**Question 1**: Identify Business Process Modelling for Online Agriculture Store

**Answer**: Business Process Modelling for Online Agriculture Store is below:

**Goal**: To bridge the gap between farmers and agriculture product manufacturer

**Inputs**: Product need of the farmers, online platform, easy to operate, train farmers how to operate, payment gateway

**Resources**: Internet connectivity, software, mobile phone, manufacturers

**Output**: availability of the agricultural products, good quality grain and vegetables, ease from the tension of getting crops destroy

**Activities**: partnership with the manufacturer, getting manufacturer’s firm registered with the app, good and trust worthy logistic partner

**Value**: connecting city to village, premium quality produce, idea of access to farmers in the open market using same software

**Question 2**: SWOT Analysis

**Answer**: Aspects to consider for SWOT analysis are as follows:

**Strengths** **Weakness**

|  |  |
| --- | --- |
| * Aid to all the other farmers in   different rural location   * Connectivity to rural areas | * Connecting different manufacturers * Finding logistic partners * Literacy rate of village farmers |

**Opportunity** **Threat**

|  |  |
| --- | --- |
| * Village farmers to compete with city farmers * Equal platform for farmers * A future opportunity of business | * Expensive than expected * Not viable for all farmers |

**Question 3**: Feasibility Study

**Answer**: **Technology**: collection of data of farmers need in systematic manner, software for app development, payment gateways, platform where all different software can work together

**Hardware**: Logistics, warehouse, containers

**Software**: product software, payment software, network connectivity, confirmation on the cart

**Resources**: software developers, project managers, business analyst, testers admin

**Budget**: 2 cr

**Timeframe**: 18 months

**Question 4**: GAP Analysis

**Answer**: **Current State**

* To build an online store to facilitate farmers
* Looking for manufacturers
* Arranging for logistics partners
* Application should work fine and be able to accept the products farmers are looking for

**Desired State**

* A full-fledged working application from where farmers can order seed, pesticides, fertilizers etc.
* Easy to operate software for farmers
* Farmers in the remote village do not have to face this problem of unavailability of basic agriculture products
* Reaching out to maximum of remote villages with the technology.

Question 5: Risk Analysis

Answer: Internal Risk:

* Dependency on the manufacturers
* Dependency on the logistics partners
* Literacy, understanding and comprehension of farmers in rural areas

External Risk:

* Internet connectivity
* Change in government regulations and policies

BA Risk:

* Change in requirement
* Lack of user involvement
* Improper planning
* Unrealistic expectations

Project based risk:

* Scope creep
* Lack of executive support

**Question 6**: Stakeholder Analysis (Raci Matrix)

|  |  |  |  |
| --- | --- | --- | --- |
| **R/A/C/I** | **Name** | **Designation** | **Details** |
| **Responsible** | Mr Pandu  Mr Dooku  Kevin  Ben  Peter | Financial Head  Project Coordinator  Stakeholder  Stakeholder  Stakeholder | Email- pandu@123.com  Phone No.- 7868773892  Reach out- 11am to 4pm IST  Email- dooku@123.com  Phone No.- 7868773892  Reach out-11am to 4pm IST  Email- kevin@123.com  Phone No.- 7868773892  Reach out-11am to 4pm IST  Email- ben@123.com  Phone No.- 7868773892  Reach out-11am to 4pm IST  Email- peter@123.com  Phone No.- 7868773892  Reach out-11am to 4pm IST |
| **Accountable** | Mr Karthik  Mr Vandanam | Delivery Head  Project Manager | Email- karthik@123.com  Phone No.- 7868773892  Reach out-11am to 4pm IST  Email- vandanam@123.com  Phone No.- 7868773892  Reach out- 11am to 4pm IST |
| **Consulted** | Ms Juhi  Mr Teyson  Ms Lucie  Mr Tucker  Mr Bravo  Mr Mike  Mr John  Mr Jason  Mr Bravo  Ms Yogita | Sr Java Developer  Java Developer  Java Developer  Java Developer  Java Developer  Network Admin  DB Admin  Tester  Tester  BA | Email- juhi@123.com  Phone No.- 7868773892  Reach out- 11am to 4pm IST  Email- teyson@123.com  Phone No.- 7868773892  Reach out- 11am to 4pm IST  Email- lucie@123.com  Phone No.- 7868773892  Reach out- 11am to 4pm IST  Email- tucker@123.com  Phone No.- 7868773892  Reach out- 11am to 4pm IST  Email- bravo@123.com  Phone No.- 7868773892  Reach out- 11am to 4pm IST  Email- mike@123.com  Phone No.- 7868773892  Reach out- 11am to 4pm IST  Email- john@123.com  Phone No.- 7868773892  Reach out- 11am to 4pm IST  Email- jason@123.com  Phone No.- 7868773892  Reach out- 11am to 4pm IST  Email- bravo@123.com  Phone No.- 7868773892  Reach out-11am to 4pm IST  Email-yogita@123.com  Phone No.- 7868773892  Reach out- 11am to 4pm IST |
| **Informed** | Mr Henry | Sponsor | Email- henry@123.com  Phone No.- 7868773892  Reach out- 10am to 6pm IST |

