Q1: BPM

 Goal:

* Create an online store to help remote farmers buy agricultural products.
* Through this Online Web / mobile Application, Farmers and Companies (Fertilizers, seeds, and pesticides manufacturing Companies) can communicate directly with each other.

Inputs:

* Application should be user friendly.
* Application should accept the product details from the manufacturers.
* Should be able to display it the farmers.
* Farmers should be able to browse through the products and select what they need.
* Farmers should be able to buy them and get them delivered.

Resources:

* Manufacturers,
* farmers,
* the platform (app/website),
* logistics partners,
* payment systems,
* development teams, and
* customer support.

Outputs:

* Farmers should be able to login/signup for app usage.
* Able to pay and get them delivered.

Activities:

* Farmers must be able to login/signup.
* Browse through the products and know the benefits and uses of the products published directly by the manufacturers.
* Should be able to pay for the item and update the delivery address.
* Invoices sent to registered emails and able to track the delivery.

Value to the end customer:

* Better availability of the product through the app.
* More choices for the farmer.
* Able to buy straight from the manufacturer.

Q2: SWOT Analysis

STRENGTHS

* One of its kind applications to a large base of customers in remote areas.
* Convenient in ordering from several vendors and vast product range from their homes being delivered to their choice.
* Ability to track deliveries.

WEAKNESS

* Hard to use the application due to little knowledge in using computers and phones.
* Internet issues with rural areas.
* Third party logistics and serviceability in rural areas.
* Third party manufacturers.

OPURTUNITIES

* Partnerships with local and international suppliers.
* NO limit in operations.
* Expanding technology into farming.
* Increase business for small suppliers and new manufacturers.
* Increase in demand for high quality and organic suppliers.

THREATS

* Supply chain disruptions.
* Competition from large e-commerce platforms and local suppliers.
* Cybersecurity risks.
* Change in government regulations.

Q3: Feasibility study

Market research:

* Mr. Henry initiated the project after engaging in several discussions with his friends who are farmers and understanding their challenges.
* The research ended up as one of a kind which has not been around yet.
* This will be benefiting the farmers as well as the companies to gain more customer base and Soonu will gain income by charging subscription fees from farmers and manufactures paying a commission on their products.

Hardware Feasibility:

For a web application on an e-commerce platform to run successfully, consider the following requirements.

* Web servers: determine the number of web servers required to handle traffic.
* Database servers: Ensure adequate server capacity for hosting databases to store product, user and other information.
* Cloud Infrastructure: Using AWS, Azure or Google cloud for scalability and flexibility. Other cloud services like Azure App, EC2 can be used to host the platform.
* Back up and disaster recovery: Plan for regular backups and critical data implement disaster recovery strategies to ensure business continuity.

Software feasibility:

* Technology Stack for Development:
	+ Java: The core technology for backend development. Mr. Karthik should ensure the development team is familiar with Java frameworks, especially Spring Boot for building scalable and secure applications.
	+ Use JavaScript frameworks like React.js or Angular for a dynamic user interface. The frontend can communicate with the Java backend via RESTful APIs.
	+ Database Management: Relational databases like MySQL or PostgreSQL for storing structured data (e.g., user profiles, orders, product catalogs).
	+ E-commerce Features: For handling shopping carts, payments, product listings, and customer management, consider integrating e-commerce frameworks or building these from scratch.
		- Payment Gateway Integration: Consider payment systems like Razorpay, Stripe, or PayPal for processing payments securely.
		- Shipping Integration: Integrate with third-party logistics APIs for shipping and tracking deliveries.
* IDE for Development: Tools like Eclipse or IntelliJ IDEA for Java development.
* Version Control System: Git and platforms like GitHub or GitLab for collaborative development.
* Continuous Integration/Continuous Deployment (CI/CD): Use Jenkins, GitLab CI, or CircleCI for automated testing and deployment.

