Question 1: BPM

Answer 1:

**Goal:** To provide a convenient and accessible platform for farmers in remote areas to purchase agricultural products online.

**Inputs:** Product details from manufacturers, Farmer registration information, Payment details, Shipping addresses

**Resources:** Online platform, Logistics and delivery partners, Software

**Outputs:** Product deliveries to farmers, Online catalog of agricultural products, Sales Report

**Activities:** Customer Support, Delivery, Registration, Inventory management

**Value Created to End Customer:** Convenience, Wide range, Timely delivery

Question 2: SWOT Analysis

Answer 2:

**STRENGTHS:**

* **Talent Pool:** APT IT SOLUTIONS has a talented pool of skilled Java developers, testers, network and database administrators.
* **Stakeholder Involvement:** Direct involvement of Stakeholders ensures that the platform meets their specific needs and requirements

**WEAKNESSES:**

* **New Concept Venture:** Online agriculture store is a new concept, requiring careful planning and execution.
* **Logistics and Delivery:** Reaching remote areas with efficient and timely delivery can be challenging.
* **Budget Constraints:** Managing the project within the 2 crore INR budget is crucial.
* **Tight Timeframe:** 18-Month tight Timeline for development and deployment, especially with new platform development.

**OPPORTUNUTIES:**

* **Growing Demand:** Increasing need for accessible agricultural inputs in remote areas
* **Expansion:** Successful implementation can lead to expansion to other market and product categories.
* **Expanding Market:** The platform can potentially be scaled to serve a larger market beyond the initial target audience.

**THREAT:**

* **Technical Issues:** Potential for technical glitches, cyberattacks, or data breaches that could disrupt operations
* **Competition:** Existing players in the agricultural product market might pose a threat.
* **Internet Connectivity Issues:** Limited internet access in remote areas may hinder platform usage

Question 3: Feasibility Study

Answer 3:

A feasibility study is an assessment of a project idea to determine if it’s practical or not.

* **Technology:** Based on the database server, Payment gateways, API’s, Security
* **Hardware:** Based on the Servers, Network infrastructure, Database and storage Requirements
* **Software:** Database management system, Operating system, Web Server, Payment gateway Software
* **Resources:** Availability of skilled Java developers, testers, database administrators, and network administrators.
* **Budget:** Total budget allocated: ₹2 Crores INR: Various costs involved such as Development Costs, Maintenance Costs
* **Time Frame:** Project duration: 18 months

Question 4: Gap Analysis

Answer 4:

**AS-IS (Current State):**

* Farmers face difficulties in procuring agricultural products.
* Farmers rely on local markets or middlemen for purchasing agricultural products.
* Limited access to information on product availability, pricing, and quality.
* Inefficient and time-consuming process for farmers to obtain inputs.
* Communication between farmers and manufacturers is inefficient.

**TO-BE (Desired State):**

* Farmers can directly purchase products from manufacturers through an online platform.
* Access to detailed information about products, including specifications, pricing, and reviews.
* Direct Communication between manufacturers and farmers
* Timely delivery of agriculture products
* Increased efficiency and convenience for farmers in obtaining products.

Question 5: Risk Analysis

Answer 5:

Risk analysis is the process of identifying and assessing risks that may threaten an organization’s success

* **Internal Risk:** Lack of Expertise like experience in e-commerce development, Team conflict, Poor project planning, monitoring, and control can lead to project failures.
* **External Risk:** Competition, Regulatory changes, Economic downturn can impact the project’s feasibility and profitability.
* **BA Risks:** Incomplete or Inaccurate requirements, Lack of stakeholder engagement, Lack of Domain knowledge, Change in requirements
* **Project based Risks:** Budget Overruns, Technology risks, Delays

Question 6: Stakeholder Analysis

Answer 6:

|  |  |  |  |
| --- | --- | --- | --- |
| Stakeholders  | Designation  |  Details  | RACI |
| Mr. Karthik | Delivery Head | Karthik@gmail.com | A/I |
| Mr. Vandanam | Project Manager | vandanam@gmail.com | R/A/I |
| Ms. Juhi | Senior Java Developer  | Juhi@gmail.com | R/I |
| Mr. Teyson | Java Developer | teyson@gmail.com | R/I |
| Ms. Lucie | Java Developer | lucie@mail.com | R/I |
|  Mr. Tucker | Java Developer | Tucket@gmail.com | R/I |
| Mr. Bravo | Java Developer  | bravo@gmail.com | R/I |
| Mr. Mike | Network Admin | mike@gmail.com | R/I |
| Mr. John | DB Admin | john@gmail.com | R/I |
| Mr. Jason | Tester | jason@gmail.com | R/I |
| Ms. Alekya | Tester | Alekya@gmail.com | R/I |
| Mr. Pandu | Financial Head | pandu@gmail.com | C/I |
| Mr. Dooku | Project Coordinator  | Dooku@gmail.com | C/I |
| Ben | Farmer | Ben@gmail.com | C/I |
| Peter | Farmer  | peter@gmail.com | C/I |
| Kevin | Farmer  | Kevin@gmail.com | C/ I |

