.
* Documents business needs:
* Should recommend food based on order history.
* Should consider trending dishes in the user’s location.
* Should personalize based on dietary preferences.
* Creating Detailed Requirements:
* Defines how the AI system will analyze past orders.
* Outlines possible integration with restaurant promotions.
* Suggests possible UI/UX designs for recommendation placement.
* Stakeholder Collaboration:
* Discusses feasibility with developers.
* Ensures compliance with business goals and technical constraints.

**Role of the Product Owner (PO):**

* **Prioritization & Business Value Analysis:**
* Reviews the BA’s findings and evaluates how **AI recommendations** impact business goals.
* Uses **prioritization techniques (MoSCoW, WSJF)** to compare AI recommendations vs. other features.
* Decides:
* AI recommendations have high complexity (CP = 13).
* **Live Order Tracking** has higher immediate business value (CP = 8)
* Prioritizes **Live Order Tracking first**, moving AI recommendations to a later sprint.
* Backlog Management:
* Updates the **Product Backlog** with the AI recommendation feature.
* Writes **User Stories** for development:
* "As a customer, I want to see personalized food suggestions based on my past orders so that I can quickly decide what to order."
* Defines **Acceptance Criteria** (e.g., must display top 3 suggestions, update based on user preferences).
* Decision-Making & Sprint Planning:
* Works with Scrum Master and development team to break down work.
* Reviews feature implementation and ensures it meets business goals before release. The **Product Owner (PO)** **decides when and how** to implement it.

Conclusion:

* A **Business Analyst** helps define **what** should be built by analysing requirements. The **Business Analyst (BA)** **analyses and defines** what’s needed.
* A **Product Owner** decides **what gets built first** by managing priorities.

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

Answer:

**[Sramana Bhattacharya]**  
[xyz@gmail.com] | [+91 9876543210] | [LinkedIn Profile] | [Bangalore, India]

**Professional Summary**

Results-driven **Product Owner** with **3 years of experience** in Agile environments, specializing in **product backlog management, stakeholder collaboration, and sprint planning**. Adept at translating business needs into actionable user stories to deliver high-value products. Proven expertise in **Scrum, Kanban, and SAFe frameworks** with a focus on **customer satisfaction and business growth**.

**Key Skills & Competencies**

* Product Backlog Management
* Agile & Scrum Methodologies
* Stakeholder Management
* User Story & Acceptance Criteria Writing
* Prioritization Techniques (MoSCoW, WSJF, RICE)
* Sprint Planning & Execution
* Data-Driven Decision Making
* Market & Competitive Analysis
* Roadmap & Strategy Development
* Jira, Confluence, Trello, Azure DevOps

**Work Experience**

**Product Owner**  
[Xipro Pvt Ltd] | [Bangalore] | [Feb/2022] – Present

* Led end-to-end **product development lifecycle** for a [describe product, e.g., “food delivery platform”] used by [number] customers.
* Collaborated with **cross-functional teams (developers, QA, UX/UI, business stakeholders)** to prioritize and refine the **product backlog**.
* Defined and maintained **user stories, acceptance criteria, and product roadmap** in **Jira/Confluence**.
* Managed **Sprint Planning, Backlog Grooming, and Retrospectives**, ensuring smooth delivery of high-value features.
* Conducted **customer research and competitor analysis** to identify product opportunities, increasing customer retention by **30%**.
* Worked closely with **Scrum Masters & Business Analysts** to streamline Agile processes.
* Used **KPIs, customer feedback, and analytics** to optimize product features, leading to a **25% increase in user engagement**.

**Education**

**Bachelor’s Degree in Mechanical engineering**  
[ABCD University] | [2020]

**Certifications & Training**

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

**Projects**

**Project: Scrum Foods (Online Food Delivery App)**

* Defined and prioritized the **product backlog**, focusing on **MVP launch**.
* Implemented **real-time order tracking, AI-based recommendations**, and seamless payment integration.
* Achieved **25% faster delivery time** by optimizing the order fulfilment workflow.

**Tools & Technologies**

* Jira, Confluence, Trello, Azure DevOps
* Google Analytics, Tableau, Power BI (for data insights)
* Figma, Adobe XD (for wireframes & UX collaboration)
* SQL (Basic knowledge for data-driven decisions)

**Soft Skills**

* Strong Communication & Negotiation
* Leadership & Decision-Making
* Analytical & Problem-Solving
* Customer-Centric Approach
* Team Collaboration & Conflict Resolution

**Languages**

* [English] (Fluent)
* [Hindi] (Fluent)
* [Bengali] (Fluent)