

COEPD - Traditional Development
Capstone Project 1 - Online Agriculture Products Store

ANSWERS

Q1: Identify Business Process Model for Online Agriculture Store

(Goal, Inputs, Resources, Outputs, Activities, Value Created to the End Customer)

Answer:

Goal:

To build an Online Agriculture Products Store (Web/Mobile Application) that bridges the gap between rural farmers and agriculture product companies, enabling seamless procurement of fertilizers, seeds, and pesticides.

Inputs:

1. Requirements from Stakeholders (Peter, Kevin, Ben, and farmers in rural areas)
2. Product catalogs from companies manufacturing fertilizers, seeds, pesticides
3. CSR initiative sponsorship from Mr. Henry
4. Technical team resources from APT IT Solutions
5. Existing pain points in procurement faced by farmers
6. Infrastructure (hardware/software/network) for application development

Resources:

Category	Resources
Human	BA, PM, Developers, Testers, Admins, SMEs, Stakeholders
Technological	Java-based framework, Databases, Network setup
Financial	CSR funding by Mr. Henry's company
Time	18 months project timeline
Knowledge-based	Domain expertise in agriculture and software development

Outputs:

1. Fully functional Web and Mobile Application for online agri-product ordering
2. Farmer login modules, product search with filtering, and order placement features
3. Company-side interfaces for product listing, pricing, and stock availability
4. Order tracking, payment gateway, and delivery coordination system
5. Reporting and feedback systems for continuous improvement

Activities:

1. Requirement Gathering & Analysis
2. Business Process Modeling
3. System Architecture Design
4. UI/UX Development
5. Back-end Java Development & API integration
6. Testing (Unit, Integration, UAT)
7. Deployment & Maintenance

Value Created to the End Customer (Farmers):

- Direct access to genuine agricultural products
- Time and cost savings by eliminating middlemen

- Increased transparency in pricing
- Better availability of seasonal products
- Improved productivity and crop health
- Empowerment of rural economy

Reasoning:

By identifying these aspects, we break down the entire solution lifecycle, showing how technology can be used to solve a real-world problem using structured business analysis techniques.

Q2: Mr. Karthik is doing SWOT analysis before he accepts this project. What aspects should he consider as Strengths, Weaknesses, Opportunities, and Threats?

Answer:

A SWOT Analysis is a strategic planning tool used to evaluate the Strengths, Weaknesses, Opportunities, and Threats involved in a project or business initiative. In the context of the Online Agriculture Products Store, here is a detailed SWOT analysis:

Strengths (Internal Positive Factors):

Strength	Explanation
Strong CSR backing from Mr. Henry	Financial support is secured, ensuring steady budget allocation throughout the 18-month duration.
Experienced Technical Team	APT IT Solutions has a skilled pool of developers, testers, admins, and a BA, which accelerates implementation.
Identified Stakeholders	Real users (Peter, Kevin, Ben) are already involved, ensuring accurate requirement gathering.
Focused Goal	Clear vision to solve a defined problem – streamlining product access for farmers.
Java as a proven tech stack	Reliable, scalable, and well-supported for web and mobile solutions.

Weaknesses (Internal Negative Factors):

Weakness	Explanation
Limited technical literacy among farmers	Many end-users may not be familiar with using mobile or web apps, requiring intuitive design or training.
No existing data or integration infrastructure	Since it's a fresh initiative, all data (products, users, companies) must be gathered from scratch.
Dependency on a CSR model	If any outside of the scheme things appear as issues in CSR continuity, project funding and sustainability may be at risk.
Geographic accessibility issues	Internet/network coverage might be poor in some remote farming areas, affecting app usability.

Opportunities (External Positive Factors):

Opportunity	Explanation
Untapped rural e-commerce market	The agricultural sector is largely underserved digitally, creating scope for innovation and scale.
Government support for agri-digitalization	Possibility of future integration with government schemes or subsidies.

Expand to B2B and B2C models	Can onboard distributors, cooperatives, and wholesalers for bulk orders.
Cross-platform reach	A responsive design can help reach both smartphone users and basic web users.
Brand reputation boost	Mr. Henry and APT IT Solutions can build significant goodwill through this impactful initiative.

Threats (External Negative Factors):

Threat	Explanation
Resistance to change by traditional farmers	Farmers used to physical markets may hesitate to trust online buying.
Cybersecurity risks	As the application will handle user data and possibly payment information, data security is critical.
Competitive imitation	Competitors may launch similar apps with more features or quicker delivery.
Regulatory constraints	Government regulations around pesticide/fertilizer sales may require compliance handling.
Logistics & Delivery issues	Ensuring timely delivery in remote areas could become a bottleneck.

Reasoning:

Conducting a thorough SWOT analysis at the initiation stage helps Mr. Karthik identify the internal capabilities and external conditions that will influence the project. This sets the foundation for mitigating risks and leveraging strengths and opportunities during planning and execution.

Q3: Mr. Karthik is trying to do a feasibility study on doing this project in Technology (Java). Please help him with points (Hardware, Software, Trained Resources, Budget, Time Frame) to consider in the feasibility study.

Answer:

A feasibility study is conducted to determine whether a project is technically, operationally, and financially viable. For Mr. Karthik to assess the feasibility of developing the Online Agriculture Products Store using Java technology, the following areas must be analyzed in depth:

1. Hardware Feasibility

Parameter	Explanation
Development Servers	High-performance servers required for development and testing phases.
Production Server Infrastructure	Needs to be scalable and reliable to handle traffic from rural users and product companies.
Cloud vs. On-Premise	Evaluate cost and convenience of hosting the application on cloud (e.g., AWS, Azure) vs. setting up local servers.
Mobile Devices for Testing	Multiple Android and iOS devices are needed to test mobile compatibility and responsiveness.
Backup and Storage Systems	Reliable backup servers and cloud storage for codebase, databases, logs, and reports.

2. Software Feasibility

Parameter	Explanation
Technology Stack	Use of Java (Spring Boot for backend), HTML/CSS/JS/React for frontend, and MySQL/PostgreSQL for database.
Development Tools	IntelliJ IDEA or Eclipse IDE, Maven/Gradle (build tools), Git (version control), Jenkins (CI/CD pipeline).
Testing Tools	JUnit for unit testing, Selenium for automation testing, Postman for API testing.
Deployment Environment	Use of Docker/Kubernetes for containerization and deployment.
OS Compatibility	Ensure compatibility with Android, iOS (via React Native or Flutter for hybrid apps), and various browsers.

