

Quy Trình Phát Triển Phần Mềm

Software Development Life Cycle – SDLC

Người trình bày:



Đoàn Thị Kim Nhung



BA Team



BATIZENS

Mục Lục

01

Vòng đời/Quy trình Phát triển phần mềm
Software Development Life Cycle

02

Các mô hình Phát triển phần mềm
Software Development Models



BATIZENS



Từ ngữ viết tắt và định nghĩa

#	Từ ngữ viết tắt	Định nghĩa
1	PTPM	Phát triển phần mềm
2		
3		
4		
5		
6		
7		
8		





01



Vòng đời/Quy trình Phát triển phần mềm



Vòng đời/Quy trình PTPM

1. Planning and Requirement Analysis

Giai đoạn lập kế hoạch và phân tích yêu cầu

2. Defining Requirements

Giai đoạn xác định yêu cầu

3. Designing Architecture

Thiết kế kiến trúc

4. Developing Product

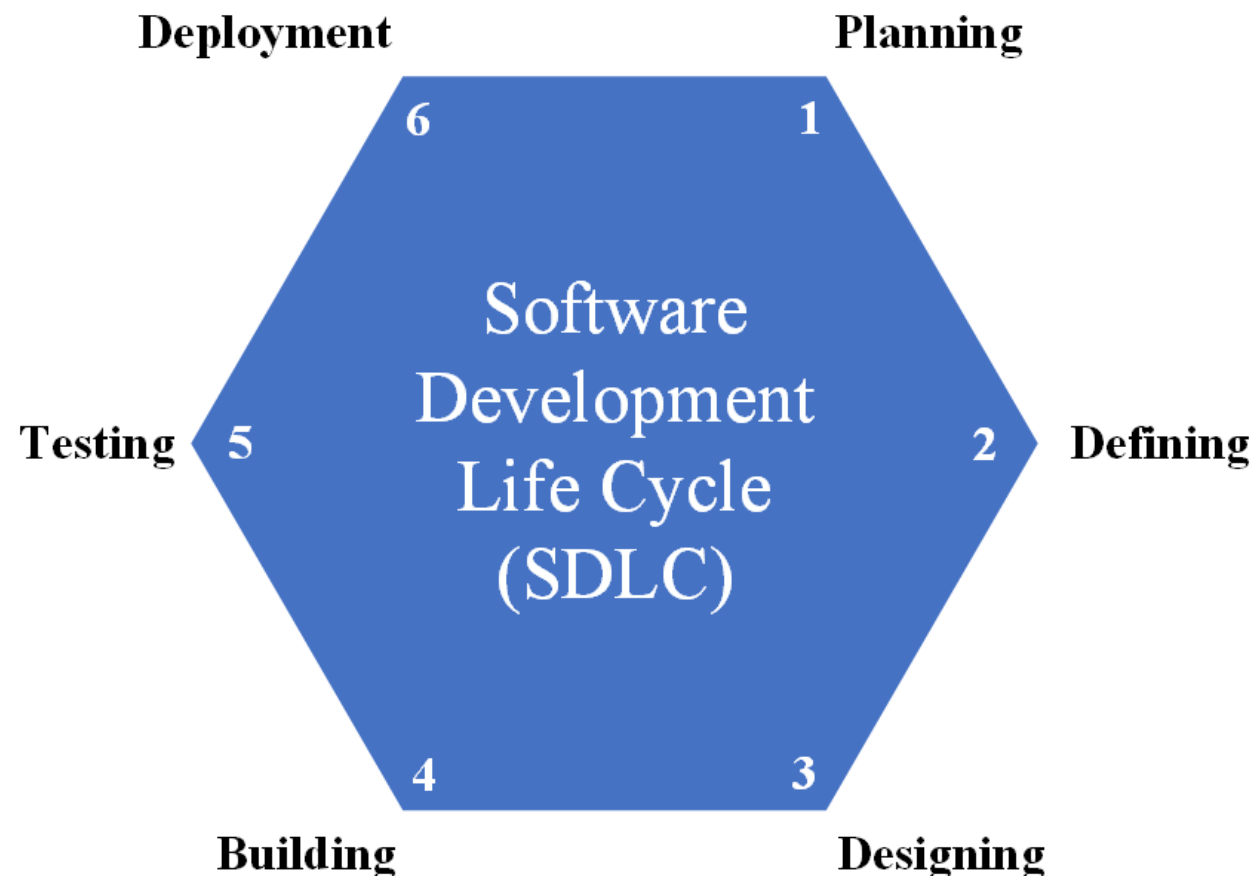
Phát triển sản phẩm

5. Product Testing and Integration

Kiểm thử và tích hợp sản phẩm

6. Deployment and Maintenance of Products

Triển khai và bảo trì sản phẩm





Vòng đời/Quy trình PTPM

1. Planning and Requirement Analysis: Giai đoạn lập kế hoạch và phân tích yêu cầu

Trong quá trình phát triển phần mềm, lập kế hoạch và phân tích yêu cầu là hai bước quan trọng. Lập kế hoạch dựa trên phản hồi từ khách hàng và khảo sát thị trường. Phân tích yêu cầu tạo nền tảng cho dự án. Chất lượng của dự án phụ thuộc vào việc lập kế hoạch và phân tích yêu cầu.

#	Đầu vào	Đầu ra
1	Phản hồi từ khách hàng: Thông tin được thu thập từ khách hàng về các yêu cầu, mong muốn và mục tiêu của họ đối với sản phẩm phần mềm.	Tài liệu yêu cầu: Bao gồm tất cả các yêu cầu của khách hàng và thị trường được thu thập và xác định rõ ràng.
2	Khảo sát thị trường: Các cuộc khảo sát thị trường được thực hiện để đánh giá nhu cầu của thị trường, xu hướng, cạnh tranh và các yếu tố khác có liên quan.	Bản kế hoạch dự án: Chứa thông tin chi tiết về lịch trình, ngân sách, tài nguyên và các nguyên tắc và quy trình quản lý dự án.





Vòng đời/Quy trình PTPM

1. Planning and Requirement Analysis: Giai đoạn lập kế hoạch và phân tích yêu cầu

#	Đầu vào	Đầu ra
3	Tài liệu yêu cầu: Các tài liệu được tạo ra trong các giai đoạn trước đó hoặc thông tin chi tiết về yêu cầu chức năng, phi chức năng, và yêu cầu khác từ khách hàng hoặc bên liên quan.	Sơ đồ dự án: Có thể bao gồm các biểu đồ và sơ đồ để minh họa quá trình phân tích yêu cầu và lập kế hoạch.
4	Kế hoạch dự án: Bao gồm lịch trình, ngân sách, phân công tài nguyên, và các thông tin quản lý dự án khác.	Báo cáo phân tích yêu cầu: Bao gồm phân tích chi tiết về yêu cầu, phân loại chúng và ưu tiên hóa để sử dụng trong quá trình phát triển tiếp theo.
5	Thông tin từ các bộ phận khác: Bao gồm thông tin từ bộ phận kinh doanh, marketing, kỹ thuật và bộ phận liên quan khác.	





Vòng đời/Quy trình PTPM

1. Planning and Requirement Analysis: Giai đoạn lập kế hoạch và phân tích yêu cầu

#	Các hoạt động
1	Lập kế hoạch (Planning): Xác định các bước cần thiết để tiến hành dự án phát triển phần mềm, bao gồm lập lịch, phân công nhiệm vụ, xác định nguồn lực cần thiết và thiết lập các tiêu chuẩn quản lý dự án.
2	Xác định Phạm vi dự án (Define Project Scope): Xác định phạm vi của dự án, bao gồm các tính năng và chức năng dự kiến của sản phẩm phần mềm, cũng như những gì không được bao gồm trong dự án.
3	Đặt mục tiêu và mục đích (Set Objectives and Goals): Xác định mục tiêu cụ thể mà dự án cần đạt được, cũng như đặt ra các mục tiêu và mục đích cho các giai đoạn của dự án.
4	Lập kế hoạch nguồn lực (Resource Planning): Xác định và phân bổ nguồn lực cần thiết cho dự án, bao gồm con người, thiết bị và tài nguyên khác, để đảm bảo dự án được thực hiện theo kế hoạch và đạt được mục tiêu.





Vòng đời/Quy trình PTPM

2. Defining Requirements: Giai đoạn xác định yêu cầu

Trong giai đoạn này, yêu cầu phần mềm được xác định thông qua BRD/User Story/URD/SRS/FSD sau khi được phê duyệt từ khách hàng và các bên liên quan.

