1. What is the difference between Brainstorming and JAD Sessions?

|  |  |  |
| --- | --- | --- |
| **Feature** | **Brainstorming** | **JAD (Joint Application Development)** |
| **Definition** | A technique used to generate a wide range of ideas by encouraging open discussion among participants. | A structured session where business and IT stakeholders collaborate to define system requirements and solutions. |
| **Purpose** | Idea generation, problem-solving, and innovation. | Requirement gathering, solution design, and consensus-building. |
| **Participants** | Anyone involved in the problem domain (e.g., business analysts, developers, end-users, and stakeholders). | Key stakeholders such as business analysts, system architects, developers, SMEs, and decision-makers. |
| **Structure** | Unstructured or semi-structured; free-flowing discussion. | Highly structured, with predefined agendas, facilitators, and documentation. |
| **Outcome** | A list of ideas, possible solutions, or approaches. | Detailed business and system requirements, functional specifications, and agreed-upon action plans. |
| **Facilitation** | Can be led by a moderator but is generally open-ended. | Led by a JAD facilitator who ensures discussions remain focused and productive. |
| **Time Commitment** | Short (a few hours to a day). | Longer (several days to weeks, depending on project complexity). |
| **Documentation** | Informal notes or mind maps. | Formal documentation, including BRD, FRD, RTM, and meeting minutes. |
| **Example Usage** | Generating ideas for a new product feature, process improvement, or marketing strategy. | Gathering detailed requirements for a new banking application, healthcare system, or enterprise software. |

1. **Why Document Analysis is one of the compulsory techniques we use in a Project? Justify**

* **Information Gathering**- Documents contain valuable information, insights and data that can be crucial for understanding the project context, requirement, scope and objectives. Analyzing documents helps project team gain a comprehensive understanding of the project’s background.
* **Requirement clarification**- Ensure a clear understanding of project goal to prevent miscommunication.
* **Risk management-** identify potential challenges and develop strategies to mitigate them.
* **Legal and regulatory compliance**- many projects need to adhere to legal and regulatory standards. Analyzing relevant documents helps ensure that the project align with these requirements, avoiding legal issues and potential penalties.
* **Historical context**- Learn from past projects success and challenges
* **Stakeholder alignment**- project involve multiple stakeholders with varying interests and perspectives. Analyzing documents related to stakeholder preferences, concerns and expectations helps in aligning everyone’s goal
* **Scope Definition**- clearly outline project scope to manage expectations
* **Communication strategy**- use documents for effective intra-team and inter-team communication
* **Change management**- evaluate impacts of changes to make informed decisions.
* Decision making
* Quality assurance

1. **In Which Context we will use Reverse Engineering**

Reverse engineering is used when we need to analyze an existing system, application, or process to understand its structure, functionality, and dependencies. Below are the key contexts where it is applied

* **Legacy System Analysis** – Understand old systems without proper documentation.
* **System Migration** – Extract business logic for seamless migration.
* **Compliance & Auditing** – Ensure adherence to regulatory requirements (e.g., HIPAA, BASEL, AML).
* **Software Maintenance & Debugging** – Fix issues in undocumented systems.
* **Data Extraction & Integration** – Understand database structures for integration.
* **Competitor Analysis** – Analyze competitor products (ethically) for benchmarking.
* **Security & Vulnerability Assessment** – Identify security loopholes.
* **Business Process Improvement** – Optimize workflows and automate redundant tasks.
* **Extracting Business Rules from Code** – Retrieve logic embedded in scripts.
* **Legacy Code Refactoring** – Improve performance of outdated applications.
* **Reverse Engineering APIs** – Integrate undocumented APIs into new systems.

