Capstone Part 2 perp 1

Q1, Write Agile Manifesto

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

Twelve Principles of Agile Software

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

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

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

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

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

1. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
2. Working software is the primary measure of progress.
3. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
4. Continuous attention to technical excellence and good design enhances agility.
5. 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 Behaviour accordingly.

**Q2,** User Stories- Acceptance Criteria-BV-CV?

|  |
| --- |
| User story 1 Tasks:1 Priority: HighSo That -I can order food onlineI want to -- Do the registration on the scrum food appAcceptance Criteria: Should have valid email Id ; Send Successful Notification to the userCP 8BV 500As a customer / user  |
| User story 2 Tasks: 1. Priority: HIGHAcceptance Criteria: User Name, Password, Mobile No, Email, Address,Phone Number.Click on Register Button.Send Successful Notification to the userCP 5BV 500So That -- I can login on the appI want to -- Login on the scrum food appAs a customer / user |
| User story 3 Tasks 1 Priority:HighAcceptance Criteria: : Should get the list of all the restaurants category wise veg and non veg and bothCP 5So That - I can choose the food and can add to cartI want to -Search and view restaurantsAs a customer / user BV 400 |
| User story 4 Tasks: :1. Priority: HighI want to – View restaurants as per the veg and Non veg categoryAcceptance Criteria: : Should get the list of all the restaurants category wise veg and non veg and bothCP 8BV 400So That - I can choose the food and can add to cartAs a customer / user |
| User story 5 Tasks: 1. Priority: High So That -I can choose the food and can add to cartAcceptance Criteria: Should get the list of all the restaurants category wise veg and non veg and bothCP 5BV 400I want to -view restaurants as per the veg and Non veg categoryAs a customer / user |
| User story 6 Tasks: 1. Priority So That -I can get food on my addressHIGHAcceptance Criteria: - once user want to order the food ; it should be delivered to the given addressCP 5BV 400I want to - Add the addressAs a Online User |
| User story 7 Tasks: 1. Priority: High Acceptance Criteria: All the selected item should get displayed in the cartCP8BV 400So That -I can get the selected foodI want to - I want to add selected item in the cartAs a Online User |
| User story 8 Tasks: 1Acceptance Criteria: App the payment option should get displayed in the payment mode ; COD - net banking - UPI – walletCP 3BV 400So That -so the I can make the payment of my choiceI want to - select the payment modeAs a Online User Priority:HIGH  |
| User story 9 Tasks: 1 Priority: HIgh I want to - track my given orderAcceptance Criteria: App should display the map and the current movement of the order;Display delivery boy mobile number ;Display delivery boy name in tracking field; Display delivery boy pictureCP 5BV 400So That - I can be able to view the current state on the mapAs a online user |
| User story 10 Tasks: 1. Priority: As a Online userHighAcceptance Criteria: App should be able to cancel the order within the 10 min of the given orderCP 5BV 400So That - if not required I can be able to cancel the whole orderI want to - cancel my order |
| User story 11 Tasks:1 Priority: High Acceptance Criteria: app should be able to show the feedback and rating option where usercan add reviewCP 5BV 400So That - I can be able to share rating and give opinion on the servicesI want to - share my feedback for food and scrum food appAs a Online User |