**3. Trained Resources (Human Resource Feasibility)**

**Key Considerations:**

* **Java Development Team**:
	+ **Senior Java Developer**: Someone with significant experience in Java and enterprise applications.
	+ **Java Developers**: A team of developers skilled in Java, Spring Boot, Hibernate, and API integration.
	+ **Frontend Developers**: Experience with **React.js**, **Angular**, or **Vue.js** for creating a responsive and interactive user interface.
	+ **Database Administrators**: A DBA to handle the design, optimization, and management of databases (MySQL/PostgreSQL).
	+ **Testing Team**: Manual and automation testers familiar with the platform to test Java-based applications.
	+ **Project Manager**: Someone experienced in managing tech projects and guiding development timelines and resources.

**Other Key Roles**:

* **DevOps Engineer**: To handle server configuration, cloud deployments, CI/CD pipelines, and ensure scalability.
* **UI/UX Designers**: To design an intuitive, farmer-friendly interface for both web and mobile applications.
* **Security Specialists**: To ensure the platform is secure, particularly for sensitive data such as payment information.

**4. Budget Feasibility**

**Key Considerations:**

* **Total Budget**: ₹2 Crores INR (as per the project specification).
* **Infrastructure Costs**:
	+ **Cloud Hosting**: Depending on the provider (AWS, Google Cloud, Azure), monthly costs for hosting, storage, and bandwidth may vary. A typical budget could be ₹50,000–₹1,00,000/month for cloud services.
	+ **Licenses**: Any required software licenses, e.g., for enterprise-level databases or security tools.
	+ **Development Tools**: Costs for tools such as IDEs, version control platforms, or project management software (e.g., Jira, Trello).
* **Development Costs**:
	+ **Personnel Costs**: Salaries for Java developers, frontend developers, testers, project managers, and other resources. These can account for a significant portion of the budget, with salaries varying based on experience (typically ₹10–₹25 Lakh/year per senior developer).
	+ **Outsourcing**: If external developers or consultants are hired, those costs should be factored in.
* **Marketing and Launch Costs**: These would cover the platform’s advertising, outreach to farmers, SEO efforts, and launch promotions.
* **Contingency Fund**: A small portion (5-10%) of the budget should be reserved for unforeseen costs, especially during the later stages of development.

**Budget Allocation Example (₹2 Crores INR)**:

* **Personnel Costs**: ₹1.2 Crores INR (development team, QA, project management)
* **Cloud Infrastructure and Hosting**: ₹15–₹20 Lakh INR (based on scaling needs)
* **Licensing & Software Tools**: ₹5–₹10 Lakh INR
* **Marketing**: ₹25–₹50 Lakh INR
* **Miscellaneous and Contingency**: ₹20 Lakh INR

**5. Time Frame Feasibility**

**Key Considerations:**

* **Project Duration**: 18 months (as per the original plan).
* **Development Phases**: The project should be broken down into the following phases:
	1. **Planning & Requirement Analysis (1-2 months)**: Define functional requirements, technology stack, design, and architecture.
	2. **System Design & Prototyping (2 months)**: UI/UX design, database schema, API design, and architectural decisions.
	3. **Backend & Frontend Development (8-9 months)**: Simultaneous development of Java-based backend services and frontend user interface.
	4. **Testing & Quality Assurance (2-3 months)**: Manual testing, unit testing, integration testing, and performance testing.
	5. **Deployment & Launch (1-2 months)**: Initial deployment to the cloud, bug fixes, and beta testing.
	6. **Post-launch Support & Enhancements (2 months)**: Address any issues post-launch, gather user feedback, and make enhancements.
* **Milestone Tracking**: Set clear milestones to ensure each phase stays on track with deadlines.
* **Buffer Time**: Allow for a buffer in the schedule (1–2 months) for unexpected delays or challenges in development, especially for testing or complex integrations.

**Q4: Gap Analysis**

 A **gap analysis** identifies the current state, the desired future state, and the gaps between them. It helps determine what is required to move from the current state to the desired state. Below is a gap analysis for the Online Agriculture Product Store project.