**Question 7**: Business Case Document

**Answer**:

* **Why is the project initiated?**
* To build an online store to facilitate farmers to buy seeds, pesticides, fertilizers etc.
* **What are the current problem?**
* Framers facing difficulties in procuring seeds, pesticides, fertilizers etc.
* **With this project, how many problems can be solved?**
* With this project farmer would not have to worry about the supplies for farming and they can produce good quality product which would eventually help them to reach the urban area as well and could provide the opportunity to compete in the urban agricultural market.
* **What are the resources required?**
* Software for the application, project team (developers, testers, project manager, BA), stakeholders, budget, manufacturers and logistics venders.
* **How much organizational change is required to adapt this technology?**
* Change in the project team, allocating team members as per the requirement of the project, for e.g., allocating developers to develop an application for farming products, testers need to check that it should be user friendly, project manager should analyze the required resources accurately.

Every project is different from another even if the product or service is same, that is the reason that it should be built differently for every another project same goes with the budget for the project.

* **What is Time frame to recover ROI?**
* 18 Months
* **How to identify the stakeholders?**
* By the needs and requirement of the project.

By the analyzing the interest in the project.

**Question 8**: SDLC and it’s 4 methodologies

**Answer**: Software Development Life Cycle (SDLC), is a systematic process followed to develop software, encompassing phases like analysis, design, development, testing, deployment and maintenance with different methodologies used to manage these phases to ensure a high quality product is delivered.

These methodologies are:

1. **Sequential- Waterfall**

Each phase must be completed in its entirety before the next phase begin. At the end of each phase a review takes place to determine if the project is on the right path and whether to continue with it or not or to discard the project.

|  |  |
| --- | --- |
| Stages of Waterfall model | Resources |
| Requirement Gathering | BA, PM |
| Requirement Analysis | BA, PM, Tech Team |
| Design | Tech Team |
| Development- coding | Programmers, developers |
| Testing | Testers |
| Unit, component, system, system integration, UAT |  |
| Process- Configure Management | Project Manager |
| Deployment & Implementation | Release Engineer (delivering source code into finished products) |

Deployment- moving code from development environment to production

Implementation- running the code for the very first time in production

After implementation, maintenance stage starts, and support team will take care.

1. **Iterative- RUP**

Rational Unified Process (RUP) is an iterative software development process framework, which is based on a set of building blocks or content element, describing what is to be produce, skills required and step by step explanation describing how specific development goals are to be achieved.

RUP is iterative meaning it involves breaking down a project into smaller cycles called iteration, where each iteration delivers a working piece of software and allows continuous feedback and improvement throughout the development process.

Building blocks or content elements are the following:

* Roles (who)- set of related skills, competency and responsibility
* Work product (what)- something resulting from the task, including all the documents and models produced while working through the project
* Tasks (how)- a unit of work assigned to a role that provide meaningful result.

Within each iteration, tasks are categorized into 9 disciplines:

6 Engineering Disciplines:

* Business Modelling- defining the business needs and objective
* Requirements- gathering and documenting functional and non-functional requirements
* Analysis and design- breaking down requirements into system architecture and detailed design
* Implementation- coding and developing the software components
* Test- executing test cases to verify the functionality and quality of the software
* Deployment- delivering the completed software to the end user

3 Supporting Disciplines:

* Configuration and change management- focus on managing changes to the system throughout the development lifecycle.
* Project Management- oversees the overall project planning, scheduling, resource allocation, risk management, and communication to ensure the project is on track and meets its goal.
* Environment- it deals with technical infrastructure needed for development, including tools, platforms and the development environment itself.