#	Đầu vào	Đầu ra
1	Thông tin từ giai đoạn trước: Các thông tin thu thập được từ giai đoạn lập kế hoạch và phân tích yêu cầu được sử dụng làm đầu vào cho giai đoạn xác định yêu cầu.	Các tài liệu Yêu cầu Phần mềm: Một tài liệu chi tiết mô tả các yêu cầu chức năng và phi chức năng của phần mềm mục tiêu.
2	Phản hồi từ khách hàng và các bên liên quan: Phản hồi từ khách hàng và các bên liên quan khác về các yêu cầu, mong đợi và mục tiêu của họ đối với sản phẩm phần mềm.	Phê duyệt từ khách hàng và các bên liên quan: Tài liệu cần được phê duyệt bởi khách hàng và các bên liên quan để xác nhận rằng tất cả các yêu cầu đã được định rõ và chấp nhận.





Vòng đời/Quy trình PTPM

2. Defining Requirements: Giai đoạn xác định yêu cầu

#	Đầu vào	Đầu ra
3	Tài liệu tham chiếu: Bất kỳ tài liệu nào từ các dự án tương tự hoặc tiêu chuẩn ngành công nghiệp có thể được sử dụng làm nguồn tham khảo để hiểu và xác định yêu cầu cho phần mềm.	Xác nhận yêu cầu: Xác nhận tài liệu đáp ứng các mục tiêu và mong đợi của khách hàng và các bên liên quan khác.
4	Tài liệu lập kế hoạch dự án: Các tài liệu lập kế hoạch dự án cung cấp thông tin về các mục tiêu, nguồn lực, lịch trình và các ràng buộc khác của dự án, giúp xác định yêu cầu phù hợp.	Sự hiểu biết rõ ràng về yêu cầu: Các nhà phát triển và các bên liên quan có sự hiểu biết rõ ràng và thống nhất về yêu cầu của phần mềm.





Vòng đời/Quy trình PTPM

2. Defining Requirements: Giai đoạn xác định yêu cầu

#	Các hoạt động
1	Xác định Yêu cầu (Defining Requirements): Thu thập và xác định các yêu cầu của khách hàng và bên liên quan, cũng như xác định mục tiêu và phạm vi của dự án chi tiết.
2	Yêu cầu Chức năng và Phi chức năng (Non/Functional Requirements): Xác định các tính năng và chức năng cụ thể mà phần mềm cần phải cung cấp để đáp ứng yêu cầu của người dùng.
3	Yêu cầu Kỹ thuật (Technical Requirements): Xác định các yêu cầu kỹ thuật, bao gồm nền tảng công nghệ, kiến trúc hệ thống, hiệu suất và bảo mật, để đảm bảo rằng phần mềm được thiết kế và triển khai một cách hợp lý.
4	Đánh giá và Phê duyệt Yêu cầu (Requirement Reviews and Approval): Tiến hành xem xét và đánh giá các yêu cầu được xác định, đảm bảo tính đầy đủ, rõ ràng và phù hợp với mục tiêu dự án, sau đó đưa ra quyết định phê duyệt hoặc điều chỉnh cần thiết.





Vòng đời/Quy trình PTPM

3. Designing Architecture: Thiết kế kiến trúc

BRD/User Story/URD/SRS/FSD là các tài liệu tham khảo cho nhà thiết kế phần mềm, giúp xây dựng kiến trúc tốt nhất cho sản phẩm. Các yêu cầu từ các tài liệu này được đưa vào DDS để thiết kế kiến trúc. DDS được đánh giá trước khi chọn thiết kế cuối cùng cho quá trình phát triển.

#	Đầu vào	Đầu ra
1	Các tài liệu Yêu cầu Phần mềm: Cung cấp các yêu cầu chức năng và phi chức năng của phần mềm, làm cơ sở cho việc thiết kế kiến trúc.	Tài liệu Thiết kế Chi tiết (DDS): DDS chứa các thiết kế chi tiết về kiến trúc của phần mềm, bao gồm cấu trúc tổ chức, giao diện và các thành phần chức năng khác.
2	Thiết kế sơ bộ (nếu có): Nếu có, các thiết kế sơ bộ từ giai đoạn trước có thể được sử dụng như là một hướng dẫn cho việc phát triển kiến trúc chi tiết hơn.	Phê duyệt từ khách hàng và bên liên quan: DDS cần được phê duyệt bởi khách hàng và các bên liên quan để xác nhận rằng thiết kế đáp ứng các yêu cầu và mong đợi.





Vòng đời/Quy trình PTPM

3. Designing Architecture: Thiết kế kiến trúc

#	Đầu vào	Đầu ra
3	Thông tin từ các bên liên quan: Phản hồi và yêu cầu từ khách hàng, các bên liên quan và các nhà phân tích thị trường cũng là một phần của đầu vào, để đảm bảo rằng thiết kế kiến trúc đáp ứng đúng nhu cầu và mong đợi.	Lựa chọn thiết kế cuối cùng: Một thiết kế cuối cùng được chọn sau khi xem xét và đánh giá các thiết kế đề xuất trong DDS.





Vòng đời/Quy trình PTPM

3. Designing Architecture: Thiết kế kiến trúc

#	Các hoạt động
1	Thiết kế (Design): Xác định yêu cầu và đề xuất giải pháp thiết kế tổng quan cho hệ thống; Đánh giá rủi ro và lựa chọn kiến trúc phù hợp.
2	Thiết kế Chi tiết (LLD): Phân rã chức năng, thiết kế cấu trúc dữ liệu, và định nghĩa thuật toán chi tiết; Tạo tài liệu mô tả chi tiết về thiết kế.
3	Thiết kế Kiến trúc cao cấp (HLD): Định nghĩa kiến trúc tổng thể của hệ thống và cách các thành phần tương tác; Phân tích tính năng và tạo tài liệu mô tả kiến trúc tổng thể.





Vòng đời/Quy trình PTPM

4. Developing Product: Phát triển sản phẩm

Trong giai đoạn này, các nhà phát triển sử dụng mã lập trình dựa trên thiết kế trong DDS để bắt đầu phát triển sản phẩm. Tuân thủ quy định của tổ chức là điều quan trọng đối với lập trình viên, và họ sử dụng các công cụ lập trình truyền thống như trình biên dịch, trình thông dịch, trình gỡ lỗi,... Các ngôn ngữ lập trình phổ biến như C/C++, Python, Java được sử dụng theo quy định của dự án.

#	Đầu vào	Đầu ra
1	Tài liệu Thiết kế Chi tiết (DDS): Tài liệu này chứa thông tin chi tiết về thiết kế của sản phẩm, bao gồm các yếu tố như cấu trúc dữ liệu, thuật toán, giao diện, và các chi tiết khác cần thiết cho việc phát triển.	Mã nguồn Phần mềm: Mã nguồn của sản phẩm phần mềm được phát triển theo thiết kế đã được xác định trong DDS.
2	Yêu cầu Phần mềm (SRS): Các yêu cầu chức năng và phi chức năng được xác định từ tài liệu yêu cầu phần mềm, cung cấp hướng dẫn cho quá trình phát triển.	Phiên bản Phần mềm: Phiên bản sẵn sàng cho việc kiểm thử và triển khai sau khi hoàn thành giai đoạn phát triển.





Vòng đời/Quy trình PTPM

4. Developing Product: Phát triển sản phẩm

#	Đầu vào	Đầu ra
3	Công cụ lập trình: Các công cụ như trình biên dịch, trình thông dịch, trình gỡ lỗi,... được sử dụng để viết và kiểm tra mã nguồn.	Tài liệu Hướng dẫn Sử dụng (nếu có): Nếu có, tài liệu hướng dẫn sử dụng sẽ được tạo ra để hướng dẫn người dùng sử dụng sản phẩm.
4	Ngôn ngữ lập trình: Sử dụng ngôn ngữ lập trình như C/C++, Python, Java,... tuân thủ các quy định của phần mềm.	





Vòng đời/Quy trình PTPM

4. Developing Product: Phát triển sản phẩm

#	Các hoạt động
1	Phát triển (Development): <ul style="list-style-type: none">Mã hóa: Viết mã nguồn theo thiết kế đã được xác định trong DDS;Kiểm tra và Sửa lỗi: Kiểm tra mã nguồn để phát hiện và sửa các lỗi cú pháp, lỗi logic và các vấn đề khác;Tích hợp và Thử nghiệm: Tích hợp các thành phần riêng lẻ và thực hiện thử nghiệm để đảm bảo hoạt động nhất quán của hệ thống.
2	Tiêu chuẩn Mã lập trình (Coding standard): Đảm bảo rằng mã nguồn tuân thủ các tiêu chuẩn mã hóa được xác định trước đó. Các tiêu chuẩn này bao gồm cách đặt tên biến, cách định dạng mã nguồn, và các quy tắc khác để làm cho mã nguồn dễ đọc, dễ hiểu và dễ bảo trì.