1. **What is the difference between Brainstorming and Focus Groups?**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Brainstorming** | **Focus Groups** |
| **Definition** | A technique used to generate creative ideas and solutions through open discussion. | A structured discussion where a small group provides feedback on a specific topic, product, or idea. |
| **Purpose** | Idea generation, problem-solving, and innovation. | Gathering opinions, preferences, and feedback from targeted users or stakeholders. |
| **Participants** | Diverse team members, stakeholders, or experts. | Selected individuals from a target audience (e.g., customers, end-users). |
| **Structure** | Informal or semi-structured; free-flowing discussions. | Structured; guided by a facilitator with predefined questions. |
| **Outcome** | A wide range of creative ideas and potential solutions. | Insights, perceptions, and feedback for decision-making. |
| **Facilitation** | Can be open-ended, often led by a moderator. | Led by a facilitator who ensures structured discussion. |
| **Use Case** | Generating new features for a product, process improvements, or solving business problems. | Understanding user needs, preferences, and concerns about a product or service. |
| **Example** | Generating ideas for improving a **credit risk assessment model** in BFSI. | Collecting feedback from **healthcare insurance customers** about a new digital claims process. |

1. **Observation Technique – Explain both Active and Passive approaches**

Observation techniques are commonly used in research, usability testing and requirement gathering to gather data by directly observing individuals, processes or systems. There are two main approaches to observation: active and passive.

1.Active Observation- In active observation, the observer actively engages with the participants, or the environment being observed. They may interact with participants, ask questions or guide the observation process.

2.Passive Observation- In passive observation, the observer takes a more non-intrusive and hands-off approach. They simply observe and record the behaviors, activities or events without directly interacting with the participants or influencing the observed context

1. **How do you conduct the Requirements Workshop?**

A Requirements Workshop is a structured meeting where stakeholders collaborate to define, refine, and validate project requirements. It ensures clarity, alignment, and stakeholder buy-in.

**Step 1: Planning the Workshop**

* Define Objectives: Identify the purpose (e.g., gather new requirements, validate existing ones, resolve conflicts).
* Identify Participants: Include key stakeholders such as business users, SMEs, developers, testers, and project sponsors.
* Prepare Materials: Gather BRDs, FRDs, process flows, existing system documentation, and business rules.
* Set the Agenda: Define topics, discussion points, and expected outcomes.
* Select a Facilitator: A business analyst or project manager ensures smooth discussion and captures key points.

**Step 2: Conducting the Workshop**

* Kick-off & Introductions: Explain the purpose, scope, and expected outcomes.
* Elicit Requirements: Use techniques like brainstorming, JAD sessions, process mapping, and use case modeling.
* Facilitate Discussions: Ensure all stakeholders participate, resolve conflicts, and clarify ambiguities.
* Capture Requirements: Document functional, non-functional, data, and compliance requirements in BRD, FRD, or User Stories.
* Validate & Prioritize: Align with business goals, feasibility, and impact analysis.

**Step 3: Post-Workshop Follow-Up**

* Document Outcomes: Summarize key requirements, action items, and decisions.
* Get Stakeholder Sign-off: Validate and obtain approval on documented requirements.
* Refine & Update Artifacts: Update BRD, FRD, RTM, and Jira/Confluence repositories.
* Plan Next Steps: Identify pending discussions, feasibility studies, or prototyping needs.

1. **In which context, Interview Technique can be conducted by a BA? How may approaches are there in conducting Interviews? (Structured – Unstructured) Explain them. Explain the difference between Open Ended Questions and Closed ended**

Interview Technique can be used to verify the fact, clarify ambiguity, trigger enthusiasm, engage end user, identify requirements, and the opinion and ideas. It is used to get more information from the people in an formal or informal setting by asking questions and documentation the responses. It involves direct communication with the individuals or a group of people who are part of an initiative; there are two basic types of interview. They are,

**Structured Interview**- in which the interviewer has the predefined set of questions. It is a structured way of interview.

**Unstructured Interview**- in which the interviewer does not have the predetermined set of questions ad it may vary based on the stakeholder response and interactions.

**Open Ended Questioned**- Open- ended questions are those that provide respondents with a question prompts and provides them a space in which to construct their own response.

**Close Ended questions**- Often the answer is a single word (e.g Yes or No) or less common a short phrase. You are not looking for an explanation or an elaboration to the question in the answer given to the question

1. **Questionnaire Technique – Where we will use? Give one example**

The Questionnaire Technique is used when a Business Analyst needs to collect information from a large group of stakeholders efficiently. It is useful in:

Requirement Gathering: Capturing business needs from multiple users.

User Feedback & Surveys: Assessing customer satisfaction, product usability, or process effectiveness.

Market Research: Understanding trends, user behavior, or preferences.