4 Project Lifecycle Phases:

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

1. **Evolutionary- Spiral**

It gives more emphasis based on risk analysis. The spiral model has 4 phases:

* Planning- gathering the requirements
* Risk analysis- identify risks and alternate solution. At the end a prototype is produced
* Engineering- software is produced along with the testing at the end of the phase.
* Evolution- this phase allows customer to evaluate the output of the project to date before the project continues to the next spiral

Advantages:

* High amount of risk analysis
* Good for large and critical mission
* Software is produced early in software lifecycle

Disadvantage:

* Can be costly
* Risk analysis require high level of expertise
* Project’s success highly depends on the risk analysis phase
* Doesn’t work well for smaller projects

1. **Agile- Scrum**

* It can be implemented where faster delivery is required.
* No documentation
* Customer retention since no documentation is required
* The code in itself forms as a documentation
* SDLC cut down by employing seasoned Developers

4 main values:

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

12 principles of Agile:

* Satisfy the customer through early and continuous delivery of valuable software
* Welcome changing requirements, even late in development. It harnesses change for the customer’s competitive advantage
* Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale
* Business people and developers must work together daily throughout the project
* Build project around motivated individuals. Give them the environment and support they need and trust them to get job done
* Effective and efficient way of conveying information to and within a development team is face-to-face conversation
* Working software is primary measure of progress
* It promotes sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely
* Technical excellence and good design enhance agility
* Simplicity- the art of maximizing the amount of work not done is essential
* The best architectures, requirements and designs emerge from self-organizing teams
* At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

Meetings:

* Sprint Planning Meeting- this happens at the beginning of each sprint and team decides on what they will be delivering in the sprint.
* Daily Scrum Meeting- this happens each day where team will just answer 3 questions:
* What did you do today?
* What will you do tomorrow?
* Are there any impediments that is slowing or stopping us?
* Sprint Review Meeting- this happens at the end of the sprint where team will demo the completed stories to product owner and get it cleared
* Sprint Retrospective Meeting – happens at the end of the sprint where team will answer these 3 questions:
* What went well in the sprint?
* What did not go well?
* What are the required areas of improvement in next sprint?

**Question 9**- SDLC models

**Answer**- Software Development Life Cycle (SDLC), is a systematic process followed to develop software, encompassing phases like analysis, design, development, testing, deployment and maintenance with different methodologies used to manage these phases to ensure a high quality product is delivered.

These methodologies are:

1. **Sequential- Waterfall**

Each phase must be completed in its entirety before the next phase begin. At the end of each phase a review takes place to determine if the project is on the right path and whether to continue with it or not or to discard the project.

|  |  |
| --- | --- |
| Stages of Waterfall model | Resources |
| Requirement Gathering | BA, PM |
| Requirement Analysis | BA, PM, Tech Team |
| Design | Tech Team |
| Development- coding | Programmers, developers |
| Testing | Testers |
| Unit, component, system, system integration, UAT |  |
| Process- Configure Management | Project Manager |
| Deployment & Implementation | Release Engineer (delivering source code into finished products) |

Deployment- moving code from development environment to production

Implementation- running the code for the very first time in production

After implementation, maintenance stage starts, and support team will take care.

1. **Iterative- RUP**

Rational Unified Process (RUP) is an iterative software development process framework, which is based on a set of building blocks or content element, describing what is to be produce, skills required and step by step explanation describing how specific development goals are to be achieved.

RUP is iterative meaning it involves breaking down a project into smaller cycles called iteration, where each iteration, where each iteration delivers a working piece of software and allows continuous feedback and improvement throughout the development process.

Building blocks or content elements are the following:

* Roles (who)- set of related skills, competency and responsibility
* Work product (what)- something resulting from the task, including all the documents and models produced while working through the project
* Tasks (how)- a unit of work assigned to a role that provide meaningful result.

Within each iteration, tasks are categorized into 9 disciplines:

6 Engineering Disciplines:

* Business Modelling- defining the business needs and objective
* Requirements- gathering and documenting functional and non-functional requirements
* Analysis and design- breaking down requirements into system architecture and detailed design
* Implementation- coding and developing the software components
* Test- executing test cases to verify the functionality and quality of the software
* Deployment- delivering the completed software to the end user

3 Supporting Disciplines:

* Configuration and change management- focus on managing changes to the system throughout the development lifecycle.
* Project Management- oversees the overall project planning, scheduling, resource allocation, risk management, and communication to ensure the project is on track and meets its goal.
* Environment- it deals with technical infrastructure needed for development, including tools, platforms and the development environment itself.