Vòng đời/Quy trình PTPM

4. Developing Product: Phát triển sản phẩm

#	Các hoạt động
3	Mã nguồn Có thể Mở rộng (Scalable code): Phát triển mã nguồn mà có thể mở rộng một cách linh hoạt, tức là có khả năng mở rộng để phục vụ cho việc thêm mới hoặc thay đổi tính năng mà không cần sửa đổi quá nhiều mã nguồn hiện có.
4	Quản lý Phiên bản (Version control): Sử dụng hệ thống quản lý phiên bản như Git để theo dõi và quản lý các thay đổi trong mã nguồn. Điều này giúp đảm bảo rằng các phiên bản mã nguồn được lưu trữ và quản lý một cách hiệu quả, và cho phép khôi phục lại các phiên bản trước đó nếu cần thiết.
5	Đánh giá Mã nguồn (Code review): Tiến hành đánh giá mã nguồn bởi các thành viên khác trong nhóm phát triển để kiểm tra và cải thiện chất lượng của mã nguồn. Quá trình này giúp phát hiện lỗi và vấn đề sớm, đồng thời cũng tạo ra cơ hội để chia sẻ kiến thức và kinh nghiệm giữa các thành viên trong nhóm.





Vòng đời/Quy trình PTPM

5. Product Testing and Integration: Kiểm thử và tích hợp sản phẩm

Sau khi phát triển sản phẩm, việc kiểm thử phần mềm là cần thiết để đảm bảo hoạt động mượt mà của nó.

Mặc dù, việc kiểm thử tối thiểu được tiến hành ở mỗi giai đoạn của vòng đời phát triển phần mềm. Do đó, ở giai đoạn này, tất cả các lỗi có thể xảy ra được theo dõi, sửa chữa và kiểm tra lại. Điều này đảm bảo rằng sản phẩm đáp ứng các yêu cầu chất lượng của các tài liệu tại bước 2&3;

#	Đầu vào	Đầu ra
1	Sản phẩm Phần mềm: Phiên bản hoặc bản phát triển gần nhất của sản phẩm phần mềm cần được kiểm thử và tích hợp.	Phiên bản Sản phẩm đã kiểm thử: Phiên bản của sản phẩm phần mềm được kiểm thử và tích hợp, sẵn sàng cho giai đoạn triển khai hoặc bản phát hành.
2	Tài liệu Thiết kế và Yêu cầu Phần mềm (DDS và SRS): Tài liệu thiết kế chi tiết và yêu cầu chức năng và phi chức năng của sản phẩm để sử dụng như một cơ sở cho việc kiểm tra và tích hợp.	Báo cáo Kiểm thử và Tích hợp: Báo cáo chi tiết về các kết quả kiểm thử và tích hợp, bao gồm các lỗi phát hiện, kết quả kiểm thử, và bất kỳ vấn đề nào được ghi nhận.





Vòng đời/Quy trình PTPM

5. Product Testing and Integration: Kiểm thử và tích hợp sản phẩm

#	Đầu vào	Đầu ra
3	Mã nguồn và Tài liệu Thiết kế: Mã nguồn của sản phẩm phần mềm và tài liệu thiết kế được sử dụng để xác định các tính năng, hàm, và cấu trúc của sản phẩm.	
4	Công cụ và Môi trường Kiểm thử: Các công cụ kiểm thử tự động và môi trường kiểm thử cần thiết để thực hiện các bài kiểm tra và tích hợp.	





Vòng đời/Quy trình PTPM

5. Product Testing and Integration: Kiểm thử và tích hợp sản phẩm

#	Các hoạt động
1	Kiểm thử Phần mềm: Thực hiện các bài kiểm thử để đảm bảo tính chính xác, hoạt động và hiệu suất của sản phẩm phần mềm.
2	Tích hợp Phần mềm: Tích hợp các thành phần riêng lẻ của sản phẩm và đảm bảo rằng chúng hoạt động một cách nhất quán khi kết hợp lại với nhau.
3	Sửa lỗi và Retest: Sửa các lỗi phát hiện trong quá trình kiểm thử và tích hợp, và thực hiện các bài kiểm thử lại để đảm bảo rằng các lỗi đã được khắc phục đúng cách.
4	Tạo và Cập nhật Tài liệu: Tạo và cập nhật tài liệu liên quan đến việc kiểm thử và tích hợp, bao gồm tài liệu hướng dẫn kiểm thử và tài liệu hướng dẫn sử dụng.
5	Đào tạo và Hỗ trợ: Cung cấp đào tạo cho người dùng cuối và cung cấp hỗ trợ sau triển khai để giúp họ sử dụng sản phẩm một cách hiệu quả.





Vòng đời/Quy trình PTPM

6. Deployment and Maintenance of Products: Triển khai và bảo trì sản phẩm

Sau khi hoàn thành kiểm thử chi tiết, sản phẩm cuối cùng được phát hành theo các giai đoạn theo chiến lược của tổ chức. Sau đó, sản phẩm sẽ được kiểm thử trên môi trường thực tế để đảm bảo hoạt động mượt mà.

#	Đầu vào	Đầu ra
1	Sản phẩm Phần mềm hoàn thiện: Phiên bản cuối cùng của sản phẩm phần mềm sau khi đã hoàn thành quá trình phát triển và kiểm thử.	Sản phẩm phần mềm được triển khai: Sản phẩm phần mềm được phát hành cho người dùng cuối hoặc khách hàng.
2	Phản hồi từ quá trình kiểm thử: Nhận phản hồi từ quá trình kiểm thử trước đó để xác định các vấn đề còn tồn đọng và các cải tiến cần thiết.	Cải tiến sản phẩm (nếu có): Các phiên bản cập nhật hoặc bản vá được phát hành sau khi thu thập và xử lý phản hồi từ người dùng.



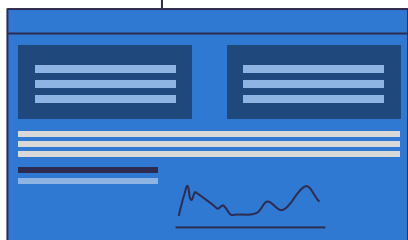


Vòng đời/Quy trình PTPM

6. Deployment and Maintenance of Products: Triển khai và bảo trì sản phẩm

#	Các hoạt động
1	Lập kế hoạch Phát hành (Release Planning): Xác định lịch trình và các tính năng sẽ được phát hành trong các phiên bản sản phẩm tiếp theo dựa trên yêu cầu của khách hàng và phản hồi từ người dùng.
2	Triển khai và Bảo trì (Deployment and Maintenance): Triển khai sản phẩm phần mềm cho người dùng cuối, và tiếp tục theo dõi và bảo trì sản phẩm sau khi đã triển khai để đảm bảo tính ổn định và hiệu suất.
3	Tự động hóa Triển khai (Deployment Automation): Tự động hóa các quy trình triển khai sản phẩm để tăng tính hiệu quả và đảm bảo sự nhất quán trong quá trình triển khai.
4	Bảo trì (Maintenance): Thực hiện các hoạt động bảo trì như sửa lỗi, cải tiến và cập nhật phần mềm để đảm bảo tính ổn định và hiệu suất của sản phẩm.
5	Thu thập Phản hồi (Feedback): Thu thập phản hồi từ người dùng cuối và các bên liên quan về trải nghiệm sử dụng sản phẩm và các vấn đề gặp phải, sau đó cải thiện sản phẩm trong các phiên bản và bản cập nhật sau này.