**1. Current State (As-Is)**

* **Product Sourcing Difficulty**: Farmers in remote areas face difficulties in procuring essential agricultural products (fertilizers, seeds, and pesticides). This is due to limited availability and access to local stores or distributors.
* **Limited Technological Infrastructure**: Many farmers in rural areas may have limited access to the internet, smartphones, or a basic understanding of technology, which could hinder them from using an online platform for purchasing agricultural products.
* **Fragmented Communication**: Communication between farmers and manufacturers (fertilizer, seed, and pesticide companies) is not streamlined, resulting in inefficiencies. Farmers may rely on local intermediaries who charge additional fees or delay the process.
* **Lack of Centralized Platform**: There is no centralized online platform that brings together agricultural product suppliers (fertilizer, seed, and pesticide manufacturers) and farmers to simplify the purchasing process.
* **Local Market Dependence**: Farmers are heavily dependent on local suppliers who may not always offer the best prices or availability of specific agricultural products.
* **Manual Operations**: The current farming supply chain relies on manual methods, with paper records or informal communication, leading to inefficiencies.

**2. Desired State (To-Be)**

* **Centralized Online Platform**: A robust online platform where farmers can browse and order agricultural products like fertilizers, seeds, and pesticides directly from manufacturers. The platform should be easy to navigate, even for tech-averse users, with simple options to view, compare, and select products.
* **Seamless Transactions and Deliveries**: The platform will enable seamless transactions with secure payment options, and the logistics system will ensure timely delivery to remote locations. The farmers will have a transparent tracking system for their orders.
* **Product Availability**: Farmers will have access to a wide variety of products (fertilizers, seeds, pesticides) from multiple manufacturers. This will eliminate dependency on local stores and provide farmers with options for better prices and quality.
* **Mobile and Web Access**: The platform will be accessible on both mobile and web versions, accommodating farmers in areas with limited internet access. A user-friendly interface, with options for voice-based inputs, will help in overcoming technological challenges.
* **Real-time Communication with Manufacturers**: Direct communication channels between farmers and manufacturers will be available for inquiries about products, pricing, and availability. This would streamline the buying process.
* **Data-Driven Insights**: The platform can offer farmers insights into their purchases, provide crop-specific product recommendations, and track the effectiveness of fertilizers, seeds, and pesticides.
* **Automated Operations**: The purchasing process, order management, and inventory updates will be automated. Manufacturers will be able to update product catalogues and stock levels easily, which will reduce manual overhead.

Q5: Risk Analysis

**BA Risks**

* **Unclear Requirements**: Farmers and manufacturers may not fully articulate their needs.
* **Changing Requirements**: Farmers might request new features after development starts.
* **User Adoption**: Farmers in remote areas might struggle with the online system.
* **Data Security**: Sensitive data of farmers and manufacturers must be protected.
* **Regulatory Compliance**: Agricultural products may have legal restrictions.

**Project Risks**

* **Budget Overrun**: Additional features may increase costs beyond 2 Crores INR.
* **Technology Challenges**: Network issues in rural areas may affect accessibility.
* **Resource Availability**: Developers, testers, and admins need proper allocation.
* **Integration Issues**: The system must seamlessly manage different manufacturers' data.
* **Timeline Risks**: Delays in requirement gathering or testing may impact the 18-month schedule.
* **Logistical challenges**: Delivering products to regional areas can be highly.
* **Scalability Issues:** Platform may not be able to handle growth, especially if it expands to more regions and more users**.**

**Q6: Stakeholder Analysis (RACI Matrix)**

The major tasks for the SOONY Online Agriculture Product Store project include:

* Requirement Gathering
* Requirement Analysis
* System Design
* Development
* Testing & Bug Fixing
* Deployment & Go-Live
* Stakeholder Communication & Feedback
* Project Budget Management

The main stakeholders involved are:

