

Question 1 – Write Agile Manifesto – 8 Marks

What is Agile?

Agile refers to a methodology based on the ability to respond to change rapidly. This groundbreaking approach to project management is gaining popularity. First emerging as a development method in the software development space, Agile Methodologies are now being converted to fit operations in various other industries.

The 4 Values of Agile

Agile Methodology is built on four basic values, which are outlined in the Agile Manifesto, first published in 2001. These values act as overarching principles to which each Agile team should adhere.

“Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan”

The four values demonstrate the importance of functionality over structure. Ultimately, each team member, each project, and each client will operate differently. Thus, Agile harnesses strengths for improved results, instead of relying on preset processes to complete work.

The 12 Principles of Agile

In addition to the four Agile values, the Agile Manifesto outlines 12 principles every Agile team member, especially the project manager, should follow.

1. Our highest priority is to 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 for 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

Write a minimum of 40 User stories and their Acceptance Criteria, along with their BV and CP

User Story No:1	Task:2	Priority:HIGHEST
Value Statement: AS A CUSTOMER I WANT TO LOGIN TO THE APPLICATION		

SO THAT I CAN BROWSE	
BV:1000	CP:2
Acceptance Criteria: Login page, User ID and password field, login option Basic Flow: Open the application Enter the credentials (User ID, Password) Click login Homepage is shown Alternative Flow: Incorrect password or user id Show error message Prompt to re-enter credentials Exceptional Flow: Showing network error Multiple incorrect credentials leads to lock the account	

User Story No:2	Task:2	Priority::HIGHEST
Value Statement: AS A CUSTOMER I WANT TO REGISTER IN THE APPLICATION SO THAT I CAN USE THE APPLICATION		
BV:1000		CP:2
Acceptance Criteria: Registration Screen		

Text Boxes for User Name, Password, Nation ID, Mobile No, Email, Address, Phone Number.

Click on Register Button.

Send Successful Notification to the user

Basic Flow:

Open login page

Click register

Enter Name, phone number, address, email ID, and other details

Create user id and password

Click submit

Alternative Flow:

Show empty fields and prompt enter details

Invalid character in fields

Fill mandatory field

Expectational Flow:

Show User already exists

User Story No:3	Task:3	Priority::HIGHEST
Value Statement: AS A DELIVERY BOY I WANT VIEW THE ORDER SO THAT I CAN ACCEPT TO DELIVER THE ORDER		
BV:1000		CP:2
Acceptance Criteria: View Order, Display List of orders in the tabular Form Basic Flow: Open order Show the assigned order Click to view the order details		

Accept the order
Update the order status like pickup,out for delivery
Deliver the order
Update order delivered

Alternative Flow:

Not able to accept or update the order

Expectational Flow:

Show no order available
Show order order cancelled
Unable to load order

User Story No:4	Task:3	Priority::HIGHEST
Value Statement: AS A RESTAURANT OWNER I WANT TO VIEW THE ORDER SO THAT I CAN TAKE ORDERS FROM CUSTOMER		
BV:1000		CP:2
Acceptance Criteria: View Order, Display List of orders in the tabular Form Basic Flow: Open order and view Prepare the order Update the order Deliver boy pickup the order Alternative Flow: Order delayed No order available		

Exceptional Flow:

Order cancelled

Unable to load order

User Story No:5	Task:3	Priority::HIGHEST
Value Statement: AS AN ADMIN I WANT TO UPDATE OR REMOVE THE RESTAURANT OR DELIVERY BOY SO THAT I CAN MANAGE RESTAURANT AND DELIVERY BOY		
BV:500		CP:2
Acceptance Criteria: Basic Flow: Open application Enter credentials Click restaurant or delivery boy details Add,remove or edit delivery boy Alternative Flow: Instead of edit details were removed Showing error Not able to update details Exceptional Flow: Active account deactivated Assign multiple restaurant to one delivery boy		

User Story No:6	Task:3	Priority::HIGHEST
-----------------	--------	-------------------

Value Statement: AS A CUSTOMER I WANT TO SEARCH FOR FOOD SO THAT I CAN ORDER IT	
BV:1000	CP:3
Acceptance Criteria: Basic Flow: Click search bar Enter food and search Showing related details Alternative Flow: If not available show other suggestions Alternative restaurant or food Other relative menu Show no food Exceptional Flow: Show error Food not found Unrelated food	

User Story No:7	Task:3	Priority:LOW
Value Statement: AS A USER I WANT TO WRITE REVIEW SO THAT I CAN RATE THE EXPERIENCE		
BV:200	CP:1	

Acceptance Criteria:

Basic Flow:

Write the review

Rate the food by selecting stars

Submit

Alternative Flow:

Show error

Expectational Flow:

Review deleted

Review already uploaded

User Story No:8	Task:3	Priority::MEDIUM
Value Statement: AS A BUSINESS OWNER I WANT TO VIEW PAYMENT SO THAT I CAN UPDATE PAYMENT FOR RESTAURANT		
BV:200		CP:1
Acceptance Criteria: Basic Flow: Open payment View,edit, the payment details Update it to complete the process Alternative Flow: Show failed Show error Exceptional Flow:		

Instead of edit details were removed

User Story No:9	Task:2	Priority:MEDIUM
Value Statement: AS A REGIONAL ADMIN I WANT TO VIEW THE FEEDBACK SO THAT I CAN MANAGE CUSTOMER FEEDBACK		
BV:200		CP:2
Acceptance Criteria: Basic Flow: Click feedback Open feedback View , collect details update it in report Manage the feedback Alternative Flow: Show error Expectational Flow: No feedback available Loading feedback		

User Story No:10	Task:2	Priority:MEDIUM
Value Statement: AS A CUSTOMER I WANT TO VIEW TRACK THE ORDER		

SO THAT I CAN SEE WHERE ORDER IS	
BV:500	CP:2
Acceptance Criteria: Basic Flow: Open order Click track order Showing order movement and status Alternative Flow: Showing error Showing delay Exceptional Flow: Loading page Try again Showing not able to track	

User Story No:11	Task:2	Priority::HIGHEST
AS A USER I WANT TO DIFFERENT PAYMENT METHOD SO THAT I CAN CHOOSE ANY MODE		
BV:1000		CP:2
Acceptance Criteria: Display payment modes, radio buttons to select payment modes, payments button. Business Rule. Can select only one payment mode Basic Flow: Click payment Show multiple payment methods		

Click credit card
Complete the payment

Alternative Flow:

Show error
Loading

Exceptional Flow:

All payment methods are not showing
Not able to proceed

User Story No:12	Task:2	Priority:LOW
Value Statement: AS A RESTAURANT OWNER I WANT TO VIEW THE REVIEW SO THAT I CAN IMPROVE THE SERVICE		
BV:200		CP:1
Acceptance Criteria: Basic Flow: Open review View the review Collect details update the report Reply the review Alternative Flow: Show error Try again Expectational Flow: Not able to reply to the review		

No reviews shown

User Story No:13	Task:2	Priority:LOW
Value Statement: AS A CUSTOMER I WANT TO VIEW THE REVIEW SO THAT I CAN DECIDE THE ORDER BASED ON IT		
BV:200		CP:1
Acceptance Criteria: Basic Flow: Open review View,read review Like the review Alternative Flow: Showing error Expected Flow: No reviews yet		

User Story No:14	Task:3	Priority::HIGHEST
Value Statement: AS A DELIVERY BOY I WANT TO VIEW ORDER PICKUP AND DELIVERY SO THAT I CAN SELECT AND ACCEPT THE ORDER		

BV:500	CP:2
Acceptance Criteria: View Order, Display List of orders in the tabular Form Basic Flow: Open order View, select order Click accept order Click confirm Alternative Flow: No order available Order cancelled Expectational Flow: Not able to accept or select order	

User Story No:15	Task:2	Priority::HIGHEST
Value Statement: AS A DELIVERY BOY I WANT TO UPDATE STATUS SO THAT THE USER CAN KNOW THE ORDER UPDATES		
BV:500		CP:2
Acceptance Criteria: Basic Flow: Click the order Edit details like pickup,out for delivery, delivered Click update and submit Alternative Flow:		

Showing error
Try again

Exceptional Flow:

Not showing the order
Show no order available

User Story No:16	Task:2	Priority:MEDIUM
Value Statement: AS A RESTAURANT OWNER I WANT TO VIEW THE REVENUE GENERATED THROUGH SCRUM FOOD APPLICATION SO THAT I CAN KNOW BUSINESS SALES THROUGH THE APPLICATION		
BV:200		CP:2
Acceptance Criteria: Select Reports Select Revenue Reports Select to and from date Select Region (can select all) Generate Report Download Report in EXCEL Basic Flow: Login to application View payment and revenue Update the report Logout Alternative Flow: Show no details Try again Show error Exceptional Flow:		

Not able to access the revenue
Show no details

User Story No:17	Task:2	Priority:MEDIUM
Value Statement: AS A RESTAURANT OWNER I WANT TO VIEW THE DELIVERY BOY DETAILS SO THAT I CAN VERIFY THE DELIVERY BOY BEFORE ACCEPTING HIM		
BV:200		CP:1
Acceptance Criteria: Basic Flow: Open order Open delivery boy details View the review Accept the delivery boy Alternative Flow: Show error Try again Show alternative delivery boy details Exceptional Flow: Show no details Show no delivery boy available		

User Story No:18	Task:2	Priority:MEDIUM
AS A DELIVERY BOY		

I WANT TO VIEW THE PAYMENT SO THAT I CAN KNOW WHETHER IT IS A COD OR ONLINE PAYMENT	
BV:100	CP:1
Acceptance Criteria: Basic Flow: Open order Open payment View payment methods See the details and amount of COD Close the application Alternative Flow: Show no details Show error Show other payment methods Exceptional Flow: Show no payment Try again	

User Story No:19	Task:3	Priority:MEDIUM
Value Statement: AS A RESTAURANT OWNER I WANT TO RAISE THE ISSUE SO THAT I CAN LET THE ADMIN KNOW AND RESOLVE THE ISSUE		
BV:200		CP:2
Acceptance Criteria: 1)BR-ALL MANDATORY 2) TEXT BOX FIELDS		

3)DISPLAY ORDER ID
4) TEXT BOX,FOR DESCRIPTION
5) SUBMIT BUTTON
6)GENERATE ISSUE ID
7)DISPLAY SUCCESSFUL

Basic Flow:

Open application
Click raise issue
Enter the details
Click submit

Alternative Flow:

Show error
Try again
Send email

Exceptional Flow:

Show already uploaded

User Story No:20	Task:3	Priority:MEDIUM
Value Statement: AS A CUSTOMER I WANT TO VIEW THE MENU SO THAT I CAN SELECT THE FOOD		
BV:500		CP:2
Acceptance Criteria: Basic Flow: Open application search restaurant Click menu View the menu		

Select the food
Place order

Alternative Flow:

Show error
Try again
Search by food name

Exceptional Flow:

Menu not available
Show no details

User Story No:21	Task:2	Priority:MEDIUM
Value Statement: AS A USER I WANT TO LOGOUT SO THAT I CAN CLOSE THE APPLICATION ONCE THE ORDER IS COMPLETED		
BV:500		CP:2
Acceptance Criteria: User can see a Logout button when logged in. Clicking Logout ends the session. User is redirected to the login page after logout. User cannot access dashboard after logging out (even using back button). Logout works for all user roles. A message is shown: "You have been logged out successfully." (optional) Basic Flow: Once completed and received order Or browse the application Click the logout Application will be closed Redirect to login page Alternative Flow:		

Show error
Try again

Exceptional Flow:

Network error
Logout failed

User Story No:22	Task:2	Priority:LOW
Value Statement: AS A DELIVERY BOY I WANT TO VIEW FEEDBACK SO THAT I CAN IMPROVE THE SERVICE		
BV:200		CP:1
Acceptance Criteria: Basic Flow: Open review View,read review Like the review Reply the review Alternative Flow: Showing error Exceptional Flow: No reviews yet		