4 Project Lifecycle Phases:

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

1. **Evolutionary- Spiral**

It gives more emphasis based on risk analysis. The spiral model has 4 phases:

* Planning- gathering the requirements
* Risk analysis- identify risks and alternate solution. At the end a prototype is produced
* Engineering- software is produced along with the testing at the end of the phase.
* Evolution- this phase allows customer to evaluate the output of the project to date before the project continues to the next spiral

Advantages:

* High amount of risk analysis
* Good for large and critical mission
* Software is produced early in software lifecycle

Disadvantage:

* Can be costly
* Risk analysis require high level of expertise
* Project’s success highly depends on the risk analysis phase
* Doesn’t work well for smaller projects

1. **Agile- Scrum**

* It can be implemented where faster delivery is required.
* No documentation
* Customer retention since no documentation is required
* The code in itself forms as a documentation
* SDLC cut down by employing seasoned Developers

4 main values:

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

12 principles of Agile:

* Satisfy the customer through early and continuous delivery of valuable software
* Welcome changing requirements, even late in development. It harnesses change for the customer’s competitive advantage
* Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale
* Business people and developers must work together daily throughout the project
* Build project around motivated individuals. Give them the environment and support they need and trust them to get job done
* Effective and efficient way of conveying information to and within a development team is face-to-face conversation
* Working software is primary measure of progress
* It promotes sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely
* Technical excellence and good design enhance agility
* Simplicity- the art of maximizing the amount of work not done is essential
* The best architectures, requirements and designs emerge from self-organizing teams
* At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

Scrum can be implemented either at the beginning of the project or when you sense that project is falling behind schedule.

* Scrum Team- it is project resources comprises of BA, Developers, Testers. Average team size of 7-8.
* Product Owner- decides what needs to be in the product and how the product needs to be. Regular interaction with the customers and BAs. Role may be played by BA or any person who worked for end users for a long time or customer himself.
* Scrum Master- he will monitor the performance of the team within the sprint. Team will raise all their issues to scrum master and he will run to look for answers. This role can be played by any person in team normally BA’s play this role.
* Product Burn down- it shows how much work was left to do at the beginning of each sprint.
* Sprint- this is the period that team decides to deliver their objective. Normally a sprint period will be for 2 weeks but may extend to 4 weeks.

Meetings:

* Sprint Planning Meeting- this happens at the beginning of each sprint and team decides on what they will be delivering in the sprint.
* Daily Scrum Meeting- this happens each day where team will just answer 3 questions:
* What did you do today?
* What will you do tomorrow?
* Are there any impediments that is slowing or stopping us?
* Sprint Review Meeting- this happens at the end of the sprint where team will demo the completed stories to product owner and get it cleared
* Sprint Retrospective Meeting – happens at the end of the sprint where team will answer these 3 questions:
* What went well in the sprint?
* What did not go well?
* What are the required areas of improvement in next sprint?

Product Back Log: all Stories- all requirements

Burn down Chart- it is a graphical view of the remaining work left versus the time in an iteration. It is often used to determine when work will be completed on a project or an iteration.

Epic- set of related user stories, also considered a “really big user story”.

Iteration- it is an iterative development concept that establishes a short time frame to deliver a set of software features or user stories. Each iteration includes typical waterfall activities such as analysis, design, development and testing, yet they are time boxed within a 1-4 weeks window. At the end of an iteration, the progress is reviewed with the business customer, and recommended changes can be incorporated into future iteration.

Planning Poker- an estimation game created by Mike Cohn of Mountain Goat Software. Used to estimate individual user stories as a team activity. The team gathers and reviews user stories one at a time. As stories are presented, the team discusses the user story and provides an estimate of the work from their own deck of cards. All estimates are presented and discussed until the team arrives at a consensus.

Release- it is a set of working software delivered to the business customer resulting from a set of iteration. During release, planning team will review a product backlog to organize user stories into the specific releases and iteration that deliver a functional product to the business customer.

Scrum- it is an iterative development methodology used to manage software projects. In scrum-based projects, there isn’t a specific project manager directing project team tasks; the team is self-directed, with collocated team members relying on communication over documentation for effective project delivery.