Compliance & Risk Assessments: Ensuring adherence to regulations like HIPAA, BASEL, or AML.

Post-Implementation Reviews: Collecting insights on system performance and user experience.

Example: In a BFSI project, a questionnaire can be used to survey branch managers about the challenges in the loan approval process to identify automation opportunities.

1. **How to Sort the Requirements – Where we will use? Give one example**

When all the requirements are gathered there are chances of redundancy in those requirements so basically all the scattered requirements are put together and the repetition of requirements are removed which is known as sorting of requirements. The process for sorting is :

1. Identification of requirements.

2. Dividing identified requirements into functional and non-functional requirements.

3. If identified requirements are similar then they are put together and removed.

We will sort the requirementsin two ways such as functional requirements and Non-functional requirements.

Functional requirements define a function that a system or system element must qualified to perform and must be documented in different forms. The functional requirements describe the behavior of the system as it correlates to the system’s functionality.

Examples of Functional requirements are authentication, business rules, Audit tracking, certification requirements, transaction correction, etc.

Non- functional requirements are not related to software’s functional aspect. They can be the necessities that specify the criteria that can be used to decide the operation instead of specific behaviors of the system.

Example- usability, reliability, security, storage, cost, flexibility, configuration, performance, legal or regulatory requirements, etc

Example:

In a **BFSI project**, sorting regulatory requirements based on **compliance deadlines** (e.g., BASEL reporting mandates must be addressed before optional feature enhancements).

1. **Prioritize the Requirements – –Where we will use? Give one example**

Where to Use: Prioritizing requirements is crucial in any project, particularly when there are limited resources or when requirements are competing in terms of importance. It is commonly used in the following contexts:

Product Development: When a product has many features or requirements, prioritization helps determine which features should be built first to maximize value for stakeholders.

Agile Projects: In Agile methodologies, where iterative development and flexibility are key, prioritizing requirements ensures that the team focuses on delivering the highest-value features first.

Project Management: When working with constrained timelines and budgets, it helps the team deliver the most important features within the available resources.

Business Analysis: To manage expectations and provide clarity on which features are essential to meet business objectives versus those that can be deferred.

How to Prioritize:

* MoSCoW Method (Must have, Should have, Could have, Won’t have)
* Kano Model (Basic needs, Performance needs, Excitement needs)
* Value vs Effort Matrix (High value, low effort vs low value, high effort)
* Story Mapping (Breaking down user stories to prioritize functionality)

Example:

Let’s consider an Employee Loan Management System. The project team has gathered a variety of requirements from HR, Accounts, and employees. However, they cannot implement everything in the initial phase due to time constraints. Here's how the requirements can be prioritized:

Requirements List:

1. Employee Loan Application Form
2. Loan Approval Workflow
3. Integration with Payroll System
4. Employee Loan Dashboard
5. Loan Repayment Schedule and Notifications

Prioritization:

* Must Have:
  + Employee Loan Application Form (critical for loan requests)
  + Loan Approval Workflow (needed to process loan applications)
* Should Have:
  + Loan Repayment Schedule and Notifications (important for the employee to track payments)
* Could Have:
  + Employee Loan Dashboard (useful but not essential for the initial launch)
* Won't Have:
  + Integration with Payroll System (can be implemented in later phases; does not impact the first release significantly)

1. **Weekly status reporting – How we will drive?**

To drive weekly status reporting effectively, follow these steps:

1. Set Clear Reporting Expectations

* Define the frequency and format of the report (weekly, same day/time each week).
* Outline the key components to be included (e.g., completed tasks, ongoing tasks, issues, risks, upcoming activities).

2. Gather Data from Team Members

* Request updates from all team members and relevant stakeholders.
* Ensure the data is accurate, concise, and includes measurable progress or outcomes.

3. Create a Consistent Template

* Design a simple, structured template for the report. Key sections might include:
  + Completed Tasks: Tasks that were finished during the week.
  + Ongoing Tasks: Work in progress, including current status and % completion.
  + Upcoming Tasks: Tasks to be completed next.
  + Issues/Risks: Any challenges that may affect progress.
  + Next Steps/Action Items: Immediate follow-up required.

4. Analyze & Summarize

* Review the data to ensure clarity and completeness.
* Summarize the key points in a way that can be quickly understood by stakeholders.