Question 7: Business Case Document

Answer 7:

**Project Name: Online Agriculture Products Store**

 **Project Description:** This project aims to develop an online web/mobile application to facilitate farmers in remote areas to purchase agricultural products like fertilizers, seeds, and pesticides. The platform will connect farmers directly with manufacturers, enabling them to browse, select, and order products for delivery.

**Q. Why is this project initiated?**

The project is initiated to address the difficulties faced by farmers in remote areas in procuring essential agricultural inputs like fertilizers, seeds, and pesticides. These challenges include limited access, high costs due to intermediaries, and lack of information about product availability and quality.

**Q, What are the current problems?**

* High costs of agricultural inputs due to intermediaries
* Limited access to information about products
* Lack of transportation and logistics infrastructure
* Difficulty in accessing quality inputs
* Low agricultural productivity

**Q. With this project, how many problems could be solved?**

* Reducing the cost of agricultural inputs by eliminating intermediaries
* Providing farmers with access to information about products, prices, and availability
* Streamlining the purchasing process
* Improving access to quality inputs
* Enhancing agricultural productivity

**Q. What are the resources required?**

* Financial resources: The project budget is INR 2 Crores.
* Human resources: A team of skilled professionals, including project managers, developers, testers, and domain experts.
* Technological resources: Infrastructure, software, and hardware resources required for the development and deployment of the application.
* Stakeholder support: Support from key stakeholders, including farmers, suppliers, and government agencies.
* Logistics and delivery resources

**Q. How much organizational change is required to adopt this technology?**

The project requires minimal organizational change as it primarily involves the adoption of new technology. However, it may necessitate training for farmers and suppliers on using the online platform and building trust in online transactions.

**Q. What is the Time frame to recover ROI?**

The return on investment (ROI) is expected to be realized within 2-3 years as farmers experience increased productivity and income due to improved access to agricultural inputs.

**Q. How to identify stakeholders?**

Stakeholders can be identified through various methods, including interviews, surveys, focus groups, and stakeholder mapping exercises

Question 8: Four SDLC Methodologies

Answer 8:

**SDLC Methodoloies:** Software Development Life Cycle (SDLC) refers to a structured process for developing, deploying, and maintaining software applications. It involves a series of phases, each with specific goals and deliverables. SDLC Methodologies are processes and practices used by software development teams in order to successfully navigate the Software Development Life Cycle (SDLC).

Here are four prominent SDLC methodologies:

1. **Sequential Model (Waterfall Model):** The waterfall model arranges all the phases sequentially so that each new phase depends on the outcome of the previous phase. Conceptually, the design flows from one phase down to the next, like that of a waterfall, as each phase must be completed before moving to the next, with no going back.



Stages of waterfall Model:

* **Requirements:** Gather and document all system requirements.
* **Design**: Create the system architecture and specifications.
* **Implementation:** Develop and unit test small program units.
* **Integration & Testing:** Combine units and test the complete system.
* **Deployment**: Release the system to the customer or market.
* **Maintenance**: Fix issues and release updates/enhancements.
1. **Iterative Model:** With the Iterative Model, only the major requirements are known from the beginning. Based on these, the development team creates a quick and cheap first version of the software. Then, as additional requirements are identified, additional iterations of the software are designed and built. The Iterative Model involves developing software in small cycles, improving it with each iteration based on feedback.

****

Phases:

* Requirement Gathering
* Initial Design
* Prototype Development
* Testing
* Evaluation & Feedback
* Enhancements in Next Iteration
1. **Evolutionary Model:** The **Evolutionary Model** focuses on creating a working prototype and refining it based on user feedback. This approach focuses on rapid prototyping and continuous evolution of the product. It emphasizes early feedback and frequent releases to gather user input and refine the product iteratively.