| User story 12 Tasks:1. Priority: High Acceptance Criteria: BR-ALL Mandatory, Text Box Fields, Display Order Id, Text Box for Description, Submit Button, Generate Issue ID, Display SuccessfulCP 2BV, 200So That – I can Request for RefundI want to Chat with Reg Admincan add reviewAs a customer / user |
| User story 13 Tasks: Show Multiple Resturants andPriority: High MenuAcceptance Criteria: The menu includes dishes , prices and descriptionsShow the restaurant is open or closedCP 2BV : 200So That - T I CAN FIND A PLACE TO ORDER FOODI want to -BROWSE DIFFERENT RESTAURANTS AND MENUMENUSAs a customer /user |
| User story 14 Tasks: 1. Priority: HIGHAcceptance Criteria: App displays relevant restaurant and dishes matching the queryCP 2BV 200So That - I CAN FIND A PLACE TO ORDER FOODI want to - TO BROWSE FOR SPECIFIC DISHES ANDCUISINESAs a customer / userCUISINES |
| User story 15 Tasks: 1. Priority: HIGH Acceptance Criteria: Filter restaurants by cuisine type and dietaryoptions(vegan, veg,nonveg,egg)CP 2BV : 200So That -I CAN FIND A PLACE TO ORDER FOODI want to -I WANT TO FILTER RESTAURANTSSO THAT I CAN FIND A PLACE TO ORDER FOODAs a customer / user |
| User story 16 Tasks: 1. Priority: HIGHAs a customer / user Acceptance Criteria: 1)App shows real time update on the order status2)Display estimated delivery timeCP 2BV 200So That - I KNOW THE TIME OF DELIVERYI want to - I WANT TO TRACK MY ORDER |
| User story 17 Tasks: 1 Priority: HIGH SO THAT I CAN RATE AND REVIEW THE RESTAURANTS I HAVE VISITEDAcceptance Criteria: )Can see reviews from other users to help me make dining decisionsCP 2BV 200I want to -TO RATE AND REVIEW RESTAURANTSAs a customer / user |
| User story 18 Tasks: 1. Priority: HIGH Acceptance Criteria: 1)Access my list of favourites easily for future ordersCP 2BV 200So That -I CAN ORDER FROM MY FAVOURITESI want to - I WANT TO SAVE FAVOURITE RESTAURANTS ANDDISHESAs a customer / userDISHES |
| User story 19 Tasks: 1 Priority:HIGH Acceptance Criteria: Can see the details such as order items, total cost and order dateCP 2BV 200So That - SO THAT I CAN ORDER AGAINI want to - VIEW PAST ORDER HISTORYAs a customer / user |
| User story 20 Tasks: 3 Priority:High Acceptance Criteria: Notifications for order confirmation, )Notification for dispatch, Notification for deliveryCP 2BV 200So That -SO THAT I CAN RECEIVE UPDATESI want to -I WANT TO RECEIVE NOTIFICATIONSAs a customer / user |
| User story 21 Tasks: 1 Priority:High Acceptance Criteria: Customer support section with contact informationCP 2BV 200So That - SO THAT I CAN SUBMIT QUERIES OR ISSUESI want to - CONTACT CUSTOMER SUPPORTAs a customer / user |
| User story 22 Tasks:2 Priority:High Acceptance Criteria: Manage order status, Notify restaurants about incoming ordersCP 2BV 200So That -I CAN UPDATE ORDER STATUSI want to - RECEIVE AND MANAGE ORDERSAs a RESTAURANT OWNER |
| User story 23 Tasks: 2 Priority: So That - I CAN VIEW AND RESPOND TO CUSTOMERREVIEWSAcceptance Criteria: Owners can address feedbackOwners can improve their servicesCP 2BV 200I want to - ACCESS TO CUSTOMER REVIEWSAs a RESTAURANT OWNER |
| User story 24 Tasks: 1 Priority: Acceptance Criteria: Active PromocodesCP 5BV 100So That -SO THAT I CAN ORDER AT LOWER PRICEI want to -APPLY PROMOCODES AND DISCOUNTSAs a customer / user |
| User story 25 Tasks: 7 Priority:High Acceptance Criteria: Order visibility, Real-time updates, Order details, Order filtering and sorting, Order map view, Order navigation, Order completion and confirmationCP 5BV 200So That - I ACCEPT THE ORDERI want to - VIEW THE ORDERSAs a Delivery Boy |
