**Bài 1:Điểm khác biệt chính giữa Quy trình phát triển phần mềm và Mô hình quy trình phát triển phần mềm:**

1. Quy trình phát triển phần mềm và Mô hình quy trình phát triển phần mềm là hai khái niệm có liên quan nhưng không hoàn toàn tương đương. Dưới đây là điểm khác biệt chính giữa chúng:

-**Quy trình phát triển phần mềm (Software Development Process):**

+, Là một phạm vi rộng hơn và tổng quát hơn, nó bao gồm tất cả các hoạt động, quy trình, phương pháp và công cụ được sử dụng trong việc phát triển phần mềm.

+, Thường là một khái niệm trừu tượng hướng dẫn cách tiến hành công việc từ khi bắt đầu dự án đến khi hoàn thành sản phẩm.

+, Có thể được tùy chỉnh và áp dụng theo yêu cầu cụ thể của dự án hoặc tổ chức.

-**Mô hình quy trình phát triển phần mềm (Software Development Process Model):**

+, Là một loại hình cụ thể hoặc một bản thiết kế cụ thể cho quy trình phát triển phần mềm.

+, Đại diện cho một chuỗi cụ thể các hoạt động, giai đoạn hoặc bước mà các nhà phát triển phần mềm phải thực hiện để tạo ra sản phẩm phần mềm.

+, Có các mô hình phổ biến như Waterfall, Agile, Spiral, và Incremental.

Tóm lại, một quy trình phát triển phần mềm là một khái niệm tổng quát về cách thức tiến hành phát triển phần mềm, trong khi một mô hình quy trình phát triển phần mềm là một phiên bản cụ thể hơn của quy trình đó, mô tả các bước cụ thể hoặc các giai đoạn mà dự án phải tuân thủ.

Bài 2:

**- Giới thiệu Mô hình Waterfall (Thác nước):**

Mô hình Waterfall (Thác nước) là Mô hình Quy trình đầu tiên được giới thiệu. Nó cũng được gọi là mô hình vòng đời tuần tự tuyến tính. Nó rất đơn giản để hiểu và sử dụng. Trong mô hình thác nước, mỗi giai đoạn phải được hoàn thành trước khi giai đoạn tiếp theo có thể bắt đầu và không có sự chồng chéo trong các giai đoạn.

Waterfall là mô hình quy trình đầu tiên và nổi tiếng nhất được giới thiệu, với cách tiếp cận Vòng đời phát triển hệ thống (SDLC) lâu đời nhất, được sử dụng để phát triển phần mềm.

Mô hình thác nước minh họa quá trình phát triển phần mềm theo dòng tuần tự tuyến tính. Điều này có nghĩa là bất kỳ giai đoạn nào trong quá trình phát triển chỉ bắt đầu nếu giai đoạn trước đó hoàn thành. Trong mô hình thác nước, các pha không chồng lên nhau.

**-các pha của mô hình waterfall:**

**Tuần tự và đơn điệu:** Mỗi giai đoạn trong mô hình Waterfall diễn ra theo thứ tự tuyến tính, không có sự chồng chéo giữa các giai đoạn. Điều này có nghĩa là một giai đoạn không thể bắt đầu cho đến khi giai đoạn trước đó hoàn thành.

**Giai đoạn rõ ràng và cố định:** Mô hình Waterfall đặt ra các giai đoạn cụ thể như thu thập yêu cầu, thiết kế, triển khai, kiểm thử và bảo trì. Mỗi giai đoạn có một tập hợp cụ thể các hoạt động và mục tiêu riêng.

**Yêu cầu được xác định trước:** Mô hình Waterfall yêu cầu rằng tất cả yêu cầu của dự án phải được xác định và chốt trước khi bắt đầu giai đoạn thiết kế. Điều này có thể là một điểm yếu khi các yêu cầu thay đổi trong quá trình phát triển.

**Khó điều chỉnh khi cần thiết:** Vì tính tuần tự và đơn điệu của nó, mô hình Waterfall không linh hoạt khi cần điều chỉnh hay thay đổi yêu cầu trong quá trình phát triển. Điều này có thể dẫn đến việc tăng chi phí và thời gian.

**Phù hợp cho các dự án có yêu cầu ổn định:** Mô hình Waterfall thích hợp cho các dự án có yêu cầu ổn định và không thay đổi nhiều trong quá trình phát triển. Đối với các dự án có yêu cầu thay đổi thường xuyên, mô hình Agile thường được ưa chuộng hơn.

**-ưu điểm và nhược điểm:**

**+ ưu điểm:**

✓ Đơn giản, dễ hiểu và sử dụng

✓ Dễ dàng quản lý do độ cứng của mô hình.

✓ Mỗi giai đoạn có các phân phối cụ thể và một quy trình xem xét.

✓ Các giai đoạn được xử lý và hoàn thành cùng một lúc.