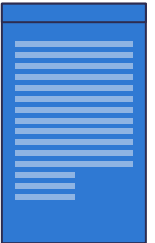
02

Các mô hình Phát triển phần mềm





Các mô hình PTPM



1. Thác nước (Waterfall)

2. Xoắn ốc (Spiral)

3. Bản mẫu (Prototyping)

4. Chữ V (V)

5. Phát triển nhanh (RAD)

6. Hợp nhất (RUP)

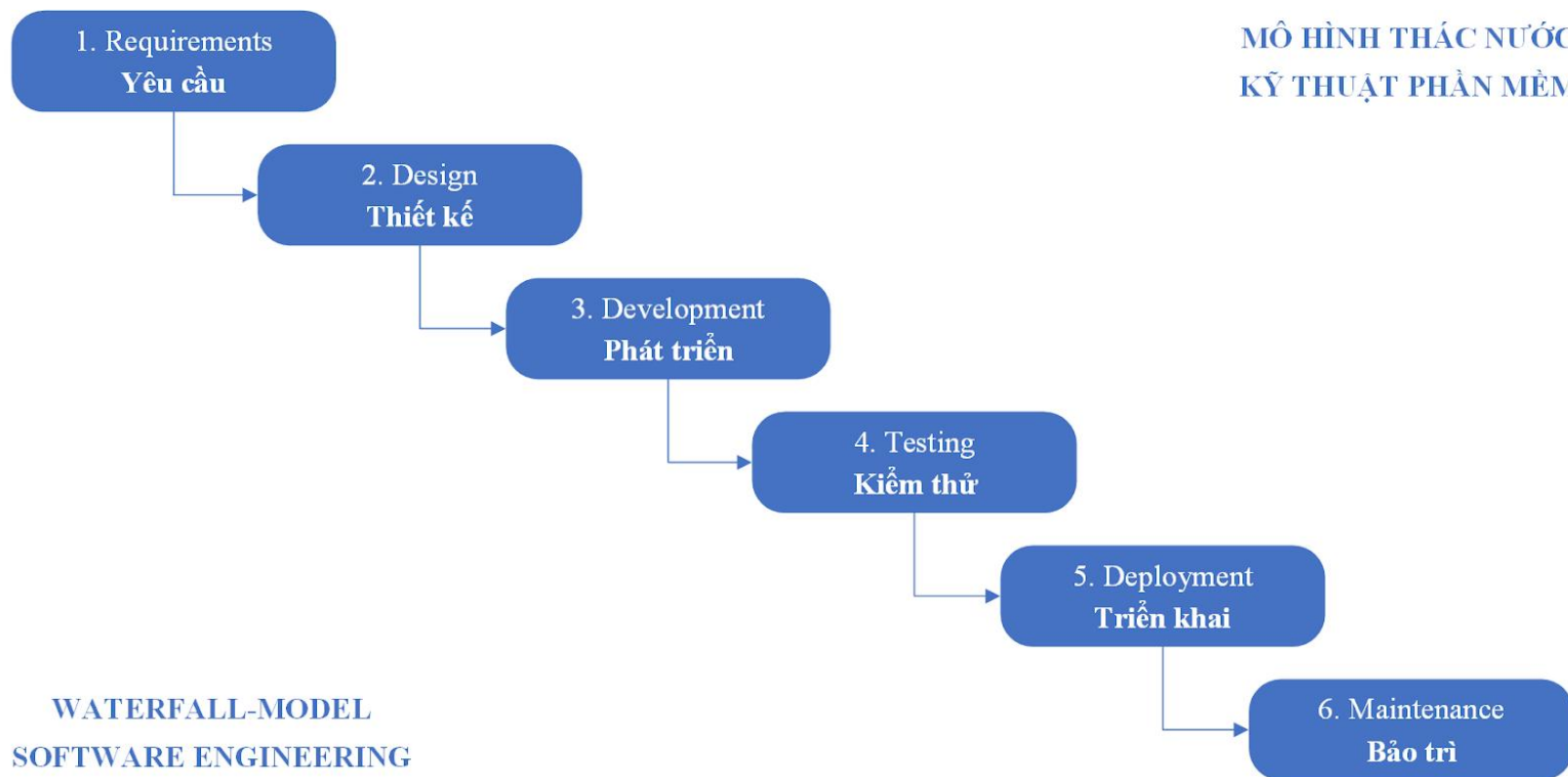
7. Scrum-Agile





1. Mô hình Thác nước (Waterfall)

Khái niệm: Mô hình thác nước (Waterfall model) là một trong những mô hình phát triển phần mềm cổ điển nhất và đơn giản nhất trong các mô hình vòng đời phát triển phần mềm. Mô hình này được tổ chức theo kiểu tuyến tính, trong đó các giai đoạn phát triển liên tục chảy từ trên xuống dưới như một dãy thác nước.



Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





1. Mô hình Thác nước (Waterfall)

Các bước thực hiện trong mô hình thác nước bao gồm:

- 1. Yêu cầu (Requirements):** Thu thập, phân tích và xác định yêu cầu của khách hàng cho sản phẩm phần mềm.
- 2. Thiết kế (Design):** Xây dựng thiết kế chi tiết của hệ thống dựa trên yêu cầu đã xác định trong giai đoạn trước.
- 3. Phát triển (Development):** Viết mã và triển khai các tính năng, chức năng và cấu trúc đã thiết kế.
- 4. Kiểm thử (Testing):** Kiểm tra và đánh giá sản phẩm phần mềm để đảm bảo rằng nó đáp ứng yêu cầu và hoạt động như mong đợi.
- 5. Triển khai (Deployment):** Phát hành sản phẩm phần mềm cho người dùng cuối hoặc môi trường sản xuất.
- 6. Bảo trì (Maintenance):** Sửa chữa, cải tiến và duy trì sản phẩm phần mềm sau khi triển khai để đảm bảo tính ổn định và hiệu suất.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile



Ưu nhược điểm:

#	Ưu điểm	Nhược điểm
1	Dễ hiểu và sử dụng	Không có đường truyền phản hồi
2	Dễ quản lý	Khó thích ứng với yêu cầu thay đổi
3	Dễ kiểm soát rủi ro	Không có phần chồng chéo giữa các giai đoạn
4	Phù hợp với các dự án có yêu cầu ổn định	Sự tham gia của các bên liên quan hạn chế
5		Phát hiện lỗi muộn
6		Chu kỳ phát triển dài
7		Không phù hợp cho các dự án phức tạp

- Waterfall
- Spiral
- Prototyping
- V
- RAD
- RUP
- Scrum-Agile

Dự án áp dụng mô hình này:

- Waterfall
- Spiral
- Prototyping
- V
- RAD
- RUP
- Scrum-Agile

#	Tiêu chí	Mô tả
1	Yêu cầu dự án	Phù hợp cho các dự án có yêu cầu rõ ràng từ đầu.
2	Thời gian và ngân sách	Đảm bảo tiến độ và ngân sách nếu yêu cầu được xác định cẩn thận từ đầu.
3	Phạm vi dự án	Dự án có phạm vi ổn định và không thay đổi nhiều (nhỏ và trung bình).
4	Tính linh hoạt	Không linh hoạt và khó thích ứng với thay đổi yêu cầu.
5	Tương tác khách hàng	Không tạo điều kiện thuận lợi cho tương tác liên tục với khách hàng.
6	Quản lý rủi ro	Giúp quản lý rủi ro hiệu quả hơn với việc xác định rõ ràng các giai đoạn.
7	Tính phù hợp	Phù hợp với nhóm phát triển có kinh nghiệm và dự án có yêu cầu ổn định.
8	Chất lượng và hiệu suất	Đảm bảo chất lượng và hiệu suất qua việc phân chia dự án thành các giai đoạn rõ ràng.

Dự án áp dụng mô hình này:

#	Tiêu chí	Mô tả
9	Khả năng tái sử dụng	Có thể tái sử dụng các thành phần phần mềm nếu được thiết kế và triển khai cẩn thận.
10	Tính đáng tin cậy	Đáng tin cậy nếu yêu cầu được xác định rõ ràng và các giai đoạn được thực hiện cẩn thận.
11	Phản ứng cộng đồng	Là một mô hình phổ biến và được sử dụng rộng rãi trong ngành công nghiệp.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





2. Mô hình Xoắn ốc (Spiral)

Khái niệm: Mô hình Xoắn ốc (Spiral) là một mô hình phát triển phần mềm dựa trên rủi ro, có nghĩa là tập trung vào quản lý rủi ro thông qua nhiều vòng lặp của quá trình phát triển phần mềm.