| User story 26 Tasks: 5 Priority:High Acceptance Criteria:User Authentication, Error Handling, Password security, Multi-factor Authentication, Compatibility and UsabilityCP 5BV 200So That - I CAN ACCEPT THE ORDERI want to -loginAs a DELIVERY BOY |
| User story 27 Tasks: 5. Priority:Medium Acceptance Criteria: Access to feed back system, Feedback Visibility, Feedback sorting and filtering, Response Mechanism, User SupportCP 5BV 200So That - I CAN KNOW THE CUSTOMERS FEEDBACKI want to - VIEW FEEDBACKAs a DELIVERY BOY |
| User story 28 Tasks: 5 Priority: Medium Acceptance Criteria: Access to feedback system, Feedback Visibility, Feedback sorting and filtering, Response Mechanism, User SupportCP 5BV 200So That - I CAN KNOW THE CUSTOMERS FEEDBACKI want to - VIEW FEEDBACKAs a Admin |
| User story 29 Tasks: 5 Priority:Medium Acceptance Criteria: Access to feed back system, Feedback Visibility, Feedback sorting and filtering, Response Mechanism, User SupportCP 5BV 200So That - I CAN KNOW THE CUSTOMERS FEEDBACKI want to - VIEW FEEDBACKAs a Restaurant owner |
| User story 30 Tasks:3 Priority:High Acceptance Criteria: )Display issue section, Sorting and filtering of issues list, Editing and modifying the issuesCP 3BV 100So That - I CAN RESOLVE THEMI want to – know the issuesAs a Admin |
| User story 31 Tasks: 3. Priority: HIGH Acceptance Criteria: Display issue section, Sorting and filtering of issues list, Editing and modifying the issuesCP 5BV 200So That -I CAN RESOLVE THEMI want to – know the issuesAs a Regional Admin |
| User story 32 Tasks: 6 Priority: HIGH Acceptance Criteria: Select Reports,Select Revenue Reports,Select to and from date,Select Region (can select all),Generate Report,Download Report in EXCELCP 5BV 200So That - I VIEW RESTAURANTS REVENUEI want to - VIEW REVENUE GENERATEDAs a RESTAURANT OWNER |
| User story 33 Tasks: 2. Priority: HIGHAcceptance Criteria: ID proofPunctuality and reliabilityCP 5BV 400So That - SO THAT I VERIFY THE DELIVERY BOYI want to - I WANT TO KNOW DELIVERY BOYAs a RESTAURANT OWNER |
| User story 34 Tasks: 2. Priority: HIGH Acceptance Criteria: Display delivery boy mobile number, Display delivery boy name in tracking field, Display delivery boy pictureCP 1BV 200So That - I CAN CONTACT DELIVERY BOY FOR THE STATUSI want to - VIEW THE CONTACT NUMBER OFDELIVERY BOYAs a customer / user |
| User story 35 Tasks: 2. Priority: Medium So That - CUSTOMER CAN CHECK OPENING AND CLOSING HOURSAcceptance Criteria: Click on restaurant dashboard,Add from time to time,Click on submit,Display updated successfullyCP 2BV 100I want to -TO PROVIDE TIME SLOTSAs a RESTAURANT OWNER |
| User story 36 Tasks:3. Priority: High Acceptance Criteria: Notifications for order confirmation, Notification for dispatch, Notification for deliveryCP 2BV 200So That - I CAN RECEIVE UPDATESI want to - RECEIVE NOTIFICATIONSAs a customer / user |
| User story 37 Tasks: 1. Priority: Medium Acceptance Criteria: Customer support section with contact informationCP 2BV 400So That - I CAN SUBMIT QUERIES OR ISSUESI want to - CONTACT CUSTOMER SUPPORTAs a customer / user |
| User story 38 Tasks:4. Priority: Medium Acceptance Criteria:CP 3BV 100So That -I CAN CANCEL ITI want to -I WANT TO VIEW THE ORDERAs a customer / user |
| User story 39 Tasks: 4. Priority:High Acceptance Criteria: Real time tracking,Security and data privacy,User friendly InterfaceCP 3BV 400So That -I CAN VIEW THE STATUS OF THE DELIVERYI want to -TRACK THE DELIVERYAs a REGIONAL ADMIN |
| User story 40 Tasks: 2. Priority: High Acceptance Criteria: Mention the list of order generated in a day ; billing done ; revenue generated ; payment received ; fetch in excelCP5BV 400So That -View revenue generated through Scrum Food appI want to -check the revenue day wiseAs a - Restaurant owner |

