

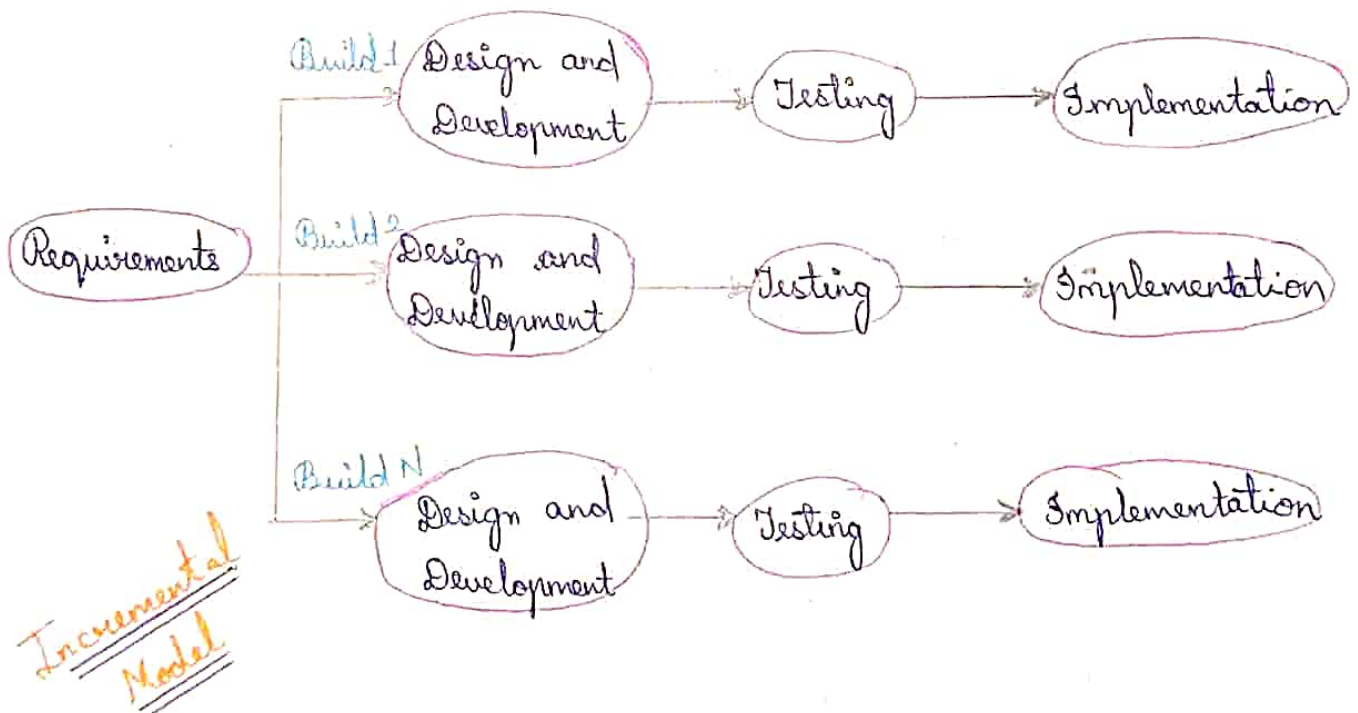
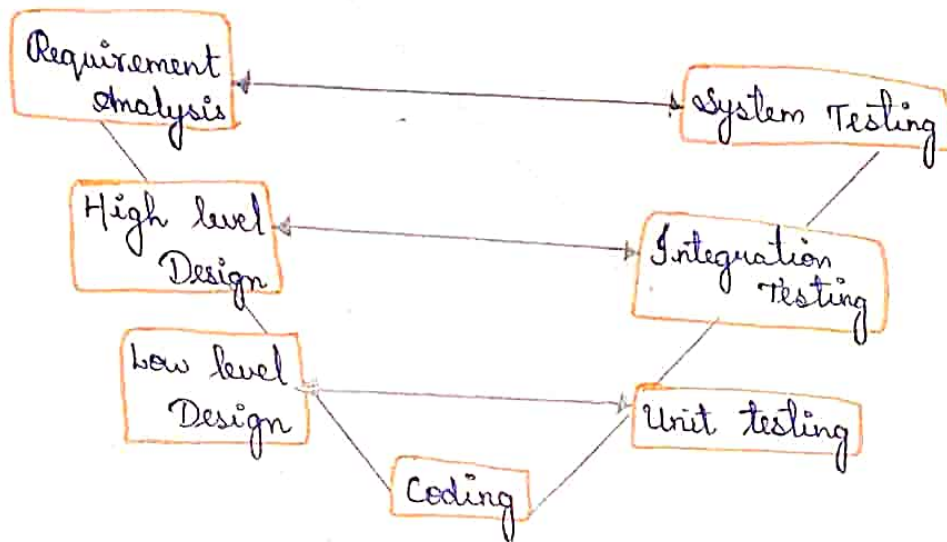
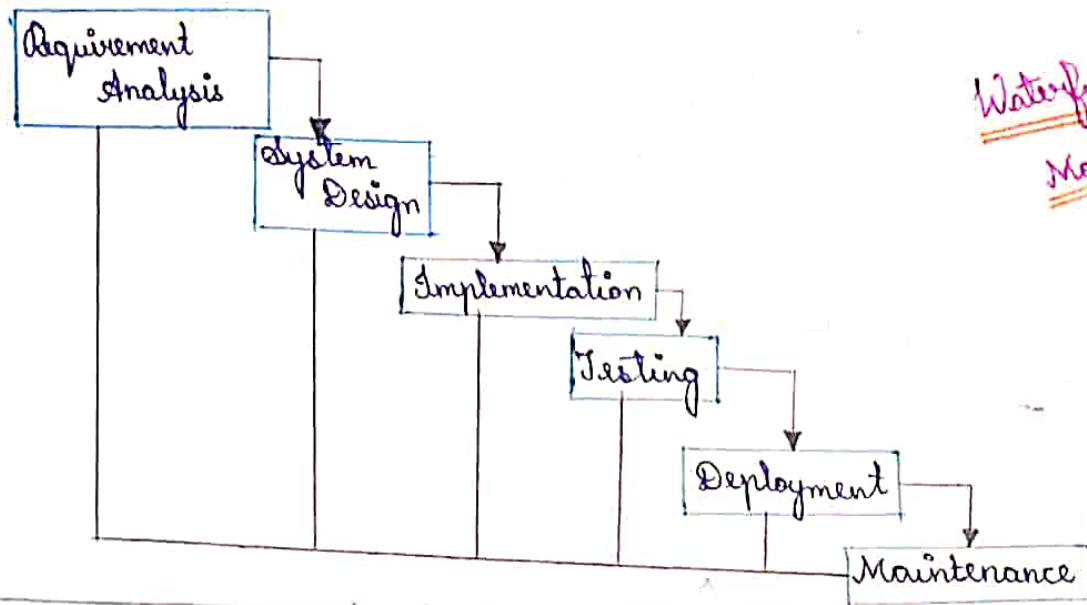
1. Compare the Various Software development process model?