5. Distribute the Report

* Send the report to relevant stakeholders such as project managers, team leads, or department heads.
* Use tools like email, Jira, Confluence, or project management platforms to share the status report.

6. Review and Address Feedback

* After distribution, collect feedback from stakeholders and adjust the format/content of future reports as needed.
* Ensure that action items and issues are tracked and followed up on.

7. Ensure Accountability

* Assign ownership of specific tasks or action items and ensure they are followed up on in the next report cycle.
* Track progress and make sure any roadblocks or delays are identified early.

8. Use Reports for Decision Making

* Use the weekly reports to review progress against project goals and timelines.
* Make informed decisions regarding resource allocation, risk management, or priority shifts based on the report.

1. **Meeting Minutes Document – prepare one Sample**

Meeting Title: Food Delivery System - Requirements Discussion  
Date: February 5, 2025  
Time: 11:00 AM – 12:00 PM  
Location: Virtual (via Zoom)  
Prepared by: Anjali Mehta (Business Analyst)  
Attendees:

* Lisa Wong (Product Manager)
* Tom Harris (Operations Manager)
* John Baker (Developer)
* Emily Carter (Marketing Manager)
* Anjali Mehta (Business Analyst)

Agenda:

1. Discussion of food delivery system requirements
2. Customer interface features
3. Delivery process flow
4. Integration with payment gateway
5. Timeline and next steps

Discussion Points:

1. Food Delivery System Requirements
   * Key Points:
     + Lisa outlined the key features for the system: customer registration, menu browsing, order placement, payment, and real-time delivery tracking.
     + Delivery personnel will be able to receive order details and track delivery status.
     + Customers will get push notifications for order updates.
     + Action: Anjali to create a detailed requirement document and circulate by February 7, 2025.
2. Customer Interface Features
   * Key Points:
     + Tom emphasized the need for an easy-to-use interface for customers.
     + Features like filtering menu items, selecting delivery time slots, and tracking orders in real-time were discussed.
     + Action: John to work on wireframes for the customer interface and share them by February 9, 2025.
3. Delivery Process Flow
   * Key Points:
     + Tom explained the delivery flow: order placed -> restaurant receives the order -> food is prepared -> delivery personnel picks it up -> delivery to customer.
     + Integration with maps for route optimization will be essential.
     + Action: Anjali to document the delivery process flow by February 8, 2025.
4. Integration with Payment Gateway
   * Key Points:
     + Emily requested integration with popular payment gateways (e.g., PayPal, Stripe) for seamless payments.
     + Action: John to research and recommend the best payment gateway integration options by February 10, 2025.
5. Timeline and Next Steps
   * Key Points:
     + Lisa set a tentative timeline for the initial prototype: system architecture by February 20, 2025, with the first version of the app ready by March 5, 2025.
     + Action: All team members to review the requirements document and provide feedback by February 7, 2025.
6. **Change Tracker – Document - – prepare one Sample**

A Change Tracker Document is used to track and manage changes to project requirements, scope, or deliverables throughout the project lifecycle. It helps ensure that all changes are documented, reviewed, and approved by stakeholders to avoid scope creep and keep the project on track.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Change Request ID** | **Change Request Title** | **Date of Change** | **Requester** | **Description of Change** | **Impact of Change** | **Priority** | **Status** | **Approver** | **Implementation Date** | **Comments** |
| CR-001 | Update Loan Eligibility Criteria | 3-Feb-25 | HR Department | The eligibility criteria for employee loans need to be revised to include additional factors. | The new eligibility check will require changes in the system’s logic and delay implementation. | High | Approved | John Doe | 10-Feb-25 | Additional resources needed for testing. |
| CR-002 | Extend Loan Repayment Period | 4-Feb-25 | Accounts Dept. | Extend the loan repayment period from 12 months to 18 months to accommodate employee needs. | Affects the loan repayment module, which will need code updates and re-testing. | Medium | In Progress | Jane Smith | 15-Feb-25 | Code changes are being implemented. |
| CR-003 | Modify Loan Approval Process | 5-Feb-25 | IT Department | Adjust the loan approval process workflow to include an additional review step. | Minor adjustments to the loan workflow module; will cause slight delay in loan approval processing. | Low | Pending Approval | John Doe | TBD | Pending stakeholder feedback. |