* Mr. Henry (Business Owner)
* Mr. Pandu (Financial Head)
* Mr. Dooku (Project Coordinator)
* Peter, Kevin, Ben (Farmer Representatives & Stakeholders)
* Mr. Karthik (Delivery Head - APT IT SOLUTIONS)
* Mr. Vandanam (Project Manager)
* Business Analyst (Me)
* Java Developers (Juhi, Teyson, Lucie, Tucker, Bravo)
* Network & DB Admins (Mike, John)
* Testers (Jason, Alekya)

| **Task** | **Mr. Henry (Business Owner)** | **Mr. Pandu (Financial Head)** | **Mr. Dooku (Project Coordinator)** | **Farmers (Peter, Kevin, Ben)** | **Mr. Karthik (Delivery Head)** | **Mr. Vandanam (Project Manager)** | **BA (You)** | **Java Developers** | **Network & DB Admins** | **Testers** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Requirement Gathering** | I | I | A | C | I | R | R |  |  |  |
| **Requirement Analysis** | I | I | A | C | I | R | R |  |  |  |
| **System Design** | I | I | A | I | I | R | C | R | R |  |
| **Development** | I | I | I | I | I | R | C | R | R |  |
| **Testing & Bug Fixing** | I | I | I | I | I | R | C | C | C | R |
| **Deployment & Go-Live** | I | I | I | I | A | R | I | C | R |  |
| **Stakeholder Communication & Feedback** | A | A | R | C | C | I | R | I | I | I |
| **Project Budget Management** | I | A | R | I | A | I | I | I | I | I |

**Q7: Business case Document**

**1. Executive Summary**

**Project Title**: Development of an Online Agriculture Product Store

**Project Sponsor**: Mr. Henry (Founder of SOONY)

**Project Manager**: Mr. Karthik (Delivery Head, APT IT SOLUTIONS)

**Project Duration**: 18 months

**Budget**: INR 2 Crores

1. Overview

A business case is a package of information, analysis, and recommendations. It is prepared by the Senior Business Analyst (Sr. BA), Business Architect, and Pre-sales Team. This document aims to identify the principal stakeholders affected by the issue and provide a systematic approach to resolving business challenges.

2. Why is This Project Initiated?

* Farmers in remote areas face significant challenges in accessing essential agricultural supplies such as fertilizers, seeds, and pesticides.
* Traditional supply chain inefficiencies caused high costs, delays, and limited quality product availability.
* This project aims to develop an Online Agriculture Product Store to address these challenges and provide a seamless platform for farmers.

3. Current Problems

* Difficulty in procuring agricultural supplies in remote areas.
* High dependency on intermediaries leading to inflated costs.
* Lack of transparency in product availability and pricing.
* Limited digital access for farmers to connect with suppliers.

4. Solutions & Impact

* Increased Accessibility: Farmers can order essential supplies anytime, anywhere.
* Direct Communication: Farmers and manufacturers can interact without intermediaries.
* Cost Savings: Eliminating middlemen reduces costs for farmers.
* Efficiency: Streamlined procurement process with fast delivery options.
* Transparency: Farmers can compare prices, read reviews, and make informed purchasing decisions.

5. Resources Required

* Human Resources: Project Team including BA, Developers, Testers, PM, Network & DB Admins.
* Technical Resources: Web & Mobile Application Development, Cloud Hosting, Payment Gateway.
* Financial Resources: 2 Crores INR Budget Allocation.
* Training & Support: Training modules for farmers and customer support teams.

6. Organizational Change & Adoption

* Digital transformation of traditional agricultural procurement methods.
* Training programs for farmers to adopt new technology.
* Collaboration with manufacturers to integrate their supply chain.

7. Time to Recover ROI

* Expected ROI within 3-5 years through increased farmer participation and platform monetization.
* Cost recovery through service charges, partnerships, and advertising.

8. Stakeholder Identification:

 Stakeholders impacted by the issue and benefiting from the solution include:

* Farmers (End Users)
* Manufacturers (Fertilizers, Seeds, Pesticides Suppliers)
* Project Sponsors (Mr. Henry & Committee)
* APT IT SOLUTIONS Project Team (PM, Developers, Testers, Network & DB Admins)
* Government & Agricultural Agencies (Potential Partners)

9. Risks & Mitigation Strategies:

Business Analyst (BA) Risks:

* Risk: Incomplete or unclear requirements from stakeholders.
* Mitigation: Conduct multiple requirements gathering sessions and stakeholder interviews.
* Risk: Resistance from farmers due to lack of digital literacy.
* Mitigation: Provide training and video tutorials for farmers.

