Capstone Project -1

Online Agriculture Products Store

# Question 1: Identify Business Process Model for Online Agriculture Store

**Goal:** Develop an online agriculture store to bridge the gap between the farmers and suppliers

**Input:** Requirements from stakeholders, Existing agricultural challenges, Market research on digital solutions, Trained employees

**Resources:** Ware house, Office Space, financial backing from Mr. Henry, Farmers and suppliers as users

**Outputs:** Fully functional web/ mobile application, Improved access to agricultural products for farmers

**Activities:** Onboard Farmers and Sellers in platform - Browse Products and Place Order – Make Payment and Order Processing – Shipping & Delivery

**Value:** Increased efficiency in procurement, Direct interaction between farmers and suppliers, Enhanced rural economy

# Question 2: SWOT Analysis on Agriculture Product Store

|  |  |
| --- | --- |
| Strengths* Strong financial backing
* Experienced IT development team
* Large market potential
 | Weakness* Limited digital literacy among farmers
* Logistics and supply chain challenges
 |
| Opportunities* Expanding to more agricultural regions
* Partnering with government initiatives
 | Threats* Competition from existing online platforms
* Cybersecurity risks
 |

Question 3: What is the Feasibility Study?

* **Technology:** Based on the database servers, Payment gateways, Security and API’s
* **Hardware:** Based on Storage, Backup systems, Cloud-based servers
* **Software:** Based on Shopping cart software, content management system and payment gateway software.
* **Trained Resources:** Project management team, Business Analysts, Experienced Java developers
* **Budget:** Various cost involved such as development costs and maintenance costs
* **Time Frame:** 18 months

# Question 4: Gap analysis on Agriculture product store

**Current State**

* Farmers struggle to find and procure fertilizers, seeds, and pesticides due to limited availability in remote areas.
* No direct communication between farmers and suppliers, leading to delays and high costs.
* Manual and time-consuming process, requiring farmers to travel long distances.

**Desired State**

* Farmers can easily access and purchase agricultural products through an online platform.
* Direct interaction between farmers and suppliers for better pricing, availability, and timely delivery.
* A streamlined online ordering and doorstep delivery system ensures convenience and efficiency.

# Question 5: Risk Analysis of Agriculture Product Store

**Internal Risks:**

* Dependence on external vendors for product supply and inventory management
* High operating expenses in technology and marketing

**External Risks:**

* Competition from offline markets
* Changes in government regulations and policies after e commerce

**BA Risks:**

* Miscommunication of requirements
* Domain
* scope creep

**Project based risks:**

* Technical failures,
* user adoption challenges

# Question 6: RACI matrix for Agriculture Product Store



**Responsible**

**Accountable**

**Informed**

**Consulted**

# Question 7: Business Case Document

**Why this project initiated ?**

* The project was initiated to solve the challenges faced by farmers in remote areas who struggle to procure essential agricultural products such as fertilizers, seeds, and pesticides. By creating an online platform, farmers can directly connect with suppliers, ensuring easy access to quality products.

**What are the current problem?**

* Farmers in remote areas face difficulties in procuring fertilizers, seeds, and pesticides.
* Lack of direct communication between farmers and suppliers leads to delays and high costs.
* Limited access to quality agricultural products affects productivity and crop yield.
* Dependency on intermediaries increases prices and reduces profit margins for farmers.

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

* Easy procurement: Farmers can directly buy from manufacturers, ensuring timely access to agricultural products.
* Cost reduction: Eliminating middlemen helps in reducing costs for farmers.
* Quality assurance: Direct purchases from reputed suppliers ensure better quality products.
* Increased efficiency: Faster order processing and doorstep delivery save time and effort for farmers.