PARAMETER	WATERFALL MODEL	V MODEL	INCREMENTAL MODEL	PROTOTYPE MODEL	SPIRAL MODEL	CONCURRENT MODEL	UNIFIED PROCESS MODEL
Definition of the Process Model	It is a sequential, plan driven process where you must plan and schedule all your activities before starting the project.	Verification & Validation model is an extension of the waterfall model. All the requirements are gathered at the start & cannot be changed.	It divides the system's function into small increments that are delivered one after the other in quick succession. The most important functionality is implemented in initial increments.	It is very famous model. In this model client is also involved at the time of designing the system. It is also modified until the client is not satisfied.	It is a risk driven iterative software process models, its steps aren't activities but phases for addressing whatever problem has the greatest risk of causing a failure.	It is a development model in which multiple processes or phases of the development process are carried out simultaneously. They are essential in software engineering.	UP is a process model that use the Unified Model language to represent models of the software systems to be developed.
Cost	Low	High	High	High	High	Low	High
Phases	Requirements, Design, Implementation, Verification, Deployment, Maintenance.	Verification phases, Validation phases.	Requirement Analysis, Design & Development, Testing, Implementation	Requirement Gathering & Analysis, Quick Design making, Building Prototype, User Evaluation	Planning, Design, Construct, Evaluation.	Requirement Analysis, High level Design, Detailed Design, Coding, Implementation.	Inception, Elaboration, Construction, Production.