 **Prototype 1 → Feedback → Prototype 2 → Refinements → Final Product**

Phases:

* Requirement Collection
* Prototype Development
* User Feedback & Refinements
* Final Product Development
* Deployment & Maintainance
1. **Agile Model:** Agile is the mainstream methodology used in modern software development, and expands its influence beyond coding into many aspects of product development, from ideation to customer experience. The Agile methodology breaks a project down into multiple cycles, each passing through some or all of the SDLC phases. The focus is on people and how they work together to get the project done. Agile calls for continuous collaboration between team members and stakeholders with regular cycles of feedback and iteration.

The Agile Manifesto’s 4 Core Values :

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



Question 9: SDLC Models

Answer 9:

1. **V Model:** The V Model is a software development process that visually represents the steps taken during the project. It’s an extension of the Waterfall model and emphasizes the importance of testing throughout the development lifecycle. The “V” shape illustrates the development phases descending on one side and the corresponding testing phases ascending on the other. Each development phase has a corresponding testing phase. This parallel structure ensures that testing is integrated into every stage, rather than being a separate activity at the end.

The Two Sides of the V:

Left Side: **Development Phases**

* Requirements Gathering: This is the foundation. The project’s goals, functionalities, and constraints are defined in detail. User stories, use cases, and other documentation methods are used. Deliverable: Requirements Specification Document
* System Design: The overall system architecture is designed, including hardware, software, network infrastructure, database design, and the system’s interaction with other systems. Deliverable: System Design Document
* Architectural Design: Focuses on the structure and components of the software. Modules, interfaces, data flow, and communication between components are defined. Deliverable: Architectural Design Document
* Module Design (Detailed Design): Each individual module or component is designed in detail. Algorithms, data structures, and interfaces within the module are specified. Deliverable: Module Specifications
* Coding: The actual code is written based on the detailed design specifications. This is where the software comes to life. Deliverable: Working Code

Right Side: **Testing Phases**

* Unit Testing: Individual modules or components are tested to ensure they function correctly in isolation. Developers typically perform this testing. Test Deliverables: Unit Test Cases, Test Results
* Integration Testing: Modules that have been unit tested are integrated together, and the interactions between them are tested. This verifies that the modules work correctly as a group. Test Deliverables: Integration Test Cases, Test Results
* System Testing: The entire system is tested as a whole to ensure it meets the requirements specified in the beginning. This is often done in a simulated environment. Test Deliverables: System Test Cases, Test Results
* Acceptance Testing: The system is tested by the end-users or clients in a real-world environment to determine if it meets their needs and expectations. This is the final stage of testing before the system goes live. Test Deliverables: Acceptance Test Cases, Test Results



**Pros**

* Early defect detection.
* Well-structured for large projects requiring strict validation.

**Cons**

* Not suitable for frequent requirement changes.
* More time-consuming than Waterfall.
1. **Rational Unified Process (RUP):** The Rational Unified Process (RUP) is a well-defined software development process framework that provides a structured approach to building software. It was originally developed by Rational Software, which is now part of IBM. RUP is known for being iterative, architecture-centric, use-case driven, and risk-focused.

RUP is based on a set of building blocks, or content elements, describing what is to be produced, the necessary skills required and the step-by-step explanation describing how specific development goals are to be achieved.

Four Project Life Cycle Phases

* Inception: agreement among the team and customer as to what will be built
* Elaboration: agreement within the team as to the architecture and design needed to deliver the agreed system behavior
* Construction: the iterative implementation of a fully functional system
* Transition: delivery, defect correction, and tuning to ensure customer acceptance



**Pros**

* Iterative, reduces risk with continuous feedback.
* Good documentation and role-based approach.

**Cons**

* More complex and requires expert teams.
* High management overhead.
1. **Spiral Model:** The Spiral Model is a software development process that emphasizes risk management. It’s an evolutionary approach that combines the iterative nature of prototyping with the controlled and systematic aspects of the waterfall model. The “spiral” represents the iterative nature of the process, with each loop or cycle representing a phase of the software project.

The spiral model gives more emphases placed on risk analysis. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation. A software project repeatedly passes through these phases in iterations (called Spirals in this model). The baseline spiral, starting in the planning phase, requirements are gathered, and risk is assessed. Each subsequent spiral builds on the baseline spiral.



**Pros**

* Handles changing requirements well.
* Reduces risk early in development.