Sprint- it is a scrum based agile methodology concept that is similar to an iteration. It is a time boxed to deliver a specific set of user stories and produce working features within a set time period. During sprint planning, the business customer or product owner specifies the user story priority, and the development team commits to the scope for a given sprint. During a sprint, user stories can be removed from the sprint scope, but new stories cannot be added, this allows project teams to focus on the goals of the sprint and deliver rapidly.

Story Points- it is a relative estimation method used to determine the size of user stories so teams can determine how much work can be done during an iteration. It can be expressed in a simple Fibonacci sequence, t-shirt sizes, or a relative number. By adding up the number of user stories and associated story points, the project team can establish its velocity for future iteration planning.

BA’s role in Agile Scrum- Once a project is kicked off, BA does the requirement planning, then conducts various requirement gathering sessions and analyze the requirements. The requirements are listed as “FEATURE LIST”, which is drafted by BA and discussed with Product owner. List will have all enhancement and existing features (if it is a migration project). From Feature list BA identifies the Epic and breaks them as Theme and then to User stories.

1. **V-model** - it is a project management framework that emphasizes thorough testing and validation at each stage of the development process, ensuring quality by linking each development phase with a corresponding testing phase, creating a visual representation resembling a “V” shape. It is structured approach to identify potential flaws early on and guarantee the final product meets requirement effectively.

Key points:

* Verification and validation- “V” shape represents the left side as the verification phase (design and planning) where requirements are reviewed, and the right side as the validation phase (testing and evaluation) where the developed product us tested against those requirements.
* Early Testing- testing begins early in the development cycle, allowing for prompt detection and correction of issues
* Sequential Phases- it progresses through distinct phases, with each step completed before moving to the next, providing clear traceability and accountability
* Application Beyond Software- it can be applied to other business analysis projects like process improvement or system implementation, where rigorous testing and validation are crucial.

As a BA I will: Analyze the situation:

* SMEs’ Preference for V-Model:
* SMEs likely value thorough testing and validation, especially in an agriculture product store where accuracy in product information, pricing and transaction is crucial
* They may be concerned about potential errors or inconsistencies that could negatively impact customer trust
* Project Team’s Preference for Waterfall Model:
* The project team might be prioritizing a straightforward, predictable development process
* They may perceive the waterfall model as simpler to manage, especially if they have limited experience with the V-Model
* They may also want to reduce the overhead of the extra documentation that the V model requires

Recommendation: Given the nature of an online agriculture product store, the V-model is likely the more suitable methodology. Here’s why:

* Emphasis on quality: The V-model focus on verification and validation aligns with the need for accurate and reliable product information and transactions
* Risk Mitigation: early and continuous testing helps identify and address potential issues before they escalate
* Stakeholder Confidence: The V-model’s structured approach and emphasis on testing can build confidence among SMEs and other stakeholders
* Handling Complexity: V-model’s structured testing phases can help manage the complexity of an e-commerce platform

While the waterfall model might seem simple, the V-model’s emphasis on quality and testing makes it a better fit for an online agriculture product store. By addressing the project team’s concerns and providing adequate support, APT IT SOLUTIONS can successfully implement the V-model and deliver a high quality product.

**Question 10**- Waterfall Vs V-Model

**Answer**-

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Waterfall** | **V-Model** |
| **Definition** | A sequential design process where progress flows in one direction, like a waterfall | A sequential development process where each phase corresponds to a testing phase |
| **Cost** | The cost of Waterfall model is low | V-model is expensive |
| **Simplicity** | Waterfall model is simple | Simplicity of V-Model is intermediate |
| **Flexibility** | Waterfall is rigid | V-Model is flexible |
| **Phases** | There is no way to return to the earlier phase | There is no such constraint in V-model |
| **Risk management** | Risks are identified late in the process | Risks are identified earlier in the process due to continuous testing |
| **Linear Movement of Steps** | It move in a linear way | It doesn’t move in linear way |
| **Reusability** | Reusability is limited | It can be Re-use for some extent |
| **User Involvement** | Only in the beginning | More involvement compared to waterfall through verification and validation |
| **Testing Activities Start** | It starts after the development activities are over | It starts with the first stage |
| **Success Guarantee** | Guarantee of success is low | Guarantee of success is high |
| **Process** | Continuous process | Simultaneous process |
| **Defects** | More defects found in Testing phase | Less defects found in testing phase |
| **Project Size** | Suited for larger project | Suited for smaller to medium sized project |
| **Suitability** | Best for projects with well-defined, stable requirements and minimal changes | Suitable for projects where quality and reliability are critical, and where requirements are relatively stable |
| **Change Management** | Changes are expensive and difficult to implement | Changes are less expensive and easier to implement |
| **Focus** | Development centric | Verification and validation centric |
| **Completion Time** | Longer development time | Shorter completion time |
| **Documentation** | Detailed documentation for each phase | Emphasizes both on development and testing documentation for each phase |
| **Examples of Usage** | Used in projects with clear, fixed requirements (e.g., hardware development, government projects) | Common in software development where each phase has a corresponding test plan (e.g., medical software development, safety-critical systems) |