User Story No:23	Task:2	Priority:MEDIUM
Value Statement: AS A DELIVERY BOY I WANT TO VIEW THE DELIVERY REPORT AND REVENUE GENERATED SO THAT I CAN TRACK MY DELIVERY PERFORMANCE AND EARNING THROUGH THE APPLICATION		
BV:100		CP:1
Acceptance Criteria: Select Reports Select Revenue Reports Select to and from date Select Region (can select all) Generate Report Download Report in EXCEL Basic Flow: Delivery boy logs into the system. Navigates to the "Delivery Report" or "My Performance" section. Selects a date range (e.g., this week, this month). System displays: List of deliveries (with status: completed, pending, cancelled) Revenue generated per delivery Total revenue for the selected period Delivery boy reviews the data. (Optional) Clicks "Download" to export the report. Alternative Flow: Search by filter Filter by date Filter by delivery Filter by status Exceptional Flow: Network issue No reports found		

User not logged in

User Story No:24	Task:3	Priority:MEDIUM
Value Statement: AS AN ADMIN I WANT TO VIEW THE RAISED ISSUE SO THAT I CAN RESOLVE THEM		
BV:100		CP:1
Acceptance Criteria: Basic Flow: Admin logs into the system. Navigates to the “Issue Tracker” or “Support Requests” section. System displays a list of raised issues with the following details: Issue ID Date and time reported Reporter (user name/role) Issue category (e.g., delivery, payment, app error) Status (e.g., new, in-progress, resolved) Admin selects an issue to view full details and any attached notes or screenshots. Admin assigns the issue to a team or takes action directly. Admin updates the issue status (e.g., to “In Progress” or “Resolved”). Alternative Flow: Filter by report or issue Filter by date Exceptional Flow: No reports found Unauthorised access		

User Story No:25	Task:2	Priority:MEDIUM
Value Statement: AS A REGIONAL ADMIN I WANT TO VIEW REGIONAL RESTAURANT AND DELIVERY BOYS SO THAT I CAN MANAGE AND UPDATE THEM		
BV:100		CP:1
Acceptance Criteria: List of delivery boy and restaurant,select Restaurants, verify restaurant details, approve button, reject button, notification to the restaurant. Basic Flow: Regional Admin logs into the system. Navigates to the "Regional Dashboard". Selects the "Restaurants" tab: Sees a list of restaurants in their assigned region. Each entry includes name, location, contact, status (active/inactive), etc. Selects the "Delivery Boys" tab: Sees a list of delivery boys assigned to the region. Includes name, assigned restaurant, contact, availability, rating, etc. Clicks on any restaurant or delivery boy to: View full profile Edit details (e.g., contact info, status, assignment) Enable/disable or reassign as needed. Saves any updates. System confirms updates and reflects them immediately. Alternative Flow: Search and filter Export report Exceptional Flow: No data found Show error Unauthorised access		

User Story No:26	Task:3	Priority:MEDIUM
Value Statement: AS A CUSTOMER I WANT TO CANCEL THE ORDER SO THAT I CAN AVOID UNWANTED PAYMENT AND ORDER		
BV:100		CP:2
Acceptance Criteria: BR-ALL MANDATORY TEXT BOX FIELDS DISPLAY ORDER ID TEXT BOX,FOR DESCRIPTION SUBMIT BUTTON GENERATE ISSUE ID DISPLAY SUCCESSFUL Basic Flow: Open order Click cancel the order below the order details Give reasons for cancellation Confirm the cancellation Get confirmation message Alternative Flow: Cancel before placing order Cancel scheduled order Exceptional Flow: Order delivered Cancel not allowed Show error		

User Story No:27	Task:2	Priority:MEDIUM
Value Statement: AS A CUSTOMER I WANT TO VIEW ALL THE RESTAURANTS SO THAT I CAN SELECT ANY RESTAURANT I WANT		
BV:100		CP:1
Acceptance Criteria: Basic Flow Customer opens the app or website and logs in (if needed). Navigates to the "Restaurants" or "Browse" section. The system fetches and displays a list/grid of all available restaurants based on location. Each restaurant shows: name, rating, cuisine type, delivery time, offers, and open/closed status. Customer scrolls through or browses the list. Clicks on a restaurant to view its menu and details. Proceeds to place an order Alternative Flow: Use filter option Search by name Search by location Exceptional Flow: Unauthorised access Show error No data found Restaurant temporarily unavailable Network issue		

User Story No:28	Task:3	Priority:HIGH
Value Statement:		

AS A CUSTOMER

I WANT TO ORDER FOOD

SO THAT I CAN ENJOY THE FOOD

BV:500

CP:2

Acceptance Criteria:

Basic Flow:

Customer logs into the app or website.

Navigates to the "Restaurants" section.

Selects a restaurant from the list.

Browses the menu and adds desired items to the cart.

Reviews the order in the cart:

Items

Quantity

Price

Special instructions (if any)

Proceeds to checkout.

Enters or confirms delivery address and selects payment method.

Clicks "Place Order".

System processes the order and shows confirmation with:

Estimated delivery time

Order tracking option

Customer receives real-time updates and eventually the delivery.

Alternative Flow:

Schedule order for later

Apply coupon

Reorder from the past order

Exceptional Flow:

Payment failed

Out of stock

Restaurant closed

Delivery not available

Network issue

User Story No:29	Task:2	Priority:MEDIUM
Value Statement: AS A REGIONAL ADMIN I WANT TO VIEW THE ISSUE SO THAT I CAN RESOLVE THEM		
BV:100		CP:1
Acceptance Criteria: Basic Flow: Regional Admin logs into the system. Navigates to the "Issues" or "Support Requests" section. System displays a list of issues specific to the admin's region, including: Issue ID Raised by (e.g., restaurant owner, customer) Date/time Status (e.g., New, In Progress, Resolved) Category (e.g., delivery issue, system bug, payment issue) Admin clicks on a specific issue to view full details. Admin adds comments, updates the status (e.g., moves to In Progress), and assigns the issue to a relevant team or resolves it directly. Once resolved, the admin marks the issue as Resolved and optionally adds a resolution note. The reporting party is notified of the resolution. Alternative Flow: Filter and search report Export report Exceptional Flow: Unauthorised access Update failed No reports found Show error		

--

User Story No:30	Task:2	Priority:MEDIUM
Value Statement: AS A REGIONAL ADMIN I WANT TO VIEW THE PAYMENT MADE THE REGIONAL RESTAURANT SO THAT I CAN UPDATE THE REPORT		
BV:100		CP:1
Acceptance Criteria: Basic Flow: Regional Admin logs into the admin portal. Navigates to the "Payments" or "Restaurant Finance" section. System displays a list of restaurants in the admin's region. Admin selects a restaurant to view its payment history. System displays: Payment date Amount paid Payment status (e.g., Completed, Pending) Payment method (e.g., bank transfer, digital wallet) Invoice or reference number Admin reviews the information and exports or copies relevant details for reporting. Admin updates internal or external reports accordingly. Alternative Flow: Filter and search payment Download payment report View summary report Exceptional Flow: No reports found Show error		

unauthorised access

User Story No:31	Task:2	Priority:MEDIUM
Value Statement: AS AN ADMIN I WANT TO VIEW THE DELIVERY BOY SO THAT I CAN VERIFY,HIRE AND UPDATE HIM		
BV:50		CP:1
Acceptance Criteria: List of delivery boy ,select delivery boy, verify delivery boy details, approve button, reject button, notification to the delivery boy. Basic Flow: Admin logs into the admin panel. Navigates to the "Delivery Personnel Management" section. System displays a list of delivery boys with key information: Name Contact info Status (e.g., Active, Pending Verification) Region/Zone Assigned restaurants or areas Background check status Admin clicks on a delivery boy's profile to view full details. Admin performs one or more actions: Verify documents or background check Hire by changing status to "Active" Update profile information (e.g., phone number, zone, vehicle details) Changes are saved, and the delivery boy is notified if applicable Alternative Flow: Filter and search delivery boy Add new delivery boy Deactivate delivery boy		

Exceptional Flow:

Unauthorised access
No data available
Incomplete profile
Duplicate profile

User Story No:32

Task:2

Priority:MEDIUM

Value Statement:

AS AN ADMIN

I WANT TO VIEW THE RESTAURANT AND REGIONAL ADMIN REQUESTS

SO THAT I CAN REVIEW,APPROVE,REJECT,UPDATE AND WORK ON THEM

BV:100

CP:1

Acceptance Criteria:

Basic Flow:

Admin logs into the admin dashboard.

Navigates to the "Requests" section.

System displays two request categories:

Restaurant Requests (e.g., registration, privilege upgrades, profile changes)

Regional Admin Requests (e.g., new admin creation, role updates, region reassignments)

Admin selects a category and views a list of requests with:

Requester name

Type of request

Submission date

Status (Pending, Approved, Rejected)

Admin clicks on a request to view full details and any attached documents.

Admin can approve, reject, or mark for follow-up, optionally adding a comment or reason.

The requester is notified of the decision.

Alternative Flow:

Filter and sort request

Bulk approval

Exceptional Flow:

Duplicate request

Unauthorised access

Show error

Invalid request

User Story No:33

Task:2

Priority:MEDIUM

Value Statement:

AS A RESTAURANT OWNER

I WANT TO LOGOUT

SO THAT I CAN CLOSE THE RESTAURANT

BV:500

CP:2

Acceptance Criteria:

User can see a Logout button when logged in.

Clicking Logout ends the session.

User is redirected to the login page after logout.

User cannot access dashboard after logging out (even using back button).

Logout works for all user roles.

A message is shown: "You have been logged out successfully." (optional)

Basic Flow:

Once completed and delivered order

Or browse the application

Click the logout

Application will be closed

Redirect to login page

Alternative Flow:

Show error

Try again

Exceptional Flow:

Network error

Logout failed

User Story No:34

Task:2

Priority:MEDIUM

Value Statement:

AS A DELIVERY BOY

I WANT TO LOGOUT THE APPLICATION

SO THAT I CAN COMPLETE THE DELIVERY JOB

BV:500

CP:2

Acceptance Criteria:

User can see a Logout button when logged in.

Clicking Logout ends the session.

User is redirected to the login page after logout.

User cannot access dashboard after logging out (even using back button).

Logout works for all user roles.