**Cons**

* Expensive due to continuous iterations.
* Requires strong risk analysis skills.
1. **Scrum Model:** Scrum is an iterative and incremental Agile framework for managing software development. It’s designed to deliver working software frequently and adapt to changing requirements quickly. Unlike traditional SDLC models like Waterfall, Scrum emphasizes flexibility, collaboration, and customer involvement.

SCRUM can be implemented either at the beginning of the project or when you sense that project is falling behind schedule. This model exercises full Admin Power.

**Scrum Team:** Project resources are grouped as Scrum teams which comprises of Bas, Developers, Testers. Each Team size will on average be 7-8.

**Product Owner:** He will decide what needs to be in the product and will be responsible for how the product has to be. He will regularly interact with customers and Bas. This role may be played by BA or any person who worked for end users for a long time or customer himself

**Scrum Master:** He will monitor the performance of the team within the sprint. Team will raise all their issues to Scrum Master and he will run to look for answers. This role can be played by any person in team normally BA’s plays this role.

**Product burndown:** It shows how much work was left to do at the beginning of each sprint.

**Sprint**: This is the period that team decides to deliver their objective. Normally a sprint period will be for 2 weeks but may extend to 4 weeks



**Pros**

* High flexibility for changing requirements.
* Faster feedback and continuous improvement.

**Cons**

* Requires high team collaboration.
* Needs experienced teams to manage iterations effectively.

**While the Waterfall model might seem appealing due to the clear requirements and fixed constraints, the V model appears to be a more suitable choice for this project. As a BA I would be choosing V-model because:**

* **This project has clear and well-defined requirements.**
* **Quality is critical as it involves financial transactions for farmers.**
* **Early testing in V-Model will reduce post-release defects, making it more reliable.**

Question 10: Waterfall Vs V Model

Answer 10:

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.**  | **Features**  | **Waterfall Model**  | **V-Model** |
| 1.  | **Basic concept** | A linear, sequential approach where each phase (requirements, design, implementation, testing, deployment, maintenance) is completed before moving to the next | An extension of Waterfall, emphasizing testing at each development stage. Development and testing phases are parallel, forming a V shape. |
| 2. | **Testing Focus** | Testing occurs primarily after the development phase is complete. | Testing is integrated throughout the process, with each development phase having a corresponding testing phase |
| 3. | **Flexibility** | Highly rigid and inflexible. Changes after a phase is completed are difficult and costly.  | More flexible than Waterfall due to early testing and verification, but still less adaptable than agile models |
| 4. | **User Involvement** | Limited user involvement, mainly during the requirements gathering phase. | More user involvement compared to Waterfall, as testing and validation stages offer opportunities for feedback. |
| 5. | **Error Detection** | Errors are typically detected late in the development cycle, making them more expensive to fix.  | Errors are identified earlier due to parallel testing, reducing the cost and effort of fixing them.  |
| 6. | **Risk Management** | Risk management is limited, as risks are often identified late in the process. | Improved risk management due to early testing and verification, allowing for early mitigation of potential issues. |
| 7. | **Project Size** | Suitable for small to medium-sized projects with well-defined requirements. | Can be used for larger projects with evolving requirements, especially where testing is critical. |
| 8. | **Cost** | Generally lower cost compared to V-Model, as it requires less extensive testing and verification. | Higher cost due to the increased focus on testing and the need for specialized testing resources. |
| 9. | **Documentation** | Heavy emphasis on documentation, with detailed documentation required for each phase.  | Also emphasizes documentation, but the focus is on linking documentation to testing and verification activities.  |
| 10. | **Complexity** | Simpler to understand and implement, making it suitable for less complex projects. | More complex than Waterfall due to the integration of testing throughout the process. |
| 11. | **Maintenance** | Maintenance is typically a separate phase that occurs after deployment. | Maintenance is considered throughout the process, as testing and verification can help identify potential maintenance issues early on. |

Question 11: As a BA, state your reason for choosing one model for this project.

Answer 11: **As a BA I would be choosing V-Model** becauseThis project has clear and well-defined requirements, Quality is critical as it involves financial transactions for farmers, Early testing in V-Model will reduce post-release defects, making it more reliable.

* **Early Risk Mitigation:**  The V-Model’s parallel testing process allows for early identification of potential issues, whether they are related to functionality, usability, or performance. This early detection minimizes the risk of encountering major problems late in the project, which could impact the budget and timeline. For a project with a fixed budget of 2 crores and an 18-month duration, risk management is paramount.
* **Clearly Defined and Stable Requirements:** The project requirements are well-documented and unlikely to change since the stakeholders have a clear vision. The V-Model is ideal for projects with fixed requirements where major changes are not expected during development.
* **SME Preference:** The Subject Matter Experts (SMEs) have already suggested using the V-Model, showing their confidence in its structured approach. As SMEs play a crucial role in requirement validation and acceptance testing, the V-Model aligns with their expectations.