Q3. What is epic? Write 2 epics

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

1. Login – Consumer should be able to login with Email ID and Password
2. Order History – Customers should be able to locate the Current orders and previous orders
3. Search and Filter – Customers should be able to search and filters all the available Food option using key word
4. Design Delivery Map – As a part of the order status users would like to view the location and arrival time for their food .

2 Epics

Ratings and reviews:

As a user, I want to view ratings and reviews for restaurants on scrum foods, so that I

can make informed decisions about where to order food from.

As a user, I want to provide ratings and reviews for restaurants on scrum foods, so that I

can share my experiences with other users and contribute to the community.

Acceptance Criteria

● Users can view average ratings and reviews for each restaurant on the

restaurants details page

● Users can read detailed reviews and comments left by the other customers

● Users can sort and filter reviews based on criteria such as rating and relevance

● Users can rate the restaurants and leave a review after placing the order

● User can edit or delete their own reviews within a specified timeframe

● Reviews are displayed in a way that provides helpful insights to other users

● The rating and review system maintains the integrity and authenticity of user

Feedback

2.Epic: Real-Time Order Tracking for Food Delivery App

Description:

The real-time order tracking epic aims to provide users with a seamless and transparent

experience by allowing them to track the status and location of their food orders in

real-time. This feature enhances customer satisfaction, reduces support inquiries, and

improves overall user engagement.

User Stories:

As a customer, I want to see the live status of my order.

● Display the current status of the order, such as "order confirmed,"

"preparing," "out for delivery," and "delivered."

● Provide real-time updates as the order progresses through various stages.

​As a customer, I want to track the location of my delivery.

● Integrate GPS or location services to show the delivery partner's real-time

location on a map.

● Allow customers to view the estimated time of arrival (ETA) based on the

delivery partner's location.

​As a customer, I want to receive notifications for order updates.

● Send push notifications or SMS updates to inform customers about order

confirmation, preparation, and delivery status changes.

● Provide delivery partner details, including name, contact information, and

a profile picture.

​As a customer, I want to contact the delivery partner directly.

● Enable in-app chat or call functionality to allow customers to communicate

with the assigned delivery partner.

● Ensure privacy by using masked phone numbers or secure messaging

channels.

​As a customer, I want to view the delivery route.

● Display the delivery route on the map, showing the path the delivery

partner will take to reach the destination.

● Allow customers to track the progress of the delivery in real-time along the

route.

​As a customer, I want to provide feedback on the delivery experience.

● Allow customers to rate the delivery partner and overall delivery

experience after the order is delivered.

● Implement a feedback system with written comments to gather valuable

insights.

​As a customer, I want to see estimated delivery time adjustments.

● Account for real-time traffic conditions and other factors that may affect the

delivery time.

● Update the estimated delivery time accordingly and inform the customer

promptly.

​As a customer, I want to have a seamless tracking experience across platforms.

● Ensure the real-time order tracking feature is available and consistent on

all supported platforms (e.g., mobile app, web).

​As an admin, I want to monitor order tracking performance.

● Provide analytics and reporting on order tracking metrics, such as average

delivery time and customer satisfaction ratings.

● Use data to identify areas for improvement and optimize the delivery

process.

Acceptance Criteria:

Real-Time Order Updates:

● The app should provide real-time updates on the status of the user's order,

such as "Order received," "Preparing," "Out for delivery," and "Delivered."

Order Location Tracking:

● The app should display the live location of the delivery driver while en

route to the user's address.

● The map should update at regular intervals to reflect the driver's

movement accurately.

Estimated Delivery Time:

● The app should provide an accurate estimated time of delivery (ETA)

based on the driver's current location, distance to the delivery address,

and traffic conditions.

Delivery Notifications:

● Users should receive push notifications or in-app alerts for significant

order updates, such as when the order is dispatched for delivery or when

it is near the delivery address.

Map Zoom and Interaction:

● Users should be able to zoom in and out on the map to view the delivery

driver's route more closely.

● The map should support standard interactions, such as panning and

rotating, to improve the user experience.

Delivery Status History:

● Users should have access to the delivery status history, allowing them to

see the timeline of their order from placement to delivery completion.

Accuracy and Reliability:

The real-time tracking information should be accurate and reliable,providing users with the most up-to-date data available.

● The system should handle location updates efficiently, minimizing delays

or inaccuracies. Privacy and Security:

Opt-Out Option:

● Users should have the option to disable real-time order tracking if they prefer not

to share their location information.

User Education:

● Provide clear instructions or tooltips to educate users on how to use the real-time

order tracking feature effectively.

Support for Multiple Orders:

● If a user places multiple orders, the app should allow them to track each order

individually with its own status and location updates.

By meeting these acceptance criteria, the food delivery app can successfully implement

real-time order tracking, providing users with a transparent and convenient way to

monitor their orders from the moment they are placed until they are delivered to their

doorstep.