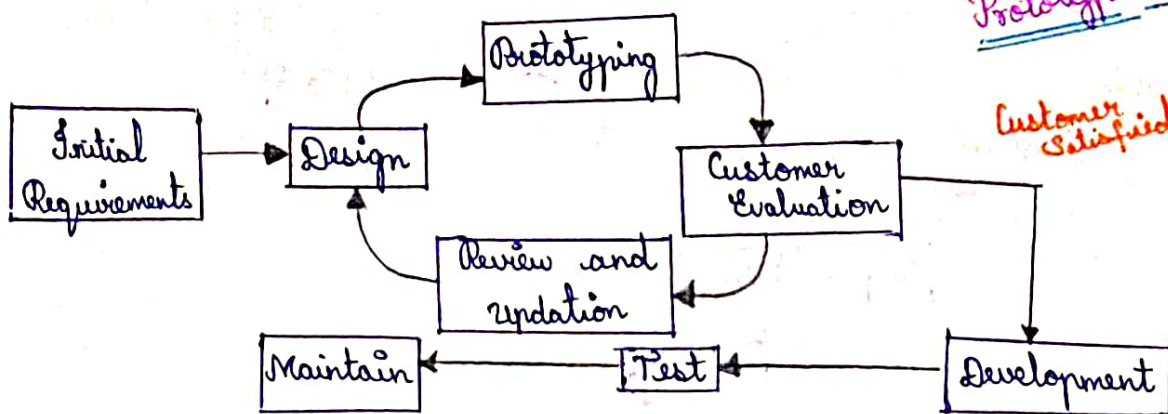
				Prototype Refinement, Building the final Product, Maintenance.			
Advantages	<ol style="list-style-type: none"> 1. Simple and easy to understand 2. Well defined stages and milestones. 	<ol style="list-style-type: none"> 1. Corresponding testing phase for each development stage. 2. Rigorous Verification and Validation throughout. 	<ol style="list-style-type: none"> 1. Delivers a working product at each increment. 2. Easier to test and debug smaller incremental change. 	<ol style="list-style-type: none"> 1. Early user feedback. 2. Enhanced Communication. 	<ol style="list-style-type: none"> 1. Incorporates risk analysis & management. 2. Flexibility to accommodate changes during the development process. 	<ol style="list-style-type: none"> 1. Early Execution. 2. Resource utilization. 	<ol style="list-style-type: none"> 1. Iterative and Incremental Development. 2. Collaboration and Communication
Disadvantages	<ol style="list-style-type: none"> 1. Lack of flexibility to accommodate changes. 2. High risk of client dissatisfaction until the end of the Project. 	<ol style="list-style-type: none"> 1. Still lacks flexibility in handling changes. 2. Testing may be delayed until the later stages 	<ol style="list-style-type: none"> 1. Can lead to issues if the overall system architecture is not well defined. 	<ol style="list-style-type: none"> 1. Risk of Miscommunication. 2. Misleading users. 	<ol style="list-style-type: none"> 1. Complex & may require expertise in risk analysis. 2. Can be time consuming and costly. 	<ol style="list-style-type: none"> 1. Resource Allocation. 2. Risk of Inconsistencies. 	<ol style="list-style-type: none"> 1. Overemphasis on planning 2. Not ideal for small Projects.