✓ Hoạt động tốt cho các dự án nhỏ hơn, nơi các yêu cầu được hiểu rất rõ.

✓ Các giai đoạn được xác định rõ ràng.

✓ Các mốc quan trọng được hiểu rõ.

✓ Dễ dàng sắp xếp các công việc.

✓ Quá trình và kết quả được ghi chép đầy đủ.

**+ nhược điểm:**

✓ Không có phần mềm đang hoạt động nào được sản xuất cho đến cuối vòng đời.

✓ Lượng rủi ro cao và không chắc chắn.

✓ Không phải là một mô hình tốt cho các dự án phức tạp và hướng đối tượng. ✓ Mô hình kém cho các dự án dài và đang diễn ra. Không phù hợp với các dự án mà các yêu cầu có nguy cơ thay đổi từ trung bình đến cao.

✓ Vì vậy, rủi ro và sự không chắc chắn là cao với mô hình quy trình này.

✓ Rất khó để đo lường sự tiến bộ trong các giai đoạn.

✓ Không thể đáp ứng các yêu cầu thay đổi.

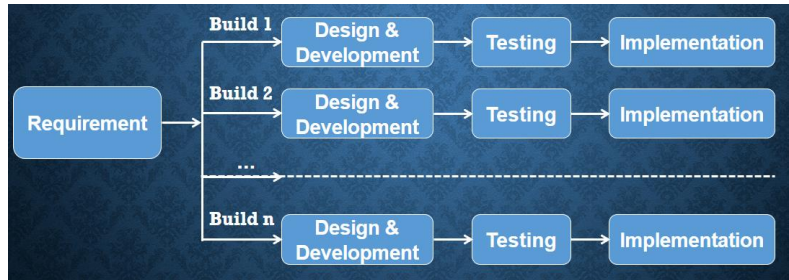
Bài 3:

**Giới thiệu mô hình Iterative :**

Mô hình tiếp cận lặp (Iterative hay Iterative and Incremental) nhấn mạnh vào việc phát triển phần mềm không phải là một quá trình tuyến tính mà là một chuỗi các vòng lặp, từng bước cải tiến, ngày càng hoàn thiện. Mô hình này bắt đầu với việc triển khai một phần nhỏ của yêu cầu phần mềm, sau đó liên tục cải tiến qua mỗi vòng lặp. Quan trọng là mỗi vòng lặp tạo ra một phiên bản mới của sản phẩm phần mềm, thường đi kèm với sự điều chỉnh và mở rộng từ phiên bản trước đó.

Điểm đặc biệt của mô hình này là không cố gắng bắt đầu với tất cả các yêu cầu kỹ thuật được xác định từ đầu. Thay vào đó, nó bắt đầu với một phần nhỏ của phần mềm và sau đó đánh giá để xác định các yêu cầu tiếp theo. Quá trình này tiếp tục được lặp lại, tạo ra các phiên bản mới của phần mềm qua mỗi vòng lặp.

**Các pha trong mô hình:**



o Requirement: Phân tích và định nghĩa yêu cầu phần mềm

o Build 1: Triển khai đơn giản với một tập nhỏ các yêu cầu.

o Build i: Cải tiến phiên bản trước đó bằng cách bổ sung thêm các yêu cầu để tạo ra một phiên bản mới.

o Build n: Giai đoạn cuối xây dựng được một phiên bản phần mềm hoàn thiện với đầy đủ các yêu cầu phần mềm

❖ **Ưu điểm**:

➢ Một số chức năng phần mềm có thể được phát triển nhanh chóng và sớm được đưa vào sử dụng trong vòng đời;

➢ Kết quả thu được sớm và theo định kỳ;

➢ Có thể lập kế hoạch phát triển song song;

➢ Ít tốn kém hơn để thay đổi phạm vi/yêu cầu, hỗ trợ thay đổi yêu cầu;

➢ Dễ dàng kiểm tra và gỡ lỗi trong quá trình lặp lại nhỏ hơn;

➢ Một số chức năng phần mềm có thể được phát triển nhanh chóng và sớm được đưa vào sử dụng trong vòng đời;

➢ Kết quả thu được sớm và theo định kỳ;

➢ Có thể lập kế hoạch phát triển song song;

➢ Ít tốn kém hơn để thay đổi phạm vi/yêu cầu, hỗ trợ thay đổi yêu cầu;

➢ Dễ dàng kiểm tra và gỡ lỗi trong quá trình lặp lại nhỏ hơn;

❖ **Nhược điểm**:

➢ Cần nhiều tài nguyên hơn;

➢ Các vấn đề về kiến trúc hoặc thiết kế hệ thống có thể phát sinh;

➢ Không thích hợp cho các dự án nhỏ;

➢ Sự phức tạp trong quản lý là nhiều hơn;

➢ Cần có nguồn lực có kỹ năng cao để phân tích rủi ro;

➢ Tiến độ dự án phụ thuộc nhiều vào giai đoạn phân tích rủi ro