**Question 11**- As a BA, state your reason for choosing one model for this project

**Answer**- Given the scenario, I would recommend an Agile methodology, specifically a Scrum-based approach, with elements of iteration development. Here’s why:

* User-centric Development:
* Allows continuous feedback from farmers, ensuring the application meets their needs and is user-friendly
* Regular sprints and demos enable early validation of features
* Flexibility and Adaptability:
* Agile accommodates changing requirements and allows for adjustments based on user feedback and market dynamics
* This is crucial for a project with potential for evolving needs
* Early and Frequent Delivery:
* Enables the delivery of working software in increments, allowing for early user testing and feedback
* This will allow the team to get feedback from the farmers as soon as possible
* Stakeholder Collaboration:
* It provides collaboration between the development team, stakeholders (farmers, committee), and manufacturers
* Regular sprint reviews and planning sessions ensure alignment and transparency
* Risk Mitigation:
* By breaking down the project into smaller, manageable sprints, and it helps mitigate risks and allows for early detection of issues
* Focus on Value:
* It focuses on delivering the highest value features first, ensuring the application provides immediate benefits to farmers
* User friendly testing:
* It processes and sprints, allow for constant user testing, and allow for quick changes to the UI/UX to make it more user friendly

Implementation Considerations:

* Sprint Planning:
* Involve farmers and the committee in sprint planning to prioritize features and ensure alignment with their needs
* Regular Demos and Reviews:
* Conduct regular demos and reviews to gather feedback and make necessary adjustments
* User Testing:
* Prioritize user testing with farmers throughout the development process
* Clear Communication:
* Establish clear communication channels between the development team, stakeholders and manufacturers
* Documentation:
* Maintain essential documentation, focusing on user stories, acceptance criteria and key decisions
* Training:
* Provide training to the farmers on how to use the application

By adopting an Agile approach, APT IT SOLUTIONS can deliver a user-friendly, valuable applications the meets the needs of remote farmers and fulfills Mr. Henry’s vision.

**Question 12**: Gantt Chart

**Answer**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Process** | **Jan-Mar** | **Apr-June** | **Jul-Sep** | **Oct-Dec** | **Jan-Mar** | **Apr-June** |
| Requirement Gathering |  |  |  |  |  |  |
| Requirement Analysis |  |  |  |  |  |  |
| Design 1 |  |  |  |  |  |  |
| Testing 1 |  |  |  |  |  |  |
| Design 2 |  |  |  |  |  |  |
| Testing 2 |  |  |  |  |  |  |
| Design 3 |  |  |  |  |  |  |
| Testing 3 |  |  |  |  |  |  |
| Design 4 |  |  |  |  |  |  |
| Testing 4 |  |  |  |  |  |  |
| UAT |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Progress |  |  |  |  |  |  |
| Achievement |  |  |  |  |  |  |