A message is shown: "You have been logged out successfully." (optional)

Basic Flow:

Once completed and delivered order

Or browse the application

Click the logout

Application will be closed

Redirect to login page

Alternative Flow:

Show error

Try again

Exceptional Flow:

Network error
Logout failed

User Story No:35	Task:2	Priority:MEDIUM
Value Statement: AS AN ADMIN I WANT TO VIEW THE PRIVILEGE OF RESTAURANT SO THAT I CAN APPROVE OR REJECT THEM		
BV:200		CP:1
Acceptance Criteria: List of restaurant,select Restaurants, verify restaurant details, approve button, reject button, notification to the restaurant. Basic Flow: Admin logs into the admin panel. Navigates to the "Restaurant Management" or "Privileges" section. Views a list of restaurants with: Name Region Current privilege level/status (e.g., Pending, Approved, Rejected) Requested privileges (e.g., access to promotions, dashboard features, advanced reports) Clicks on a restaurant to view detailed privilege request. Reviews the request, attached documents (if any), and justification. Clicks "Approve" or "Reject". System updates the privilege status and notifies the restaurant. Admin can optionally add a comment explaining the decision. Alternative Flow: Filter and search Edit details and privilege Exceptional Flow:		

Show error
Unauthorised access
Duplicate request

User Story No:36	Task:2	Priority:MEDIUM
Value Statement: AS A BUSINESS OWNER I WANT TO VIEW THE REPORT SO THAT I CAN KNOW APPLICATION PERFORMANCE		
BV:100		CP:1
Acceptance Criteria: Basic Flow: Business owner logs into the system. Navigates to the "Reports" or "Analytics" section. Selects the type of report (e.g., Sales, Orders, Revenue, Customers, Feedback). Chooses a time range (e.g., Today, This Week, Last Month, Custom Range). Clicks "Generate Report". System displays the report in graphical and tabular format (e.g., charts, KPIs, trends). User reviews the data and optionally: Downloads as PDF/Excel Shares the report Prints the report User logs out or continues working. Alternative Flow: Schedule report Filter report Exceptional Flow: No reports found Unauthorised access Show error		

User Story No:37	Task:2	Priority:LOW
Value Statement: AS AN ADMIN I WANT TO VIEW THE CUSTOMER FEEDBACK SO THAT I CAN UPDATE THE APPLICATION		
BV:50		CP:1
Acceptance Criteria: Basic Flow: Open review View,read review Update the report Alternative Flow: Showing error Exceptional Flow: No reviews yet		

User Story No:38	Task:3	Priority:MEDIUM
Value Statement: AS A REGIONAL ADMIN I WANT TO VIEW CANCELLED ORDER SO THAT I CAN UPDATE THE REFUND		
BV:500		CP:2

Acceptance Criteria:

Basic Flow:

Click the cancelled order
Order details
Filter results and search
Update refund

Alternative Flow:

Auto approved refund

Exceptional Flow:

No cancelled order available
Show error

User Story No:39	Task:3	Priority:HIGH
Value Statement: AS A DELIVERY BOY I WANT TO REGISTER TO THE APPLICATION SO THAT I CAN DO THE DELIVERY JOB		
BV:1000		CP:3
Acceptance Criteria: Registration Screen Text Boxes for User Name, Password, Nation ID, Mobile No, Email, Address, Phone Number. Click on Register Button. Send Successful Notification to the user Basic Flow: Open login page		

Click register
Enter Name, phone number, address, email ID, and other details
Create user id and password
Click submit

Alternative Flow:

Show empty fields and prompt enter details
Invalid character in fields
Fill mandatory field

Expectational Flow:

Show User already exists

User Story No:40	Task:3	Priority:HIGH
Value Statement: AS A RESTAURANT OWNER I WANT TO REGISTER TO THE APPLICATION SO THAT I CAN DO THE SALES IN THE APPLICATION		
BV:1000		CP:3
Acceptance Criteria: Registration Screen Text Boxes for User Name, Password, Nation ID, Mobile No, Email, Address, Phone Number. Click on Register Button. Send Successful Notification to the user Basic Flow: Open login page Click register Enter Name, phone number, address, email ID, and other details Create user id and password Click submit		

Alternative Flow:

Show empty fields and prompt enter details

Invalid character in fields

Fill mandatory field

Expectational Flow:

Show User already exists

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

Epic

An agile epic is a body of work that can be broken down into specific tasks (called user stories) based on the needs/requests of customers or end-users.

When adopting agile and DevOps, an epic serves to manage tasks. With a task tracking template, teams can organize, prioritize, and monitor their tasks effectively.

Epics are almost always delivered over a set of sprints. As a team learns more about an epic through development and customer feedback, user stories will be added and removed as necessary.

Epics are a helpful way to organize work and to create a hierarchy. The idea is to break work down into manageable pieces so that large projects can get done and one can continue to share value to the customers regularly. Epics help teams break their work down while continuing to work towards a bigger goal.

Epic 1: Login

As a User, I want to login to Scrum Food with my login credentials, so that I can use the application.

As a user, I want to login with my email ID and password ,so that I can use the application

As a user, I want to logout,so that I can close the application

As a user, I want notified when I enter wrong credentials, so that I can re-enter the correct one

As a user, I want forgot password in the login page,so that I can reset the password and use the application

As a user I want to stay logged in in the application until I log out so that I don't have to login frequently

As a user, I want to login page show remember the credentials so that I don't have to enter the credentials every single time

Acceptance criteria:

User can login using the credentials created during registration

User password will be masked while entering in the login page

User can remove the masking option for password

User can log out using logout option

User can reset password using forgot password option

User will be notified while wrong credentials by turning the column red

User will be notified whether password or user id is wrong while entering wrong credentials

User can make the login page save the credentials by clicking remember me

Epic 2: Payment

As a user I want to have multiple payment methods so that I can use any method to complete the order

As a user I want to be notified when the payment is done so that I can know the payment is completed

As a user I want to have OTP for online payment so that I can have a secured payment

As a user I want to save the card details in the application so that I can use it for future transactions

As a user I want to be redirected to the payment options so that I can use other methods to complete the payment

Acceptance criteria:

Error message displayed for failed payment

User can save the card details in the application

User will be notified when the payment is completed

Card details will be masked in column

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

Aspect	Business Value	Complexity Points (Story Points)
Definition	The benefit or impact a feature brings to the business or customer	The effort, difficulty, or uncertainty involved in delivering a feature
Focus	Focused on "What do we gain?"	Focused on "How hard is it to build?"
Who assigns it	Assigned by Product Owner or Business Stakeholders	Assigned by Development Team
Purpose	To prioritize features based on impact	To estimate effort and plan sprints

Measured by	Value metrics like revenue potential, risk reduction, customer satisfaction	Abstract units (e.g., Fibonacci scale: 1, 2, 3, 5, 8, etc.)
Helps with	Prioritization of backlog items	Sprint planning and capacity estimation

Business Value

Any product, business or organization creates a product OR resource that has some “value”. This is the “value” that make lives easy, that produces better products, that establishes the organizations trust, vision and drive business goals.

A business value gives “that value” to end users and businesses to develop a “feature” withstanding the changing market needs, demands, competition and foresight. Given the focal point and decision-making stance, a business value (here after called as “BV”) became a core principle in any methodology a project/ program/ enterprise adopts or implement

The term business value crops up often in agile literature. One of the most common forms of user story templates references it:

As a (role), I want (function) so that (business value).

However, the business value is more than just the closing clause of a user story. The first

And who is that person to maximize the BV?... Well, the Scrum Guide says, it is the “product owner” that steers the BV, measures ROI, drive the features in meeting the BV. Now, this realization about BV drives us towards establishing the value.

A Product Owner’s will establish and maximize the business value through strategic skills of product roadmap, product discovery, vision and tactical skills of preparing the product backlog, prioritizing, ensuring user stories are in place and well written.

Meaningful techniques are employed to customize and deliver the business outcomes

An organization will establish a BV considering these parameters in order to stay competitive and deliver value:

Legal

Stakeholders: what end users needs are and will buy.

Risk/ change management/ capability building: embracing marketplace and IoT.

Process/ service/ product: what the team is considering as excitement.

Cost/ market value.

Some techniques (consolidated) that maximize, measure, assess Business value at a project level, program level and enterprise levels:

Epic- feature-user story-tasks-estimations-Requirement Traceability matrix

Product backlog- slicing-Prioritization-validation

Empathy- personas-Stakeholder management

In Agile development, "BV" typically stands for Business Value. It refers to the worth or benefit that a project or feature brings to the business, often measured in terms of increased revenue, reduced costs, or enhanced customer satisfaction.

Elaboration:

Focus on Value:

Agile development emphasizes delivering value to the end-user and the business, rather than just focusing on feature completion.

Prioritization Tool:

BV helps prioritize work items in the backlog, ensuring that the most valuable items are addressed first.

Decision-Making Guide:

It guides decision-making about which features to develop and which to defer, based on their potential impact on the business.

Communication and Collaboration:

BV facilitates communication and collaboration between development teams, business owners, and stakeholders, ensuring everyone is aligned on the project's goals.

Measuring Success:

BV can be used to track and measure the success of Agile projects by assessing the actual value delivered versus the planned value.

Examples of BV:

Increasing revenue by introducing a new product feature.

Reducing operational costs by automating a business process.

Improving customer satisfaction by enhancing product usability.

Importance in Agile:

BV is a core principle in Agile, driving the vision and goals of Agile projects and ensuring that they are focused on delivering the most valuable results.

MoSCoW Agile Prioritization Technique

- Must– The must requirements is given the topmost priority
- Should– Next priority is given to the requirements that are highly desirable, though not mandatory
- Could– The next priority is given to the requirement that is nice to have
- Won't– And the final consideration is given to the requirements which will not work in the process at that point of time.

Currency Notes Technique

The Currency Notes Technique is a collaborative, hands-on prioritization method used in Agile to assign business value to backlog items, based on the metaphor of distributing money (typically fake currency notes) among features.

Also known as "Buy a Feature" or "Agile Dollar Voting", this technique involves giving stakeholders or team members a fixed amount of virtual currency (e.g., \$100, \$1,000, or tokens) and asking them to "spend" it on the features or user stories they value the most.

It simulates real-world resource constraints, forcing people to prioritize what matters most.

Process

Prepare the Features or Backlog Items

- List features or epics on index cards, sticky notes, or a digital board.
- Each item should have a short description, and optionally, estimated effort (e.g., story points).

Assign a Virtual Budget

- Each stakeholder receives the same amount of play money (e.g., \$100 in notes or tokens).

Spend the Money

- Stakeholders "spend" their money on the items they consider most valuable.

- They can allocate all to one item or divide it among multiple.
- This can be done individually or as a group discussion.

Tally the Results

- Add up the amounts each feature received.
- The features with the highest total funding are considered the highest business value.

Prioritize the Backlog

- Use the results to rank items in the backlog.
- Combine with complexity points or effort for further refinement (e.g., via WSJF).

Complexity Points

In Agile development, complexity points, often called "story points", are a way to estimate the relative effort required to complete a user story or task. They represent a combination of complexity, effort, risk, and uncertainty, rather than a specific time estimate. Story points are used in agile to help teams plan sprints and manage their workload.

Story points are an estimation technique used in Agile project management methodologies to help your team scope the effort required to complete a task. Story points account for factors like task complexity and uncertainty, which makes them more accurate than other estimation techniques such as time-based estimation.

What are story points?

Story points are a way to estimate the amount of effort required to complete a user story in your product backlog. Usually estimate story points before a sprint planning meeting, since that's when the team determines how much work they can carry out in an upcoming sprint.

Story points are units of measure for expressing an estimate of the overall effort required to fully implement a product backlog item or any other piece of work. Teams assign story points relative to work complexity, the amount of work, and risk or uncertainty. Values are assigned to more effectively break down work into smaller

pieces, so they can address uncertainty. Over time, this helps teams understand how much they can achieve in a period of time and builds consensus and commitment to the solution. It may sound counter-intuitive, but this abstraction is actually helpful because it pushes the team to make tougher decisions around the difficulty of work.

Story Point Estimation

Story point estimation is a technique used to estimate the effort required to implement a user story. User stories are short, simple descriptions of a feature or functionality that a user wants. They are often used in Agile software development to break down larger tasks into smaller, more manageable pieces.

The process of story point estimation involves assigning a numerical value to each user story, based on its level of effort. These values are usually represented in points, with larger numbers indicating more effort. The goal is to create a common understanding of the relative complexity of different user stories, and to provide a basis for planning and scheduling work.

One of the key benefits of story point estimation is that it allows teams to estimate the size of a project without getting bogged down in the details of individual tasks. Instead, the team can focus on the project

Planning Poker

Planning Poker is a popular technique for conducting story point estimation. It involves a group of team members, usually including developers, product owners, and project managers. The team works together to assign points to each user story, based on its level of complexity.

The process begins with the product owner presenting a user story to the team. The team then discusses the story, asking questions and clarifying any details that are unclear. Once the team has a good understanding of the story, they each select a card from a deck of cards that represents a point value. The cards typically range from 0 to 100 points, with larger numbers indicating more effort.

After everyone has selected a card, the cards are revealed simultaneously. If there is a wide range of values, the team discusses the reasons for the differences and then

repeats the process until a consensus is reached. The process continues until all the user stories have been estimated.

One of the benefits of Planning Poker is that it encourages collaboration and discussion among team members. By discussing the details of each story and coming to a consensus on the level of effort required, the team develops a shared understanding of the project and the work involved

Purpose

Foster discussion and shared understanding of work.