Waterfall

Spiral

Prototyping

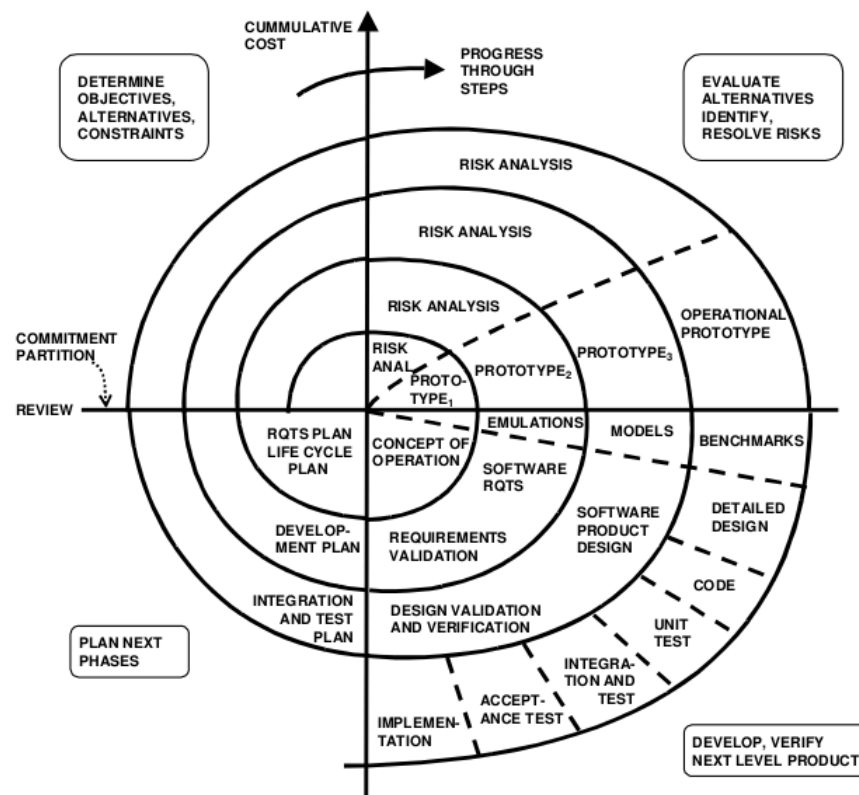
V

RAD

RUP

Scrum-Agile

- Mô hình Spiral tập trung vào việc phát triển theo từng vòng lặp, mỗi vòng lặp đều bao gồm các giai đoạn trên để đảm bảo rằng sản phẩm phần mềm được phát triển một cách linh hoạt và chất lượng, và rủi ro được quản lý một cách hiệu quả.
- Mô hình Spiral là sự kết hợp của một số phương pháp phát triển phần mềm như Waterfall, Prototype, và Iterative.





2. Mô hình Xoắn ốc (Spiral)

Các bước thực hiện trong mô hình thác nước bao gồm:

1. Giai đoạn Xác định mục tiêu và đề xuất các phương án thay thế (Determine objectives, Alternatives, Constraints): Yêu cầu được thu thập từ khách hàng và các mục tiêu được xác định, phân tích chi tiết ở đầu mỗi giai đoạn.

2. Giai đoạn Xác định và giải quyết rủi ro (Evaluate alternatives identify, resolve risks): Tất cả các phương án thay thế được đánh giá để chọn ra phương án tốt nhất. Sau đó, các rủi ro liên quan đến phương án đó được xác định và giải quyết bằng cách sử dụng chiến lược tốt nhất. Cuối cùng, một mô hình nguyên mẫu được xây dựng cho phương án tốt nhất.

3. Giai đoạn Phát triển phiên bản tiếp theo của sản phẩm (Develop, Verify next level product): Các tính năng được xác định được phát triển và kiểm thử. Cuối cùng, phiên bản tiếp theo của phần mềm sẵn có.

4. Giai đoạn Đánh giá và lập kế hoạch cho giai đoạn tiếp theo (Plan next phases): Trong giai đoạn thứ tư, khách hàng đánh giá phiên bản phần mềm đã phát triển đến thời điểm hiện tại. Cuối cùng, việc lập kế hoạch cho giai đoạn tiếp theo được bắt đầu.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





2. Mô hình Xoắn ốc (Spiral)

Ưu nhược điểm:

#	Ưu điểm	Nhược điểm
1	Xử lý rủi ro hiệu quả	Phức tạp
2	Phù hợp cho các dự án lớn và phức tạp	Tốn kém về tài nguyên
3	Linh hoạt trong việc điều chỉnh yêu cầu	Tốn kém về thời gian
4	Cải thiện giao tiếp giữa các bên liên quan	Phụ thuộc quá nhiều vào phân tích rủi ro
5	Cải thiện chất lượng và độ tin cậy của sản phẩm	Khó khăn trong quản lý thời gian

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





2. Mô hình Xoắn ốc (Spiral)

Dự án áp dụng mô hình này:

#	Tiêu chí	Mô tả
1	Yêu cầu dự án	Dự án với yêu cầu biến đổi và không xác định rõ từ đầu.
2	Thời gian và ngân sách	Dự án có yêu cầu phức tạp và cần phải quản lý rủi ro một cách hiệu quả để kiểm soát thời gian và ngân sách.
3	Phạm vi dự án	Dự án có phạm vi linh hoạt và cần phải điều chỉnh theo yêu cầu mới hoặc thay đổi từ khách hàng.
4	Tính linh hoạt	Dự án yêu cầu khả năng thích ứng với sự thay đổi và điều chỉnh trong quá trình phát triển.
5	Tương tác khách hàng	Dự án cần liên tục tương tác với khách hàng để đảm bảo sự hài lòng và phản hồi đúng đắn.
6	Quản lý rủi ro	Dự án với nhiều rủi ro không xác định và cần phải được quản lý và giảm thiểu trong quá trình phát triển.
7	Tính phù hợp	Dự án cần một phương pháp phát triển linh hoạt và có khả năng thích ứng với môi trường thay đổi.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





2. Mô hình Xoắn ốc (Spiral)

Dự án áp dụng mô hình này:

#	Tiêu chí	Mô tả
8	Chất lượng và hiệu suất	Dự án cần cải thiện chất lượng và hiệu suất thông qua việc lặp lại và kiểm soát chất lượng định kỳ.
9	Khả năng tái sử dụng	Dự án cần phải thiết kế và triển khai các thành phần có thể tái sử dụng cho các dự án tương lai.
10	Tính đáng tin cậy	Dự án cần đảm bảo sự đáng tin cậy thông qua việc quản lý rủi ro và kiểm soát chất lượng định kỳ.
11	Phản ứng cộng đồng	Dự án cần phản hồi tích cực và hỗ trợ từ cộng đồng phát triển và khách hàng.

Waterfall

Spiral

Prototyping

V

RAD

RUP

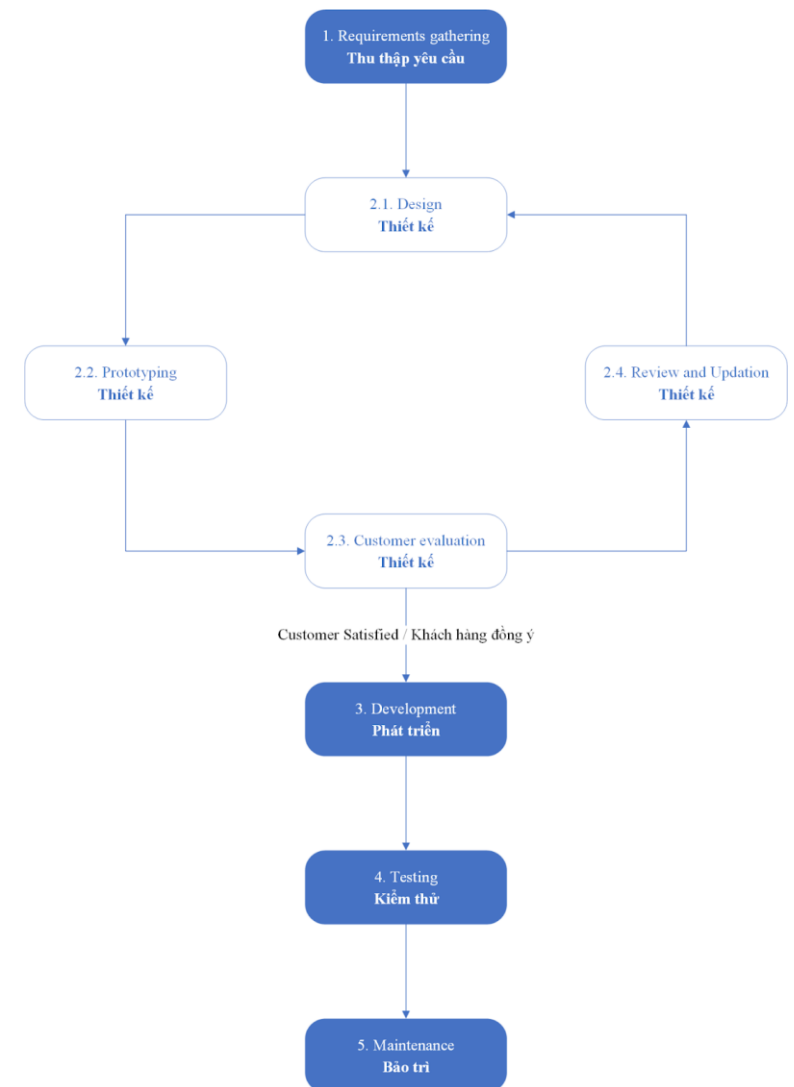
Scrum-Agile



MHPTPM

3. Mô hình Bản mẫu (Prototyping)

Khái niệm: Mô hình Bản mẫu (Prototyping) là một phương pháp phát triển phần mềm tập trung vào việc xây dựng nhanh chóng các phiên bản prototype của sản phẩm cuối cùng, cho phép người dùng cuối đánh giá và cung cấp phản hồi sớm trong quá trình phát triển. Mô hình này thường được sử dụng khi yêu cầu không rõ ràng hoặc thay đổi thường xuyên.



Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile

MHPTPM

3. Mô hình Bản mẫu (Prototyping)

Các bước thực hiện trong mô hình thác nước bao gồm:

- 1. Thu thập yêu cầu (Requirements gathering):** Thu thập yêu cầu từ khách hàng và các bên liên quan để hiểu rõ mục tiêu và yêu cầu của dự án.
- 2. Thiết kế (Design):** Xây dựng một thiết kế ban đầu dựa trên yêu cầu đã thu thập gồm Thiết kế (Design); Tạo bản prototype (Prototyping); Đánh giá của khách hàng (Customer Evaluation); Xem xét và Cập nhật (Review and Updation). Sau đó, tạo ra một phiên bản prototype dựa trên thiết kế này.
- 3. Phát triển (Development):** Phát triển prototype dựa trên thiết kế ban đầu để hiển thị các tính năng và chức năng cơ bản của hệ thống.
- 4. Kiểm thử (Testing):** Tiến hành kiểm thử prototype để xác nhận tính đúng đắn của các tính năng và chức năng đã phát triển và đảm bảo rằng chúng đáp ứng được yêu cầu của khách hàng.
- 5. Bảo trì (Maintenance):** Đánh giá và thu thập phản hồi từ khách hàng. Quá trình này có thể lặp lại cho đến khi đạt được một phiên bản prototype hoặc sản phẩm cuối cùng mong muốn.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile

Ưu nhược điểm:

#	Ưu điểm	Nhược điểm
1	Phản hồi sớm từ khách hàng	Tăng chi phí
2	Hiểu rõ hơn về yêu cầu	Thời gian bị chậm trễ trong quá trình phát triển
3	Thúc đẩy sự sáng tạo	Rủi ro về hiểu biết sai lầm về yêu cầu và mục tiêu
4	Giảm rủi ro trong quá trình phát triển	Quản lý phiên bản prototype sẽ phức tạp dần
5	Tăng cường sự tương tác, gia tăng sự hài lòng	

- Waterfall
- Spiral
- Prototyping
- V
- RAD
- RUP
- Scrum-Agile

MHPTPM

3. Mô hình Bản mẫu (Prototyping)

Dự án áp dụng mô hình này:

#	Tiêu chí	Mô tả
1	Yêu cầu dự án	Yêu cầu không rõ ràng hoặc thay đổi thường xuyên.
2	Thời gian và ngân sách	Thời gian và ngân sách linh hoạt để tạo và thử nghiệm các phiên bản prototype.
3	Phạm vi dự án	Phạm vi nhỏ đến trung bình, có thể chia thành các phần nhỏ để phát triển từng phần một.
4	Tính linh hoạt	Cần phải thích ứng nhanh chóng với thay đổi trong yêu cầu hoặc thị trường.
5	Tương tác khách hàng	Cần tương tác chặt chẽ với khách hàng để hiểu rõ nhu cầu và cung cấp phản hồi sớm từ các phiên bản prototype.
6	Quản lý rủi ro	Rủi ro cao hoặc không chắc chắn về yêu cầu, thích ứng với việc tạo và kiểm thử prototype giúp giảm thiểu rủi ro và giải quyết vấn đề sớm hơn.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile



MHPTPM

3. Mô hình Bản mẫu (Prototyping)

Dự án áp dụng mô hình này:

#	Tiêu chí	Mô tả
7	Tính phù hợp	Cần có phương pháp phát triển linh hoạt và có thể điều chỉnh để phản ánh sự thay đổi trong nhu cầu của khách hàng và các bên liên quan.
8	Chất lượng và hiệu suất	Đảm bảo chất lượng và hiệu suất cao thông qua việc phản hồi sớm từ khách hàng và cải thiện liên tục.
9	Khả năng tái sử dụng	Tối ưu hóa tái sử dụng và giảm thiểu thời gian và chi phí phát triển thông qua việc xây dựng các thành phần có thể tái sử dụng.
10	Tính đáng tin cậy	Đạt được mức độ đáng tin cậy cao hơn thông qua việc phản hồi sớm từ khách hàng và giải quyết các vấn đề sớm hơn.
11	Phản ứng cộng đồng	Tạo điều kiện cho sự tương tác chặt chẽ và phản hồi từ cộng đồng người dùng và các bên liên quan khác để cải thiện sản phẩm.

Waterfall

Spiral

Prototyping

V

RAD

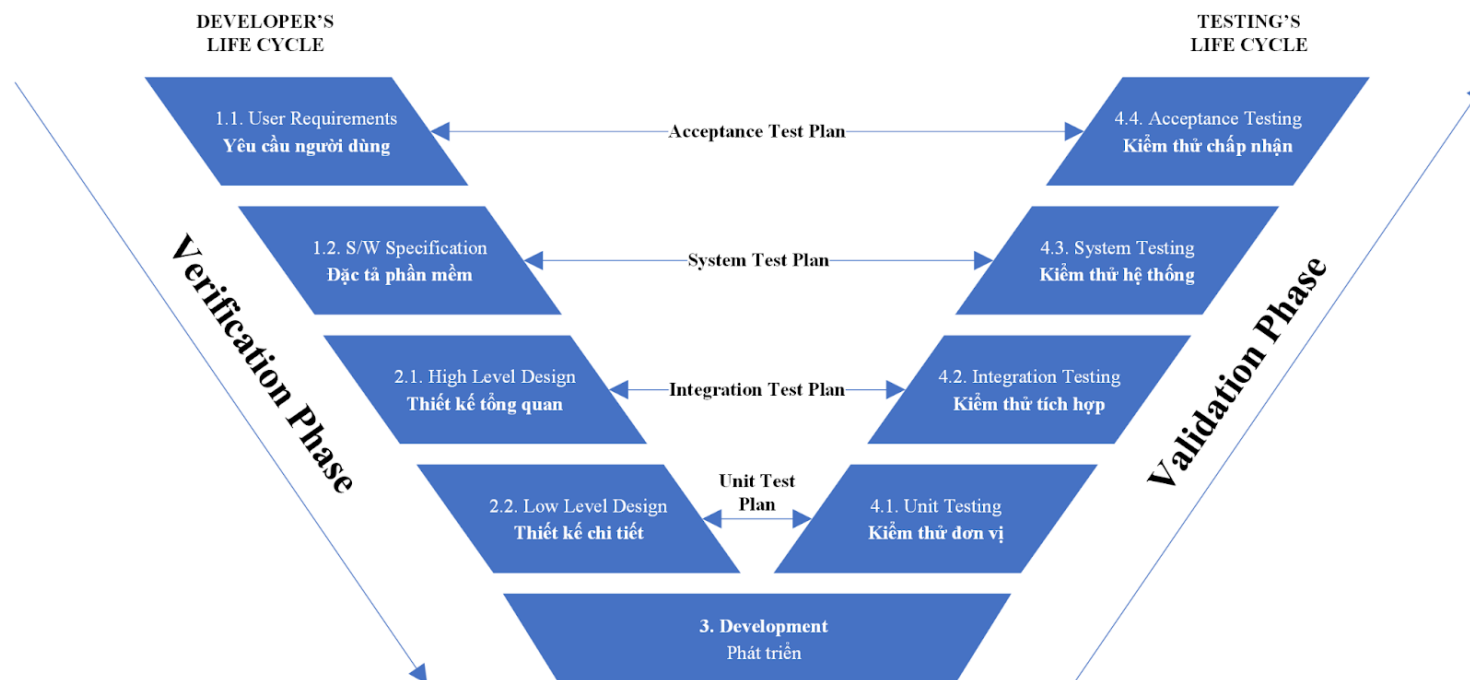
RUP

Scrum-Agile



4. Mô hình chữ V (V)

Khái niệm: Mô hình "chữ V" là phương pháp phát triển phần mềm trong đó quá trình kiểm thử được xem xét cùng với quá trình phát triển. Tên gọi "chữ V" xuất phát từ việc biểu diễn các giai đoạn phát triển và kiểm thử dưới dạng các dãy thời gian hình chữ V, trong đó các bước phát triển ở phía trên của "V" và các bước kiểm thử ở phía dưới. Mô hình này giúp đảm bảo rằng từng phần của phần mềm được phát triển đúng cách và kiểm thử một cách cẩn thận để đảm bảo chất lượng.