Q4, What is the difference between BV and CP

**Business Value** is the vision where in product owner uses to order the product backlog. It can be derived by lowering costs, increasing revenue, growing customer satisfaction, reducing risk or enhancing capability.

* Business value is what delivered to end customer and how they received it. Feedback from customer is good measure to know the value delivered.

**Story Points – (CP)**

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

**Q5**, Explain about Sprint?

In the context of software development and project management, a sprint is a

time-boxed, iterative development period during which a specific set of tasks and goals

are worked on by a development team. Sprint is a core concept in Agile methodologies,

such as Scrum, which emphasizes flexibility, collaboration, and delivering value to the

customer in shorter cycles.

Here are the key characteristics and components of a sprint:

● Time Frame: A sprint typically has a fixed duration, often ranging from 1 to 4

weeks. The duration is consistent across all sprints to provide a predictable

cadence for development and planning.

● Goals and Objectives: At the beginning of each sprint, the development team,

along with stakeholders, selects a set of user stories, features, or tasks to work

on during that sprint. These items are collectively referred to as the sprint

backlog.

● Planning: During sprint planning, the development team breaks down the

selected items from the product backlog into smaller tasks and estimates the

effort required for each task. The team commits to completing these tasks within

the sprint duration.

● Daily Stand-ups: Throughout the sprint, the team holds daily stand-up meetings

(also known as daily scrums) to discuss progress, obstacles, and plans. Each

team member shares what they've accomplished, what they're working on, and

any challenges they're facing. These meetings foster communication and

collaboration.

● Development: The development team works on the tasks identified in the sprint

backlog. They collaborate closely, often using techniques like pair programming

and frequent code reviews to ensure high-quality work.

● Continuous Integration: Developers integrate their code changes into the main

codebase regularly, ensuring that the software remains functional and stable

throughout the sprint.

● Testing: Testing is an integral part of a sprint. Automated tests are run to validate

code changes, and manual testing may be conducted to ensure the quality of the

software.

● Review and Demo: At the end of the sprint, the development team conducts a

sprint review and demo. They showcase the completed work to stakeholders,

gathering feedback and validation. This helps ensure that the delivered features

align with expectations.

● Retrospective: Following the review and demo, the team holds a sprint

retrospective. They reflect on what went well during the sprint, what could be

improved, and actions to take in the next sprint. The retrospective encourages

continuous improvement.

● Incremental Development: Each sprint results in a potentially shippable product

increment, meaning that at the end of each sprint, a new version of the software

is available with additional features or improvements.

● Adaptability: Agile methodologies emphasize adaptability and the ability to

respond to changing requirements. If new priorities or insights emerge,

adjustments can be made in subsequent sprints.

Sprints allow development teams to iteratively deliver value to customers and

stakeholders in a controlled and predictable manner. By breaking down the work into

manageable chunks and continuously seeking feedback, Agile teams can enhance

collaboration, reduce risk, and improve the overall quality of the software being

developed.

WIP

PBI

Task

Don

e

Logi

n

Login user with email and password. A temporary

password will be sent to the admin's registered email

address with a link for a password reset.

View, filter previously

added data

DOD

Select reporting

period, Indicators

Collect Scope 1 data

Collect Scope 2 data

**Q6,** Explain Product backlog and sprint back log?

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

**Sprint back Log** is the subset of product backlog

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

**Q7,** What is impediments log? write 2 impediments

An Agile impediment log is used to record, track and resolve a list of obstacles, challenges that delay the team's performance

An Impediment is anything that keeps the Team from getting work Done and that slows Velocity.

* Ex- sick team member, a missing resource, lack of management support Business or customer issues; Unresolved dependencies
* Organizational Impediments – issues that are dependent on others to solve. These issues include but are not limited to:
	+ Slow internet
	+ Issues with obtaining input from other teams or divisions o Lack of training

**Q8,** Explain Velocity of the Team**?**

Velocity refers to the measure of the amount of work a development team can complete

during a sprint.

The calculation of velocity is performed by the development team itself, as they are

responsible for estimating the effort required to complete each user story or backlog

item.

Story point estimation:

Story point estimation is a technique used in agile software development to estimate the

effort required for a specific task or user story. It's a relative measure of complexity

rather than a fixed time unit. Team members assign story points based on their

understanding of the work involved, considering factors like complexity, effort, and

uncertainty.