Question 12: Gantt Chart

Answer 12:



Question 13: Explain the difference between Fixed Bid and Billing projects

Answer 13:

1. **Fixed Bid Projects:** (sometimes called “Fixed Price”) A fixed bid project involves a pre-determined, set price for the entire project or a specific phase of it. The client agrees to pay this amount regardless of the actual time or resources required to complete the work. In other words, In a Fixed Bid Model, the project budget is determined upfront, and the service provider (in this case, APT IT SOLUTIONS) agrees to deliver the project within that budget. The client (Mr. Henry, Mr. Pandu, and Mr. Dooku’s committee) pays a fixed amount for the entire project, regardless of the actual time and resources expended.

**Pros**:

* Predictable cost for the client.
* Clear deadlines and deliverables.
* Lower risk for the client as the service provider absorbs cost overruns.

**Cons:**

* Less flexibility for changes.
* May lead to lower quality if vendors try to cut costs.

1. **Billing Projects: (**which often refers to “Time & Materials” billing) , particularly when referring to “Time & Materials,” involve charging the client based on the actual time and resources spent on the project. This typically includes hourly rates for labor and the cost of materials.

**Pros**:

* Flexible and adaptable to changes.
* Better suited for complex or ongoing projects.

**Cons**:

* Unpredictable costs for the client.
* No fixed deadline, which may lead to project delays.
* Requires active monitoring of time and expenses.

Question 14:

Answer 14:

|  |
| --- |
|  **Requirement** **Gathering Phase** |
| S.No. | **Tasks** | **Actionable Items** | **Start Time**  | **End Time**  | **Duration**  |
|  | Stakeholder Identification -Meeting  | Meeting to list down the stakeholders  | 10:00 AM  | 11:00 AM  | 1 hour |
|  | Client Interaction  | A ZOOM call to update the Client on the MOM  | 11:00 AM  | 12:30 PM | 1.5 hours |
|  | Documentation | Analyze existing documentation for insights | 12:30 PM  | 1:30 PM  | 1 hour |
|  | **Lunch break**  |  | **1:30 PM**  | **2:30 PM** |  **1 hour**  |
|  | Summarization | Summarize interview notes, identify key points | 2:30 PM  | 4:00 PM  | 1.5 hours |
|  | Requirements Sorting  | Working on the template  | 4:30 PM  | 5:30 PM  | 1 hour |
|  | Team Meeting  | Discussion on the day inputs  | 5:30 PM  | 7:00 PM  | 1.5 hours |

|  |
| --- |
|  **Requirement Analysis Phase**  |
| S.No. | **Tasks** | **Actionable** **Items**  | **Start** **Time**  | **End Time**  | **Duration**  |
|  | Requirement Categorization | Group collected requirements by category | 10:00 AM  | 11:30 AM | 1.5 hours |
|  | Use Case Diagram Creation | To visualize interactions b/w users and the system | 11:30 AM  | 1:30 PM  | 2 hours  |
|  | **Lunch break**  |  | **1:30 PM**  |  **2:30 PM** | **1 hour**  |
|  | User Story Creation | Write user stories based on identified use cases | 2:30 PM  | 3:30 PM  | 1 hour |
|  | BRD Drafting and Review | Documenting detailed requirements and review the BRD | 3:30 PM  | 6:00 PM  | 2.5 hours |
|  | RTM Setup | Setting up the RTM to track requirements throughout the project | 6:00 PM  | 7:00 PM  | 1 hour |

|  |
| --- |
|  **Design Phase** |
| **S.No.** | **Tasks** | **Actionable Items**  | **Start Time** | **End Time** | **Duration** |
|  | UI/UX Collaboration Meeting | Meet with UI/UX team to discuss user flows and initial design concepts | 10:00 AM | 11:30 AM  | 1.5 hours |
|  | Wireframe Review | Review wireframes created by the UI/UX team, provide feedback | 11:30 AM  | 12.30 PM | 1 Hour |
|  | User Flow Diagram Creation | Create user process flow diagrams | 12:30 PM | 1:30 PM  | 1 hour |
|  | **Lunch break**  |  | **1:30 PM**  | **2:30 PM**  | **1 hour**  |
|  | Prototype Feedback Session | Provide feedback on prototype | 2:30 PM  | 4:00 PM  | 1.5 hours  |
|  | Data Flow Diagram Creation | Map data flow for features | 4:00 PM  | 5:30 PM  | 1.5 hours |
|  | Design Documentation | Document design decisions, rationale | 5:30 PM  | 6:00 PM  | 0.5 hour |
|  | Finalize Design Documents | Finalize UI/UX design, review usability | 6:00 PM  | 7:00 PM  | 1 hour |

