**Question No – 1) What is Brainstorming?**

Brainstorming is a way for a group of people to come together and think of as many ideas as possible to solve a problem. The goal is to encourage creativity without judging ideas at first. Everyone shares their thoughts freely, and later, the best ones are selected. It is commonly used in business, product development, and problem-solving to generate fresh and innovative solutions.

**Question No – 2) What is JAD Sessions?**

A Joint Application Development (JAD) session is a meeting where business owners, developers, and other stakeholders come together to discuss and finalize project requirements. Instead of gathering requirements separately, all key people work together in a structured session to ensure everyone is aligned. This method saves time, reduces misunderstandings, and helps teams build better solutions.

**Question No – 3) Why Document Analysis is one of the compulsory techniques we use in a Project?**

Document analysis is important because it helps teams understand existing systems, processes, and rules before making changes or building something new. By reviewing documents like policies, user manuals, and reports, businesses can avoid mistakes, ensure accuracy, and maintain consistency. Without document analysis, there’s a risk of missing key details, leading to problems later in the project.

**Question No – 4) In Which Context we will use Reverse Engineering?**

Reverse engineering is used when we need to understand how an existing product or system works without having detailed information about it. It is commonly used to analyse old software, improve outdated systems, fix security issues, or learn from competitor products. For example, if a company wants to update an old application but lacks documentation, they can reverse-engineer it to understand how it was built.

**Question No – 5) what is Observation Technique– Explain both Active and Passive approaches?**

Observation is a technique where someone watches a process or people’s behavior to gather information. This helps in understanding real-world issues without relying only on what people say.

* Active Observation: The observer participates in the process, asking questions and interacting with users. Example: A business analyst testing an app alongside real users.
* Passive Observation: The observer simply watches without interfering. Example: A researcher silently observing how customers interact with a new website.

Both approaches help in identifying problems, improving user experiences, and refining business processes.

**Question No – 6) How do you conduct the Requirements Workshop?**

A Requirements Workshop is a structured meeting where stakeholders, business analysts, and developers collaborate to gather, refine, and validate project requirements. The key steps in conducting a requirements workshop are:

1. Planning: Define objectives, identify participants, and prepare agendas.
2. Facilitation: Conduct discussions using techniques like brainstorming, JAD sessions, or role-playing.
3. Documentation: Capture key insights, decisions, and open questions.
4. Validation: Review and confirm gathered requirements with stakeholders.
5. Follow-up: Share workshop outcomes and define next steps.

These workshops ensure alignment, reduce rework, and accelerate requirement gathering in projects

**Question No – 7) Prioritise the Requirements––Where we will use?**

Prioritizing requirements is crucial in project management, especially when resources, time, or budget are limited. It helps teams focus on delivering the most valuable features first, ensuring efficient product development.

Where Do We Use It?

* Agile Development: Helps in backlog refinement and sprint planning.
* Waterfall Projects: Used in scope definition and project planning.
* MVP (Minimum Viable Product) Development: Prioritizes essential features for early market release.
* Risk Management: Identifies critical vs. optional requirements.

Example of MoSCoW Prioritization

The MoSCoW method is a popular prioritization technique that classifies requirements into four categories:

1. Must-Have – Critical for project success.
*Example:* User authentication in a banking app.
2. Should-Have – Important but not immediately necessary.
*Example:* Dark mode feature in a mobile app.
3. Could-Have – Nice to have but not a priority.
*Example:* Multiple themes for UI customization.
4. Won’t-Have (This Time) – Not included in the current release but may be considered later.
*Example:* AI-based chatbot for customer service in phase 1.

This approach ensures that essential features are delivered first while allowing flexibility for future enhancements.

**Question No – 8) Difference between Traditional Development Model and Agile Developm ent Models**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Traditional Development Model (Waterfall)** | **Agile Development Model** |
| **Approach** | Sequential, phase-wise | Iterative, incremental |
| **Flexibility** | Rigid, hard to change | Adaptive to changes |
| **Delivery** | Delivered at project end | Continuous deliveries |
| **Stakeholder Involvement** | Minimal after requirements phase | Ongoing collaboration |
| **Testing** | Performed at the end | Continuous testing |
| **Risk Management** | High risk of failure if wrong assumptions are made | Lower risk due to early |

**Question No – 9) what Roles and responsibility of scrum Master in agile process?**

A Scrum Master ensures the Agile team follows the Scrum framework effectively. They act as a coach, facilitator, and problem solver to improve team productivity and collaboration.

Key Responsibilities:

1. Facilitating Scrum Events – Organizes daily stand-ups, sprint planning, reviews, and retrospectives.
2. Removing Impediments – Identifies and resolves blockers affecting team progress.
3. Coaching the Team – Guides team members on Agile principles and best practices.
4. Ensuring Agile Adoption – Promotes Agile values within the team and organization.
5. Protecting the Team – Shields the team from distractions and external pressures.
6. Promoting Continuous Improvement – Encourages regular feedback and process improvements.

Conclusion

The Scrum Master is a servant-leader who ensures smooth workflow, helps the team stay focused, and fosters a culture of continuous improvement in Agile projects.