Leverage team knowledge to estimate complexity, effort, and risk.

Prevent bias or anchoring in estimation.

Who Participates

Development Team (required)

Scrum Master (facilitator)

Product Owner (provides clarification on stories)

Preparation

1. Refined Backlog: Product Owner ensures that backlog items are clearly written and well understood.

2. Estimation Scale: Most teams use Fibonacci sequence (1, 2, 3, 5, 8, 13, 21...), but others like modified Fibonacci or T-shirt sizes are also acceptable.

3. Poker Cards or Apps: Each estimator has a set of cards or access to a digital Planning Poker tool (e.g., PlanningPoker.com, ScrumPoker Online, Jira plugins).

Step-by-Step Process

Select a Story

The Product Owner presents a user story or task from the backlog and gives a brief description, including:

- Acceptance criteria
- Dependencies
- Assumptions
- Risks

Discussion

Team members can ask questions and discuss:

- What needs to be done?
- What are the unknowns?
- Are there testing, integration, or design efforts involved?

Estimation

Each team member privately selects a card that represents their estimate of the effort required.

On cue (e.g., “1-2-3... Show!”), all cards are revealed simultaneously to prevent influence.

Discuss Differences

If estimates differ widely, the team discusses the reasoning:

- Lower estimates might reflect a simpler understanding.
- Higher estimates might reveal hidden complexity or risk.
- The Product Owner may provide further clarification.

Re-estimate

After discussion, the team votes again. This is repeated until a consensus is reached or the team agrees on an average.

Record the Estimate

The agreed-upon story point value is recorded in the backlog management tool

Question 5 –Explain about Sprint– 5 Marks

Sprint

In Agile product development and software project management, a sprint is a set period of time during which specific work has to be completed and made ready for review. Incorporating sprints into the software development lifecycle (SDLC) encourages iterative development and continuous improvement, ultimately helping organizations to produce quality software faster and often at lower cost.

Sprint timebox

In Scrum, a sprint timebox is a fixed period of time, typically between one and four weeks, during which a Scrum team works towards a specific sprint goal. This fixed timeframe, also known as a timebox, helps maintain focus and manage expectations within the team and with stakeholders.

Sprint goal

In Scrum and other agile frameworks, a sprint goal is a concise description of what the team aims to achieve during a specific sprint. It's a high-level objective, not a list of tasks, that provides purpose and direction for the entire team. Sprint goals are typically defined collaboratively by the Product Owner and the development team during sprint planning.

Sprint planning.

Setting up a sprint requires proper sprint planning. This is the project manager's chance to decide on objectives and actions for an upcoming sprint. The project manager gathers the team to determine:

How much work can be completed within one sprint

And how will it be completed

This list of objectives is known as the sprint backlog — an important aspect of sprint project management.

There must be enough work to fill the period, but not too much.

Not planning enough work. Can derail the project and lead to budget and timeline overages.

Planning too much work. Can lead to team burnout and missed deadlines.

Daily Scrum.

Also known as stand-up meetings, daily Scrum meetings ensure sprints are running on schedule and all team members are in the loop when problems pop up. Sprint stand-ups typically only last 15 minutes and require each team member to:

Discuss what they've accomplished since the last meeting

Explain what they'll work on before the next meeting

Detail any obstacles that are standing in their way

Daily stand-ups should be quick touchpoints. If more in-depth meetings are required, they should be scheduled outside of stand-ups.

Sprint workflow and process

The sprint workflow is intended to help team members evaluate their work and communicate with each other throughout the entire process. The workflow is followed for each sprint and usually includes these steps:

Create the product backlog and define the product goal. Create a list of tasks that must be completed before the product is released. The product goal, which describes the future state of the product under development, is included in the product backlog. The backlog is built by the product owner, who provides a list of prioritized items to the Scrum master and Scrum team. The backlog is based on user stories, which focus on features that consider the type of end user, what they want in the product, and why. It is refined as needed during the sprint.

Plan a sprint. The team discusses top-priority user stories and decides what can be delivered in the sprint.

Create a sprint backlog and define the sprint goal. Agreed upon by the entire team, the sprint backlog is a list of items that finalizes and defines what the development team will complete during the sprint. This provides the focus necessary to complete the sprint and its backlog items.

Execute the sprint.

The time frame in which the work defined in the sprint backlog must be completed. Ideally, the period should be long enough to give the team flexibility and freedom to do the work. It should also be short enough to minimize complexity and risk and to facilitate frequent feedback and fast changes.

Hold a daily scrum. Led by the Scrum master, the team comes together for short daily meetings, in which they discuss what they have completed, what they are working on, and any issues that are blocking the work.

Determine the outcome. The outcome of a sprint is a hypothetically usable product. The product owner can decide if the product is ready for release or if additional features are needed.

End sprint.

At the end of a sprint, two meetings are held:

Sprint review.

Once a sprint is completed, the project manager hosts a sprint review meeting with all team members and stakeholders to:

Demonstrate sprint outputs

Determine what was accomplished and what wasn't

Review project forecasts

Untested or incomplete work is not shown, but is instead saved for the next sprint's planning round.

Sprint retrospective.

The final step in the sprint project management process is the sprint retrospective. This takes place after the sprint review and before the next sprint planning session.

This collaborative session allows team members to discuss accomplishments and challenges during the previous sprint so that processes can be altered, if needed. The goal is to fix one thing at a time and make small, incremental changes from sprint to sprint

Sprint roles

A variety of roles are involved in a sprint, with each working on different parts of the process. These roles include the following:

Product owner

This person represents the business or user community and is a liaison between the development team and customers. The product owner is in charge of working with the user group to define, prioritize, and adjust what features the product release should include. They also accept or reject work results and keep customers informed on the project's status.

Scrum master.

This person is the main facilitator for the project's development team. They manage the process for how information is exchanged during the sprint, including leading stand-up meetings and helping the team stay on track by mediating problems and removing obstacles. Their main aim is to observe the sprints, maintain informational transparency during every sprint, provide coaching to the rest of the team, and identify opportunities to improve workflows as a sprint progresses.

Scrum team.

This group of people is responsible for executing the work, specifically, i.e., the product backlog items that must be completed to achieve the sprint goal. The team may also include testers, architects, designers, and personnel from IT operations who work together to complete a set of work items within the time-constrained period of a sprint. While the Scrum master is charged with protecting the team and ensuring that it stays on focus during every sprint, the team itself is self-managed and ultimately responsible for collectively determining how to reach their goals.

Sprint artifacts and ceremonies

In the Agile Scrum methodology, artifacts provide the key information that a Scrum team needs to understand the product under development, as well as completed and planned activities for the project.

Examples of artifacts are the following:

Product backlogs.

Sprint backlogs.

Burndown charts.

Planning meeting increments.

Sprint reviews and retrospectives.

User stories.

Together, these artifacts promote transparency in the sprint and provide useful insights into a sprint's performance. They may be created when the team, product owner and Scrum master are planning the work, setting sprint goals, creating or organizing tasks to achieve those goals, or analyzing results and comparing them to goals.

Ceremonies are the meetings that are held for every sprint. The main participants in these ceremonies are the development team, the product owner, and the Scrum master.

Backlog refinement

Backlog refinement is the process of reviewing, ranking, and editing your product backlog. Backlog refinement is an important tool in your product development process because it helps your development team build only the features and functionalities that the customer wants and the business needs.

A sprint may include many ceremonies, including the following:

Sprint planning meeting.

Daily scrum or daily stand-up meeting.

Sprint review.

Sprint or Agile retrospective.

Question 6 – Explain Product backlog and sprint backlog– 5 Marks

What is a Backlog?

The backlog in product development is the list of actions intended to enhance the features and improve the quality of products. The list contains steps concerning bugs, tasks, user stories and other similar activities that provide well-researched and detailed information on actions that should be taken.

Product Backlog

A product backlog is a prioritized, living document that contains all the features, enhancements, and fixes needed for a product. It's essentially a "to-do" list for the development team, and it's constantly refined and updated as the product evolves. The Product Owner is responsible for managing the backlog, ensuring it reflects the product's vision and priorities. The product, after use, is associated with certain unimaginable or unpredictable issues brought forward after the user experience. It is

aimed to refine the product to make it more user-friendly and accessible depending on the type and level of customers. Various such backlogs are taken for action based on the complexity, the value they contribute to the product, associated finances, and their place in the market. It also considers feasibility, risks, effect on performance, and corporate values.

It is a highly flexible component of the Scrum framework. The reason is owed to market shifts, necessities, budgets, business objectives, and feedback. Agile methodology assists in combating the changes based on such problems. It is also a regular and mandatory process due to the previously mentioned reasons and requirements for product refinement.

Sprint Backlog

A sprint backlog is a dynamic document used in Agile and Scrum project management to define the work a development team will complete during a specific sprint (a time-boxed iteration). It's created at the start of each sprint and consists of product backlog items, tasks, and a plan for delivering the sprint goal. It is a part of the product backlog and is mainly associated with allocating a particular time to complete the sprint, which deals with product refinement. It includes the part that can not be completed during the sprint. The related discussions are done during the sprint planning meetings. The sprint backlog must involve participation and updating information for every member. The sprint backlog can change, unlike the product backlog. The primary aim of the backlog is to maintain smooth functioning and progress of the sprint goals.

Feature	Product Backlog	Sprint Backlog
Definition	A prioritized list of everything needed in the product	A list of tasks selected for delivery in a specific sprint
Scope	Full product scope (long-term)	Sprint scope (short-term)
Owner	Product Owner	Development Team
Content	Epics, user stories, features, bugs, technical tasks	User stories/tasks selected for the sprint, with work breakdown

Changes during Sprint	Can be updated anytime	Should not change after Sprint Planning (unless agreed)
Time Frame	Product lifecycle	One sprint (1–4 weeks typically)
Purpose	To guide overall product development	To deliver the sprint goal
Level of Detail	High-level to medium; refined over time	Detailed tasks are ready for execution
Updated by	Product Owner (with team inputs)	Development Team (daily updates during the sprint)
Examples	"Build user login feature", "Add reporting module"	"Design login UI", "Code API for login", "Write unit tests"

Question 7 – What is an impediments log? Write 2 impediments – 5 Marks

Impediment Log:

An impediments log is a document or record used in Agile methodologies, particularly Scrum, to track and manage obstacles or "blockers" that hinder a team's progress in achieving their sprint goals. It's a systematic way to identify, record, and address issues that prevent the team from completing work. It is an important tool in which all the impediments affecting the project are documented. An Impediment is usually described as an obstacle, hindrance, or hurdle that can decrease the productivity and performance of the Scrum team. They must be identified as soon as possible, a solution found in a quick time, and they should be removed for the team to contribute effectively. They can be classified into two types: Internal and External. Internal Impediments can be classified as either improper communication or reduction in performance of the workforce, whereas External impediments could involve various factors such as the requirement of unnecessary documents or issues in the software license. An organization can suffer from unwanted costs if it fails in identification or does not find an appropriate solution in dealing with this factor. The Scrum Master is responsible for recording the impediments in the Impediment Log, and these issues can be discussed and sorted in Daily Standup Meetings and Sprint Review Meetings.