3. Trained Resources Feasibility

Role	Resource Available	Skill Requirements
Project Manager	Mr. Vandanam	Project Planning, Gantt charts, Risk Management
Business Analyst	You (New BA)	Requirement gathering, process modeling, stakeholder communication
Developers	Ms. Juhi (Sr. Java), Mr. Teyson, Ms. Lucie, Mr. Tucker, Mr. Bravo	Java, Spring Boot, API integration, UI/UX development
Testers	Mr. Jason and Ms. Alekya	Manual and Automation Testing
DB Admin	Mr. John	Database design, optimization, indexing
Network Admin	Mr. Mike	Server setup, network security, access control

Conclusion: The talent pool at APT IT Solutions is well-equipped with the required Java expertise to execute this project effectively.

4. Budget Feasibility

Expense Head	Estimate	Notes
Resource Cost	Medium to High	Cost for developers, testers, BA, PM for 18 months
Infrastructure	Medium	Cloud hosting, storage, testing devices
Training/Workshops	Low to Medium	Farmer and company training sessions
Contingency	10–15% of total budget	To cover unexpected costs
Licenses (if any)	Minimal	Most Java tools and frameworks are open-source

Conclusion: Backed by CSR funding, the budget seems feasible given open-source tools and in-house resources.

5. Time Frame Feasibility

Phase	Duration Estimate	Activities
Requirement Gathering & Analysis	1–1.5 months	Stakeholder interviews, SRS, process modeling
Design (UI/UX + System)	1 month	Wireframes, architecture
Development (D1–D4)	9–10 months	Backend, frontend, APIs
Testing (T1–T4 + UAT)	3–4 months	Unit, integration, UAT
Deployment & Training	1 month	App launch, farmer orientation

Conclusion: The 18-month duration is adequate, provided proper planning and milestone tracking are followed.

Overall Feasibility Conclusion:

- Technically feasible using Java and open-source tools
- Operationally feasible with trained and available team
- Financially feasible under CSR funding
- Time feasible within the 18-month schedule

This green light allows Mr. Karthik to proceed to the next stage preparing the Gap Analysis for Mr. Henry.

Q4. Mr. Karthik must submit a Gap Analysis to Mr. Henry to convince him to initiate this project. What points (compare AS-IS existing process with TO-BE future process) should be showcased in the GAP Analysis?

Answer:

A Gap Analysis identifies the differences between the current state (AS-IS) and the desired future state (TO-BE) of the business. In this case, the purpose of the Gap Analysis is to highlight the challenges rural farmers currently face in procuring agricultural products and demonstrate how the Online Agriculture Products Store will bridge these gaps.

1. GAP ANALYSIS TABLE – Online Agriculture Product Store

Aspect	AS-IS Process (Current State)	TO-BE Process (Proposed Future State)	Gaps Identified
Procurement Method	Farmers rely on local markets or middlemen to buy seeds, fertilizers, and pesticides.	Farmers can directly purchase products from manufacturers via online platform.	No direct access to manufacturers; presence of intermediaries increases cost and delays.
Availability of Products	Limited or inconsistent availability in remote areas. Farmers often have to travel long distances.	24x7 access to a wide range of agriculture products through the website/mobile app.	Lack of consistent availability and product variety locally.

Pricing Transparency	Prices vary by seller; no price standardization or comparison available.	Farmers can compare prices from multiple companies and choose the best offer.	Lack of price clarity and negotiation power for farmers.
Information Access	No clear product descriptions, usage instructions, or reviews.	Detailed product catalog, usage guides, customer reviews provided online.	Information asymmetry; farmers lack product awareness.
Order & Delivery Process	Manual purchase, travel, or reliance on middlemen.	Online ordering, doorstep delivery, and order tracking.	Time-consuming and inefficient manual process.
Communication Between Farmers and Suppliers	No direct interaction; communication happens via middlemen.	Direct digital communication between farmers and product companies.	No reliable platform for two-way communication.
Customer Support	Difficult to reach support or complaint resolution mechanisms.	In-app customer support, chatbot, and escalation matrix available.	Lack of proper grievance redressal system.
Payment Mechanism	Mostly cash-based; no digital payment infrastructure.	Integration with UPI, net banking, wallets, and COD options.	No digital payment flexibility; cash-only transactions are risky.
Record Keeping	Manual invoices or none; no transaction history.	Digital invoices, order history, and wallet integration for refunds or offers.	Lack of transaction history or proof of purchase.
Reach and Market Expansion for Companies	Companies cannot directly reach small/remote farmers.	Companies can list and sell products nationwide without intermediaries.	Limited market reach for product companies.

2. Summary of Gaps Identified

Category	Gap Identified
Operational Gap	Delays and inefficiencies in product procurement and delivery.
Technical Gap	Lack of digital infrastructure for e-commerce in remote areas.
Communication Gap	Absence of real-time interaction between farmers and companies.
Information Gap	No access to product knowledge or comparative pricing.
Support Gap	Lack of grievance redressal, support, and guidance.

3. Value of TO-BE State to Stakeholders

Stakeholder	Value Delivered by TO-BE Process
Farmers	Convenience, lower costs, informed decision-making, better access
Manufacturers (Companies)	Direct-to-customer sales, brand exposure, nationwide reach
Mr. Henry (Sponsor)	Positive social impact, improved livelihood of farmers

APT IT Solutions	Successful delivery of a CSR-driven, impactful project
Government/CSR Committee	Contribution to rural upliftment and digital India mission

4. Justification to Initiate the Project

- The gap between the current challenges and the digital solution is significant.
- Implementing the TO-BE state will empower rural farmers, eliminate middlemen exploitation, and establish a transparent, scalable, and sustainable system.
- The TO-BE model aligns with Mr. Henry's CSR goals, and the solution is technically and financially feasible as established earlier.

Conclusion:

The Gap Analysis clearly showcases the inefficiencies and pain points in the AS-IS process and the strategic value the TO-BE solution will bring to all stakeholders. This supports Mr. Karthik's proposal to initiate the project without delay.

Q5. Project Development Approach:

Mr. Karthik explained to Mr. Henry about SDLC and four methodologies like Sequential, Iterative, Evolutionary, and Agile. Please share your thoughts and clarity on these methodologies.

