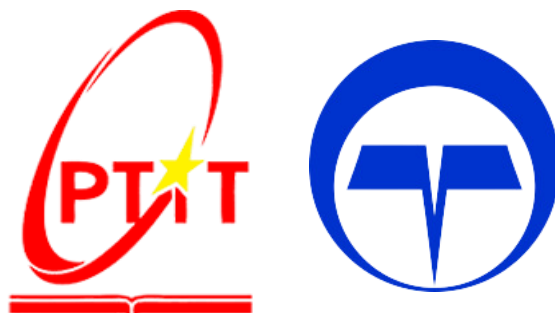


HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
VIỆN KHOA HỌC KỸ THUẬT BƯU ĐIỆN



BÁO CÁO BÀI TẬP CÁ NHÂN
MÔN HỌC: QUẢN LÝ DỰ ÁN PHẦN MỀM
CHỦ ĐỀ: TÌM HIỂU CÁC MÔ HÌNH PHÁT TRIỂN
PHẦN MỀM

Sinh viên thực hiện : Nguyễn Khả Phong

Mã sinh viên : B23DCCC129

Lớp học phần : INT1450

Giảng viên hướng dẫn : Nguyễn Việt Dũng

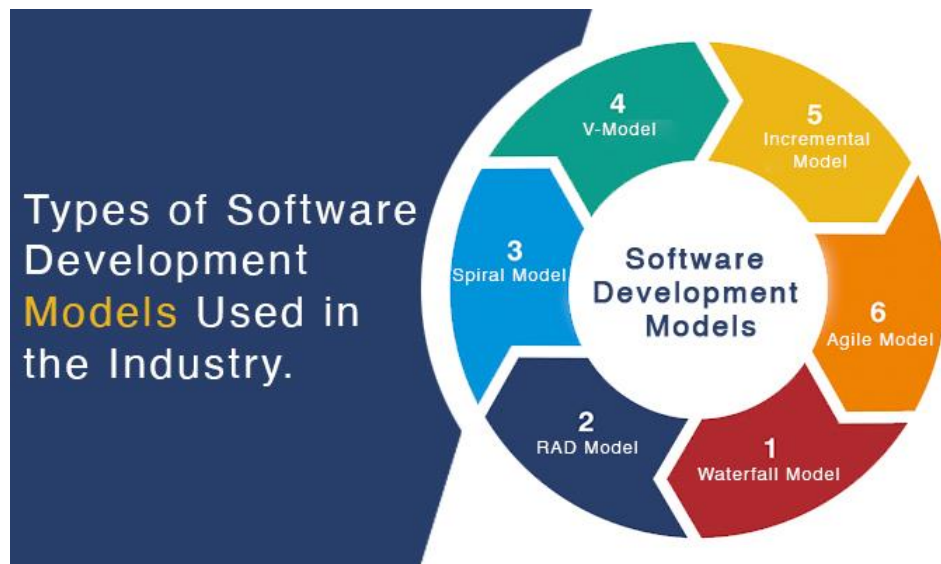
Hà Nội – 2/2025

1. Mô hình phát triển phần mềm

Mô hình phát triển phần mềm là một quy trình hoặc khuôn khổ hướng dẫn cách xây dựng và triển khai phần mềm một cách có tổ chức. Nó bao gồm các bước, pha, hoạt động và vai trò của các thành viên trong nhóm phát triển để đảm bảo việc phát triển phần mềm thành công và đáp ứng được yêu cầu của khách hàng.