1. **Difference between Traditional Development Model and Agile Development Models**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Traditional Development Model (Waterfall)** | **Agile Development Model** |
| **Approach** | **Sequential and Linear** – Follows a fixed, step-by-step process. | **Iterative and Incremental** – Focuses on continuous improvement through repeated cycles (sprints). |
| **Process Structure** | Divided into distinct phases (e.g., Requirements, Design, Development, Testing, Deployment). | Phases (e.g., Planning, Design, Development, Testing) are conducted iteratively in cycles (sprints). |
| **Flexibility** | Low flexibility – Changes are difficult to accommodate after the planning phase. | High flexibility – Changes can be incorporated throughout the project. |
| **Documentation** | Heavy documentation required at each stage. | Minimal documentation – Focuses on working software over comprehensive documentation. |
| **Customer Involvement** | Limited customer involvement – Customer feedback is gathered mainly at the start and end. | Continuous customer collaboration and feedback are integral throughout the project. |
| **Project Phases** | Defined and fixed phases that must be completed before moving to the next. | Short, iterative cycles (sprints) with frequent deliveries. |
| **Risk Management** | Risks are identified upfront and mitigated as the project progresses. | Risks are managed iteratively – they are addressed in each sprint. |
| **Delivery** | Final product is delivered at the end of the project. | Working software is delivered in increments, often at the end of each sprint. |
| **Cost Estimation** | Difficult to estimate accurately upfront – Changes lead to cost overruns. | Easier to estimate on a per-sprint basis, as development is broken into small parts. |
| **Time Management** | Fixed timeline – Delays in one phase affect the entire project timeline. | Flexible timeline – Each sprint is short, and adjustments are made at the end of each iteration. |

1. **Explain Brainstorming Technique – Where to use? 5 Marks**

Brainstorming is a creative technique used to generate many ideas or solutions to a problem in a short period. It encourages free-thinking and collaboration, fostering an open environment where participants can share their ideas without judgment.

Where to Use Brainstorming:

1. Idea Generation in Product Development:  
   Brainstorming is useful in the early stages of product or service development to generate innovative ideas.  
   Example: Creating new features for a mobile app or coming up with ideas for a marketing campaign.
2. Problem-Solving Sessions:  
   When facing a challenge or roadblock, brainstorming allows teams to generate multiple potential solutions.  
   Example: Finding solutions for reducing operational costs or resolving technical issues in a software development project.
3. Marketing and Branding:

Brainstorming helps teams develop creative concepts for advertising, branding, and promotional activities.  
Example: Coming up with a new tagline, campaign theme, or brand identity for a company.

1. Team Collaboration and Innovation:  
   It promotes collaboration across different teams, encouraging diverse perspectives and ideas.  
   Example: Teams from different departments (e.g., sales, marketing, R&D) coming together to brainstorm new ideas for a product launch.
2. Event Planning and Organization:  
   It’s effective for planning large events, ensuring all aspects, such as venue, catering, and activities, are considered.  
   Example: Organizing a corporate event, where brainstorming is used to decide themes, activities, and guest speakers.
3. **What reports Accounts Departments will generate?**

Key reports the Accounts Department would generate for the Employees Loan Management System:

1. Loan Request Report: Tracks all loan requests (employee details, amount, status, rejection reasons).
2. Loan Approval Report: Lists approved loans with details (loan amount, terms, approval date).
3. Loan Disbursement Report: Tracks disbursement details (employee, amount, date, balance).
4. Loan Repayment Schedule Report: Details repayment schedule (employee, loan terms, frequency, start date).
5. Loan Repayment Report: Tracks repayments made (amount paid, outstanding balance, missed payments).
6. Outstanding Loan Balance Report: Displays remaining loan balances and due dates.
7. Loan Default and Arrears Report: Identifies overdue loans (missed payments, overdue amounts).
8. Loan Interest Report: Tracks interest on loans (rate, amount charged, total interest paid).
9. Employee Loan Summary Report: Overview of all loans (total amount, repaid, outstanding, status).
10. Financial Impact Report: Assesses financial impact (disbursed amount, repayments, overdue, payroll impact).
11. Loan Program Performance Report: Measures loan program success (applications, approvals, repayment rate).
12. Audit Report: Tracks loan transactions for compliance (approvals, disbursements, repayments).
13. **What is the structure of the message/mail communicated from the HR department to the employee in case the Loan is rejected?**

In case of a loan rejection, the HR department needs to communicate the decision professionally, while providing clear information regarding the reasons for the rejection. Here's a structured format for the message/mail:

Subject: Loan Request Status – Rejection Notification

Dear [Employee Name],

I hope this message finds you well.