Project Risks:

* Risk: Delays in development due to unforeseen challenges.
* Mitigation: Implement Agile checkpoints within the V-Model to ensure continuous progress.
* Risk: Security threats and data breaches.
* Mitigation: Use encryption, two-factor authentication, and regular security audits.
* Risk: Low adoption rate among farmers.
* Mitigation: Launch awareness campaigns and offer initial incentives.

10. Conclusion

The SOONY Online Agriculture Product Store is a crucial initiative aimed at transforming the way farmers procure agricultural supplies. By leveraging technology, the platform will bridge the gap between farmers and manufacturers, ensuring a more efficient and cost-effective supply chain. With strategic planning, risk mitigation, and stakeholder collaboration, this project has the potential to revolutionize the agricultural sector and empower farmers.

Q8: Four SDLC Methodologies

**Software Development Life Cycle (SDLC)** is a structured approach to software development that outlines the steps involved in creating software applications. It assists in managing the complexity of developing and maintaining software, ensuring the project is completed on schedule, within budget, and meets the required standards.

There are different **SDLC methodologies** used in software development, each with its approach to organizing the phases of development. Here are the four methodologies

**1. Sequential (Waterfall Model)**

* **Definition:** A **linear** and **structured** development methodology where each phase is completed before moving to the next.
* **Phases:** Requirement Gathering → Design → Development → Testing → Deployment → Maintenance.
* **Best Suited For:** Projects with **clearly defined** and **stable requirements** (e.g., government projects, banking systems).
* **Pros:**
	+ Easy to manage and track progress.
	+ Well-documented approach ensures clarity.
* **Cons:**
	+ No flexibility to accommodate changes mid-development.
	+ Late discovery of defects increases rework costs.

**2. Iterative (RUP - Rational Unified Process)**

* **Definition:** A development process where the system is built **in increments**, with each iteration refining previous versions.
* **Process:** Initial requirements → Develop a working model → Get feedback → Improve in the next iteration.
* **Best Suited For:** Medium to large projects where changes are expected but must be controlled (e.g., ERP software, business applications).
* **Pros:**
	+ Issues are identified and fixed early.
	+ Allows continuous user feedback and refinement.
* **Cons:**
	+ Requires detailed initial planning.
	+ Can lead to scope creep if not managed well.

**3. Evolutionary (Spiral Model)**

* **Definition:** A **risk-driven** approach that combines elements of **iterative** and **sequential** models. Focuses on **prototyping** and **risk analysis** in each phase.
* **Phases:** Planning → Risk Analysis → Development → Evaluation (repeat in cycles).
* **Best Suited For:** Large, complex, and high-risk projects (e.g., defense systems, aerospace applications).
* **Pros:**
	+ Early identification and mitigation of risks.
	+ Frequent validation ensures product alignment with business needs.
* **Cons:**
	+ Requires skilled risk management.
	+ Can be costly due to repeated cycles.

**4. Agile (Scrum, Kanban, XP)**

* **Definition:** A **flexible and collaborative** approach that emphasizes **quick delivery** and **continuous user involvement**.
* **Process:** Product Backlog → Sprint Planning → Development → Review & Feedback → Deployment (repeat in short cycles).
* **Best Suited For:** Dynamic environments where requirements change frequently (e.g., e-commerce, mobile applications).
* **Pros:**
	+ Quick adaptability to changes.
	+ Continuous customer involvement improves user satisfaction.
* **Cons:**
	+ Can lead to poor documentation.
	+ Requires high team collaboration and discipline.