Impediments

Two resources are in unplanned long leave

Log ID	1
Description	Two resources are on unplanned leave
Impact	Delay in development of product
Priority	Low(minor delay in development)
Assigned to	scrum master
Status	Open
Action taken	sprint scope is extended
Resolution	Other team members are working on part of unavailable resources place

Business stakeholder is not available for clarification

Log ID	2
Description	Business stakeholder is not available for clarification
Impact	Delay in getting clarity in requirements
Priority	Medium (requirements related questions are in pending)
Assigned to	Product owner
Status	In progress
Action taken	Product owner notified by email about the issue
Resolution	Team members working on other requirements

One resource is leaving the project

Log ID	3
Description	One resource is leaving the project
Impact	Delay in project progress. High work pressure on other resources
Priority	High
Assigned to	scrum master
Status	open
Action taken	New resource request raised
Resolution	Other resources are working on the other resource part and new resource request initiated

Question 8 – Explain Velocity of the Team – 1 mark

Velocity

In Scrum and other Agile project management frameworks, velocity serves as an Agile metric used to estimate the amount of work a Scrum team can complete within a specific time frame, typically a single sprint. Velocity is a measure of the amount of work a Team can tackle during a single Sprint and is the key metric in Scrum. Velocity is calculated at the end of the Sprint by totaling the Points for all fully completed User Stories. Velocity is a key feedback mechanism for the Team. It helps them measure whether process changes they make are improving their productivity. It also facilitates very accurate forecasting of how many stories a Team can do in a Sprint. (In Scrum this is called using Yesterday's Weather.) For forecasting purposes the average of the last three Sprint's Velocity should be used. Of course, this means it takes three Sprints of experience for a Team to determine its Velocity accurately, which can sometimes be difficult to explain to impatient stakeholders.

Without Velocity, Release Planning is impossible. By knowing Velocity, a Product Owner can figure out how many Sprints it will take the Team to achieve a desired level of functionality that can then be shipped. Depending on the length of the Sprint, the Product owner can fix a date for the release.

Many factors affect the number of story points a team member can complete during a two-week sprint, including individual experience, task complexity, and team dynamics. New Scrum teams usually average 5–10 story points per person per two-week sprint.

Understanding the team's velocity can help with continuous improvement. It allows teams to forecast future sprints, plan projects, and set realistic goals. This metric helps develop a stable work rhythm, predict project timelines, and manage stakeholder expectations. It is also crucial for effective sprint planning and managing stakeholder expectations.

How to calculate sprint velocity in Scrum

You typically calculate sprint velocity at the end of each sprint by totaling the story points or other units of measurement for all fully completed user stories.

Here is the step-by-step process of how to calculate velocity in Scrum:

Plan a sprint

Before a sprint begins, outline and assign points to all the user stories in the product backlog. For instance:

Assign user authentication: 5 points

Add payment gateway integration: 8 points

Implement search functionality: 3 points

Develop a user profile page: 13 points

Implement email notifications: 2 points

Optimize database queries: 21 points

Create an admin dashboard: 5 points

The team should commit to completing user stories in the upcoming sprint based on the average velocity from previous sprints and other factors, such as holidays or external dependencies. For example, if the average velocity is 15 points with no holidays or external dependencies, the team could commit to user stories totaling around 15 points for the next sprint.

List completed user stories

Create a list of all the fully completed user stories at the end of each sprint. These should be stories that have met their acceptance criteria and that the Scrum Master and Product Owner have approved.

If a user story is 90% done, it is not fully complete. The team should move it to the next sprint and re-evaluate the points based on the remaining tasks.

Check story points

The team should have already assigned story points to each completed user story. If story points need re-evaluation, this is the time to do it.

For instance, let's say the team has completed three user stories in the current sprint—assign user authentication, add payment gateway integration, and implement search functionality. You could assign those tasks with the following story points:

Assign user authentication: 5 points

Add payment gateway integration: 8 points

Implement search functionality: 3 points

Sum points to find velocity

Next, you must total the story points for all the completed user stories. The sum of the story points represents the sprint velocity.

In the above scenario, the total would be 5 points + 8 points + 3 points = 16 points, which is the velocity for this sprint.

Average velocity

Calculating the average sprint velocity over the number of sprints the team completes can provide a more reliable measure for future sprints. This measure benefits newly formed teams or those that have changed in size or structure.

For example, if the velocities for the last three sprints were 14, 16, and 15, the average velocity would be $(14 + 16 + 15) / 3 = 15$ points.

Factors that can affect scrum velocity

Various factors can influence scrum metrics and velocity. Understanding these can help in planning and continuously improving the team's performance.

Team size and skill level

The number of individuals on a development team and their respective skill levels can influence the work the team can complete during a sprint. A larger team can complete

more story points in a sprint. However, more people can lead to high communication overhead and coordination challenges.

Conversely, a small, highly skilled team could outperform a large, less skilled team by efficiently handling complex tasks.

Team stability and experience

When Scrum team members work together for multiple sprints, they'll likely iron out many of the kinks that hinder new teams. They'll have established communication patterns and know who is good at what.

These teams have shared experiences to draw on when problems arise. This familiarity can significantly improve velocity.

Complexity of user stories

A sprint filled with complex stories will usually result in a low velocity. The velocity figure will be misleading if the complexity doesn't accurately reflect the assigned story points.

To maintain a consistent velocity, some teams aim for a balance of "quick win" tasks and more complex tasks within a sprint.

External dependencies and constraints

If your team relies on another team to complete database updates or API integrations and that team runs late, it can directly lower your team's velocity. Being aware of these dependencies and planning for them through effective inter-team communication can mitigate negative impacts on velocity.

Likewise, public holidays or mandatory company events should be factored into sprint planning, as they reduce the available working time.

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

Sprint Burndown Chart

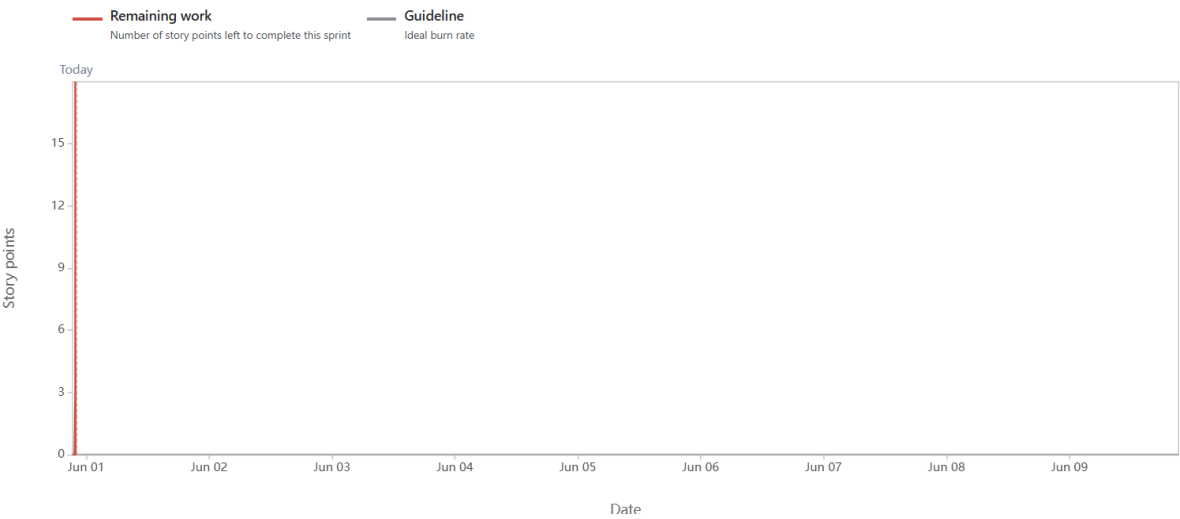
The sprint burndown chart tracks the amount of work remaining in a single sprint, typically lasting between one to four weeks. This type of burndown chart is used to monitor progress during the sprint and helps teams gauge whether they are on target to complete all the tasks by the sprint's end. It is a critical tool for Scrum teams that work in iterative cycles.

Product Burndown Chart

A product burndown chart tracks the total work remaining in a project or across multiple sprints. Unlike the sprint burndown chart, which is limited to a single sprint, the product burndown chart shows the progress of the entire product backlog. This is useful for long-term projects and gives stakeholders a broader view of when the project will likely be completed.

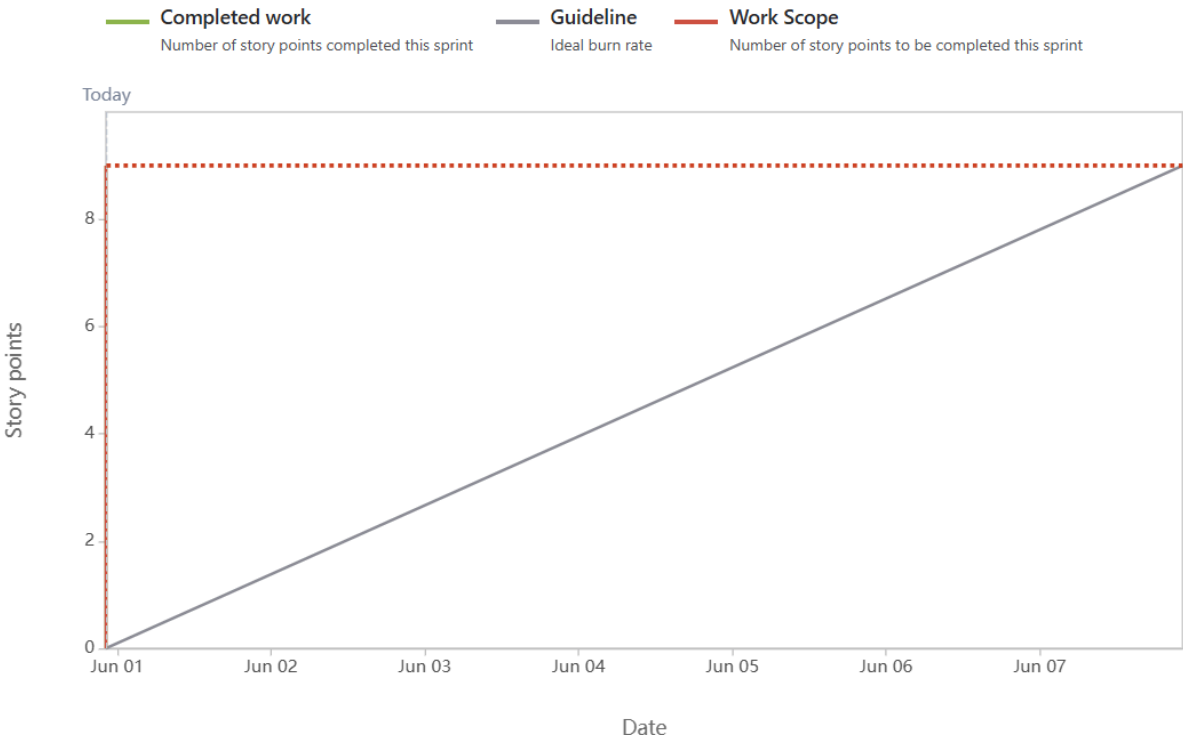
Date - May 31st, 2025 - June 9th, 2025

Sprint goal - create registration for the application

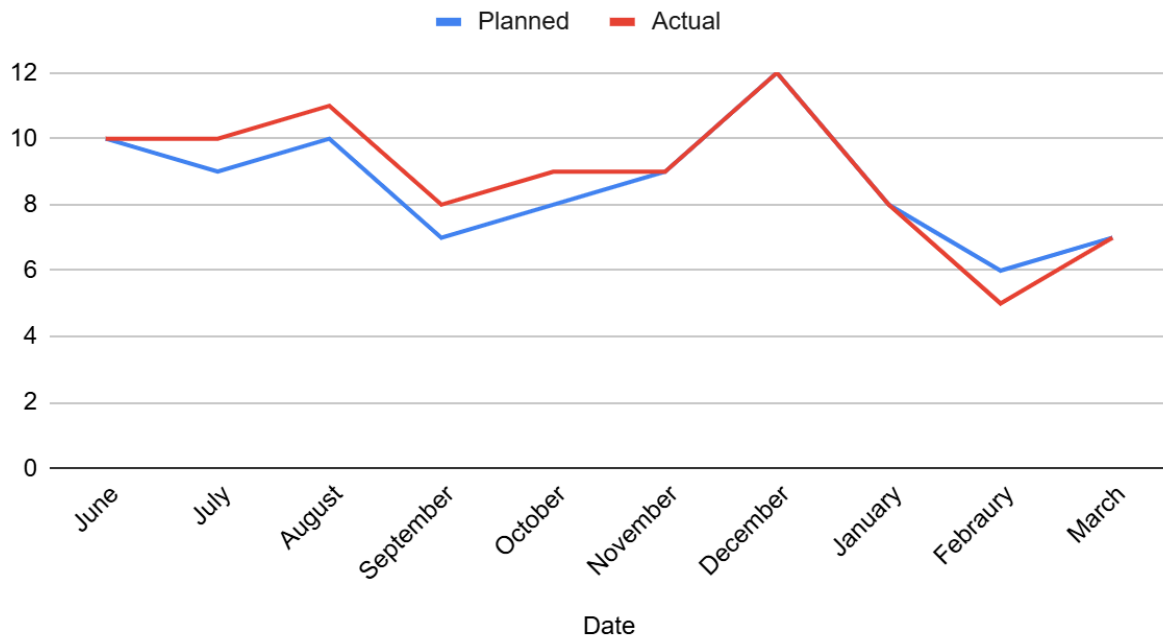


Date - May 31st, 2025 - June 7th, 2025

Sprint goal - create login



Planned and Actual



Question 10 – Explain about Product Grooming – 2 Marks

Product grooming

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. Additionally, the process helps product managers explain and align the organization behind the strategy that informs the backlog items.