Mô hình phát triển phần mềm giúp quản lý dự án, phân chia công việc, xác định các bước tiến trình và đảm bảo rằng sản phẩm phần mềm cuối cùng có chất lượng tốt và đáp ứng yêu cầu của khách hàng. Có nhiều mô hình phát triển phần mềm phổ biến được sử dụng trong ngành công nghiệp phần mềm, bao gồm:

- Mô hình thác nước (Waterfall model)
- Mô hình xoắn ốc (Spiral model)
- Mô hình agile
- Mô hình tiếp cận lặp (Iterative model)
- Mô hình tăng trưởng (Incremental model)
- Mô hình chữ V (V model)
- Mô hình Scrum
- RAD model (Rapid Application Development)



Những lợi ích mô hình phát triển phần mềm mang lại:

- Tổ chức và quản lý

Mô hình phát triển phần mềm cung cấp một khuôn khổ cụ thể và có cấu trúc để tổ chức và quản lý quá trình phát triển phần mềm. Nó định rõ các bước, pha và hoạt động cần thiết để hoàn thành dự án phần mềm. Nhờ đó, các thành viên trong nhóm phát triển có thể hiểu rõ công việc của mình, phân công công việc một cách rõ ràng và phối hợp hiệu quả với nhau.

- Đáp ứng yêu cầu

Đảm bảo rằng quá trình xác định yêu cầu được thực hiện cẩn thận từ đầu. Điều này giúp sản phẩm cuối cùng sẽ đáp ứng được các yêu cầu và nhu cầu của khách hàng. Việc xác định yêu cầu đầy đủ và rõ ràng từ đầu cũng tránh được các thay đổi yêu cầu không cần thiết và tiết kiệm thời gian, công sức, tài nguyên.

- Kiểm soát rủi ro

Mô hình phát triển phần mềm thường bao gồm các bước kiểm thử và đánh giá tiến độ, giúp nhận biết và quản lý rủi ro trong quá trình phát triển. Điều này giúp tránh các vấn đề và sai sót không mong muốn, và giúp đảm bảo dự án được thực hiện một cách ổn định và hiệu quả.

- Tiết kiệm thời gian và tài nguyên

Mô hình phát triển phần mềm giúp tối ưu hóa việc sử dụng thời gian và tài nguyên. Việc sắp xếp công việc theo các giai đoạn và pha cho phép nhóm phát triển tiến hành công việc một cách có hệ thống và tuần tự. Điều này giúp tránh sự rối loạn và lãng phí, từ đó giảm thiểu chi phí không cần thiết và tăng hiệu quả trong việc sử dụng tài nguyên.

- Đảm bảo chất lượng

Các mô hình phát triển phần mềm thường bao gồm các bước kiểm thử và đảm bảo chất lượng. Việc thực hiện kiểm thử thường xuyên giúp phát hiện và sửa chữa lỗi ngay từ khi chúng xuất hiện, giúp đảm bảo rằng sản phẩm cuối cùng có chất lượng tốt và đáp ứng yêu cầu của khách hàng.

- Dễ dàng duy trì và mở rộng

Mô hình phát triển phần mềm thường cung cấp sự cân nhắc và chi tiết trong việc xây dựng kiến trúc và thiết kế phần mềm. Điều này giúp dễ dàng duy trì, sửa đổi và mở rộng phần mềm sau khi hoàn thành. Việc có kiến trúc và thiết kế rõ

ràng từ đầu giúp giảm thiểu sự rối loạn và rủi ro khi thực hiện các thay đổi và mở rộng phần mềm sau này.

2. Phân tích 2 mô hình phát triển phần mềm phổ biến và so sánh

2.1. Mô hình thác nước (Waterfall model)



2.1.1. Mô tả

- Đây được coi như là mô hình phát triển phần mềm đầu tiên được sử dụng.
- Mô hình này áp dụng tuần tự các giai đoạn của phát triển phần mềm.
- Đầu ra của giai đoạn trước là đầu vào của giai đoạn sau. Giai đoạn sau chỉ được thực hiện khi giai đoạn trước đã kết thúc. Đặc biệt không được quay lại giai đoạn trước để xử lý các yêu cầu khi muốn thay đổi.