Flow Diagram Of Process Models

3



Prototype Model

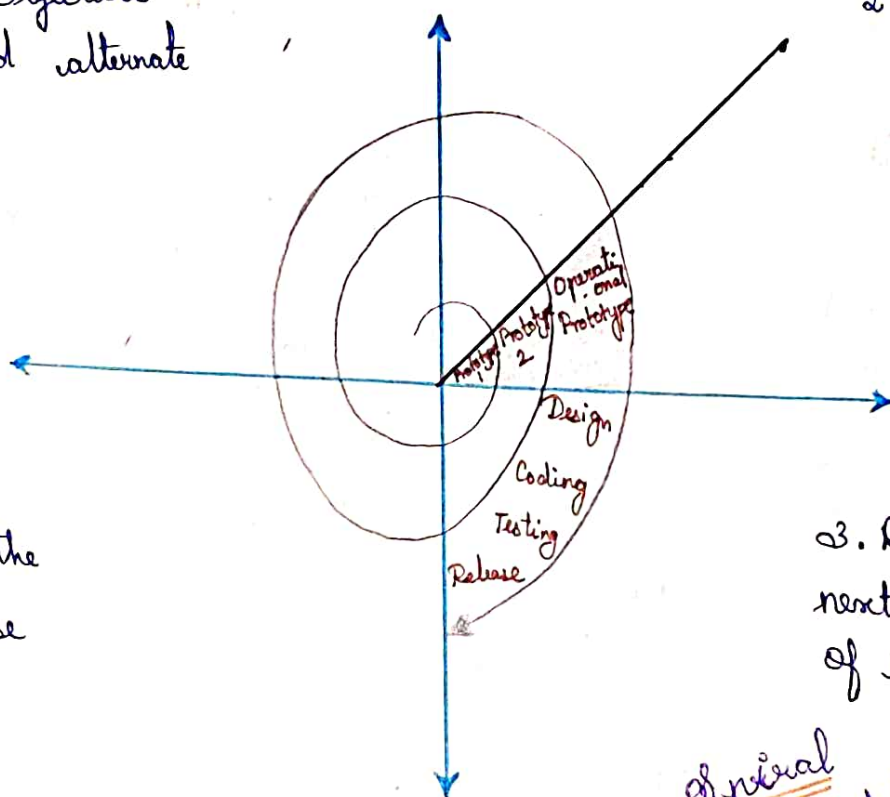


1. Plan Objectives and find alternate solutions

2. Risk Analysis and Resolving

4. Plan the next phase

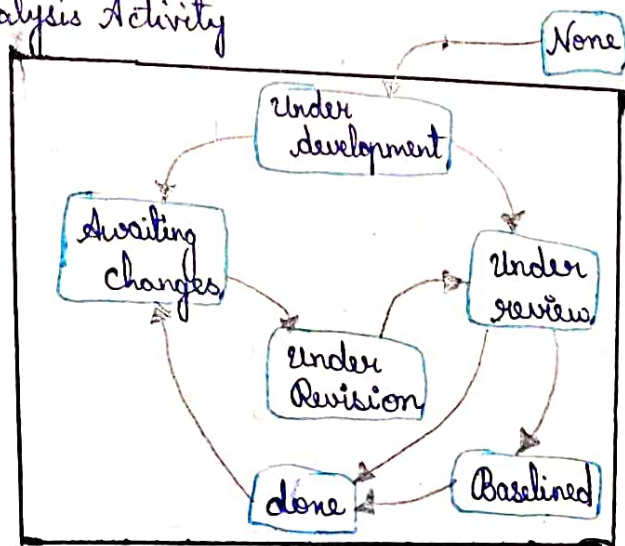
3. Develop the next Version of the Product.

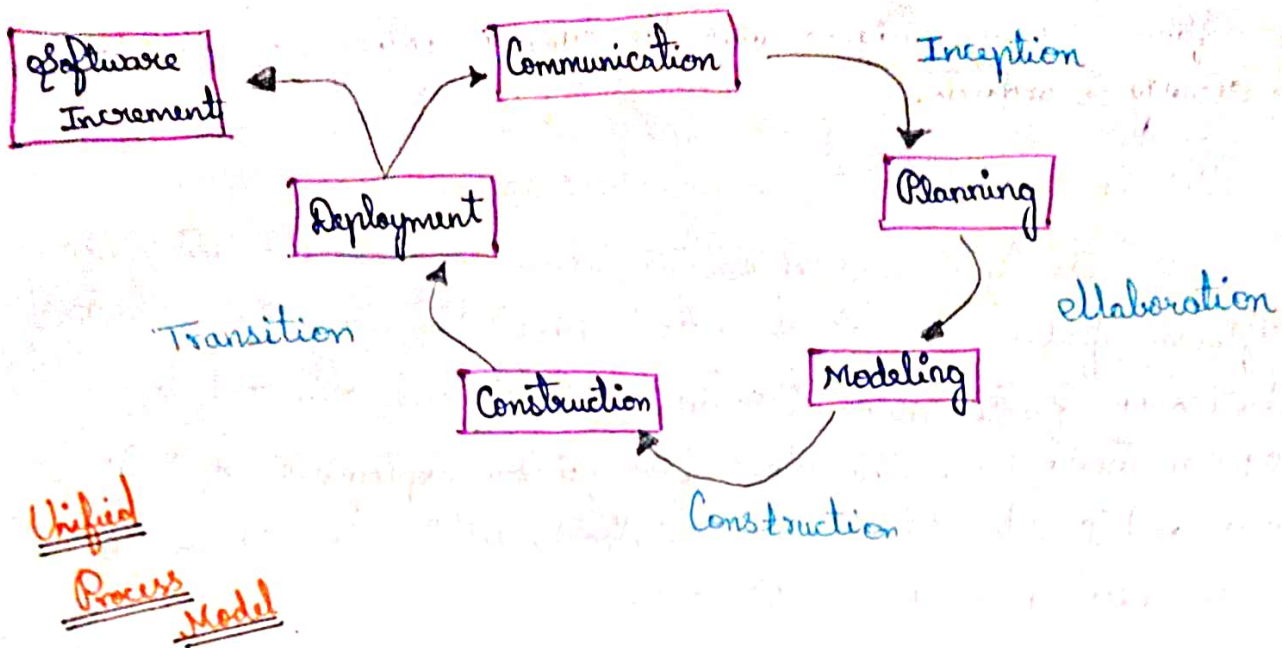


Spiral Model

Analysis Activity

Concurrent Model





Q. Assume you are a software designer involved in the development of library authentication software. What could be your choice of software process model for developing library authentication software. Justify your choice.

