

IT-314

Software Engineering

LAB-1 Report

Choosing Software Process Models



July 31, 2023

Aayushkumar Patel(202101476)

Software Development Models for Various Situations

(a) A simple data processing project:

Model: Waterfall Model.

Reason: The Waterfall Model is suitable for straightforward projects with well-defined requirements, as it follows a linear and sequential approach, ensuring each phase is completed before moving to the next.

(b) A data entry system for office staff who have never used computers before. The user interface and user-friendliness are extremely important:

Model: Prototyping Model.

Reason: The Prototyping Model allows for iterative development and continuous feedback, making it ideal for creating user-friendly systems. By involving end-users in the prototyping process, the system's user interface can be refined until it meets their needs.

(c) A spreadsheet system that has some basic features and many other desirable features that use these basic features:

Model: Incremental Model (Evolutionary Model).

Reason: The Incremental Model enables the development of a system in smaller, manageable increments. It is suitable for projects where additional features are built on top of a stable core, making it a good fit for enhancing a spreadsheet system.

(d) A web-based system for a new business where requirements are changing fast and where an in-house development team is available for all aspects of the project:

Model: Time Boxing Model.

Reason: The Time Boxing Model sets fixed time frames for each development phase, allowing for rapid iterations. It suits projects with rapidly changing requirements and a dedicated in-house team, ensuring a continuous flow of deliverables.

(e) A Website for an online store which has a long list of desired features it wants to add, and it wants a new release with new features to be done very frequently:

Model: Incremental Waterfall Model.

Reason: The Incremental Waterfall Model combines the iterative nature of incremental development with the structured approach of the Waterfall Model. It is suitable for projects requiring frequent releases of new features while maintaining stability.

(f) A system to control anti-lock braking in a car:

Model: Synchronize & Stabilize Model.

Reason: The Synchronize & Stabilize Model is often used in safety-critical systems like anti-lock braking. It emphasizes rigorous synchronization of development activities to ensure proper integration and stabilization before deployment.

(g) A virtual reality system to support software maintenance:

Model: Throw Away Prototyping Model.

Reason: The Throw Away Prototyping Model allows for the rapid creation of prototypes to explore new ideas and validate concepts. It fits well with projects like virtual reality systems, where experimentation and proof of concept are essential.

(h) A university accounting system that replaces an existing system:

Model: Evolutionary Prototyping Model.

Reason: The Evolutionary Prototyping Model focuses on continuous improvement and refinement. It is appropriate for replacing existing systems, as it allows for gradual enhancements and aligns with evolving user needs.

(i) An interactive system that allows railway passengers to find train times from terminals installed in stations:

Model: Spiral Model.

Reason: The Spiral Model is ideal for projects involving high risk and uncertainty, such as interactive systems with varying user needs. It facilitates frequent prototyping, testing, and adjustments to meet passenger requirements effectively.

(j) Company has asked you to develop software for a missile guidance system that can identify a target accurately:

Model: Water-Flow Model.

Reason: The Water-Flow Model emphasizes a systematic and thorough approach, which is crucial for developing highly reliable and accurate systems like missile guidance.

(k) When emergency changes have to be made to systems, the system software may have to be modified before changes to the requirements have been approved. Choose a process model for making these modifications that ensure that the requirements documents and the system implementation do not become inconsistent:

Model: Synchronize & Stabilize Model.

Reason: The Synchronize & Stabilize Model provides a disciplined approach to handle emergency changes while ensuring consistency between requirements and system implementation.

(l) Software for an ECG machine:

Model: Time Boxing Model.

Reason: The Time Boxing Model, with its fixed time frames for development phases, is suitable for developing software for medical devices like an ECG machine, ensuring timely and efficient delivery with a focus on safety and quality.

(m) A small-scale, well-understood project (no changes in requirement will be there once decided):

Model: Waterfall Model.

Reason: For small, straightforward projects with stable and well-understood requirements, the Waterfall Model provides a predictable and systematic approach to development, meeting the project's needs without the need for extensive iterations.