4. Mô hình chữ V (V)

Các bước thực hiện trong mô hình thác nước bao gồm:

- 1. Thu thập yêu cầu (Requirements Gathering)** bao gồm User Requirements và System Specification: Thu thập yêu cầu từ khách hàng và người dùng cuối để hiểu rõ mục tiêu và phạm vi dự án.
- 2. Thiết kế hệ thống (System Design/Architecture Design):** Xây dựng kiến trúc tổng quan của hệ thống để xác định cách các thành phần sẽ tương tác với nhau và cách chúng sẽ hoạt động cùng nhau để đạt được mục tiêu của dự án.
- 3. Thiết kế chi tiết (Detail Design/Module Design):** Thiết kế chi tiết của từng module hoặc thành phần trong hệ thống để xác định cách chúng sẽ thực hiện các chức năng cụ thể.
- 4. Phát triển (Development/Coding):** Viết mã nguồn dựa trên thiết kế đã được xác định để chuyển đổi ý tưởng và thiết kế thành mã máy thực tế.
- 5. Kiểm thử đơn vị (Unit Testing):** Kiểm tra từng đơn vị mã nguồn để đảm bảo tính đúng đắn và chức năng của từng module hoặc thành phần.
- 6. Kiểm thử thành phần (Component Testing):** Kiểm tra các thành phần đã được phát triển và xác định tính chính xác và tính tương thích của chúng.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





4. Mô hình chữ V (V)

Các bước thực hiện trong mô hình thác nước bao gồm:

7. Kiểm thử tích hợp (Integration Testing): Kiểm tra tích hợp các thành phần để đảm bảo tính tương thích và hoạt động đúng đắn của toàn bộ hệ thống.

8. Kiểm thử hệ thống (System Testing): Kiểm tra toàn bộ hệ thống để đảm bảo rằng tất cả các thành phần đã hoạt động đúng đắn và đạt được mục tiêu yêu cầu.

9. Kiểm thử chấp nhận (Acceptance Testing): Kiểm tra hệ thống từ quan điểm của người dùng cuối hoặc khách hàng để đảm bảo rằng nó đáp ứng các yêu cầu và mong đợi của họ.

10. Triển khai và bảo trì (Deployment and Maintenance): Triển khai hệ thống vào môi trường sản xuất và duy trì nó sau khi đã hoàn thành để đảm bảo tính ổn định và hiệu suất của hệ thống trong thời gian dài.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





4. Mô hình chữ V (V)

Ưu nhược điểm:

#	Ưu điểm	Nhược điểm
1	Quản lý và theo dõi tiến độ một cách chính xác	Rủi ro và không chắc chắn cao do xác nhận tất cả trước khi bắt đầu
2	Thích hợp đối với các dự án nhỏ và rõ ràng	Không phù hợp với dự án phức tạp và hướng đối tượng
3	Đơn giản và dễ hiểu	Không phù hợp với dự án có yêu cầu không rõ ràng
4	Tập trung xác nhận và xác minh sớm yêu cầu	Không lặp lại các giai đoạn
5	Rõ ràng và có cấu trúc	Không dễ dàng xử lý các sự kiện đồng thời
6	Tập trung vào kiểm thử	
7	Tăng cường khả năng theo dõi các thay đổi	
8	Cải thiện giao tiếp giữa khách hàng và nhóm PT	

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





4. Mô hình chữ V (V)

Dự án áp dụng mô hình này:

#	Tiêu chí	Mô tả
1	Yêu cầu dự án	Dự án có yêu cầu rõ ràng và đã được xác định cụ thể từ đầu.
2	Thời gian và ngân sách	Dự án có thời gian và ngân sách cố định, không có nhiều biến động.
3	Phạm vi dự án	Dự án có phạm vi cụ thể và không thay đổi quá nhiều trong quá trình phát triển.
4	Tính linh hoạt	Dự án không cần phải thay đổi linh hoạt trong quá trình phát triển, có thể tuân thủ một quy trình phát triển cụ thể và tiêu chuẩn.
5	Tương tác khách hàng	Khách hàng tham gia chủ động trong quá trình xác định yêu cầu và kiểm tra kết quả, có thể cung cấp phản hồi đúng thời điểm để cải thiện dự án.
6	Quản lý rủi ro	Dự án có khả năng quản lý và giảm thiểu rủi ro hiệu quả trong quá trình phát triển.
7	Tính phù hợp	Dự án có phù hợp với một quy trình phát triển cụ thể và yêu cầu tuân thủ các tiêu chuẩn và quy định.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





4. Mô hình chữ V (V)

Dự án áp dụng mô hình này:

#	Tiêu chí	Mô tả
8	Chất lượng và hiệu suất	Dự án đặt nặng vào việc đảm bảo chất lượng và hiệu suất của sản phẩm, từ quá trình thiết kế đến kiểm thử và triển khai.
9	Khả năng tái sử dụng	Có yêu cầu và nhu cầu tái sử dụng các thành phần phần mềm đã phát triển trong các dự án sau này.
10	Tính đáng tin cậy	Dự án đặt mục tiêu đảm bảo sự đáng tin cậy và ổn định của sản phẩm, không chấp nhận các lỗi và sự cố không mong muốn.
11	Phản ứng cộng đồng	Dự án cần có sự hỗ trợ và phản hồi tích cực từ cộng đồng, đặc biệt trong việc sử dụng và triển khai sản phẩm.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





5. Mô hình Phát triển nhanh (RAD)

Khái niệm: Mô hình RAD (Rapid Application Development) là một phương pháp phát triển phần mềm tập trung vào việc phát triển nhanh chóng các ứng dụng thông qua việc sử dụng các giai đoạn phát triển ngắn và tính tương tác cao với người dùng cuối.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





5. Mô hình Phát triển nhanh (RAD)

Các bước thực hiện trong mô hình thác nước bao gồm:

1. Mô hình nghiệp vụ (Business modeling): Xác định các luồng thông tin giữa các chức năng khác nhau của công việc.

2. Mô hình dữ liệu (Data modeling): Các thông tin thu thập được từ công đoạn 1 sẽ được sử dụng để định nghĩa ra các đối tượng dữ liệu nó cần thiết cho công việc.

3. Mô hình quy trình (Process modeling): Sau khi đã định nghĩa ra các đối tượng dữ liệu thì chúng được chuyển đổi để trở thành các dòng thông tin công việc, tiến đến một vài đối tượng công việc riêng.

4. Tạo ứng dụng (Application generation): Đây là công đoạn coding giống như trong các quy trình khác, khi dữ liệu và tiến trình thực hiện đã được vạch ra thì công đoạn này là sử dụng các công cụ lập trình, phân trình biên dịch, và các ngôn ngữ lập trình để tiến hành code lên các chức năng phần mềm.

5. Kiểm thử và nghiệm thu (Testing và Turnover): Là công đoạn test sản phẩm được tạo, sau khi việc coding đã hoàn thành, và cuối cùng đưa vào thử nghiệm trong thực tế, bán SP.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





5. Mô hình Phát triển nhanh (RAD)

Ưu nhược điểm:

#	Ưu điểm	Nhược điểm
1	Giảm thời gian phát triển	Không lý tưởng cho ngân sách thấp
2	Dễ hiểu	Yêu cầu các nhà phát triển có kỹ năng cao
3	Giảm rủi ro	Không thích hợp cho các dự án lớn
4	Ít nỗ lực hơn	Khó theo dõi tiến trình và các vấn đề
5	Tăng khả năng sử dụng lại của các thành phần	Hoạt động trên hệ thống module hóa
6	Giảm thiểu lỗi	Phụ thuộc nhiều vào hiệu suất của nhóm
7	Dễ thu thập và phản hồi từ khách hàng	Chỉ sử dụng cho những module phát triển 2-3 tháng
8	Tốt nhất cho các dự án vừa và nhỏ	Mô hình thiết kế phải sẵn có để có thể xây dựng nhiều nhóm cùng phát triển

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





5. Mô hình Phát triển nhanh (RAD)

Dự án áp dụng mô hình này:

#	Tiêu chí	Mô tả
1	Yêu cầu dự án	Dự án có yêu cầu không rõ ràng hoặc có thể thay đổi trong quá trình phát triển.
2	Thời gian và ngân sách	Dự án có thời gian triển khai nhanh và ngân sách hạn chế.
3	Phạm vi dự án	Dự án có phạm vi nhỏ hoặc có thể chia nhỏ thành các phần nhỏ hơn để phát triển theo các vòng lặp.
4	Tính linh hoạt	Dự án cần có khả năng thích ứng với thay đổi và cập nhật nhanh chóng từ phản hồi của khách hàng.
5	Tương tác khách hàng	Dự án yêu cầu sự tham gia tích cực của khách hàng và người dùng trong quá trình phát triển để cung cấp phản hồi và sửa đổi sản phẩm dễ dàng.
6	Quản lý rủi ro	Dự án với mức độ rủi ro không cao và có thể kiểm soát được trong quá trình phát triển.
7	Tính phù hợp	Dự án phù hợp khi cần phát triển và triển khai một ứng dụng hoặc hệ thống một cách nhanh chóng.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