Q 9: My understanding of the models are as follows:

**1. Waterfall Model (Traditional & Sequential)**

* **Approach:** A **linear, step-by-step** development model where each phase must be completed before moving to the next.
* **Phases:** Requirements → Design → Development → Testing → Deployment → Maintenance
* **Best For:** Projects with **clear and stable** requirements (e.g., government, banking systems).
* **Pros:**
✔️ Simple and structured
✔️ Clear documentation and deliverables
* **Cons:**
❌ No flexibility; requirement changes are difficult to handle
❌ Late detection of defects increases rework costs

🔹 **My Understanding:** Waterfall works best when the requirements are well-defined and unlikely to change. However, for a dynamic project like the **SOONY Agriculture Product Store**, where user feedback is important, Waterfall may be too rigid.

**2. Rational Unified Process (RUP - Iterative & Structured)**

* **Approach:** **Iterative development** with four key phases—each phase refines the product step by step.
* **Phases:**
	1. **Inception** (Requirement gathering, business case)
	2. **Elaboration** (System design, risk analysis)
	3. **Construction** (Development, iterative builds)
	4. **Transition** (Deployment, user training)
* **Best For:** Medium-to-large projects with **changing requirements** (e.g., enterprise applications).
* **Pros:**
✔️ Balances structure with flexibility
✔️ Identifies risks early in the process
* **Cons:**
❌ Can be complex to implement
❌ Requires strong project management

🔹 **My Understanding:** RUP is **more flexible than Waterfall** but still follows a structured approach. It could work for the SOONY project if the requirements are moderately flexible but need controlled iterations.

**3. Spiral Model (Risk-Driven & Evolutionary)**

* **Approach:** A **risk-driven model** that combines iterative development with risk management.
* **Phases:**
	1. **Planning** (Requirements, objectives)
	2. **Risk Analysis** (Identifying risks, creating mitigation strategies)
	3. **Engineering** (Prototyping, development)
	4. **Evaluation** (Customer feedback, next iteration planning)
* **Best For:** Large, high-risk projects (e.g., aerospace, financial systems).
* **Pros:**
✔️ Early risk identification
✔️ Allows flexibility in changing requirements
* **Cons:**
❌ Expensive due to multiple iterations
❌ Requires strong risk assessment expertise

🔹 **My Understanding:** Spiral is a great option for **high-risk, high-budget** projects. If SOONY has **uncertain requirements and needs risk management**, this model could be useful. However, it may **increase costs** due to repeated prototyping.

**4. Scrum Model (Agile & Fast-Paced)**

* **Approach:** **Agile methodology** that follows short, iterative development cycles called **Sprints**.
* **Process:**
	1. **Product Backlog** (Requirements gathered in small tasks)
	2. **Sprint Planning** (Work divided into 1-4 week iterations)
	3. **Sprint Execution** (Development, Testing, Review)
	4. **Sprint Review & Retrospective** (Feedback, improvements)
* **Best For:** Projects with **frequent changes and evolving requirements** (e.g., startups, web apps).
* **Pros:**
✔️ Highly adaptable to changes
✔️ Continuous customer feedback improves the product
* **Cons:**
❌ Requires active team collaboration and discipline
❌ Lack of proper documentation can be a challenge

 **My Understanding:** **Scrum is ideal for fast-changing projects** like the **SOONY Online Agriculture Store**, where new features and improvements will be needed based on user feedback.

Q10: Waterfall vs V-Model

**Differences Between Waterfall Model and V-Model**

| **Feature** | **Waterfall Model** | **V-Model (Verification & Validation Model)** |
| --- | --- | --- |
| **Approach** | Sequential and linear | Sequential but follows a V-shaped structure |
| **Testing Phase** | Testing starts after development is completed | Testing is done at every stage (parallel to development) |
| **Verification & Validation** | Verification (design, coding) happens first, then validation (testing) | Both verification and validation occur simultaneously |
| **Flexibility** | Rigid; changes are difficult to accommodate | More flexible as testing happens early |
| **Risk Handling** | Risks are identified late in the development cycle | Risks are identified and mitigated early |
| **Error Detection** | Errors are detected late (during testing phase) | Errors are detected early due to parallel testing |
| **Cost of Fixing Bugs** | Higher, as issues are found later | Lower, as defects are identified earlier |
| **Best Suited For** | Small, well-defined projects with fixed requirements | Critical applications where early defect detection is essential (e.g., medical, aerospace) |
| **Complexity** | Simple and easy to follow | More complex due to parallel validation |
| **Documentation** | Heavy documentation required | Also requires detailed documentation but more testing-focused |