Answer:

Software Development Life Cycle (SDLC) is a systematic process used to develop software, ensuring high quality, performance, and cost-efficiency. Within SDLC, several development methodologies define how a project should progress through its phases — from requirement gathering to deployment and maintenance.

Mr. Karthik explained the following four key methodologies to Mr. Henry:

1. Sequential Methodology (Waterfall Model)

Definition:

A linear and step-by-step approach where each phase must be completed before the next begins.

Flow:

Requirements → Analysis → Design → Development → Testing → Deployment → Maintenance

Characteristics:

- Rigid structure.
- No going back once a phase is completed.
- Well-suited for projects with fixed requirements and low uncertainty.

Use Case:

Useful when the requirements are clear, stable, and unlikely to change (e.g., Banking software, Industrial automation).

Pros:

- Easy to manage and understand.
- Clearly defined stages and deliverables.
- Suitable for documentation-heavy projects.

Cons:

- Poor flexibility for changes.
- Late discovery of issues during testing.
- Not suitable for complex, evolving needs.

2. Iterative Methodology

Definition:

Development is done in small iterations or cycles. Each cycle builds a part of the system, and feedback is incorporated in the next cycle.

Flow:

Plan → Design → Build → Test → Evaluate → Repeat

Characteristics:

- Product is developed piece by piece.
- Encourages feedback and continuous improvement.
- Each version evolves with improvements.

Use Case:

Ideal for projects where requirements are evolving or not fully known upfront (e.g., mobile apps, customer portals).

Pros:

- Flexibility to change requirements.
- Partial implementation available early.
- Risk is reduced early.

Cons:

- Might require more time and resources.
- Requires good planning for iteration cycles.

3. Evolutionary Methodology

Definition:

A combination of iterative and incremental methods where the software evolves with each release, becoming more complete and functional over time.

Flow:

Initial Version → Enhanced Version 1 → Enhanced Version 2 → Final Version

Characteristics:

- Begins with a basic version of software.
- Functionalities are added step by step.
- Highly adaptable to user feedback.

Use Case:

Ideal for innovative or research-based projects (e.g., AI solutions, start-up products).

Pros:

- End users get early access to usable features.
- Responds well to user feedback.
- Allows early error detection.

Cons:

- May lack clear deadlines or scope.
- Can lead to project scope creep if not managed well.

4. Agile Methodology

Definition:

An incremental and iterative approach focusing on collaboration, customer feedback, and flexibility.

Key Frameworks: Scrum, Kanban, XP

Flow (in Scrum):

Product Backlog → Sprint Planning → Sprint (2–4 weeks) → Review → Retrospective

Characteristics:

- Customer collaboration over contracts.
- Responding to change over following a rigid plan.
- Cross-functional teams and continuous delivery.

Use Case:

Best suited for dynamic environments where requirements may change frequently (e.g., e-commerce, customer-focused web apps).

Pros:

- Continuous delivery and frequent releases.
- High customer involvement.
- Rapid response to changes and bugs.

Cons:

- Requires experienced team and strong discipline.
- May lack documentation.
- Challenging in fixed-bid contracts.

Comparison Snapshot:

Criteria	Sequential	Iterative	Evolutionary	Agile
Flexibility	Low	Moderate	High	Very High
User Involvement	Low	Moderate	High	Very High
Cost of Change	High	Medium	Low	Low
Risk Handling	Poor	Better	Very Good	Excellent
Documentation	High	Medium	Medium	Low to Medium
Delivery Speed	Slow	Moderate	Moderate	Fast

Conclusion:

Each methodology has its own strengths and limitations. As a Business Analyst, you must assess:

- Complexity of the project
- Stability of requirements
- Timeline and budget constraints
- Stakeholder involvement
- Team experience and readiness

This helps in recommending the most suitable model for development.

Q6. They discussed models in SDLC like Waterfall, RUP, Spiral, and Scrum. You put forth your understanding on these models.

Answer:

Understanding the Software Development Life Cycle (SDLC) models is essential for any Business Analyst to guide the team in selecting the right development strategy based on the project needs, risk levels, and stakeholder expectations.

Here's a comprehensive explanation of the four SDLC models discussed: Waterfall, RUP, Spiral, and Scrum.

1. Waterfall Model

Definition:

The Waterfall Model is a linear and sequential development model where each phase must be completed before the next one begins. It is the earliest SDLC model and is based on a well-defined structure.

Phases:

1. Requirement Gathering
2. System Design
3. Implementation
4. Testing
5. Deployment
6. Maintenance

Characteristics:

- Each phase has distinct deliverables.
- No overlapping of phases.
- Documentation-heavy.

Pros:

- Easy to understand and manage.
- Suitable for small projects with clear requirements.
- Strong documentation.

Cons:

- Rigid; changes are hard to accommodate.
- Testing occurs late in the cycle.
- High risk if requirements are misunderstood.

Use Case Example:

A government tender project with fixed scope, deadlines, and budget.

2. RUP (Rational Unified Process)

Definition:

RUP is a customizable and iterative software development process framework created by Rational Software (IBM). It emphasizes object-oriented design and heavy documentation.

Phases:

1. Inception
2. Elaboration
3. Construction
4. Transition

Characteristics:

- Use-case and risk-driven development.
- Allows iterative development.
- Heavy on UML modeling and design practices.

Pros:

- Adaptable to different project sizes.
- Risk is identified early.
- Continuous integration and testing.

Cons:

- Complex and resource-intensive.
- Steep learning curve for teams.
- Overhead in documentation and process control.

Use Case Example:

Large-scale enterprise solutions like ERP systems or insurance claim processing platforms.

3. Spiral Model**Definition:**

The Spiral Model combines iterative development with the systematic aspects of the waterfall model, focusing heavily on risk assessment.

Phases (repeated in spirals):

1. Planning
2. Risk Analysis
3. Engineering (Development & Testing)
4. Evaluation

Each spiral results in a progressively more complete version of the software.

Characteristics:

- Emphasizes risk analysis.
- Ideal for complex and high-risk projects.
- Incorporates customer feedback in each loop.

Pros:

- Excellent for risk-prone projects.
- Accommodates change well.
- Delivers prototypes early.

Cons:

- Requires expertise in risk analysis.
- Not suitable for small projects.
- May become expensive due to extended cycles.

Use Case Example:

Military, aerospace, or high-security government software.

4. Scrum (Agile Framework)**Definition:**

Scrum is a lightweight, Agile framework that organizes development into small, time-boxed iterations called Sprints (usually 2–4 weeks).

