NAME:- ANAS KAPADIA ROLL NO- 22DCO01

**Experiment No. 02 - Traditional Process Model**

**Aim:** To apply traditional process model for the selected case study.

**Theory:**

A software process model is defined as simplified process representation of software process each methods represent a process from a specific perspective.

Here are some common traditional process models:

1. **Waterfall Model:** The Waterfall model is a linear, sequential approach where each phase of the software development lifecycle (requirements, design, implementation, testing, deployment, and maintenance) is completed before moving on to the next. It is well-suited for projects with well-defined requirements and stable environments.
2. **V-Model (Validation and Verification Model):** The V-Model is an extension of the Waterfall model. It adds a testing phase corresponding to each development phase. It emphasizes the importance of testing and validation at each step of development.
3. **Incremental Model:** The Incremental model divides the development process into smaller, manageable parts or increments. Each increment delivers a portion of the complete system. It is useful for projects where early delivery of certain functionalities is important.
4. **Spiral Model:** The Spiral model is iterative and involves cycles of planning, risk assessment, engineering, and evaluation. Each cycle results in a more refined version of the software. It is suitable for projects with evolving or uncertain requirements.

**Procedure:**

* 1. Explain the selected process model along with the diagram, pros, and cons.



* Incremental Model

Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation, and testing phases. Every subsequent release of the module adds function to the previous release. The process continues until the complete system achieved.

## Advantage of Incremental Model

* Errors are easy to be recognized.
* Easier to test and debug
* More flexible.
* Simple to manage risk because it handled during its iteration.
* The Client gets important functionality early.

## Disadvantage of Incremental Model

* Need for good planning
* Total Cost is high.
* Well defined module interfaces are needed.
  1. Mention the suitability reasons or justifications for selecting the process model.

The Incremental Process Model can be a suitable choice for developing an Agriculture website due to several reasons and justifications that align well with the nature of the agriculture industry and the specific needs of such a website:

**Output:**

Applying the Incremental Model to develop an Agriculture- software solution involves breaking down the development process into incremental stages, each delivering a set of features relevant to the agricultural domain. Here is how you can apply the Incremental Model to develop software solutions for Agriculture:

1. **Requirements Analysis:** Define the specific goals of your Agriculture software solution. Identify the key functionalities that address the needs of the agriculture industry, such as crop management, soil analysis, weather monitoring, inventory tracking, and more.
2. **Increment Identification:** Divide the requirements into logical increments. Begin by selecting the most critical features that form the foundation of your Agriculture solution. These features should provide value and solve key challenges faced by farmers.
3. **Development and Testing: Increment 1:** Develop the core features identified in the first increment. This could involve creating interfaces for crop information, basic data collection, and simple analytics. Thoroughly test these features to ensure their accuracy and functionality.
4. **User Feedback and Iteration: Increment 1:** Release the first increment to a subset of users, such as farmers or agricultural experts. Gather feedback on the usability, effectiveness, and usefulness of the implemented features. Use this feedback to refine and enhance the features in the first increment.
5. **Increment 2 Selection:** Based on user feedback and evolving agricultural needs, select the features for the second increment. These features should build upon the foundation established in the first increment and provide more advanced functionalities.
6. **Development and Testing: Increment 2:** Develop the features identified in the second increment. This could involve enhancing data analysis capabilities, expanding crop management tools, integrating with weather APIs, and incorporating more advanced reporting.
7. **User Feedback and Iteration: Increment 2:** Repeat the feedback and iteration process for the second increment. Release the new features to users, gather feedback, and make necessary improvements based on the feedback received.
8. **Repeat for Subsequent Increments:** Continue the iterative process by selecting, developing, testing, and iterating on subsequent increments. Each increment should introduce new features and enhancements while maintaining the overall coherence of the Agriculture software solution.
9. **Integration and System Testing:** Ensure that the features developed in each increment integrate seamlessly with the existing ones. Perform system-level testing to ensure all increments work harmoniously together as a comprehensive Agriculture software solution.
10. **Final Testing and Deployment:** After all increments have been developed and tested, perform a comprehensive testing of the entire software solution. Validate that all features function accurately, data is analyzed effectively, and the software provides value to users. Once satisfied, deploy the complete Agriculture solution.
11. **Continuous Improvement and Adaptation:** Maintain the software solution by monitoring its performance and addressing any issues. Continuously gather user feedback and data to drive ongoing improvements, updates, and enhancements that cater to the changing needs of the agriculture industry.

Top of Form

**Conclusion:**

By following the Incremental Model, we can ensure that your Agriculture software solution evolves iteratively, incorporating user feedback and adapting to the dynamic requirements of the agriculture sector while delivering meaningful value throughout the development process.