EXPERIMENT NO. : 01

Date of Performance:

Date of Submission:

Aim: Application of at least two traditional process models.

Software Used: Ms Word

Theory: Software Processes is a coherent set of activities for specifying, designing, implementing and testing software systems. A software process model is an abstract representation of a process that presents a description of a process from some particular perspective. There are many different software processes but all involve:

- Specification – defining what the system should do;
- Design and implementation– defining the organization of the system and implementing the system;
- Validation – checking that it does what the customer wants;
- Evolution – changing the system in response to changing customer needs.

Types of Software Process Model

Software processes, methodologies and frameworks range from specific prescriptive steps that can be used directly by an organization in day-to-day work, to flexible frameworks that an organization uses to generate a custom set of steps tailored to the needs of a specific project or group. In some cases a “sponsor” or “maintenance” organization distributes an official set of documents that describe the process.

Software Process and Software Development Lifecycle Model

One of the basic notions of the software development process is SDLC models which stands for Software Development Life Cycle models. There are many development life cycle models that have been developed in order to achieve different required objectives. The models specify the various stages of the process and the order in which they are carried out. The most used, popular and important SDLC models are given below:

- Waterfall model
- V model
- Incremental model
- RAD model
- Agile model
- Iterative model
- Spiral model
- Prototype model

Student can draw two traditional process models and write its application in details (1.Which process model used in Mini project. 2. Anyone other than mini project).

1. Topic Name : Blog platform

   Model Used : Incremental Model

   Incremental Model:

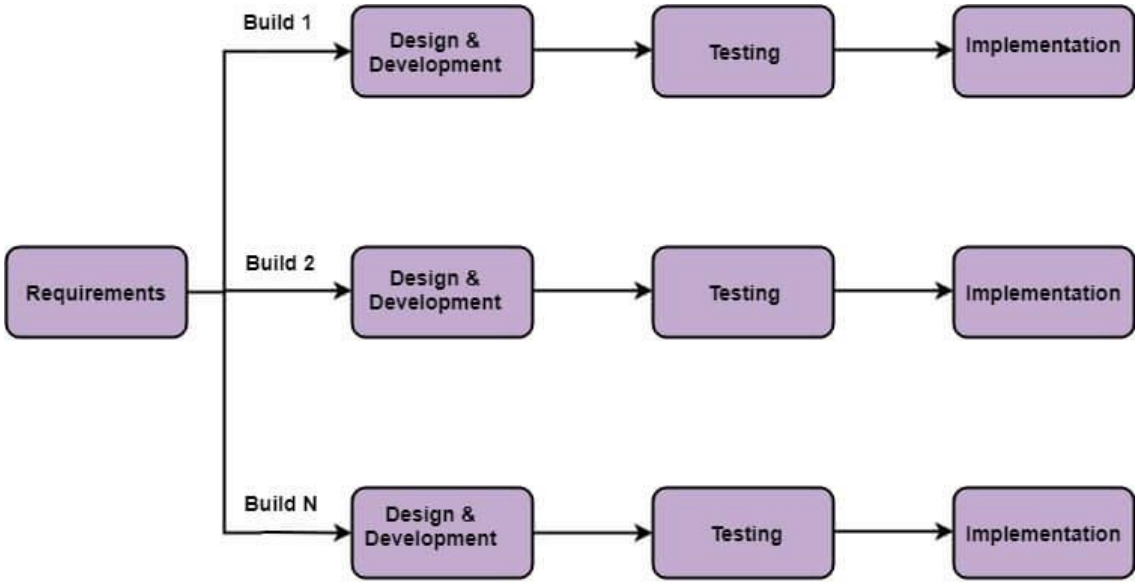


Fig: Incremental Model

The Incremental model involves breaking down the software development process into smaller, manageable components or increments. Each increment represents a portion of the complete system that is developed, tested, and delivered in stages.