Roles:

- Product Owner: Defines the product backlog and priorities.
- Scrum Master: Facilitates the process and removes blockers.
- Development Team: Builds and delivers the product increment.

Key Components:

- Product Backlog
- Sprint Backlog
- Daily Scrum
- Sprint Review & Retrospective

Characteristics:

- High customer collaboration.
- Incremental delivery.
- Change-friendly environment.

Pros:

- Early and frequent delivery of value.
- High stakeholder engagement.

- Quick detection of issues.

Cons:

- Requires experienced, cross-functional teams.
- Less documentation.
- Harder to scale for very large projects.

Use Case Example:

Customer-facing applications like e-commerce websites, mobile apps, and SaaS platforms.

Summary Comparison:

Criteria	Waterfall	RUP	Spiral	Scrum (Agile)
Approach	Sequential	Iterative	Risk-driven + Iterative	Iterative & Incremental
Flexibility to Changes	Low	Moderate	High	Very High
Documentation	High	Very High	Medium	Low to Medium
Risk Management	Low	Medium	High	Medium
Customer Involvement	Low	Medium	High	Very High
Best For	Fixed-scope projects	Large enterprise systems	High-risk, R&D projects	Dynamic and evolving projects

Business Analyst Viewpoint:

As a BA, understanding these models helps in:

- Choosing the right methodology based on project nature.
- Managing stakeholder expectations.
- Planning requirement elicitation and change management accordingly.
- Identifying where and how feedback loops, documentation, testing, and collaboration occur.

Q7. When the APT IT SOLUTIONS company got the project to make this online agriculture product store, there is a difference of opinion between a couple of SMEs and the project team regarding which methodology would be more suitable for this project. SMEs are stressing on using the V Model and the project team is leaning more onto the side of the Waterfall model. As a Business Analyst, which methodology do you think would be better for this project? Justify your choice.

Answer:

As a Business Analyst, after analysing the project requirements, constraints, and nature of stakeholders involved in the Online Agriculture Products Store, I strongly recommend using the V-Model (Validation & Verification Model) over the Waterfall model for the following reasons:

V-Model Overview:

The V-Model is an extension of the Waterfall model, but with a key distinction — each development stage has a corresponding testing phase. This approach allows early testing planning, improving the quality of the final product.

Reasons Why V-Model is Better for This Project:

1. Nature of the Project Requires High Reliability:

- The system connects farmers and agriculture companies and involves financial transactions and logistics coordination.
- Errors in the final application could cause real-time business losses or mistrust in the system.
- The V-Model ensures testing at every stage, reducing the risk of late discovery of defects.

2. Defined and Stable Requirements:

- Since Mr. Henry and his friends (Peter, Kevin, Ben) are actively involved in sharing detailed requirements upfront, and the project is part of a CSR initiative, the requirements are less likely to change frequently.
- V-Model works best when the requirements are well understood and fixed.

3. Early Test Planning = Quality Output:

- In V-Model, test plans and test cases are prepared alongside each phase of development, which helps in:
 - Minimizing the defect rate
 - Ensuring verification and validation from the beginning

4. Alignment with Committee Expectations:

- Stakeholders like Mr. Henry, Mr. Pandu, and Mr. Dooku may not be tech-savvy and would prefer predictable timelines and high quality over frequent changes and iterations.
- V-Model provides clear deliverables and phase-wise validation aligned with stakeholder checkpoints.

5. Easy Traceability:

- The V-Model provides one-to-one mapping between requirements and test cases.
- This traceability ensures that no requirement is left untested, which is very important for compliance and audit purposes in CSR projects.

6. Team Readiness & Skill Set:

- The project team includes defined roles (BA, PM, Java Developers, Testers, DB Admin, NW Admin), with no explicit mention of Agile coaches or Scrum Masters.
- V-Model fits this traditional resource structure better than Agile/Scrum.

Why Not Waterfall?

Although the Waterfall model is simple and structured, it has the following limitations in this case:

- Late Testing Phase: All testing happens only after development ends, which could lead to the discovery of critical bugs at a very late stage.
- No Early Bug Detection: There's no validation phase during requirement and design stages.
- High Risk of Rework: If issues are identified late, the rework cost and effort can be high.

Summary: Waterfall vs V-Model for This Project

Criteria	Waterfall	V-Model
Testing Involvement	Late phase only	At each development phase
Requirement Stability Required	Yes	Yes
Risk Mitigation	Less	High – early defect detection
Best For	Simple, one-time delivery	CSR, compliance, quality-driven
Feedback Mechanism	Post development	Continuous validation

Conclusion as a BA:

Considering the project scope, requirement clarity, stakeholder involvement, team composition, and the criticality of accuracy and reliability, the V-Model is the most appropriate methodology for this project. It will ensure early detection of defects, structured validation, and high-quality output – all essential for the success of the Online Agriculture Product Store under a CSR initiative.

Q8. Write down the differences between Waterfall model and V model. As a BA, state your reason for choosing one model for this project.

Answer:

1. Differences Between Waterfall Model and V-Model

Aspect	Waterfall Model	V-Model (Verification & Validation Model)
Process Flow	Linear and sequential flow from requirements → design → development → testing → deployment	V-shaped flow with corresponding testing phase for each development phase
Testing Phase	Testing happens only after the development phase is complete	Testing is planned and executed in parallel with each development phase
Requirement Validation	Requirements are validated after implementation	Requirements are validated at early stages with corresponding test cases
Risk Handling	Higher risk due to late detection of defects	Lower risk due to early and continuous testing
Flexibility to Change	Rigid; changes are difficult and costly once development starts	Less flexible but allows early detection of issues
Traceability	Limited traceability between requirements and tests	Strong traceability between development stages and tests
Feedback Loop	Feedback mostly at the end of the project	Feedback and validation at every stage of development
Documentation	Heavy documentation for each phase	Emphasis on documentation with corresponding test cases
Suitability	Suitable for simple and well-understood projects	Suitable for projects where quality and reliability are critical
Complexity	Simpler to understand and implement	More complex due to the integration of testing phases

Cost & Time	Can be costlier if defects found late	Can save cost and time by catching defects early
------------------------	---------------------------------------	--

2. Reason for Choosing V-Model for Online Agriculture Product Store Project

As a Business Analyst, I recommend the V-Model for this project because:

1. Emphasis on Early Testing & Validation

- This project involves critical agricultural products where errors could impact farmers' livelihoods.
- The V-Model's early test planning ensures defects are caught early, improving product reliability and trust.