# Question 7: Business Case Document

* **What are the resources required?**
* **Technology Stack**: Java-based platform, cloud hosting, secure payment gateway.
* **Human Resources**:
* **Project Team**: Business Analyst, Project Manager, Developers, Testers, Network Admin, and Database Admin.
* **Stakeholders**: Business sponsors, committee members, and suppliers.
* **Financial Resources**: Budget for platform development, maintenance, and marketing.
* **How much organization change is required to adopt this technology?**
* **Moderate Change**: Since it is an online platform, farmers will need basic digital literacy to place orders.
* **Training & Awareness**: Some awareness programs or guides may be required to help farmers use the platform effectively.
* **Supplier Onboarding**: Companies need to register and integrate their product catalogs on the platform.
* **What is the Time frame to recover ROI?**
* **Short-Term (6-12 months)**: Initial revenue from supplier partnerships and advertisements.
* **Mid-Term (1-2 years)**: Increased adoption by farmers leads to consistent revenue generation.
* **Long-Term (2+ years)**: Expansion into other agricultural services like advisory, bulk orders, and financing.
* **How to identify Stakeholder?**
* **Primary Stakeholders**: Mr. Henry (Project Sponsor), Mr. Pandu & Mr. Dooku (Committee Members), Mr. Karthik (Delivery Head), and the APT IT Solutions Team.
* **Secondary Stakeholders**: Agricultural product suppliers, logistic partners, and end-users (farmers).
* **Influencers**: Government bodies, agricultural associations, and NGOs supporting digital farming initiatives.

# Question 8: SDLC Methodologies

**1. Sequential Methodology:** A step-by-step approach where each phase must be completed before moving to the next. Example: Waterfall Model

**Characteristics**:

* Requirements are gathered at the beginning and remain fixed.
* Each phase (requirement analysis, design, development, testing, deployment) follows a strict order.
* Suitable for projects with well-defined requirements and minimal changes.

**Pros**:

* Clear documentation and structured workflow.
* Easy to manage due to its linear nature.

**Cons**:

* Difficult to adapt to changes once development has started.
* Late-stage defects are costly to fix.

**2. Iterative Methodology:** Development occurs in repeated cycles, with refinements made in each iteration. Example: Rational Unified Process (RUP).

**Characteristics**:

The project is broken into smaller iterations, with each iteration refining previous work.

Feedback is incorporated throughout the development cycle.

Requirements can evolve over time.

**Pros**:

* Better risk management as issues are identified early.
* Provides working software at the end of each iteration.

**Cons**:

* Requires more communication and documentation.
* May lead to scope creep if not properly managed.

# Question 8: SDLC Methodologies

**3.Evolutionary Methodology**: Software is developed in increments, allowing early delivery of functional components while evolving based on feedback. Example: Spiral Model.

**Characteristics**:

* Emphasizes risk analysis and iterative refinement.
* Each phase includes planning, risk evaluation, development, and testing.
* Suitable for projects where requirements are unclear or likely to change.

**Pros**:

* Allows for early risk identification and mitigation.
* Can accommodate changing requirements.

**Cons**:

* Requires skilled resources for effective risk assessment.
* Can be expensive and time-consuming if not managed properly.

**4. Agile Methodology**: A flexible, customer-centric approach where software is developed in short cycles called sprints. Example: Scrum Model.

**Characteristics**:

* Focuses on iterative development with regular customer feedback.
* Prioritizes working software over extensive documentation.
* Encourages collaboration between cross-functional teams.

**Pros**:

* Quick adaptability to changing requirements.
* Delivers functional software frequently.

**Cons**:

* Requires high team involvement and constant communication.
* Not ideal for projects with strict regulatory requirements.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Features** | **Pros** | **Cons** |
| Waterfall | Sequential | Structured | Inflexible |
| Model | Testing in parallel | High quality | Costly |
| Spiral | Risk assessment | Adaptable | Complex |
| Scrum | Agile framework | Fast delivery | Requires continuous engagement |