We regret to inform you that your loan request has been rejected after careful consideration. Below are the details regarding the decision:

Loan Request Details:

* Loan Amount Requested: [Amount]
* Requested Loan Type: [Type of loan, e.g., personal, home, etc.]
* Request Date: [Date of submission]

Reason for Rejection:

* [State the specific reason(s) for rejection, e.g., insufficient tenure, ineligible salary, incomplete documentation, etc.]

We understand that this may be disappointing, and we encourage you to review the eligibility criteria or contact the HR department if you need further clarification or assistance. You may also consider reapplying in the future if your circumstances change.

Thank you for your understanding.

Best regards, [Your Name]  
[Your Position]  
HR Department  
TTS Company  
[Contact Information]

1. **What is the structure of the message/mail communicated from the HR department to the employee in case the Loan is approved?**

In case of a loan approval, the HR department needs to communicate the decision clearly, outlining the loan terms and providing the employee with all necessary details. Here's a structured format for the message/mail:

Subject: Loan Request Approval – [Loan Amount] Approved

Dear [Employee Name],

I hope this message finds you well.

We are pleased to inform you that your loan request for [Loan Amount] has been approved. Below are the details of your loan and the terms and conditions associated with it:

Loan Details:

* Loan Amount: [Loan Amount]
* Loan Type: [Type of loan, e.g., personal, home, etc.]
* Approval Date: [Approval Date]

Terms and Conditions:

* Interest Rate: [Interest Rate] (if applicable)
* Repayment Period: [Repayment Period] (e.g., 12 months, 24 months, etc.)
* Monthly Repayment Amount: [Monthly Amount]
* Repayment Start Date: [Start Date]
* Repayment Method: Salary deduction (or other agreed method)
* Other Terms: [Additional terms if any]

Please review the terms carefully. If you are in agreement with the terms and conditions, kindly confirm by replying to this email by [Confirmation Deadline]. Once confirmed, the loan will be processed and disbursed accordingly.

Should you have any questions or need further clarification, please do not hesitate to contact the HR department.

We wish you the best and thank you for being a valued part of TTS Company.

Best regards, [Your Name]  
[Your Position]  
HR Department  
TTS Company  
[Contact Information]

1. **Design a sample report on the Loans applications Received by the accounts department**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Employee Name** | **Employee ID** | **Loan Type** | **Requested Amount** | **Application Date** | **Status** | **Approval Date** | **Repayment Period** | **Monthly Repayment Amount** | **Rejection Reason (If Any)** |
| 1 | John Doe | E1001 | Personal | $5,000 | 2/1/2025 | Approved | 2/3/2025 | 12 months | $500 | N/A |
| 2 | Jane Smith | E1002 | Home Loan | $20,000 | 2/2/2025 | Pending | N/A | N/A | N/A | N/A |
| 3 | Alice Johnson | E1003 | Education | $10,000 | 2/2/2025 | Rejected | N/A | N/A | N/A | Insufficient salary |
| 4 | Michael Lee | E1004 | Personal | $8,000 | 2/3/2025 | Approved | 2/4/2025 | 18 months | $450 | N/A |
| 5 | Sarah Davis | E1005 | Car Loan | $12,000 | 2/4/2025 | Pending | N/A | N/A | N/A | N/A |

1. **Which reporting Tools we will use for generating reports.**

Reporting Tools for Generating Reports:

1. Microsoft Excel

Simple, ad-hoc reports, data analysis, charts, and pivot tables.

1. Power BI

Interactive dashboards, real-time data, and business intelligence reports.

1. Tableau

Advanced visual reports and dashboards, interactive data exploration.

1. QlikView/Qlik Sense

Self-service reporting, interactive dashboards, in-memory analytics.