2. Requirement Stability and Clear Scope

- The CSR initiative has well-defined requirements gathered from stakeholders like Peter, Kevin, and Ben.
- V-Model thrives when requirements are stable and well-documented, making it easier to link requirements to tests.

3. Quality Assurance is Crucial

- The online platform will be used by diverse stakeholders including farmers and companies.
- High quality, reliability, and compliance are non-negotiable — V-Model's strong focus on validation ensures this.

4. Clear Traceability

- Traceability matrices can be created easily between requirements, design, and test cases.
- This is beneficial for audit trails and verifying that all stakeholder needs are met.

5. Stakeholder Confidence

- Non-technical stakeholders prefer clear milestones and predictable deliverables.
- The V-Model's stage-wise validation aligns well with these expectations.

6. Risk Mitigation

- Since errors in agricultural products procurement could have serious consequences, early testing minimizes business risks.

Summary:

While the Waterfall model is simpler and sequential, it lacks early testing and feedback loops, which can lead to costly fixes later on. The V-Model improves upon Waterfall by integrating testing phases at every stage, which is essential for this CSR-driven, quality-sensitive project.

Q9. The Committee of Mr. Henry, Mr Pandu, and Mr Dooku discussed with Mr Karthik and finalised on the V Model approach (RG, RA, Design, D1, T1, D2, T2, D3, T3, D4, T4 and UAT). Mr Vandanam is mapped as a PM to this project. He studies this Project and Prepares a Gantt chart with V Model (RG, RA, Design, D1, T1, D2, T2, D3, T3, D4, T4 and UAT) as development process and the Resources are PM, BA, Java Developers, testers, DB Admin, NW Admin. Explain the finalized V-Model phases and resource allocation in detail.

Answer:

1. V-Model Phases and Their Explanation

Phase Code	Phase Name	Explanation
------------	------------	-------------

RG	Requirement Gathering	Collect detailed business and technical requirements from stakeholders (farmers, companies, committee). BA leads this activity with PM support.
RA	Requirement Analysis	Analyze, clarify, and document the gathered requirements into a formal Requirements Specification Document. BA works closely with PM and technical leads.
Design	System and Technical Design	Architects and developers design system architecture, database schema, network setup, and UI design based on requirements.
D1	High-Level Design	Creating high-level design documents detailing system modules, data flow, and interactions. Leads: Senior Developers & BA.
T1	High-Level Design Testing	Review and validate high-level design documents. Testers and BA ensure design meets requirements.
D2	Detailed Design	Prepare detailed design including class diagrams, interface details, database design, and API specifications. Java Developers lead.
T2	Detailed Design Testing	Testing detailed design for completeness and consistency. Testers and BA collaborate on test plans.
D3	Coding/Development	Developers write actual code, implement features as per design documents.
T3	Unit Testing	Testers and developers perform unit testing on individual modules for functionality and correctness.
D4	Integration and System Testing	Combine modules and test the integrated system to validate end-to-end workflows and data flow. Testers lead.
T4	System Testing	Conduct comprehensive testing including functional, non-functional, security, and performance testing. Testers and BA involved.
UAT	User Acceptance Testing	End users (farmers, company representatives) validate the system in a real-world scenario before final deployment. BA coordinates UAT.

2. Resource Allocation & Roles

Resource	Role and Responsibilities	Phases Involved
Project Manager (PM) - Mr. Vandanam	Oversees project planning, execution, resource management, risk mitigation, and communication with stakeholders. Coordinates all phases.	Entire lifecycle (RG to UAT)
Business Analyst (BA) - You	Leads requirement gathering and analysis, clarifies needs, creates BRD/FRD, liaises between stakeholders and technical teams, helps prepare test cases, supports UAT.	RG, RA, T1, T2, T4, UAT
Senior Java Developer - Ms. Juhi	Leads design and development, guides junior developers, ensures coding standards and architecture adherence.	Design, D1, D2, D3

Java Developers - Mr. Teyson, Ms. Lucie, Mr. Tucker, Mr. Bravo	Develop system modules, perform unit testing with developers and testers, fix defects.	D2, D3, T3
Tester - Mr. Jason and Ms. Alekya	Prepare test plans and test cases, perform design reviews, execute unit, integration, system, and UAT testing, log defects, ensure quality.	T1, T2, T3, T4, UAT
Database Admin - John	Design and manage database schema, ensure data integrity and backup strategies, support database related testing and deployment.	Design, D2, D3
Network Admin - Mr. Mike	Manage network infrastructure, ensure secure connectivity, support deployment environment, troubleshoot network issues.	Design, D4, Deployment

3. Phase-wise Timeline and Gantt Chart Insights

- The PM (Mr. Vandanam) creates a Gantt chart scheduling these phases sequentially but allowing some overlaps, for example:
 - While BA works on Requirement Analysis (RA), Developers can start preliminary System Design.
 - Testing phases (T1, T2, etc.) start immediately after corresponding development phases (D1, D2, etc.).
 - UAT is scheduled near the end, involving real users coordinated by BA.

4. Why This Allocation and Approach?

- Clear roles ensure accountability at every phase.
- BA's involvement throughout ensures business requirements are correctly translated into technical solutions.
- Parallel testing with development phases (V-Model strength) reduces late defect discovery.
- PM oversees to keep the project on schedule and manages resource dependencies.
- Specialist resources (DB & Network Admin) ensure technical infrastructure stability.
- This structure suits the project's fixed timeline (18 months CSR initiative) and quality expectations.

Q10. Explain the difference between Fixed Bid and Billing Projects.

Answer:

1. Fixed Bid Projects

Definition:

- A Fixed Bid Project (also called Fixed Price Project) is where the client and the service provider agree upon a predefined fixed price for the entire project before starting the work.

Key Characteristics:

- Scope: Clearly defined upfront; changes are limited or charged separately.
- Price: Fixed and agreed upon before the project begins.
- Timeline: Usually fixed or estimated based on the agreed scope.
- Risk: Most risk lies with the service provider if the project takes longer or costs more than expected.

- Payments: Often milestone-based or full payment on delivery.
- Suitability: Best for projects with well-defined requirements and low expected changes.

Pros:

- Predictable cost for the client.
- Clear expectations on deliverables.
- Encourages efficient work from the service provider.