Application of the Incremental Model on the Blog platform

Phase 1: Initial Planning and Requirements

1. **Define Scope and Objectives:**
   - **Develop a blog platform website where users can create, manage, and view blog posts.**
   - **Key features include creating and editing blog posts, categorizing content, searching for specific posts, and providing a user-friendly interface.**
2. **Requirements Gathering:**
   - **Identify core functionalities (e.g., creating posts, managing posts, categorizing posts).**
   - **Define user interface requirements and layout.**
   - **Determine user roles and permissions (e.g., authors, editors, readers).**
   - **Establish content management needs (e.g., media uploads, post scheduling).**

3. **Technology Stack:**
    - **Frontend: HTML, CSS, and JavaScript for the user interface, or frameworks like React or Angular for a more dynamic experience.**
    - **Backend: Java (or other technologies like Node.js, Django) for server-side logic.**
    - **Database: Use a database system such as MySQL or PostgreSQL for managing posts and user data.**

Phase 2: Incremental Development and Implementation

Increment 1: Basic Blog Post Creation and Display

- **Objective: Create a basic interface for creating and displaying blog posts.**
- **Tasks:**
    - **Set up the project structure with your chosen technology stack.**
    - **Develop backend APIs to handle CRUD (Create, Read, Update, Delete) operations for blog posts.**
    - **Create a basic frontend interface to display a list of blog posts.**
- **Testing: Ensure that users can create, view, and edit blog posts successfully.**

Increment 2: Post Categorization and Tagging

- **Objective: Implement functionality for categorizing and tagging blog posts.**
- **Tasks:**
    - **Add options for categorizing and tagging posts in the backend.**
    - **Update the frontend to allow users to filter posts by category or tag.**
- **Testing: Verify that posts are correctly categorized and tagged, and that filtering works as intended.**

Increment 3: Search Functionality

- **Objective: Add a search feature to find blog posts by keywords.**
- **Tasks:**
    - **Implement search functionality in the backend to filter posts based on user input.**
    - **Update the frontend to include a search bar and display search results.**
- **Testing: Ensure that the search functionality returns relevant posts based on keywords.**

Increment 4: Enhanced User Interface and Experience

- **Objective: Improve the user interface for a better user experience.**
- **Tasks:**
    - **Revamp the design of the blog platform to make it more intuitive and visually appealing.**

    - **Implement responsive design if applicable to ensure compatibility across devices (desktop, tablets, smartphones).**
- **Testing: Check that UI enhancements improve the user experience without affecting existing functionality.**

Increment 5: Advanced Features and Optimization

- **Objective: Add advanced features and optimize performance.**
- **Tasks:**
    - **Implement user authentication and different user roles (e.g., authors, editors).**
    - **Add features such as post scheduling, drafts, and media management.**
    - **Optimize database queries and improve site performance.**
    - **Implement error handling and logging mechanisms.**
- **Testing: Ensure all advanced features work as expected and that performance improvements are effective.**

Benefits of Using the Incremental Model for this Project

- **Flexibility: Allows for iterative development and adjustments based on user feedback and testing results.**
- **Risk Management: Reduces risks by delivering small, manageable modules incrementally.**
- **User Feedback: Provides opportunities for early user feedback, leading to a platform that better meets user needs.**
- **Resource Management: Efficient use of resources by focusing on one module at a time, allowing for parallel development and testing.**

Utilizing the incremental model for developing the blog platform website enables a well-organized and adaptable approach, fostering continuous improvements and ensuring responsiveness to user needs.