Backlog grooming, also referred to as

Backlog management, backlog refinement, pre-planning, or story time, is a widely adopted activity by Scrum and agile product teams. The most common tactical activities that occur during backlog management include:

Removing outdated
user stories and tasks.

Adding new user stories that reflect newly discovered

user insights.

Breaking down broad user stories into smaller items.

Reordering user stories based on their priority.

Explaining and clearly defining user stories and tasks to avoid uncertainty and “black box” communication.

Assigning or re-assigning story points and estimates.

Identifying roadblocks and minimizing risks related to backlog items.

Benefits of backlog grooming

When done effectively, recurring backlog grooming sessions can keep your backlog in check and improve the organization of the items listed in it.

There are several important reasons to adopt backlog grooming:

Keeps the backlog clean

Depending on how big your organization is, a

For product development teams using the Agile project management methodology, Agile backlog grooming is a critical process that should be performed regularly for maximum impact. Just as the Agile framework itself calls for constant iteration, product backlog management enables development teams to consistently and continuously evaluate their backlogs and update them to reflect ever-evolving customer needs.

The goal of backlog grooming

The main goals of backlog grooming are to examine outstanding user stories in the backlog, confirm they are properly prioritized, and ensure that they're ready for sprint preparations. At the end of the session, you should have an organized and prioritized list of user stories.

Many agile practitioners say that a “DEEP” product backlog is the key outcome of a backlog refinement session. The DEEP acronym highlights some important traits associated with the product backlog:

- **Detailed appropriately.** Stories and other backlog items should contain enough contextual information to be understood and discussed by the cross-functional team.
- **Emergent.** Adding new items and stories should be simple as new information comes up, so nothing final is decided.

- **Estimated.** The effort required for each user story should be estimated through a measurement that's agreed upon by the team.
- **Prioritized.** Items in the backlog should be ranked depending on their tactical purpose and the value they provide.

Who owns the backlog grooming process?

The product owner or product manager should own the backlog grooming process, but they may not solely be responsible for hosting backlog grooming sessions, depending on the hierarchical structure of your organization. The Scrum Master, a project manager, or a different team member could also lead these sessions.

Some of the core responsibilities of the backlog grooming process owner include:

- Scheduling the session and ensuring the right people are invited (and attend)
- Keeping conversations focused and on topic
- Playing timekeeper and moving the conversation forward if the team gets stuck
- Sending follow-up communication to the team after the session

Who should attend backlog grooming sessions

These events are meant to be collaborative, so the entire cross-functional team should be represented at refinement sessions.

But at a minimum, the following people need to be involved in backlog grooming sessions:

- **Facilitator.** This should be someone who facilitates the session. It could be a product owner, product manager, Scrum Master, project manager, or even an Agile coach or consultant.
- **Engineers or developers.** These people own the narratives stored in the backlog.
- **Delivery team.** While this entire team plays a key role in bringing your company's products to your customer base, if the group is too large to include everyone, you can consider only inviting members of management.
- **Quality assurance representatives.** These team members can share user insights related to backlog inputs.

The key is to only invite those people who are critical for the task at hand, since too many collaborators and ideas could overwhelm the session. Also, be sure to have conversations with stakeholders before backlog refinement, not during the sessions.

How long should backlog grooming sessions be

While sessions will vary based on team priorities, the optimal length for backlog grooming is 45 minutes to an hour. However, there's no universal timeframe.

Some teams set time limits on each user story to help the meeting move efficiently. The best thing to do is to prevent side conversations and ensure your team stays on task.

1. Review and Analyze the Product Backlog:

- Identify items not ready for the next sprint:

 - Opens in new tab

- Determine which items are not yet detailed enough or prioritized correctly.

- Assess the overall health of the backlog:

 - Opens in new tab

- Check for items that are outdated, irrelevant, or not aligned with the product roadmap.

2. Refine Backlog Items:

- Add new user stories: Include user stories that reflect newly discovered insights or unmet needs.

- Break down large items: Decompose complex user stories into smaller, more manageable tasks.

- Clarify requirements: Ensure that user stories are clearly defined and that acceptance criteria are testable.

- Estimate effort: Assign story points or estimates to user stories to gauge the effort required.

- Update the backlog: Reflect any changes made to user stories, priorities, or estimates.

3. Prioritize and Order the Backlog:

- Align with strategic objectives:

- Ensure that the prioritized items align with the overall product roadmap and strategic goals.

Consider value and dependencies:

Prioritize items based on their potential value and any dependencies they may have on other items.

Communicate with stakeholders:

Involve stakeholders in the prioritization process to ensure alignment and buy-in.

4. Prepare for Sprint Planning:

Make sure user stories are ready:

Ensure that the user stories selected for the next sprint are well-defined, estimated, and prioritized.

Identify potential roadblocks:

Anticipate any potential issues or risks associated with the selected user stories.

Communicate with the development team:

Provide the development team with the necessary information and context for sprint planning.

5. Execute and Refine Continuously:

Regular grooming sessions: Schedule regular grooming sessions to maintain the health of the backlog.

Adapt to changes: Be prepared to adjust the backlog as new information or insights emerge.

Learn from experience: Review the results of each sprint and use that feedback to improve the grooming process.

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

Areas	Scrum Master	Product Owner
Roles	Works as a leader to facilitate team growth.	Works as the customer's representative and works on delivering the customer's needs.
Responsibilities	Ensures Scrum processes are abided by the team.	Strikes a balance between the clients and the team in terms of understanding the need and delivering it.

Authority	Scrum Masters can schedule and carry out meetings while ensuring team productivity	Product Owners make decisions regarding product backlogs.
Interactions	Scrum Masters generally interact with the team to work on strengthening the Scrum process.	The product Owner works as a bridge between the team, stakeholders, and clients.
Communication	Scrum Masters offer product updates to clients.	The Product Owner analyzes the available information to project how the product meets the client's requirements.
Time Commitment	Scrum Masters are committed to facilitating the Scrum process.	Product Owners are committed to prioritizing the product backlog and aligning it with the client's vision.
Skills	Skilled in agile scrum principles and have strong communication and problem-solving skills.	Skilled at agile development and have strong communication and leadership abilities.
Role Summary	Facilitates the Scrum process and ensures Agile adoption	Owns the product backlog and maximizes product value
Primary Focus	Team efficiency, process improvement	Product success, delivering customer/business value
Key Responsibilities	- Facilitate Scrum ceremonies - Remove blockers - Coach the team - Protect the team from distractions	- Manage product backlog - Define user stories - Prioritize features - Communicate with stakeholders
Owns	Scrum process and team dynamics	Product backlog, product roadmap

Authority	No authority over backlog or product scope	Full authority over product scope and backlog prioritization
Decision Making	Process and team support decisions	Business and feature decisions
Reports To	Agile Coach, Delivery/Program Manager	Business owner, Product Director, or similar
Stakeholder Interaction	Limited and indirect	Frequent and direct
Interacts With	Developers, Product Owner, and other Scrum Masters	Developers, Scrum Master, customers, and business stakeholders
Meetings Facilitated	Daily Scrum, Sprint Planning, Sprint Review, Retrospective	Sprint Planning, Backlog Refinement, Sprint Review
Tools Used	Jira, Confluence, Miro, Trello	Jira, Productboard, Aha!, Confluence, Excel
Success Metrics	Team velocity, impediment resolution, process adherence	Business value delivered, backlog health, stakeholder satisfaction
Skills Needed	Facilitation, Agile knowledge, conflict resolution, coaching	Product strategy, stakeholder management, prioritization, communication
Daily Activities	- Conduct standups - Remove blockers - Support team - Maintain the Scrum board	- Refine backlog - Write/clarify user stories - Prioritize tasks - Review sprint deliverables

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

Scrum meetings

Scrum meetings help teams, stakeholders, and management to stay aligned throughout a project.

Scrum it's a framework based on the Agile project management methodology. One of the Agile principles states that development teams should have frequent in-person conversations (or even video conferencing for virtual teams) to exchange information and react to changes quickly. So, Scrum meetings are an essential part of the framework.

There are various types of Scrum meetings, such as Sprint Planning, Daily Stand-ups, and Sprint Reviews. Each meeting has a different agenda, but they all serve a similar purpose: to keep the team aligned, identify and resolve issues quickly, and continuously improve the product development process while following the Scrum framework.

Who attends a Scrum meeting?

Most Scrum meetings should include the entire team, while other stakeholders may attend some meetings as observers. A Scrum team is a small team of 10 or fewer people with three defined roles:

Product owner: Controls the overall vision and objectives, liaises with stakeholders, and directs the team.

Scrum master: Reinforces Scrum values during meetings, dictates the timelines, and keeps team members on task.

Developers (team members): Execute the agreed work from the sprint planning. Based on the type of meeting, each Scrum team member plays a different role and participates differently.

What are the different types of Scrum meetings?

There are five different types of Scrum meetings. Here's a breakdown showing the purpose of each meeting, how often it is needed, and who must attend/run it.

1. Sprint planning meeting

When: Held at the start of each sprint.

Attendees: The full Scrum team — product owner, Scrum master, and developers.

Duration: They usually last about two hours for every week the sprint lasts. So, a two-week sprint would involve a four-hour planning session before work commences.

Purpose: The team reviews the product backlog, sets the sprint goal, and plans the work for the upcoming sprint.

The product owner conducts the meeting before the start of every sprint, following the Sprint review and the Sprint retrospective for the previous sprint. They present the product backlog, identify or reiterate the company's goals and desired outcomes, and answer any questions.

The development team looks at the product backlog tasks and predicts how much they can realistically deliver in the sprint. Time-boxing, or assigning a fixed and maximum amount of time for an activity, is a part of this process.

The work that the development team commits to is known as the sprint goal, and the collection of tasks that make up that goal is known as the sprint backlog. By the end of the meeting, the Scrum master ensures everyone agrees on the sprint's scope.

2. Daily Scrum (or Stand-up) meeting

When: Daily, typically first thing in the morning.

Attendees: All Scrum team members — the product owner, Scrum master, and developers.

Duration: No longer than 15 minutes.

Purpose: A short daily meeting where team members share progress, identify blockers, and plan their work for the day.

The daily Scrum meeting — also known as the daily stand-up meeting — allows the team to review goals and address any potential bottlenecks. (Other stakeholders may attend but are there only to listen, not participate.)

As the name suggests, the meeting takes place standing up and should last at most 15 minutes. Many teams prefer to meet at the beginning of the day, but it can occur anytime as long as the duration remains consistent.

The Scrum master guides the meeting by asking three questions:

What did you complete yesterday?

What tasks are in the pipeline today?