The actual time a story point represents can vary from team to team. For some, it might

equate to hours, while for others, it might represent days. It's important to establish a

consistent baseline within the team so that story point estimates can be used effectively

for planning and prioritization.

Tracking completed work:

Tracking completed work in Agile development typically involves calculating the total

story points completed by the team over a specific time frame, usually a sprint or

iteration. Here's how you can calculate completed work:

● Identify Completed Stories: At the end of the sprint or iteration, review the user

stories or tasks that were completed and accepted as done.

● Sum Story Points: Add up the story points assigned to all the completed user

stories. Exclude any story points that were not fully finished or accepted during

the sprint.

● Calculate Total Completed Work: The sum of story points completed

represents the total completed work for that sprint.

This completed work can be used to calculate the team's velocity for that specific sprint,

as mentioned in the previous response. It provides insights into the team's capacity and

helps with future sprint planning and estimation.

Summing story points:

Summing story points involves adding up the numerical values assigned to individual

user stories or tasks during the estimation process in Agile development. Story points

are used to represent the effort, complexity, and size of a piece of work relative to other

items on the backlog. Here's how you can sum story points:

● List Completed User Stories: Gather a list of user stories or tasks that have

been completed during a specific sprint or iteration.

● Identify Story Point Values: Each user story or task should have a story point

value assigned to it during the estimation process. These values are usually

relative, such as 1, 2, 3, 5, 8, 13, etc., representing increasing levels of

complexity or effort.

● Add Up Story Point Values: Sum up the story point values for all the completed

user stories or tasks. For example, if you completed user stories with story point

values of 3, 5, and 8, the sum would be 16.

The sum of story points provides a quantitative measure of the work completed by the

team during a sprint. This sum is often used to calculate the team's velocity, which helps

in future sprint planning and estimation.

Average velocity:

Average velocity in Agile development refers to the average amount of work, measured

in story points, that a team completes during a series of sprints or iterations. It's a key

metric used for planning and estimating future work. Here's how to calculate average

velocity:

● Select a Time Frame: Choose a specific number of past sprints or iterations for

which you want to calculate the average velocity. For example, you might choose

the last 5 sprints.

● Sum Completed Story Points: Add up the total story points completed by the

team in each of the selected sprints. This will give you the total completed work

for the chosen time frame.

● Calculate Average: Divide the total completed story points by the number of

sprints or iterations you selected. This will give you the average velocity for that

period.

Formula: Average Velocity = Total Completed Story Points / Number of Sprints

Use for Planning: The average velocity can serve as a guideline for future sprint

planning. It helps the team estimate how much work they can commit to in upcoming

iterations based on their historical performance.

Keep in mind that average velocity is a rough estimate and can fluctuate based on

various factors. It's important to consider the team's capacity, any changes in team

composition, and improvements in estimation accuracy over time.

**Q9,** Draw Sprint Burn Charts n Product Burn Down Charts? Sprint Burn down chart



**Sprint Burn Up Chart**

**Product Grooming**

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

A sprint burndown chart includes:

* X-axis- The horizontal axis of the graph represents the remaining amount of time to complete the project usually depicted in days. Sprints
* Y-axis- The vertical axis of the graph represents the effort needed to complete the project. Story Pt
* Actual work line - This represents the actual number of tasks remaining. It might be straight in some cases; however, it often changes due to unforeseen issues in the project or an increase in the number of tasks.

Q10, Explain about Product Grooming?

Product grooming, also known as backlog grooming or refinement, is a crucial activity in

Agile development that involves preparing and refining items in the product backlog to

ensure they are well-understood, prioritized, and ready for development. Let's break

down the process step by step:

● Setting the Context: At the beginning of the backlog grooming process, the

team and relevant stakeholders come together to understand the overall goals

and objectives of the project. This helps set the context for the work to be done

and aligns everyone's understanding.

● Backlog Review: The product owner and the development team review the

items in the product backlog. This involves assessing the user stories, tasks, and

other items to ensure they are accurate, up-to-date, and still relevant to the

project's goals.

● Prioritization: During backlog grooming, the team collaboratively prioritizes the

backlog items based on their value to the product and the needs of the users or

customers. This helps ensure that the most important and valuable work is

addressed first.