|  |
| --- |
|  **Development Phase** |
| **S.No.** | **Tasks**  | **Actionable Items**  | **Start Time**  | **End Time**  | **Duration**  |
|  | Developer Clarification | Answer developer requirement queries | 10:00 AM | 11:30 AM | 1.5 hours |
|  | Update Requirement Documents  | Adjust based on feedback | 11:30 AM | 1:30 PM  |  2 hours  |
|  | **Lunch break**  |  | **1:30 PM**  | **2:30 PM**  | **1 hour**  |
|  | API Documentation Review | Review API integration documents | 2:30 PM  | 3:30 PM  | 1 hour  |
|  | Track Development Progress | Monitor feature status | 3:30 PM | 5:00 PM | 1.5 hours |
|  | Status Meeting Updates  | Update requirement-related progress | 5:00 PM | 6:00 PM | 1 hour  |
|  | Feature testing support | Validate developed functionalities | 6:00 PM | 7:00 PM | 1 hour |

|  |
| --- |
|   **Testing Phase**  |
| **S.No.** | **Tasks** | **Actionable Items**  | **Start Time**  | **End Time**  | **Duration**  |
|  | Test Case Review | Review test cases for coverage | 10:00 AM | 11:30 AM | 1.5 hours |
|  | Validate Requirements | Ensure completeness | 11:30 AM | 12:30 PM | 1 hour  |
|  | Bug Review | Check reported issues | 12:30 PM  | 1:30 PM  | 1 hour  |
|  | **Lunch break**  |  | **1:30 PM**  | **2:30 PM**  | **1 hour**  |
|  | Test Execution | Run system test cases | 2:30 PM  | 4:00 PM  | 1.5 hours |
|  | Review Fixes  | Validate Bug resolutions | 4:00 PM  | 5:30 PM | 1.5 hours |
|  | Test Sign-off | Final testing approval | 5:30 PM  | 7:00 PM  | 1.5 hours  |

|  |
| --- |
|  **User Acceptance Testing Phase** |
| **S.No.** |  **Tasks**  | **Actionable Items**  | **Start Time**  | **End Time**  | **Duration**  |
|  | UAT Planning Meeting | Review UAT plan with stakeholders | 10:00 AM  | 11:00 AM  | 1 hour  |
|  | User Training | Train users for UAT testing | 11:00 AM  | 12:30 PM  | 1.5 hours |
|  | Conduct UAT | Gather user feedback | 12:30 PM | 1:30 PM  | 1 hour  |
|  | **Lunch break**  |  | **1:30 PM**  | **2:30 PM**  | **1 hour**  |
|  | Log UAT Defects  | Record usability issues | 2:30 PM  | 3:30 PM  | 1 hour  |
|  | Review Fixes with Team  | Validate bug resolutions | 3:30 PM  | 5:30 PM  | 2 hours  |
|  | Sign-Off UAT  | Get final approvals | 5:30 PM  | 7:00 PM  | 1.5 hours |

|  |
| --- |
|  **Deployment Phase** |
| **S.No.** | **Tasks**  | **Actionable Items**  | **Start Time**  | **End Time**  | **Duration**  |
|  | Deployment Readiness  | Ensuresystemsetup | 10:00 AM | 11:30 AM  | 1.5 hours |
|  | Deployment Testing | Validate deployment | 11:30 AM  | 12:30 PM  | 1 hour  |
|  | Monitor Live System | Track performance | 12:30 PM  | 1:30 PM  | 1 hour  |
|  | **Lunch break**  |  | **1:30 PM**  | **2:30 PM**  | **1 hour**  |
|  | Validate Live System | Ensure system stability | 2:30 PM  | 4:30 PM  | 2 hours  |
|  | Gather Initial Feedback | Collect post-launch issues | 4:30 PM  | 6:00 PM  | 1.5 hours  |
|  | Final Deployment Report | Document deployment details | 6:00 PM  | 7:00 PM  | 1 hour  |