Q11: Justify your choice

As a **Business Analyst (BA)**, selecting the right **Software Development Life Cycle (SDLC) model** is crucial for the success of the **SOONY online agriculture product store**. Considering the project’s nature, **Scrum (Agile)** is the most suitable model. Here’s why:

**1. Requirements May Evolve Over Time**

* Farmers, manufacturers, and other stakeholders may refine their needs after seeing initial versions of the platform.
* Agile allows for continuous feedback and updates, making it ideal for a dynamic business environment.

**2. Faster Delivery with Incremental Releases**

* Instead of waiting **18 months** for a final product (as in Waterfall), Scrum allows us to **deliver features in sprints (every 2-4 weeks)**.
* Core features (e.g., product catalogue, order placement) can be built first, while additional enhancements (e.g., AI-based recommendations) can be added later.

**3. Continuous Stakeholder Involvement**

* The **SOONY committee (Mr. Henry, Mr. Pandu, Mr. Dooku)** and farmer representatives (Peter, Kevin, Ben) will provide regular feedback.
* Scrum ensures **stakeholder engagement** through sprint reviews and demos, improving the final product.

**4. Risk Management & Flexibility**

* Traditional models (Waterfall, V-Model) lock down requirements early, making mid-project changes expensive.
* Scrum embraces change, allowing adjustments based on real user feedback.

**5. Better Collaboration & Team Efficiency**

* **APT IT Solutions** has a **talented development team**, including Java developers, testers, and DB/network admins.
* Scrum fosters **daily communication (Daily Stand-ups), sprint planning, and retrospectives**, ensuring better coordination.

**6. Higher Product Quality**

* Testing is **continuous** in Scrum, unlike Waterfall, where testing happens at the end.
* Issues are identified **early**, reducing bug-fixing costs and improving reliability.

Q12 Grantt chart

Here's the **Gantt Chart** displaying the **tasks and their timelines** in months. Let me know if you need any modifications, such as adding dependencies or adjusting the duration for any phase! 😊

**Q13 Difference between fixed bid and billing.**

**1. Fixed Bid Project**

A **Fixed Bid project** is one where the **scope, cost, and timeline** are agreed upon before the project begins.

**Characteristics:**

* **Predefined Scope:** Requirements are well-documented and finalized before the project starts.
* **Fixed Cost:** The client pays a fixed amount, regardless of the actual time and effort spent.
* **Strict Timeline:** Deadlines and milestones are predetermined.
* **Less Flexibility:** Changes in scope may require contract renegotiation.
* **Higher Risk for Vendors:** The development company (APT IT Solutions in our case) bears the risk if the project takes more effort than estimated.

**Best for:**

✅ Well-defined projects with clear requirements and minimal expected changes.
✅ Government or enterprise projects where budget approval is required upfront.
✅ Short-term projects with predictable deliverables.

**Example:**

The **SOONY Online Agriculture Product Store** project is a **Fixed Bid project** with a budget of **2 Crore INR and an 18-month duration** under the CSR initiative.

**2. Billing (Time & Material) Project**

A **Time & Material (T&M) project** is one where the client pays for the actual time spent by resources and the materials used.

**Characteristics:**

* **Flexible Scope:** The project can evolve based on ongoing requirements.
* **Variable Cost:** Payment is based on the number of hours/days the team works.
* **Rolling Timeline:** Project duration is not fixed; it depends on requirements.
* **More Control for Clients:** The client can adjust priorities and add features on the go.
* **Lower Risk for Vendors:** The development company gets paid for actual work done.

**Best for:**

✅ Complex, long-term projects where requirements are likely to change.
✅ Agile projects that require continuous iterations and improvements.
✅ Startups or R&D projects where outcomes are uncertain.

**Example:**

A software product company developing a new **AI-based recommendation engine** and needing ongoing improvements might opt for a **T&M contract** instead of a fixed bid.