2.1.2. Phân tích mô hình

- Requirement gathering: Thu thập và phân tích yêu cầu được ghi lại vào tài liệu đặc tả yêu cầu trong giai đoạn này.
- System Analysis: Phân tích thiết kế hệ thống phần mềm, xác định kiến trúc hệ thống tổng thể của phần mềm.
- Coding: Hệ thống được phát triển theo từng unit và được tích hợp trong giai đoạn tiếp theo. Mỗi Unit được phát triển và kiểm thử bởi dev được gọi là Unit Test.

- Testing: Cài đặt và kiểm thử phần mềm. Công việc chính của giai đoạn này là kiểm tra và sửa tất cả những lỗi tìm được sao cho phần mềm hoạt động chính xác và đúng theo tài liệu đặc tả yêu cầu.
- Implementation: Triển khai hệ thống trong môi trường khách hàng và đưa ra thị trường.
- Operations and Maintenance: Bảo trì hệ thống khi có bất kỳ thay đổi nào từ phía khách hàng, người sử dụng.

2.1.3. Ứng dụng

Mô hình thường được áp dụng cho các dự án phần mềm như sau:

- Các dự án nhỏ , ngắn hạn.
- Các dự án có ít thay đổi về yêu cầu và không có những yêu cầu không rõ ràng.

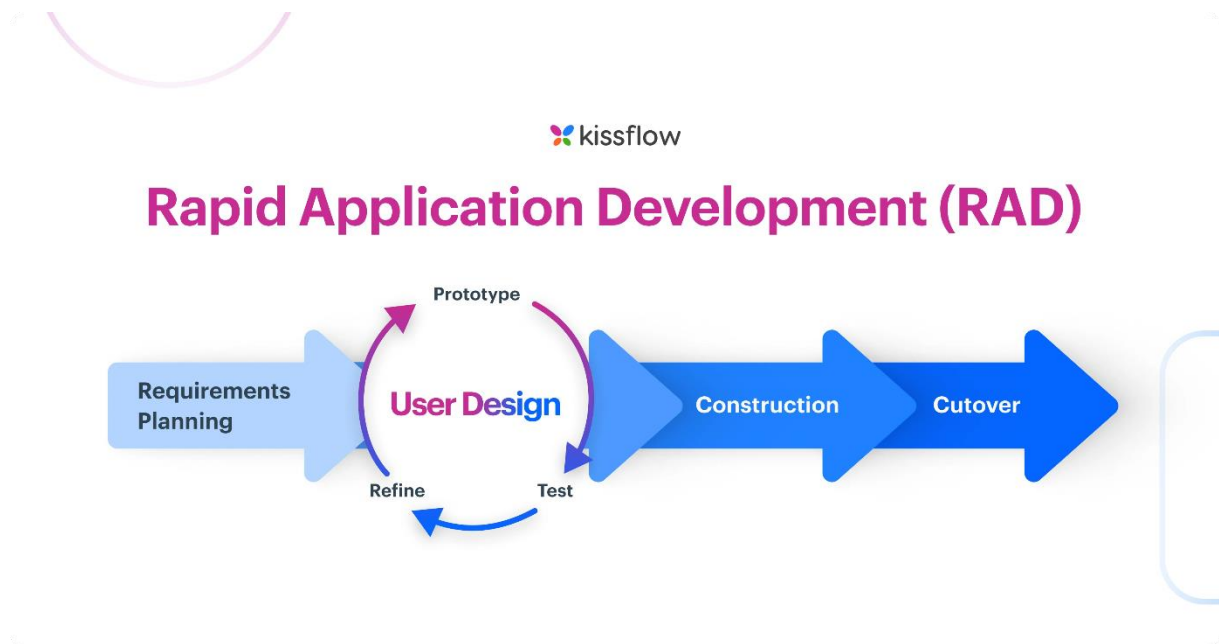
2.1.4. Ưu điểm

- Dễ sử dụng, dễ tiếp cận, dễ quản lý.
- Sản phẩm phát triển theo các giai đoạn được xác định rõ ràng.
- Xác nhận ở từng giai đoạn, đảm bảo phát hiện sớm các lỗi.