Cons:

- Less flexibility to accommodate changes.
- Risk of compromised quality if the provider tries to cut corners to save cost.
- Requires thorough upfront planning and requirement gathering.

2. Billing Projects

Also known as Time and Material (T&M) or Hourly Billing Projects.

Definition:

- The client pays based on the actual time spent and materials used during project development.

Key Characteristics:

- Scope: Flexible and can evolve during development.
- Price: Variable, based on hourly or daily rates and material costs.
- Timeline: Flexible; can extend based on scope changes.
- Risk: Mostly lies with the client as costs can increase if project extends.
- Payments: Periodic billing based on hours worked or deliverables completed.
- Suitability: Best for projects where requirements are unclear, evolving, or exploratory.

Pros:

- Flexibility to change requirements anytime.
- Easier to add new features or make modifications.
- Encourages collaboration and iterative development.

Cons:

- Uncertainty in final cost.
- Requires active client involvement to control scope and budget.
- Risk of scope creep if not managed well.

3. Summary Table

Aspect	Fixed Bid Project	Billing (Time & Material) Project
Cost	Fixed upfront	Variable, based on actual effort
Scope	Defined and frozen at start	Flexible and can evolve during development
Risk	Provider bears risk of overruns	Client bears risk of increased costs
Flexibility	Low	High
Client Control	Low after contract signing	High, with active involvement
Best for	Well-defined, stable requirements	Unclear, evolving, or exploratory projects

4. Why This Understanding is Important for This Project

- The Online Agriculture Product Store is a CSR initiative with a fixed timeline (18 months) and predefined scope gathered from stakeholders.

- A Fixed Bid Model would likely be preferred here due to:
 - Clear initial requirements from farmers and companies.
 - Fixed budget and schedule constraints.
 - Need for predictable deliverables and cost control.
- However, if changes or additional features are anticipated due to user feedback during UAT, a Billing Model or a hybrid approach could be considered for post-deployment enhancements.

Q11. Design Timesheet of a BA (Business Analyst)

Answer:

Focus: Design Phase Activities

Understanding the Role of a BA in the Design Phase

During the Design Phase, the BA plays a crucial bridging role between stakeholders and technical teams. The BA ensures that business requirements are accurately translated into solution designs, wireframes, and system architecture, working closely with the UI/UX designers and development leads.

Business Analyst Design Phase Timesheet – Sample (Weekly View)

Day	Task Description	Time Spent (Hours)	Stakeholders/Teams Involved	Purpose
Monday	Review finalized Functional Requirements Document (FRD) with development team	2 hrs	Dev Team, Tech Lead	Ensure clarity of features before UI design begins
	Conduct Design Kick-off meeting with UI/UX team	1.5 hrs	UI/UX Designers, PM	Align UI design with business needs and priorities
	Identify Key User Stories and screen requirements	2 hrs	Internal	Define screens and flows aligned to features
	Support Wireframe development	2.5 hrs	UI Team, Devs	Provide business logic inputs to support visual layout
Tuesday	Collaborate with UI team to validate user interface mock-ups	3 hrs	UI Designers	Ensure designs match business flows and usability
	Create/Update Process Flow Diagrams (BPMN or swimlanes)	2 hrs	Internal	Represent AS-IS/TO-BE process visually
	Hold Review Meeting with Stakeholders to approve design concepts	1.5 hrs	Business Stakeholders	Align with client expectations and validate alignment

Wednesday	Document Screen-Level Requirements & Field Validations	3 hrs	Internal	Support Dev and QA teams in future development & testing
	Define User Roles & Permissions Matrix	2 hrs	PM, QA, Tech Leads	Align role-based access with functionality
Thursday	Support Logical Data Design – identify key data elements for screen/database mapping	2 hrs	DB Admin, Dev Team	Enable data integrity and smooth UI-backend flow
	Validate UI Prototypes/Clickable Mock-ups	2 hrs	UI/UX Team	Ensure alignment to business goals and ease of use
	Internal Review & Feedback Loop	1.5 hrs	Internal	Iterate on design changes
Friday	Consolidate BA Deliverables (Design Spec doc, updated FRD, user journey flows)	3 hrs	PM, QA	Formal documentation for sign-off
	Attend PM Review Meeting on Design Readiness	1 hr	PM, Dev, QA, UI/UX	Final checkpoint before Design Sign-Off
	Provide Sign-Off Support/Approvals	1 hr	Stakeholders	Final validation from client side

Total Hours per Week (Design Phase):

35-40 Hours depending on project complexity.

Deliverables from BA in Design Phase

Deliverable	Purpose
Updated Functional Requirement Document	Reflects refined scope after requirement validation
UI/UX Review Comments Sheet	Track all UI suggestions and client feedback
Design Specification Document	Maps each UI component to functional logic and data fields
User Roles and Permission Matrix	Defines access control and visibility rules
Business Process Models (BPMN/Flowcharts)	Represents process flows for development reference
Field-Level Validation Rules	Defines constraints for form fields or inputs
Traceability Matrix (Requirements ↔ Design)	Ensures all requirements are traceable in the design

Why These Tasks Are Important

- Accuracy in Design → Better Development Output
A BA ensures that technical designs are aligned with business logic and user needs.

- Reduces Rework in Later Phases
Poorly validated designs can lead to scope creep or major rework during development/UAT.
- Improves Communication Across Teams
The BA acts as a translator between business terms and technical concepts.
- Ensures Compliance & Usability
Especially in projects like the Online Agriculture Store, good design means farmers (end users) can easily navigate and use the platform.

Q12. Development Timesheet of a BA (Business Analyst)

Answer:

Focus: *Development Phase Activities*

Understanding the Role of a BA in the Development Phase

In the Development Phase, the Business Analyst acts as a *bridge between the developers and stakeholders*, ensuring the implementation strictly follows the approved requirements and design. BAs support developers by clarifying requirements, updating documentation when necessary, ensuring traceability, and proactively managing requirement deviations or change requests.