# Question 9: Write about SDLC Models

**Recommended Model:** V Model, due to parallel testing benefits.

# Question 10: Waterfall model vs V model

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Waterfall Model** | **V Model** |
| **Definition** | A linear and sequential SDLC approach where each phase must be completed before the next begins. | An extension of the Waterfall Model where testing is planned parallel to the development phase. |
| **Process Flow** | Follows a strict step-by-step sequence (Requirement → Design → Development → Testing → Deployment). | Has a V-shaped structure, with validation (development) on one side and verification (testing) on the other. |
| **Testing Approach** | Testing is done only after the development phase is complete. | Testing is planned at each stage and runs parallel to development. |
| **Flexibility** | Rigid model with minimal flexibility once requirements are finalized. | Slightly more flexible due to early testing, but still requires well-defined requirements. |
| **Risk Management** | High risk as defects are identified late in the process. | Lower risk since defects are detected early in the development cycle. |
| **Cost of Fixing Errors** | High, as defects are found at the later stages of the development. | Lower, as issues are identified early through corresponding test phases. |
| **Best Suited For** | Projects with well-defined, stable requirements that are unlikely to change. | Projects where testing needs to be emphasized at every stage. |
| **Parallel Testing** | Not supported; testing is a separate final phase. | Supported; each development phase has a corresponding testing phase. |
| **Documentation** | Requires extensive documentation. | Requires even more documentation as test plans are created for every stage. |

# Question 11: Justify your choice

**Why V model is suitable?**

* **Early Testing & Quality Control**

Testing happens at **every phase** (D1 → T1, D2 → T2, etc.), so **issues are caught early** before moving forward.

* **Clear Structure & Well-Defined Requirements**

Since the project's requirements are **fixed**, the V-Model ensures **everything is properly validated** before development.

* **Minimizes Risks & Delays**

Since **testing and development go hand in hand**, major errors don’t pile up at the end, avoiding last-minute surprises.

* **Better for Fixed Timeline & Budget**

The project is **CSR-funded** with a strict **18-month deadline**—V-Model ensures **systematic progress** without unexpected delays.

* **Ensures Security & Reliability**

The platform involves **financial transactions and supply chain processes**, so **rigorous testing is necessary** to avoid failures.

**Conclusion**

The V-Model is best because it ensures early error detection, structured development, and a reliable system within a fixed timeline. It’s ideal for stable requirements and minimizes risks by testing at every step.

# Question 12: Gnatt Chart for Project Agriculture Product Store

# Question 12: Gnatt Chart for Project Agriculture Product Store

# Question 13: Fixed Billing vs Billing

|  |  |  |
| --- | --- | --- |
| **Feature** | **Fixed Bid Model** | **Billing (T&M) Model** |
| **Budget** | Fixed, agreed upon before the project starts. | Variable, depends on hours worked and resources used. |
| **Scope** | Predefined and fixed. | Flexible and can evolve over time. |
| **Risk Ownership** | Vendor bears the risk of underestimation. | Shared between client and vendor, depending on project execution. |
| **Client Involvement** | Minimal after initial requirement gathering. | High involvement throughout the project. |
| **Change Management** | Difficult; requires contract modifications. | Easy; changes can be accommodated dynamically. |
| **Time Estimation** | Defined upfront, with a fixed deadline. | Can change based on project complexity and scope expansion. |
| **Best for** | Small to medium projects with well-defined requirements. | Large, evolving projects with uncertain requirements. |

# Question 14: BA Timesheets for SDLC Stages

1. **Design Phase Timesheet**

|  |  |  |
| --- | --- | --- |
|  | **Design Phase Timesheet** |  |
| **S. No** | **Tasks** | **Actionable Items** | **Start Time** | **End Time** | **Duration** |
| 1 | Requirement brainstorming | Discussing high-level features with stakeholders | 10.00 AM | 11.30 AM | 1.5 hours |
| 2 | Wireframe creation | Drafting UI mockups with designers | 11.30 AM | 1.00 PM | 1.5 hours |
| 3 | Reviewing design documents | Aligning with business goals | 2.00 PM | 3.30 PM | 1.5 hours |
| 4 | Design feedback session | Internal review with the team | 3.3 PM | 4.30 PM | 1 hour |
| 5 | Finalizing design elements | Updating the wireframes based on feedback | 4.30 PM | 6.00 PM | 1.5 hours |
| **Total** **Duration** |  |  |  |  | **7 hours** |