**Question No – 10) What is sprint meeting and it's type explain in short every meeting?**

Sprint meetings are a crucial part of Agile development, ensuring teams stay on track and aligned. Here's a brief overview of each type:

**1. Sprint Planning Meeting**

- Held at the beginning of each sprint

- Purpose: Plan and prioritize work for the upcoming sprint

- Attendees: Development team, Product Owner, Scrum Master

- Agenda: Review backlog, set sprint goals, assign tasks

**2. Daily Scrum Meeting (Daily Stand-up)**

- Held daily, typically at the same time

- Purpose: Sync team members, discuss progress, and plan day's work

- Attendees: Development team

- Agenda: Share progress, discuss obstacles, plan day's work

**3. Sprint Review Meeting**

- Held at the end of each sprint

- Purpose: Review and demo completed work to stakeholders

- Attendees: Development team, Product Owner, stakeholders

- Agenda: Demo completed work, gather feedback, discuss next steps

**4. Sprint Retrospective Meeting**

- Held after the Sprint Review Meeting

- Purpose: Reflect on the sprint, identify improvements, and adjust processes

- Attendees: Development team, Scrum Master

- Agenda: Discuss what went well, what didn't, and identify improvements

**5. Backlog Refinement Meeting**

- Held as needed, typically bi-weekly

- Purpose: Review and refine the product backlog

- Attendees: Product Owner, Development team

- Agenda: Review backlog items, clarify requirements, estimate effort

These meetings ensure teams stay aligned, focused, and continuously improving.

**Question No – 11) What is velocity?**

Velocity is a measure of a team's productivity and progress in Agile development. It represents the amount of work a team can complete during a sprint or iteration.

**Calculation:**

Velocity is typically calculated by measuring the number of story points or hours completed during a sprint.

**Example:**

- If a team completes 20 story points in a sprint, their velocity is 20.

- If a team completes 100 hours of work in a sprint, their velocity is 100.

**Purpose:**

- To estimate the team's capacity for future sprints

- To plan and forecast upcoming work

- To measure team performance and progress over time

Velocity helps teams to better plan, estimate, and deliver their work, and to continuously improve their processes and productivity.

**Question No – 12) What is sprint burn down chart ?**

A Sprint Burn-Down Chart is a visual representation of the progress made by a team during a sprint in Agile development. Here's a brief overview:

**What it shows:**

- The total work remaining in the sprint (usually measured in hours or story points)

- The progress made by the team each day

- The rate at which the team is completing work (burn-down rate)

**Key components:**

- Ideal Burn-Down Line: A straight line showing the ideal rate of progress

- Actual Burn-Down Line: A line showing the actual progress made by the team

- Work Remaining: The total work remaining in the sprint

**Purpose:**

- To track progress and identify any deviations from the plan

- To help the team adjust their pace and stay on track

- To provide transparency and visibility to stakeholders

By using a Sprint Burn-Down Chart, teams can visualize their progress, identify potential issues, and make adjustments to ensure a successful sprint.

**Question No – 13) What is product burn down chart?**

A Product Burn-Down Chart is a visual representation of the progress made by a team in delivering a product or project over time. Here's a brief overview:

**What it shows:**

- The total scope of work remaining in the product backlog

- The progress made by the team in delivering working software

- The rate at which the team is completing work (burn-down rate)

**Key components:**

- Product Backlog: The total scope of work to be delivered

- Work Remaining: The total work remaining in the product backlog

- Burn-Down Rate: The rate at which the team is completing work

**Purpose:**

- To track progress and identify trends

- To help the team and stakeholders understand the project's status

- To provide a forecast of when the project will be completed

By using a Product Burn-Down Chart, teams can visualize their progress, identify potential issues, and make adjustments to ensure successful project delivery.

**Question No – 14) What is user stories and also define task, priority, value statement, BV, CP, acceptance criteria?**

**User Story**

A user story is a natural-language description of a software feature or requirement from the end-user's perspective. It's a concise statement that captures the essence of the feature.

**Components of a User Story:**

**1. Task**: A specific activity or feature to be developed.

Example: "As a user, I want to login to the system."

**2. Priority:** The level of importance assigned to the task.

Example: High, Medium, Low

**3. Value Statement:** A brief description of the value the task will provide to the end-user.

Example: "So that I can access my account information."

**4. BV (Business Value**): A numerical value assigned to the task to represent its business value.

Example: 1-10, where 10 is highest value

**5. CP (Complexity Points):** A numerical value assigned to the task to represent its complexity.

Example: 500,200,100,50,20,10, where 500 is highest complexity

**6. Acceptance Criteria:** A set of conditions that must be met for the task to be considered complete.

Example: "The system shall authenticate the user and redirect them to the dashboard."

**Example of a complete User Story:**

**"**As a user, I want to login to the system (Task) so that I can access my account information (Value Statement).

Priority: High

BV: 500,

CP: 5

Acceptance Criteria: The system shall authenticate the user and redirect them to the dashboard."