Business Analyst Development Phase Timesheet – Sample (Weekly View)

Day	Task Description	Time Spent (Hours)	Stakeholders/Teams Involved	Purpose
Monday	Conduct Sprint Planning or Development Kickoff Meeting	1.5 hrs	PM, Dev Team, QA	Outline development tasks per module & user story
	Support developers with requirement clarifications	2.5 hrs	Java Devs, DB Admin	Prevent blockers, ensure correct feature implementation
	Update/Refine Traceability Matrix	1.5 hrs	Internal	Link requirements to development tickets
	Review system architecture documents and identify alignment with business goals	2 hrs	Dev Leads, Network Admin	Validate technical feasibility with business objectives
Tuesday	Conduct Daily Standup Meeting and track progress against requirements	0.5 hrs	All Team Members	Track blockers, progress, and BA involvement needed
	Review Technical Design Document (TDD) and provide inputs	2 hrs	Tech Leads	Ensure logical implementation of business rules
	Manage Change Requests (CRs), Update FRD & Recommunicate to Developers	2 hrs	Stakeholders, Devs, QA	Handle evolving scope or enhancements smoothly

	Participate in Code Review Discussions (High-Level View)	1.5 hrs	Dev Team	Ensure business rules have been interpreted correctly
Wednesday	Prepare Data Mapping Sheets (UI to Backend to DB)	2.5 hrs	DB Admin, Backend Dev	Clarify data flow and integrations
	Maintain Issue/Clarification Log	1 hr	Internal	Log repetitive queries and resolutions
	Document Business Rule Implementation Status	1.5 hrs	PM, Dev Team	Track what logic has been implemented, what's pending
Thursday	Validate API Contracts or Integration Requirements	2 hrs	Dev Team, API Team	Ensure API input/output matches expected business parameters
	Prepare for SIT (System Integration Testing) by verifying development completeness	2 hrs	QA Team, PM	Ensure readiness for integrated testing
	Clarify functional logic in integration components	1.5 hrs	Java Dev, API Team	Resolve integration mismatches early
Friday	Conduct Internal Review of Completed Modules	2 hrs	Dev Team, QA	Validate module flow with functional use cases
	Conduct Stakeholder Demo on Developed Features	1.5 hrs	Mr. Henry Committee, PM	Confirm delivery is aligned with expectations
	Update BA Documents – CR Log, Issue Tracker, Functional Notes	2 hrs	Internal	Keep all BA assets current and aligned

Total Hours per Week (Development Phase):

35 to 40 Hours, adjusted based on sprint cycle or complexity of modules.

Key Deliverables by BA in Development Phase

Deliverable	Purpose
Updated Functional Requirement Document (FRD)	Reflects any approved changes or clarifications
Change Request Log (CR Log)	Track all modifications and approvals
Requirements Traceability Matrix (RTM)	Track each requirement's implementation status
Issue Clarification Sheet	Logs questions from developers with responses
Data Mapping/Field Mapping Sheet	Maps UI fields to database and APIs
Business Rule Implementation Tracker	Track rule coverage per module
Development Status Reports (Feature-wise)	For internal PM and stakeholder reporting

Why These Tasks Are Important

1. Maintains Requirement Fidelity

Without BA involvement, developers may misinterpret business requirements, especially in domain-specific platforms like the Online Agriculture Products Store.

2. Eliminates Rework

Proactive clarifications and traceability reduce defects, rework, and post-development change cycles.

3. Bridges the Communication Gap

BA translates technical questions into business-friendly terms and vice versa—crucial for progress tracking and expectation setting.

4. Ensures Accurate Development

With BA's support, all user stories, data flows, and business rules are properly coded and mapped—resulting in fewer defects during testing.

Example From Your Project: Online Agriculture Store

- For the “Add Fertilizer to Cart” feature, the BA ensures:
 - All UI fields (product, quantity, price) match FRD.
 - The backend handles inventory correctly.
 - API passes correct data.
 - Business rules like “limit per user” are implemented.

This work happens not by coding, but by the BA continuously validating logic, data flow, and business rules through communication, documentation, and demos.

Q13. Testing Timesheet of a BA (Business Analyst)

Answer:

Focus: Testing Phase Activities (System Testing, Integration Testing, Functional Testing, Defect Support)

Understanding the Role of a BA in the Testing Phase

During the Testing Phase, the Business Analyst plays a *critical support and validation role*. While testers perform test case execution, the BA ensures the requirements are correctly implemented, validates test coverage, supports defect triaging, and provides clarifications to both testers and developers. This ensures that what’s built truly meets business expectations.

Business Analyst Testing Phase Timesheet – Sample (Weekly View)

Day	Task Description	Time Spent (Hours)	Stakeholders/Teams Involved	Purpose

Monday	Participate in Test Case Review Meeting	2 hrs	QA, PM, Dev Team	Ensure all business scenarios are tested
	Validate Requirements Traceability Matrix (RTM) with Test Scenarios	1.5 hrs	QA	Confirm all requirements are covered
	Provide Clarifications on Test Case Logic, UI Flows, and Expected Results	2 hrs	QA, Developers	Help testers understand business workflows
	Attend Defect Triage Meeting (discuss severity, root cause, business impact)	1.5 hrs	QA, Dev, PM	Prioritize defects based on business impact
Tuesday	Review System Test Execution Reports	1.5 hrs	QA	Track progress of test execution
	Retest/Validate Fixed Defects (only functional/business validation)	2 hrs	QA, Dev	Ensure bug fixes align with business needs
	Update Requirement Documents based on Valid Defects or Changes	1.5 hrs	Internal	Keep FRD and CR Logs updated
	Help QA Team in Preparing SIT (System Integration Testing) Scenarios	2 hrs	QA	Guide testers in cross-module and API testing
Wednesday	Conduct Requirement Coverage Review	2 hrs	PM, QA	Map requirement-to-test case using RTM
	Review and Validate Test Data (Boundary values, invalid inputs, real-case values)	2 hrs	QA, DB Admin	Ensure realistic test case input data
	Functional Demo Review of Tested Modules	2 hrs	QA, Dev, Stakeholders	Business-side validation of tested functionality
Thursday	Conduct Regression Testing Support	1.5 hrs	QA	Confirm existing functionality remains intact
	Update Issue Tracker and Defect Analysis Sheet	2 hrs	QA, PM	Maintain log of functional/business-impacting bugs
	Approve/Test Readiness for UAT (post successful internal QA phase)	1.5 hrs	PM, Stakeholders	Ensure UAT entry conditions met

Friday	Coordinate Business Walkthrough/Demo of Tested Features	1.5 hrs	Mr. Henry's Committee, QA, Dev	Stakeholder validation of real-world usability
	Document Business Feedback and Testing Observations	1.5 hrs	Internal, Stakeholders	Ensure business feedback captured before UAT
	Conduct BA-QA Sync-up to Finalize Open Points and Exit Criteria	1.5 hrs	QA, PM	Align on closure checklist and pending clarifications

Total Hours per Week (Testing Phase):

~35 to 38 Hours, depending on defect volume and scope.