# Question 14: BA Timesheets for SDLC Stages

1. **Development Phase Timesheet**

|  |  |  |
| --- | --- | --- |
|  | **Development Phase Timesheet** |  |
| **S. No** | **Tasks** | **Actionable Items** | **Start Time** | **End Time** | **Duration** |
| 1 | Requirement clarification | Resolving queries from developers | 10.00 AM | 11.00 AM | 1 hour |
| 2 | Reviewing functional documents | Ensuring alignment with business requirements | 11.00 AM | 12.30 PM | 1.5 hours |
| 3 | Attending stand-up meetings | Status updates with the development team | 2.00 PM | 3.00 PM | 1 hour |
| 4 | Documenting change requests | Tracking modifications requested during development | 3.00 PM | 4.30 PM | 1.5 hours |
| 5 | Internal discussion | Aligning with project managers and teams | 4.30 PM | 6.00 PM | 1.5 hours |
| **Total** **Duration** |  |  |  |  | **6.5 hours** |

# Question 14: BA Timesheets for SDLC Stages

**3. Development Phase Timesheet**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Testing Phase Timesheet** |  |  |  |
| **S. No** | **Tasks** | **Actionable Items** | **Start Time** | **End Time** | **Duration** |
| 1 | Test case review | Ensuring test cases cover all scenarios | 10.00 AM | 11.30 AM | 1.5 hours |
| 2 | Functional validation | Cross-checking functionality against requirements | 11.30 AM | 1.00 PM | 1.5 hours |
| 3 | Defect triage meeting | Reviewing and prioritizing defects with QA | 2.00 PM | 3.00 PM | 1 hour |
| 4 | Updating requirement traceability matrix | Mapping test cases to business needs | 3.00 PM | 4.30 PM | 1.5 hours |
| 5 | Retesting & closure | Ensuring fixes meet requirements | 4.30 PM | 6.00 PM | 1.5 hours |
| **Total** **Duration** |  |  |  |  | **7 hours** |

# Question 14: BA Timesheets for SDLC Stages

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

|  |  |  |
| --- | --- | --- |
| **UAT (User Acceptance Testing) Phase Timesheet** |  |  |
| **S. No** | **Tasks** | **Actionable Items** | **Start Time** | **End Time** | **Duration** |
| 1 | UAT planning | Identifying test scenarios | 10.00 AM | 11.00 AM | 1 hour |
| 2 | UAT execution | Supporting users in test execution | 11.00 AM | 1.00 PM | 2 hours |
| 3 | UAT defect tracking | Documenting issues reported by users | 2.00 PM | 3.00 PM | 1 hour |
| 4 | Requirement validation | Ensuring changes align with business needs | 3.00 PM | 4.30 PM | 1.5 hours |
| 5 | Final UAT report preparation | Summarizing findings for approval | 4.30 PM | 6.00 PM | 1.5 hours |
| **Total** **Duration** |  |  |  |  | **7 hours** |

# Question 14: BA Timesheets for SDLC Stages

**5. Deployment & Implementation Phase Timesheet**

|  |  |  |
| --- | --- | --- |
| Deployment & Implementation Phase Timesheet |  |  |
| **S. No** | **Tasks** | **Actionable Items** | **Start Time** | **End Time** | **Duration** |
| 1 | Pre-deployment validation | Ensuring all requirements are met | 10.00 AM | 11.30 AM | 1.5 hours |
| 2 | Deployment monitoring | Coordinating with the IT team | 11.30 AM | 1.00 PM | 1.5 hours |
| 3 | Issue resolution support | Assisting in troubleshooting | 2.00 PM | 3.30 PM | 1.5 hours |
| 4 | Training & documentation | Creating user manuals and training end users | 3.30 PM | 4.30 PM | 1 hour |
| 5 | Go-live support | Addressing post-deployment issues | 4.30 PM | 6.00 PM | 1.5 hours |
| **Total** **Duration** |  |  |  |  | **7 hour** |