**Question No – 15) What is product vision including Target group, needs, product, value?**

Product Vision is a concise statement that defines the overall direction and goals for a product. Here's a breakdown of the key components:

**1. Target Group**

- Who are the customers/users?

- What are their characteristics, needs, and goals?

**2. Needs**

- What problems or pain points does the target group face?

- What are their needs, desires, and expectations?

**3. Product**

- What is the product or solution?

- What features and functionalities will it have?

**4. Value**

- What value will the product provide to the target group?

- How will it solve their problems, meet their needs, or improve their lives?

**Question No – 16) What is DOR and DOD?**

DOR and DOD are two important concepts in Business Analysis and Requirements Management:

DOR (Definition of Ready)

The Definition of Ready (DOR) is a checklist or a set of criteria that ensures a requirement or a user story is well-defined, feasible, and ready for development.

Typically, a DOR includes criteria such as:

- Clear and concise description

- Well-defined acceptance criteria

- Estimated effort and complexity

- Prioritized and aligned with business goals

- Reviewed and approved by stakeholders

**DOD (Definition of Done)**

The Definition of Done (DOD) is a checklist or a set of criteria that ensures a requirement or a user story is fully implemented, tested, and meets the acceptance criteria.

Typically, a DOD includes criteria such as:

- Code is written and reviewed

- Unit tests are passed

- Integration tests are passed

- Acceptance criteria are met

- Documentation is updated

- Code is deployed to production

By having a clear DOR and DOD, teams can ensure that requirements are well-defined, feasible, and meet the acceptance criteria, which ultimately leads to delivering high-quality solutions that meet business needs.

**Question No – 17) Explain Agile process?**

The Agile process is a flexible and iterative approach to software development and project management. It focuses on continuous delivery, customer collaboration, and adapting to changes quickly.

**Key Principles of Agile:**

1. Iterative Development – Work is divided into small cycles called sprints (typically 1-4 weeks).
2. Customer Collaboration – Frequent feedback from customers ensures the product meets their needs.
3. Adaptability – Requirements can change based on feedback and evolving business needs.
4. Self-Organizing Teams – Teams work independently and take ownership of tasks.
5. Continuous Improvement – Regular reviews and retrospectives help enhance efficiency.

**Agile Frameworks:**

* Scrum – Uses sprints, daily stand-ups, and roles like Scrum Master & Product Owner.
* Kanban – Focuses on workflow visualization and limiting work-in-progress.
* SAFe (Scaled Agile Framework) – Used for large organizations managing multiple Agile teams.

**Conclusion:**

Agile helps teams deliver high-quality products faster by breaking work into small, manageable pieces and continuously improving based on real-world feedback. It is widely used in software development, product management, and other industries.

**Question No – 18) What is difference between sprint backlogs & Product backlogs?**

|  |  |
| --- | --- |
| **Product Backlog** | **Sprint Backlog** |
| A prioritized list of all features, enhancements, and fixes for the product. | A subset of the product backlog items selected for a specific sprint. |
| Covers the entire project or product vision. | Focused on a single sprint (1-4 weeks). |
| Managed by the Product Owner. | Managed by the Development Team. |
| Continuously refined and updated throughout the project. | Updated daily during the sprint if needed. |
| Can change frequently based on business needs and feedback. | Fixed for the duration of the sprint (no changes mid-sprint). |
| Defines long-term goals and overall requirements. | Defines short-term deliverables for immediate development. |

**Question No – 19) what is Product grooming?**

Product Grooming, also known as Backlog Grooming or Refinement, is the process of continuously reviewing, prioritizing, and updating the Product Backlog to ensure it is well-organized, clear, and ready for future sprints.

**Key Activities in Product Grooming:**

1. Prioritization – Reordering backlog items based on business value and urgency.
2. Clarification – Refining requirements, breaking down large items into smaller tasks.
3. Estimation – Assigning effort estimates (e.g., story points) to backlog items.
4. Adding or Removing Items – Ensuring the backlog is relevant and up to date.
5. Aligning with Business Goals – Making sure the backlog reflects customer and stakeholder needs.

**Who is Involved?**

* Product Owner (leads grooming sessions)
* Scrum Team (provides input on feasibility and effort)
* Stakeholders (offer business insights)

**Conclusion:**

Product Grooming helps teams stay prepared for upcoming sprints, reducing last-minute confusion and improving development efficiency.

**Question No – 20 – Which reporting Tools we will use for generating reports**.

For generating reports on loan applications, the following reporting tools can be used:

1. **Microsoft Excel**
* Features: Customizable templates, data analysis, pivot tables, charts, and graphs.
* Use Case: Ideal for ad-hoc reporting, simple data manipulation, and visualizing data in tables and charts.
1. **Microsoft Power BI**
* Features: Interactive dashboards, advanced data visualization, real-time data updates, and integration with multiple data sources.
* Use Case: Suitable for creating dynamic, interactive reports and dashboards with drill-down capabilities.
1. **Tableau**
* Features: Data visualization, interactive dashboards, data blending, and advanced analytics.
* Use Case: Best for creating complex visualizations and interactive reports with intuitive drag-and-drop interfaces.