● Refinement and Estimation: In this step, the backlog items are refined to

provide clear and detailed descriptions. The team breaks down user stories into

smaller tasks and discusses the technical requirements. Estimation involves

assigning story points or other sizing metrics to each item, indicating the relative

effort needed for implementation.

● Dependency Analysis: The team examines potential dependencies between

backlog items. Identifying and understanding dependencies helps in planning the

order of implementation and managing potential bottlenecks.

● Acceptance Criteria: Well-defined acceptance criteria are established for each

backlog item. These criteria outline the conditions that must be met for the item

to be considered complete and ready for delivery. Clear acceptance criteria help

prevent misunderstandings and ensure a shared understanding of what is

expected.

● Backlog Grooming Meetings: These are recurring meetings where the product

owner and the development team come together to perform the activities

mentioned above. These meetings often occur before sprint planning sessions to

ensure that the upcoming sprint backlog is well-prepared.

Backlog grooming is an iterative process that helps maintain a healthy and

well-organized product backlog. It ensures that the development team always has a

prioritized list of well-defined, estimated, and ready-to-develop items. This, in turn,

supports the efficient planning and execution of sprints and helps the team deliver value

to customers in a more predictable and effective manner

Q11, Explain the roles of Scrum Master and Product Owner?

 **A Scrum Master** popularly known as a coach, motivator and leader of an Agile team. The role of a Scrum Master is to educate the team on Agile processes and help team members follow Scrum practices religiously. Facilitation scrum event as and when it is required .

The Scrum Master collaborates both with the Product Owner (PO) who focuses on building the right product, and the development team that focuses on building the product right. A Scrum Master’s job is essentially to help everyone understand and imbibe Scrum values, principles, and practices and get the best product out to the customer

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

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

**Q12,** Explain all Meetings Conducted in Scrum Project?

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

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

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

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

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

tasks and estimates the effort required.

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

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

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

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

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

informed about the progress and challenges.

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

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

the Product Owner. The team demonstrates the potentially shippable product

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

can update the backlog.

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

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

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

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

effectiveness.

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

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

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

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

Product Backlog Refinement: This meeting focuses on refining the product

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

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

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

Release planning:This meeting occurs at the start of the project or major

release and involves the product owner, development team, and stakeholders. It

aims to discuss and plan the high level scope, timeline, and goals for the project.

Ad hoc meetings:These meetings may be schedules as needed to address

specific topics or issues, such as resolving impediments, discuss technical

challenges, or conducting additional planning or collaboration sessions

**Q13,** Explain Sprint Size and Scrum Size?

**Sprint Size-** Sprints are the soul of Scrum methodology within Agile Project Management.

A Sprint is a time-boxed event of weeks in which your Scrum team focuses only on a sprint goal. The goal is typically a product increment or iteration, often an updated, improved version of your product or software. Normally a sprint happens for two weeks.

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

Q14, Explain DOR and DOD?

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

**Q15,** Explain Prioritization Techniques and MVP?

Prioritization Technique –

Prioritization techniques are methods used to determine the order in which tasks,

features, or items should be addressed in a project. These techniques help teams

allocate resources effectively and focus on delivering the most valuable work first. Some

common prioritization techniques include:

MoSCoW: This technique categorizes items into Must have, Should have, Could

have, and Won't have categories. It helps clarify essential features from those

that are optional or lower priority.

Weighted Shortest Job First (WSJF): WSJF assigns a priority score to each

item based on factors like business value, time sensitivity, and risk. Items with

higher scores are considered more important to work on.

Kano Model: This model categorizes features into Basic Needs, Performance

Needs, and Delighters. It helps prioritize based on how features impact user

satisfaction.

Value vs. Effort Matrix: Items are plotted on a matrix based on their potential

value and effort required. This helps identify quick wins and high-value tasks.

Relative Prioritization: Teams compare items pairwise to determine which is

more important. This helps create a relative ranking of items.

Buy a Feature: Stakeholders are given a budget to 'buy' features, which helps

prioritize features based on how much value they see in them.

“FURPS” Technique

This technique is used to validate must requirement.

F-Functionality

U-Usability

R-Reliability

P-Performance

S-Supportability (Extendable, Testable & Enhance-able)

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

**Q16,** Difference between Business Analyst and Product Owner?

**Product Owner role**

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

**Business Analyst role**

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

**Q17,** Prepare a sample Resume of 3yrs exp Product Owner?

****