Answer:

The choice of a software process model depends on various factors, including the project's complexity, requirements clarity and development team's experience. For library authentication software which typically involves security considerations and may have evolving requirements, an iterative and incremental process model such as the agile model, would be a suitable choice. Justification for selecting the agile model.

1. Flexibility and Adaptability

Library authentication software often involves evolving requirements especially in the context of security updates or changing user needs. The agile model allows for frequent reassessment and adaptation to changing requirements, ensuring that

(6)

the Software remains aligned with the dynamic nature of authentication and security standards.

2. Incremental Development

The agile model encourages incremental development and delivery of software features. For library authentication, this means that essential authentication functionalities can be developed and delivered in smaller, manageable increments. This allows for quicker deployment of critical features and facilitates regular user feedback, enabling the software to meet the users needs more effectively.

3. Frequent user feedback

Security software like library authentication systems, requires a keen understanding of user needs and potential vulnerabilities. The agile model emphasizes customer collaboration and regular feedback, allowing for continuous improvement based on real-world usage & user input. This iterative approach helps in identifying and addressing security concerns promptly.

4. Risk Management

Security is a critical role of library authentication system software. Agile's iterative and incremental development approach enables the identification and mitigation of risks early in the development process. Regularly assessing and adapting to changes ensures that potential security vulnerabilities can be addressed proactively.

5. Transparent Communication

The agile model encourages open and transparent communication within the development team and with stakeholders. This is crucial for security-related projects where clear communication is essential to understand and implement security measures effectively.

6. Quick Time-To-Market

Libraries often need authentication systems that can be deployed rapidly to meet user needs. Agile's iterative development and incremental releases allow for quicker time-to-market compared to more traditional, sequential models, ensuring that the software can be delivered in a timely fashion.

The Agile model is well-suited for the development of library authentication software due to its flexibility, adaptability, emphasis on frequent user feedback, effective risk management, transparent communication & quicker time-to-market. These characteristics align well with the evolving & security-sensitive nature of authentication systems.

3. Describe the steps in the requirement gatherings.

Solution:

Step 1: Assign roles

The first step in requirements gathering is to assign roles in your project. This is when you identify your project stakeholders. A stakeholder is anyone invested in the project, whether they're internal or external partners. For example, a customer is an external stakeholder. Identifying these roles first will help you determine who should analyze your project scope later on.

Step 2: Meet with Stakeholders

Once you've identified your project stakeholders, meet with them to get an idea of what they're hoping to get out of project. Understanding what stakeholders want matters because they're ultimately the ones you're creating your deliverables for.

Some Questions you can ask include:

(8)

1. What is your goal for this project?
2. What do you think would make this project successful?
3. What are your concerns about this project?
4. What do you wish this product or service would do that it doesn't already?
5. What changes would you recommend about this project?

The stakeholders are the people you're ultimately developing the Project for, so you should ask them questions that can help you Create your list of requirements.

Step 3: Gather and Document

Step three in the process happens at the same time as Step 2. You'll gather information as you ask your stakeholders questions. The goal is to document everything you can, so have all of the answers you need to start your project. Use a project management tool to collect and document this information. You can keep your project plan, project requirements and project communication all in one place. The document might have

1. Stakeholder answer to interview questions.
2. Stakeholder questions
3. Stakeholder requests
4. Stakeholder comments
5. Questions & comments that arise during interviews.

Step 4: List Assumptions and Requirements

⑨

Create your requirements management plan based on the information you've gathered. Consider the questions you initially set out to answers during the requirements gathering process. Then, use them to Create your requirements goals including

1. Length of Project Schedule
2. People involved in the Project
3. Project risks.

Step 5: Get Approval

Once you formalize your project requirements, you'll need approval from stakeholders to ensure you're meeting user needs. Encouraging clear communication can also prevent scope creep by ensuring your stakeholders know the limits of the project from beginning.

Step 6: Monitor Progress

The last part of the process is monitoring the progress of your project. You can use project management software to track your project budget and other requirements as you move through project execution.