Key Deliverables by BA in Testing Phase

Deliverable	Purpose
Updated Requirements Traceability Matrix (RTM)	Ensure full testing coverage of all requirements
Test Case Review Comments	Validate logical flow of test cases and scenarios
Defect Triage Sheet	Categorize bugs by severity and business impact
Business Rule Validation Log	Confirm each rule is implemented and tested
Updated CR Log / FRD	Reflect any approved changes from testing outcomes
Test Data Sheet (supporting realistic values)	Help testers validate edge cases, boundary conditions, and real data flows

Why These Tasks Are Important

1. Ensures Accurate Validation

BA involvement ensures that test cases reflect the true intention behind each business requirement, avoiding misaligned testing.

2. Supports Quick Defect Resolution

BAs help QA and Dev teams resolve issues faster by explaining the logic behind functionality and expected behaviour.

3. Strengthens UAT Readiness

By supporting the QA team during internal testing, BAs ensure a smoother transition to UAT by reducing business-facing bugs.

Example From Your Project: Online Agriculture Product Store

Let's say testers are validating the "View Product Inventory" feature:

- Without BA: Testers might just check if inventory shows up.
- With BA: You guide them to validate:
 - Correct stock units display.
 - Stock visibility rules (e.g., disable "Add to Cart" if out of stock).
 - Segregation by product type (fertilizers, pesticides).
 - Filter & sorting logic per business rules.

Also, if testers raise a bug saying, "Product not displayed in Cart," the BA helps identify if it's a UI issue, business logic fault, or backend fault—thus assisting triage and proper assignment.

Q14. UAT Timesheet of a BA (Business Analyst)

Answer:

Focus: Activities during the User Acceptance Testing (UAT) Phase

Understanding the Role of a BA in UAT

User Acceptance Testing (UAT) is the final testing phase before product go-live, where actual business users validate the system against their expectations. A Business Analyst (BA) plays a pivotal role by acting as the bridge between users and the technical team, facilitating the UAT process, preparing test scenarios, coordinating schedules, tracking feedback, and driving defect resolution.

Business Analyst UAT Phase Timesheet – Sample (Weekly View)

Day	Task Description	Time Spent (Hours)	Stakeholders/Teams Involved	Purpose
Monday	Prepare and Review UAT Test Plan & Scenarios	2 hrs	PM, QA, Client Committee (Mr. Henry, Peter, etc.)	Define real-world business test flows
	Create and Share UAT Test Data with Stakeholders	2 hrs	QA, DB Admin, Stakeholders	Provide realistic input data for testing
	Conduct UAT Kickoff Meeting	1.5 hrs	Stakeholders, PM, QA	Align business users on process, timelines, and expectations
	Explain Business Flows, Navigation, and Expected Outcomes to UAT Users	2 hrs	Stakeholders	Help non-technical users understand system flow
Tuesday	Support UAT Users During Execution (Functional Clarifications)	3 hrs	Stakeholders, QA, Dev	Assist users during real-time test case execution
	Track UAT Issues/Bugs Raised	1.5 hrs	QA, Dev	Maintain issue tracker
	Conduct Daily Status Update Call	1 hr	PM, QA, Stakeholders	Share progress, risks, blockers
Wednesday	Analyze and Validate UAT Defects for Business Impact	2 hrs	QA, Dev	Classify bugs as critical/non-critical based on business rules
	Provide Resolution or Workaround Suggestions to Dev Team	2 hrs	Dev, PM	Speed up bug fixing using domain knowledge
	Prepare Traceability Report for UAT	1.5 hrs	Internal, PM	Ensure all critical flows were tested

Thursday	Help Business Users Retest After Fixes	2.5 hrs	QA, Stakeholders	Ensure bugs are fixed as per business expectations
	Gather Feedback from Stakeholders on Usability, Navigation, Business Logic	2 hrs	Mr. Henry's Committee	Capture end-user insights for final changes
	Document Sign-off Checklist	1 hr	PM, Stakeholders	List of criteria to be met for UAT closure
Friday	Final UAT Closure Meeting	1.5 hrs	PM, Stakeholders, QA	Formally sign off on UAT phase
	Collect UAT Sign-off from Stakeholders	1 hr	Mr. Henry, Peter, Kevin, Ben	Final approval of business validation
	Document UAT Summary Report	1.5 hrs	PM, QA	Provide insights, status, pending issues, and lessons learned

Total Hours per Week (UAT Phase):

35 to 38 Hours, based on defect volume and user engagement level.

Key Deliverables by BA in UAT Phase

Deliverable	Purpose
UAT Plan	Detailed schedule and scope of UAT
UAT Scenarios Document	Real-world business scenarios for testing
Test Data Sheet	Practical values that mimic real user behavior
Issue Tracker (with Severity & Status)	Centralized list of business-level bugs/feedback
UAT Sign-off Checklist	Conditions to declare successful UAT completion
UAT Feedback Document	Comments and concerns raised by users during testing
Final UAT Summary Report	Wrap-up report summarizing test coverage, issues, fixes, and approvals

Why These BA Tasks Are Critical in UAT

1. Bridging Communication Gaps

- BAs ensure that non-technical stakeholders can communicate effectively with developers and testers.
- They help translate feedback like *"I don't see the Add to Cart button after filtering"* into actionable bug reports.

2. Ensuring Business Value

- UAT validates that the application aligns with the actual business operations and not just functional specifications.
- BAs help map each feature back to its business objective.

3. Minimizing Go-Live Risks

- Ensures business users accept and are satisfied with the product before deployment.
- Any major issues discovered now are less expensive to fix than post-release.

Real-Time Example – From Your Case Study

Scenario: Farmer placing order for pesticide

- **BA guides UAT users to test:**
 - Can the user search pesticide by type?
 - Are regional language filters working?
 - Does stock availability and price display correctly?
 - Can the farmer add to cart and complete payment?

If a UAT issue arises where payment confirmation isn't showing, the BA helps determine:

- Is it due to payment gateway config?
- Or does the UI not reflect confirmation status?
- Or was there an expectation mismatch on design?

BA resolves it by consulting backend/API or suggesting UI fix — streamlining UAT.