2.1.5. Nhược điểm

- Ít linh hoạt, phạm vi điều chỉnh hạn chế.
- Rất khó để đo lường sự phát triển trong từng giai đoạn.
- Mô hình không thích hợp với những dự án dài, đang diễn ra, hay những dự án phức tạp, có nhiều thay đổi về yêu cầu trong vòng đời phát triển.
- Khó quay lại khi giai đoạn nào đó đã kết thúc.

2.2. Mô hình RAD (Rapid Application Development)



2.2.1. Mô tả

- Mô hình RAD là một phương pháp phát triển phần mềm sử dụng quy hoạch tối thiểu có lợi cho việc tạo mẫu nhanh.
- Các mô-đun chức năng được phát triển song song như nguyên mẫu và được tích hợp để tạo ra sản phẩm hoàn chỉnh để phân phối sản phẩm nhanh hơn.
- Đảm bảo rằng các nguyên mẫu được phát triển có thể tái sử dụng được.

2.2.2. Ứng dụng

Mô hình RAD có thể được áp dụng thành công cho các dự án:

- Module hóa rõ ràng. Nếu dự án không thể được chia thành các mô-đun, RAD có thể không thành công.
- RAD nên được sử dụng khi có nhu cầu để tạo ra một hệ thống có yêu cầu khách hàng thay đổi trong khoảng thời gian nhỏ 2-3 tháng.
- Nên được sử dụng khi đã có sẵn designer cho model và chi phí cao.

2.2.3. Ưu điểm

- Giảm thời gian phát triển.
- Tăng khả năng tái sử dụng của các thành phần.
- Đưa ra đánh giá ban đầu nhanh chóng.
- Khuyến khích khách hàng đưa ra phản hồi.

2.2.4. Nhược điểm

- Trình độ của nhóm cần có một kỹ năng nhất định.
- Chỉ những hệ thống có module mới sử dụng được mô hình này.

2.3. So sánh

Tiêu chí	Waterfall	RAD
Quy trình phát triển	Tuần tự, từng bước một và không thể quay lại giai đoạn trước. Các giai đoạn được thực hiện theo thứ tự cố định.	Phát triển song song các module, linh hoạt hơn với việc tạo mẫu nhanh và tích hợp các thành phần.
Thời gian phát triển	Thường mất nhiều thời gian hơn do phải hoàn thành từng giai đoạn mới được chuyển sang giai đoạn tiếp theo.	Giảm thời gian phát triển nhờ việc phát triển song song.
Phạm vi dự án	Phù hợp với dự án nhỏ, ngắn hạn, yêu cầu ít thay đổi và rõ ràng.	Phù hợp với dự án có thể chia thành các module rõ ràng, thời gian ngắn (2-3 tháng), và có thể có yêu cầu thay đổi.
Sự tham gia của khách hàng	Khách hàng chủ yếu tham gia ở giai đoạn đầu (thu thập yêu cầu) và cuối (kiểm thử, triển khai).	Khuyến khích sự tham gia và phản hồi thường xuyên từ khách hàng trong suốt quá trình phát triển.
Tính linh hoạt	Ít linh hoạt, khó điều chỉnh khi đã sang giai đoạn sau.	Linh hoạt hơn, dễ dàng thích ứng với thay đổi yêu cầu.
Nhân sự	Không đòi hỏi cao về kỹ năng đặc biệt của team.	Yêu cầu team có trình độ và kỹ năng nhất định, cần có sẵn designer.
Khả năng tái sử dụng	Không đề cập đến việc tái sử dụng các thành phần.	Chú trọng đến việc tái sử dụng các thành phần và module.
Quản lý dự án	Dễ quản lý do có các giai đoạn rõ ràng, nhưng	Phức tạp hơn trong quản lý do phát triển song

khó đo lường tiến độ trong từng giai đoạn.	song, nhưng có thể đánh giá kết quả nhanh chóng.
---	--