What is hindering the flow?

This ensures everyone is aligned on the day's tasks.

3. Sprint review meeting

When: At the end of a sprint.

Attendees: All the Scrum team members, plus key stakeholders invited by the product owner.

Duration: About an hour for every week of a sprint. So, a two-week sprint should be followed by a two-hour review session.

Purpose: The team demonstrates the work completed and gathers feedback from stakeholders.

The development team leads the sprint review meeting at the end of every sprint and presents the work they accomplished to the entire Scrum team and often outside stakeholders.

The Sprint review — also known as the Sprint Demo — showcases the value the project brings to the company and demonstrates the new functionality. The stakeholders also provide feedback on the user stories implemented in the sprint. If that feedback is accepted, it's added to a new product backlog, which is reviewed and prioritized during the next Sprint planning session.

4. Sprint retrospective meeting

When: At the end of a sprint.

Attendees: The whole Scrum team — product owner, Scrum master, and developers.

Duration: Typically, 45 minutes per sprint week. For example, the average length of a Sprint retrospective would be 90 minutes for a two-week sprint.

Purpose: The team reflects on what went well, what didn't, and identifies areas for improvement.

Continuous improvement is a major part of the Scrum framework, and the sprint retrospective facilitates this by reflecting on the previous sprint, and addressing:

What went right

What went wrong

What teams could do differently to improve collaboration in the future

No one outside the Scrum team attends these meetings, as they could hinder the open communication needed. During the meeting, the team can constructively criticize various elements of the sprint without assigning blame to other members to help the team work better in their subsequent sprints.

5. Backlog refinement meeting

When: Before the sprint planning meeting.

Attendees: The product owner and some or all of the developers and Scrum master.

Duration: No fixed time.

Purpose: The team discusses and prioritizes items in the product backlog.

The backlog refinement meeting — also known as “product backlog grooming” — allows Scrum teams to review and organize the product backlog to ensure it contains appropriate, prioritized, and well-estimated items. During the meeting, they identify and remove irrelevant items, create new items, re-prioritize, and split large items into smaller ones.

Unlike the time-boxed sprint planning and sprint retrospective meetings, backlog refinement is an ongoing process, and meetings can take place as often as needed.

Optional or as-needed meetings in Scrum

A backlog refinement meeting

A backlog refinement meeting is a regular meeting in Agile methodologies where the product owner and development team review, update, and refine the product backlog. This meeting ensures the backlog is ready for the next sprint, with items prioritized, estimated, and broken down into smaller tasks. It helps clarify the team's workload, align on priorities, and prepare for sprint planning.

A release planning meeting

A release planning meeting is a collaborative session used to create a release plan, which outlines the overall project scope, timelines, and priorities for a product release. It involves stakeholders, including the product owner, development team, and business representatives, to collectively define the release's vision, goals, and roadmap.

An ad hoc meeting

An ad hoc meeting is a spontaneous, one-off meeting called to address a specific issue or opportunity that requires immediate attention and action. It's different from regular, recurring meetings that have planned agendas and scheduled times. Ad hoc meetings

are often called when urgent decisions need to be made or when unexpected situations arise.

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

What is Sprint length?

The duration of a sprint is called Sprint length. In Scrum, a sprint is a time-boxed period of work, typically lasting between one and four weeks, during which a team delivers a potentially shippable increment of product. The length of a sprint should be consistent across the team and should be based on the team's needs, the complexity of the product, and the business's need for feedback.

Who determines the Sprint length?

In Scrum, the Scrum Team, comprising the Product Owner, Development Team, and Scrum Master, collectively determines the sprint length. While the Scrum Master facilitates the discussion and helps the team reach a consensus, the ultimate decision rests with the self-organizing team. People doing the work should decide how they should manage structure or manage the work.

What is a Scrum Team?

As described in the Scrum guide, a Scrum Team is a cohesive unit of professionals focused on one objective at a time, the Product Goal. The Scrum Team comprises 3 Scrum roles – the Developers, a Product Owner, and a Scrum Master. All of these contribute different responsibilities that help deliver valuable Increments.

A Scrum team should ideally be small, typically between 3 and 9 members, not including the Product Owner and Scrum Master. This helps maintain communication, collaboration, and agility. While 5-7 members are often cited as ideal, the specific size can vary based on project needs and company culture.

Contrary to teams that use Kanban boards, this small team of people is a fundamental part of Scrum. As such, it is an important aspect of successful Scrum implementation.

The Recommended Size for a Scrum Team

According to the Scrum guide, any Scrum Team should consist of 10 or fewer people. However, this estimation doesn't include a product owner or scrum master. Therefore, the typical scrum team size should be around 11 members (9 developers, 1 product owner, and 1-2 scrum masters).

So, in a small project, you would have a Scrum team of around 5 people—3 Developers, 1 Product Owner, and 1 Scrum Master. And in a large project, your team would consist of 9 people—7 Developers, 1 Product Owner, and 1 Scrum Master.

Who determines the Scrum team size?

In Scrum, the Scrum team itself determines its size, although there are guidelines and considerations. The Scrum Guide suggests that a Scrum team should consist of 10 or fewer people. The Scrum Master and Product Owner are part of this team, but the Developers (the core development team) are the ones who ultimately decide on the optimal size.

Question 14 – Explain DOR and DOD – 2 Marks

Definition of Done

The Definition of Done is a shared understanding among the Scrum team of what it means for an increment of work to be considered complete and potentially shippable. It represents the quality standards and criteria that must be met for a user story, feature, or product increment to be considered "done" at the end of a sprint. The DoD provides transparency, accountability, and a clear definition of the team's expectations for completing work.

The DoD is a comprehensive checklist of necessary criteria that ensures the increment is at the desired quality and is truly "done".

This includes all the work that must be done for the working increment to be useful and releasable and could consist of things like writing code, testing, documentation, and meeting regulatory requirements.

It is crucial for transparency and ensuring the Increment is respecting the standards of the organisation or the product, and also satisfies the Scrum Team's quality standards.

The DoD is often refined and agreed upon by the entire Scrum Team, and it's revisited as the team learns and the product evolves.

The DoD may include criteria such as:

All code has been written, reviewed, and meets the team's coding standards.

All tests (unit, integration, regression, etc.) have been written and passed successfully.

Documentation, both technical and user-facing, has been completed and updated.
The feature or user story has been accepted by the product owner and meets the desired level of quality and functionality.

The work has been demonstrated and accepted in the sprint review.

The DoD ensures that the team delivers a potentially releasable product increment at the end of each sprint, providing value to stakeholders and maintaining a high level of quality.

Definition of Ready

The Definition of Ready represents the criteria that a user story or product backlog item must meet before it can be considered ready to be pulled into a sprint. It serves as a checklist to ensure that the team has all the necessary information, requirements, and dependencies clarified and ready before work begins. The DoR helps prevent disruptions and delays during the sprint and promotes a smoother flow of work.

The DoR, on the other hand, is a complementary practice not mentioned in the Scrum Guide. It is commonly used by Scrum Teams as part of their working agreements.

The DoR helps the team determine when a Product Backlog item is ready to be moved to the stage where development takes place. It's a checklist that might include criteria, like dependencies identified and resolved, designs completed, having been through a refinement session, or clear acceptance criteria.

It ensures that work is sufficiently prepared, clear, and feasible before the team can consider moving it to the next stage of development.

The DoR aims to prevent bringing in work that is too ambiguous or large, which would impede the flow of value and could lead to impediments or blockers during the Sprint.

The DoR may include criteria such as:

The user story is well-defined, including clear acceptance criteria.

All necessary stakeholders have provided input and signed off on the user story.

Any dependencies or prerequisites have been identified and addressed.

The user story has been estimated and prioritized appropriately.

The development team has the capacity and necessary skills to complete the work.

By establishing a clear DoR, the team can ensure that they have a shared understanding of what needs to be done and are well-prepared to begin work on a user story or product backlog item.

Difference between DoD and DoR

The main difference between the DoD and the DoR is that the DoD is a list of criteria that must be met in order for a user story or Product Backlog Item (PBI) to be considered "done," while the DoR is a list of criteria that must be met in order for a PBI to be considered "ready" for development.

The DoD is typically more specific than the DoR. This is because the DoD is used to ensure that the product team is building high-quality software that meets the needs of users. The DoR is typically less specific than the DoD because it is used to ensure that the product team is working on the right things.

It's important to note that the DoD and DoR are not static and may evolve throughout the project as the team gains experience and learns from previous sprints. They are living documents that should be continuously refined and improved to meet the changing needs of the project and the organization.

Question 15 – Explain Prioritization Techniques and MVP – 3 Marks

What is an MVP?

MVP stands for Minimum Viable Product “The version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort.”. An MVP is a version of a product with just enough features to satisfy early customers and provide feedback for future product development.

A Minimum Viable Product (MVP) is a version of the product with minimal features, built to validate product ideas.

The Minimum Viable Product allows you to collect user feedback at the early stages of development and facilitates validated learning.

MVP reduces the time to market and allows a quicker release of the product.

MVP is a great risk reduction tool: it allows products or features to fail fast without investing in their development.

Here's why MVPs are crucial:

Quick validation: MVPs allow to test product assumptions with real users quickly.

Cost efficiency: They reduce the potential losses of developing features that customers do not want.

Faster learning cycles: By receiving immediate feedback, companies can iterate more rapidly

Key Features:

Basic and Minimal Features in an MVP:

Core Functionality:

The MVP should have the essential features that address the primary problem or need it aims to solve for the target audience.

Usability:

The product should be easy to understand and use, allowing users to get a sense of its value without excessive effort.

Feasibility:

The MVP should be achievable with available resources and time, focusing on core functionality rather than striving for perfection.

Feedback Mechanism:

The MVP should include ways for users to provide feedback, such as surveys, analytics, or direct user testing.

Value Proposition:

The MVP should clearly demonstrate the value of the product to users, motivating them to engage with it and provide feedback.

An MVP focuses on the core functionality that addresses a specific user need or problem. It's not a polished or feature-rich product, but rather a basic version designed to be easily tested and iterated upon.

What is MVP Features Prioritization?

Before work on an MVP begins, the product team will have created a full list of all the features they want their product to eventually have, and various methods called MoSCoW, FP Matrix, Story Mapping, and Kano. Features Prioritization helps to define

boundaries between wants and needs. It helps designers think dispassionately about which features are core and which aren't. It's like an editing process for features

What is the MoSCoW prioritization method?

MoSCoW prioritization is a tool for creating a hierarchy of priorities before and during a project. It stems from the Agile project management method, which aims to establish elements like product cost, quality, and project requirements as early as possible. MoSCoW is an acronym for "must-have," "should-have," "could-have," and "won't-have (this time)." Each item in the acronym denotes a category of prioritization. The idea is that items are categorized at the beginning of a project to clarify what is strictly necessary, what is desirable, and what the project can do without. Before we get deeper into the categories, let's take a look at where this method came from.

The MoSCoW method can be used to prioritize project requirements, product features, or any other project elements. When using the MoSCoW method, each element is assigned to one of the four prioritization categories based on its importance to the project. Now, let's take a closer look at the prioritization categories and how to use them.

What are the categories within the MoSCoW prioritization method?

As mentioned previously, the acronym "MoSCoW" stands for "must-have," "should-have," "could-have," and "won't-have (this time)."

- **Must-have:** These items are essential for the success of the project. There can be no compromise on whether they are included, because without them, the entire project would be meaningless. In short, this is a top-priority MoSCoW requirement.
- **Should-have:** These items are those that are important but not absolutely essential like those in the "must-have" category. Elements in this category are considered a secondary priority; that is, they are important, but not crucial to success.
- **Could-have:** These items would be nice to have but are not essential. Still less important than the two preceding categories, these elements are considered a third-level priority. If including them will have negative consequences on cost or meeting deadlines, they should be omitted. It is only when they don't negatively affect other project elements that they should be included.
- **Won't-have (this time):** These items are those that are not essential and can be excluded from the project without jeopardizing its success. Being the lowest priority category, omitting them won't hurt the project and they can be included when project conditions are more favorable.