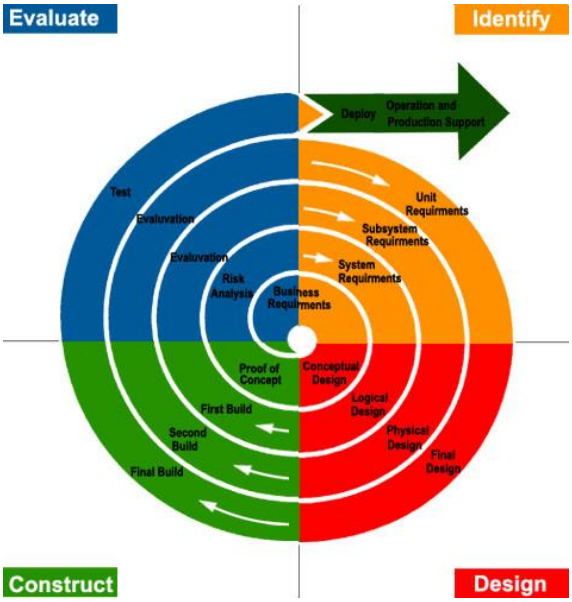
2.Spiral Model:

The spiral model is a risk-driven software development process model that combines iterative development with the systematic aspects of the waterfall model. Each cycle in the spiral involves planning, risk analysis, engineering, and evaluation, allowing for continuous improvement and risk management.

Cycle 1: Concept Development and Initial Prototype

- **Planning:**
    - Define high-level requirements and objectives for the initial prototype.
    - Establish a basic project structure.

- **Risk Analysis:**
    - Identify risks related to NLP accuracy and user interaction handling.
    - Plan mitigation strategies, such as using pre-trained models.



- **Risk Analysis:**
    - Identify risks related to NLP accuracy and user interaction handling.
    - Plan mitigation strategies, such as using pre-trained models.
- **Engineering:**
    - Develop a simple chatbot prototype that can greet users and respond to basic queries.
    - Implement a basic frontend interface.
- **Evaluation:**
    - Test the initial prototype with a small group of users.
    - Gather feedback on user interaction and response accuracy.

Cycle 2: Enhanced Functionality and Context Management

- **Planning:**
    - Define detailed requirements for enhanced functionalities, such as contextual conversation and FAQ handling.
- **Risk Analysis:**
    - Assess risks related to maintaining context and managing complex conversations.
    - Plan to use context management techniques and frameworks.

- **Engineering:**
    - Implement enhanced NLP capabilities for contextual understanding.
    - Develop modules for handling FAQs and common user tasks.
- **Evaluation:**
    - Conduct user testing to evaluate the improved functionalities.
    - Collect feedback on conversation flow and context management.

Cycle 3: Integration with External APIs and Services

- **Planning:**
    - Identify external APIs and services for integration (e.g., weather, news, task management).
    - Define integration requirements and objectives.
- **Risk Analysis:**
    - Analyze risks related to API reliability and data handling.
    - Develop fallback mechanisms for API failures.
- **Engineering:**
    - Integrate the chatbot with selected external APIs and services.
    - Implement API request and response handling.
- **Evaluation:**
    - Test the integrated functionalities with real users.
    - Evaluate the chatbot's ability to perform tasks using external services.

Benefits of Using the Spiral Model for this Project

- **Risk Management:** Continuous risk analysis and mitigation throughout the development process.
- **Flexibility:** Ability to incorporate feedback and make adjustments in each cycle.
- **User Involvement:** Regular evaluation and user testing ensure the product meets user needs.
- **Incremental Development:** Allows for incremental releases and continuous improvement.

By applying the spiral model to this chatbot app, we can effectively manage risks, incorporate user feedback, and ensure a high-quality product through iterative development and continuous refinement.

Conclusion: Hence, by using these traditional methods, we can ensure gradual development and delivery of the quick news app through the incremental model, effectively incorporating user feedback and managing risks for continuous improvement and adaptability. Simultaneously, the

spiral model facilitates iterative development of the chatbot app, balancing flexibility and structure, ensuring continuous risk management, and integrating user feedback for a robust, user-centered application.

Sign and Remark:

R1	R2	R3	Total Marks	Signature
(5)	(5)	(5)	(15)	