**Comparison Table**

| **Factor** | **Fixed Bid** | **Billing (Time & Material)** |
| --- | --- | --- |
| **Scope** | Fixed | Flexible |
| **Cost** | Fixed | Variable (based on actual effort) |
| **Timeline** | Predefined | Rolling (depends on progress) |
| **Risk for Vendor** | High (if effort exceeds estimate) | Low (client pays for effort) |
| **Flexibility** | Low (scope changes require contract updates) | High (easy to change scope) |
| **Best for** | Well-defined projects with little scope change | Evolving projects with dynamic requirements |

Q14: BA time sheets

**1. Design Timesheet (Requirement Gathering & Analysis Phase)**

| **Date** | **Task Description** | **Start Time** | **End Time** | **Total Hours** |
| --- | --- | --- | --- | --- |
| Day 1 | Kick-off meeting with stakeholders | 9:00 AM | 10:30 AM | 1.5 hrs |
| Day 1 | Gathering functional requirements | 10:30 AM | 12:30 PM | 2 hrs |
| Day 1 | Analyzing business needs | 1:30 PM | 3:30 PM | 2 hrs |
| Day 1 | Documenting BRD (Business Req. Doc) | 3:30 PM | 5:00 PM | 1.5 hrs |
| **Total Hours** |  |  | **7.5 hrs** |  |

**2. Development Timesheet (BA during Development Support)**

| **Date** | **Task Description** | **Start Time** | **End Time** | **Total Hours** |
| --- | --- | --- | --- | --- |
| Day 15 | Explaining requirements to developers | 9:00 AM | 10:30 AM | 1.5 hrs |
| Day 15 | Reviewing system design documents | 10:30 AM | 12:30 PM | 2 hrs |
| Day 15 | Answering queries & change requests | 1:30 PM | 3:30 PM | 2 hrs |
| Day 15 | Internal meeting with PM & Dev team | 3:30 PM | 5:00 PM | 1.5 hrs |
| **Total Hours** |  |  | **7.5 hrs** |  |

**3. Testing Timesheet (BA during Testing Phase)**

| **Date** | **Task Description** | **Start Time** | **End Time** | **Total Hours** |
| --- | --- | --- | --- | --- |
| Day 30 | Creating test scenarios | 9:00 AM | 10:30 AM | 1.5 hrs |
| Day 30 | Reviewing test cases with testers | 10:30 AM | 12:30 PM | 2 hrs |
| Day 30 | Performing functional validation | 1:30 PM | 3:30 PM | 2 hrs |
| Day 30 | Reporting & tracking defects | 3:30 PM | 5:00 PM | 1.5 hrs |
| **Total Hours** |  |  | **7.5 hrs** |  |

**4. UAT Timesheet (User Acceptance Testing)**

| **Date** | **Task Description** | **Start Time** | **End Time** | **Total Hours** |
| --- | --- | --- | --- | --- |
| Day 45 | Coordinating with users for UAT | 9:00 AM | 10:30 AM | 1.5 hrs |
| Day 45 | Validating test results | 10:30 AM | 12:30 PM | 2 hrs |
| Day 45 | Gathering feedback from farmers | 1:30 PM | 3:30 PM | 2 hrs |
| Day 45 | Updating UAT report & documentation | 3:30 PM | 5:00 PM | 1.5 hrs |
| **Total Hours** |  |  | **7.5 hrs** |  |

**5. Deployment & Implementation Timesheet**

| **Date** | **Task Description** | **Start Time** | **End Time** | **Total Hours** |
| --- | --- | --- | --- | --- |
| Day 60 | Final review of the system | 9:00 AM | 10:30 AM | 1.5 hrs |
| Day 60 | Coordinating with developers | 10:30 AM | 12:30 PM | 2 hrs |
| Day 60 | Supporting live deployment | 1:30 PM | 3:30 PM | 2 hrs |
| Day 60 | Updating stakeholders & reports | 3:30 PM | 5:00 PM | 1.5 hrs |
| **Total Hours** |  |  | **7.5 hrs** |  |