When should you use the MoSCoW prioritization method?

The advantage of the MoSCoW method is that it can be applied to a wide range of situations, both in personal and professional contexts. For instance, an individual might use the method to prioritize their workload for the week or a team might use it to identify which tasks need to be completed first to meet a looming deadline. In general, the MoSCoW method is most useful when an individual or project team is facing a large number of tasks and needs help deciding which ones to focus on first. Additionally, if you're starting a project and are unsure which elements should take priority, you can get your project team together and use the MoSCoW method as a discussion tool in your prioritization process.

However, it should be noted that the method does have some limitations. For instance, it doesn't account for task dependencies or other factors that might impact the order in which tasks should be completed. As such, it should be used as one tool in a broader arsenal of task-management strategies.

MoSCoW method: the pros and cons

Because the framework makes it possible to manage the requirements for a given product release, it's extremely common among Agile projects with set timeboxes. Within project management, this prioritization technique has demonstrated its effectiveness and dependability. Of course, it isn't flawless — an objective look can help highlight shortcomings as well as benefits of the MoSCoW method. Let's examine its pros and cons.

Pros

- **Easy to master:** The method is based on very simple principles that are easy to understand, so you won't need to do much background research before getting started.
- **Helps prioritize:** It helps to clearly visualize priorities and sort them into a hierarchy. This way, you'll never need to be confused about which elements are more or less important than others.
- **Useful for team discussions:** Using the MoSCoW method in group discussions can be a great way to get team members to open up about their ideas. It can serve as a conversation starter, putting everyone on the same page.
- **Helps achieve stakeholder consensus:** When used among stakeholders, it can be a good way to reach an agreement between stakeholders on which elements should take priority. When stakeholders are present during the categorization process, this improves their understanding of the project.
- **Can prevent scope creep:** Scope creep is when unintentional changes occur during project execution. By setting clear and fixed priorities at the beginning, scope creep can be prevented.

Cons

- **Priority requirements can be subjective:** Since the categorization procedure is not based on numerical data, it leaves room for subjective interpretation. This can

manifest as conflicting ideas about which elements should take the lead and which are less important.

- Items require background context: To accurately categorize each element, you'll need to provide context for each of them. This may be time-consuming and tedious.
- Doesn't account for possible change: Putting items into a fixed category doesn't allow you to account for possible changes that occur during project execution. For example, elements that are not necessary at the start may become crucial down the line based on extraneous changes in circumstance.

Feature Priority Matrix

Feature Priority Matrix is similar to MoSCoW in that it has four priority categories. Where it differs is that it introduces effort as a second axis of decision-making.

The second axis allows for the categorization of features that are both useful and easy (or useless and difficult). As such, the four buckets differ somewhat from their MoSCoW equivalents.

Big Bets are features that add great value to the product but take considerable time and effort to develop. Quick Wins also add great value but are relatively easy to accomplish.

Maybes are roughly analogous to the "could-haves": features that are easy to develop but of limited usefulness. And Time Sinks are low-impact features that are nevertheless high-effort.

The inclusion of effort increases applicability to real-world situations that a team might encounter. For instance, when discussing priorities with a client, it is easier to make the argument that a certain feature they want is considered a time sink, and thus should be excluded from the MVP. Or, with two weeks until launch, you can prioritize high-impact features with a short development time.

On the downside, without technical representation, effort estimates can be inaccurate.

Story Mapping

Story Mapping is my favorite method and the one I've used most frequently. The method introduces User Journey into prioritization. Features are added under their respective activities and ranked according to necessity. Story Mapping provides a broader perspective on the product while allowing for more granular features than in prior methods.

The topmost features are designated as the Walking Skeleton. A line is then drawn between the columns that designates features for inclusion in the MVP.

One advantage of Story Mapping is that, thanks to clear representation of user activities, it's easier to spot missing steps. A short list of features under one User Journey indicates that that journey might not have been mapped out fully.

On the downside, an example story map is one thing, but a real-world story map can be a huge, evolving thing. Duplicates will be common and the map will require dedicated effort just to keep it up-to-date.

Kano

Named after the Japanese professor who invented it, Kano is the last method of Features Prioritization I'm going to discuss. It's the most complex of the methods and the first to include money in the prioritization equation. And as you may well know, there are no short conversations when money is concerned.

Kano maps satisfaction against effort/money, and identifies four types of product features.

Basic features are similar to the Must-have MoSCoW features, but basic does not necessarily mean cheap. You will be spending a lot of money on things that are so expected that they do not contribute positively to satisfaction. That's why the Basic curve does not even touch the neutral satisfaction line. A car, for instance, has a huge number of expensive, basic features that have to be included as given.

Neutral features don't add much value. Perhaps they have limited use-cases, such as the extremely niche features that can be found in Excel that barely anyone uses.

Performance is where we plot features whose quality can be a differentiator. Common examples include screen resolution and fuel consumption/battery life—high resolution and battery life would be great; low resolution and battery life would be unacceptable.

Lastly, Delighters are features that provide a key competitive advantage and make up your Unique Value Proposition. They are novel features that people want to show their friends. Every phone needs a screen, but a really high-quality screen can be a major selling point. For instance, color, high-definition, touch, ultra-HD, and foldable screens have been delighter features at one point or another over the last 20 years. But screens

also demonstrate how delighter features drift towards basic over time: the even ultra-HD is pretty much a given on modern mobile phones, leaving folding screens as the only one that people might want to show off to their friends.

Question 16 – Difference between Business Analyst and Product Owner – 3 Marks

The main difference between a business analyst (BA) and a product owner (PO) lies in their focus and responsibilities. While both roles aim to deliver value, the BA focuses on understanding business processes, capturing requirements, and analyzing data to improve internal operations. The PO, on the other hand, focuses on representing customer needs, prioritizing features, and managing the product backlog to maximize business value and ROI.

Here's a more detailed breakdown:

Category	Product Owner	Business Analyst (BA)
Primary Purpose	To maximize the value of the product by owning the product vision, managing the product backlog, and ensuring the development team builds the right features that meet business goals and user needs.	Understand business problems and translate them into detailed requirements
Primary Focus	Business value, product goals, and maximizing return on investment.	Requirement gathering, stakeholder needs, solution design
Role in Scrum Team	Owns the product vision, defines priorities, and manages the product backlog.	A proxy to stakeholders helps clarify requirements, supports the Product Owner
Core Responsibilities	-Define and prioritize user stories. -Communicate with stakeholders and development teams. -Accept or reject completed work. -Ensure alignment with business strategy.	- Elicit requirements - Write user stories - Clarify backlog items - Analyze data/processes

Who They Work With	Developers, User, Agile Coach, Stakeholders	Stakeholders, Product Owner, Developers, QA
Deliverables	- Product backlog - Release planning - Stakeholder update	- User stories - Acceptance criteria - Business Process Documents (BPD/BRD)
Decision Power	Has the authority to make decisions on what to build and when.	Input on product scope, but not the final decision-maker
Interaction with Stakeholders	The Product Owner talks to stakeholders to understand their needs, set priorities, and make decisions about what features to build.	Frequent and direct — gathers needs and feedback
Common Tools	Jira, Confluence, Trello, Miro, VersionOne, Azure Boards	Jira, Confluence, Visio, Excel, Lucidchart, Balsamiq, SQL
Technical Knowledge Required	Moderate — Agile/Scrum and basic project tools	High—domain knowledge, sometimes SQL/data/querying, process modeling
Reports To	The Product Owner usually reports to a Product Manager, Head of Product, or sometimes directly to business leadership	Product Owner / Business Manager / Project Manager
Collaboration	Supports the team in facilitating sprints	Supports the team in refining the backlog and communicating business needs
Involvement in Backlog	The Product Owner is responsible for the product backlog. They create, prioritize, and maintain it to ensure the team works on the most valuable items.	Writes/refines user stories, may help prioritize with PO
In Agile Ceremonies	Facilitates all (Sprint Planning, Review, Retro, Daily Scrum)	Participates in Planning, Review, Refinement sessions; clarifies scope and intent

Daily Activities	- Attend daily stand-up to stay updated with the team. -Prioritize and refine the product backlog. -Clarify user stories or requirements for the team.	- Write user stories - Meet stakeholders - Clarify acceptance criteria - Review progress
Goal	Build a high-performing Agile team	Deliver solutions that solve business needs
Primary Role and Focus	PO is responsible for ensuring that the Scrum team follows the product backlog and business goal	A Business Analyst focuses on understanding business needs, gathering requirements, and analyzing data to recommend solutions that address business problems.
Principles	Following the Scrum framework of iterative development, value-based prioritization, and collaboration.	Understand business needs and define solutions.
Stakeholder Interaction	Primarily interacts with the Scrum team, and stakeholders as needed to update product backlog.	Interacts extensively with various stakeholders, including business users, product managers, developers, and testers.
Reports and Documentation	Focuses on process-related documentation, such as sprint plans, road map, and product backlog.	Generates detailed documentation, including requirements documents, user stories, use cases, and process flow diagrams.
Project Lifecycle	Typically works within Agile frameworks (e.g., Scrum, Kanban) and focuses on iterative development.	May work in various project management methodologies (Waterfall, Agile, Hybrid) depending on the organization's approach.
Tools and Techniques	Requires a strong understanding of Agile principles and Scrum practices	Requires a deeper understanding of the technical aspects of the project, including system architecture and databases.

Question 17 – Prepare a sample Resume of 3 years ' exp a Product Owner – 3 Marks

Monisha Moudivender

Chennai,TN,609602

8754080525

monisha4@gmail.com

Dynamic Product Owner at TCS Siruseri with a proven track record in driving product success through data-driven decision making and effective stakeholder management. Expert in roadmap development and agile frameworks, I successfully optimized business strategies, enhancing brand effectiveness and securing long-term accounts while mentoring teams to maximize productivity.

Adept at deconstructing and planning stories in order to promote improvements to product design and implementation. Excellent relationship-building, organizational, and project management skills with a focus on overcoming technical challenges and implementing successful resolutions.

Dynamic individual with hands-on experience in business analysis and talent for navigating challenges. Brings strong problem-solving skills and proactive approach to new tasks. Known for adaptability, creativity, and results-oriented mindset. Committed to making meaningful contributions and advancing organizational goals.

Skills

- Story planning
- Data-driven decision making
- Problem resolutions
- Use case study generation
- Business analysis
- Risk management
- Roadmap development
- Stakeholder relationship management
- Scrum methodology

- User acceptance testing
- Product planning
- Release planning
- Agile frameworks

Work History

Product Owner

TCS Siruseri, Chennai

- Owned and prioritized product backlog.
- Oversaw entire product deployments from vision and creation to rollout and delivery.
- Analyzed customer feedback to identify areas for improvement.
- Coordinated with software developers to create custom applications and trackers to achieve client goals.
- Managed implementation of new features by outlining plans and specifications such as how, where, and when each component would work.
- Focused on building critical features up front in order to identify constraints and technical challenges.
- Continually maintained and improved company's reputation and positive image in markets served.
- Captured new customers by optimizing business strategies and launching products to diversify offerings.
- Analyzed and reported on KPIs to validate and demonstrate success of marketing campaigns.
- Mentored local personnel on best practices and protocols to maximize productivity.
- Managed budget allocation and resource utilization to maximize marketing ROI.
- Established, initiated and optimized business development strategies based on company targets, product specifications, market data, and budget factors.
- Secured long-term accounts by recommending strategies to promote brand effectiveness and highlight product benefits.

Education

Master of Arts in English Language And Literature, May 2019

Madras Christian College at Chennai

Certifications

- CM - Certified Manager Certification
- Certified Scrum Product Owner, CSPO-[COEPD] - [2025]

Languages

- English - Full Professional
- French - Limited Working
- Hindi - Limited Working