**Question 13**- Fixed bid vs Billing projects

**Answer**-

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Fixed bid** | **Billing Project** |
| **Pricing Agreement** | The total price is agreed upon upfront | Based on actual time and materials used (hourly rates or per unit rates) |
| **Risk** | Service provider bears the risk of overrun | Client bears the risk of high costs due to extended time or resources |
| **Scope Definition** | The scope must be clearly defined and agreed upon at the beginning | The scope can evolve throughout the project flexibility is built into the pricing model |
| **Cost Predictability** | Predictable and fixed | Unpredictable and fluctuate based on the actual work done |
| **Budget Control** | The budget is fixed providing tight control for the client | Client has less control over the budget, as it can increase based on the time and materials used |
| **Flexibility for Changes** | It requires renegotiations and a change order | Change can be accommodated easily without a need for renegotiation |
| **Incentive for Efficiency** | Service provider has an incentive to work efficiently, as they are paid a fixed amount | Less incentive for efficiency since they are paid by the hour or by materials used |
| **Time Management** | Must manage time efficiently to avoid losing money on delays or inefficiency | Time management is less critical for the provider since they are paid for actual tome worked |
| **Payment Terms** | Tied to milestones or final lump sum upon completion | Payments are typically made regularly based on hours worked or material used |
| **Control over Timeline** | Timeline is usually agreed upon and is part of the contract | Timeline is flexible and can be adjusted as the work progresses |
| **Financial Transparency** | Limited financial transparency since the client pays a fixed price regardless of actual cost | Financial transparency is higher as client can see exactly how many hours were worked and what material were used |
| **Suitability for Complex Projects** | Works well for projects with clearly defined outcomes and a well understood scope | Best for projects with evolving or unclear scopes |
| **Client Involvement** | Less ongoing involvement | Clients are more involved |
| **Payment Risk for Client** | Client may feel more secure as the price is agreed upon upfront | Client assumes more risk, as cost can grow over time, based on actual hours or resources used |
| **Service Provider Motivation** | Providers are incentivized to complete the project as quickly and efficiently as possible to maximize profitability | Providers are incentivized to continue working as long as possible since they are paid for the time spent |
| **Dispute Resolution** | Disputes often arises when the project scope changes, requiring renegotiation or additional payments | Disputes may occur over the number of hours worked or materials used, but it is easier to resolve with ongoing records |
| **Profitability for Service Provider** | Depends on how well the provider can estimate and control the cost | Profitability is tied directly to the amount of time and resources used |
| **Scope Creep** | It can lead to project overruns that are the service provider’s responsibility | It can be accommodated and the client pays for any additional time or materials required |
| **Use in Different Phases of Project** | Used for well-defined project phases | Often used for continuous or long term work |
| **Project Closure** | The project typically ends when all milestones are completed and the final payment is made | The project is open ended and the provider continues working as long as the client needs additional work |

**Question 14:**

**Answer**: a. Design Timesheet of BA

|  |
| --- |
| **Stage Name: Design** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Stage/Process Name** | **Start Time** | **End Time** | **Total Hours** | **Task Description** | **Remarks** |
| DD/MM/YY | Design | HH:MM | HH:MM | HH:MM | Description about the task | Remark for the work done |
| DD/MM/YY | Design |  |  |  |  |  |
| DD/MM/YY | Design |  |  |  |  |  |

b. Development Timesheet of a BA

|  |
| --- |
| **Stage Name: Development** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Stage/Process Name** | **Start Time** | **End Time** | **Total Hours** | **Task Description** | **Remarks** |
| DD/MM/YY | Development | HH:MM | HH:MM | HH:MM | Description about the task | Remark for the work done |
| DD/MM/YY | Development |  |  |  |  |  |
| DD/MM/YY | Development |  |  |  |  |  |

c. Testing Timesheet of a BA

|  |
| --- |
| **Stage Name: Testing** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Stage/Process Name** | **Start Time** | **End Time** | **Total Hours** | **Task Description** | **Remarks** |
| DD/MM/YY | Testing | HH:MM | HH:MM | HH:MM | Description about the task | Remark for the work done |
| DD/MM/YY | Testing |  |  |  |  |  |
| DD/MM/YY | Testing |  |  |  |  |  |

d. UAT Timesheet of a BA

|  |
| --- |
| **Stage Name: UAT** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Stage/Process Name** | **Start Time** | **End Time** | **Total Hours** | **Task Description** | **Remarks** |
| DD/MM/YY | UAT | HH:MM | HH:MM | HH:MM | Description about the task | Remark for the work done |
| DD/MM/YY | UAT |  |  |  |  |  |
| DD/MM/YY | UAT |  |  |  |  |  |

e. Development and Implementation Timesheet of a BA

|  |
| --- |
| **Stage Name: Development and Implementation** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Stage/Process Name** | **Start Time** | **End Time** | **Total Hours** | **Task Description** | **Remarks** |
| DD/MM/YY | Development and Implementation | HH:MM | HH:MM | HH:MM | Description about the task | Remark for the work done |
| DD/MM/YY | Development and Implementation |  |  |  |  |  |
| DD/MM/YY | Development and Implementation |  |  |  |  |  |