5. Mô hình Phát triển nhanh (RAD)

Dự án áp dụng mô hình này:

#	Tiêu chí	Mô tả
8	Chất lượng và hiệu suất	Dự án yêu cầu đảm bảo chất lượng và hiệu suất cao thông qua việc sử dụng các phương pháp phát triển nhanh chóng và liên tục cải tiến dựa trên phản hồi.
9	Khả năng tái sử dụng	Dự án có thể tận dụng các thành phần và module được xây dựng sẵn từ các dự án trước đó hoặc có thể tái sử dụng các phần của ứng dụng đã được phát triển một cách nhanh chóng.
10	Tính đáng tin cậy	Dự án yêu cầu sản phẩm được phát triển nhanh chóng nhưng vẫn đảm bảo tính đáng tin cậy và ổn định trong quá trình sử dụng và triển khai.
11	Phản ứng cộng đồng	Dự án có thể triển khai sản phẩm nhanh chóng và nhận được phản hồi tích cực từ cộng đồng người dùng để cải thiện và phát triển sản phẩm theo hướng tốt nhất.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile

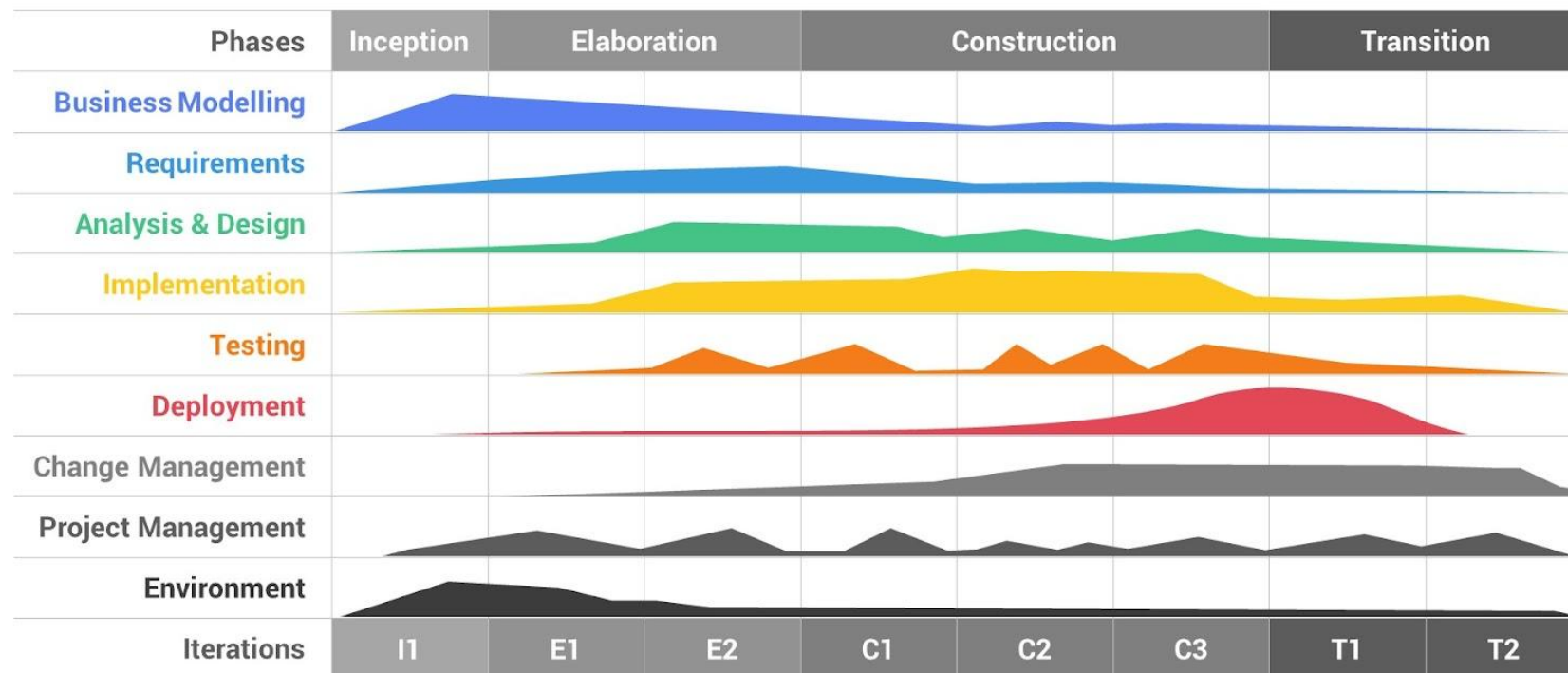




6. Mô hình Hợp nhất (RUP)

Khái niệm: Mô hình RUP (Rational Unified Process) là một framework phát triển phần mềm, RUP là một mô hình phát triển phần mềm linh hoạt, có cấu trúc và tập trung vào việc đảm bảo chất lượng và đáp ứng yêu cầu của khách hàng.

Rational Unified Process (RUP)





6. Mô hình Hợp nhất (RUP)

Các bước thực hiện trong mô hình thác nước bao gồm:

- 1. Khởi động (Inception):** Thành lập các đầu công việc trong một dự án.
- 2. Phác thảo (Elaboration):** Nghiên cứu lĩnh vực đang thực hiện trong dự án và kiến trúc hệ thống để có thể thực hiện dự án.
- 3. Xây dựng (Construction):** Thiết kế hệ thống, khung kế hoạch để thực hiện dự án.
- 4. Chuyển giao (Transition):** Triển khai thực hiện dự án.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





6. Mô hình Hợp nhất (RUP)

Ưu nhược điểm:

#	Ưu điểm	Nhược điểm
1	Tính linh hoạt	Phức tạp
2	Tập trung vào giá trị	Chi phí cao
3	Phát triển theo từng giai đoạn	Yêu cầu nhân sự có kỹ năng cao
4	Tập trung vào kiểm soát chất lượng, quản lý rủi ro	Tính linh hoạt bị hạn chế khi có thay đổi đột ngột
5	Tương tác và phản hồi với các bên liên quan	Tăng thời gian triển khai ban đầu
6	Tài liệu hóa và minh bạch	Rủi ro về tính ổn định và khả năng triển khai do phụ thuộc vào công nghệ.
7	Tính tái sử dụng	
8	Tăng cường hiệu suất	

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





6. Mô hình Hợp nhất (RUP)

Dự án áp dụng mô hình này:

#	Tiêu chí	Mô tả
1	Yêu cầu dự án	RUP thích hợp cho dự án có yêu cầu phức tạp và thay đổi thường xuyên.
2	Thời gian và ngân sách	RUP cung cấp các công cụ và quy trình để quản lý thời gian và ngân sách một cách hiệu quả.
3	Phạm vi dự án	RUP cho phép định rõ phạm vi dự án và quản lý các biến động phạm vi một cách linh hoạt.
4	Tính linh hoạt	RUP có thể điều chỉnh linh hoạt với các yêu cầu thay đổi và sự phát triển của dự án.
5	Tương tác khách hàng	RUP đặt khách hàng ở trung tâm của quá trình phát triển và thường xuyên tương tác với họ.
6	Quản lý rủi ro	RUP tập trung vào quản lý rủi ro từ giai đoạn đầu đến cuối của dự án.
7	Tính phù hợp	RUP có thể được điều chỉnh và tùy chỉnh cho phù hợp với nhu cầu cụ thể của dự án.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





6. Mô hình Hợp nhất (RUP)

Dự án áp dụng mô hình này:

#	Tiêu chí	Mô tả
8	Chất lượng và hiệu suất	RUP đặt nhiều sự chú ý vào việc đảm bảo chất lượng và hiệu suất của sản phẩm phát triển.
9	Khả năng tái sử dụng	RUP thúc đẩy việc tái sử dụng các thành phần và mã nguồn, giúp tăng tốc quá trình phát triển.
10	Tính đáng tin cậy	RUP cung cấp các phương tiện và quy trình để đảm bảo tính đáng tin cậy của sản phẩm.
11	Phản ứng cộng đồng	RUP thường được sử dụng và hỗ trợ bởi cộng đồng lớn của các nhà phát triển và chuyên gia.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





7. Mô hình Scrum-Agile

Khái niệm: Mô hình Agile là phương pháp phát triển phần mềm linh hoạt để làm sao đưa sản phẩm đến tay người dùng càng nhanh càng sớm càng tốt. Scrum là 1 dạng của mô hình Agile và là Framework phổ biến nhất khi thực hiện mô hình Agile. Scrum là mô hình phát triển phần mềm lặp đi lặp lại. Những khoảng lặp cố định thường kéo dài 1-2 tuần được gọi lại Sprint hoặc Iteration.

Waterfall

Spiral

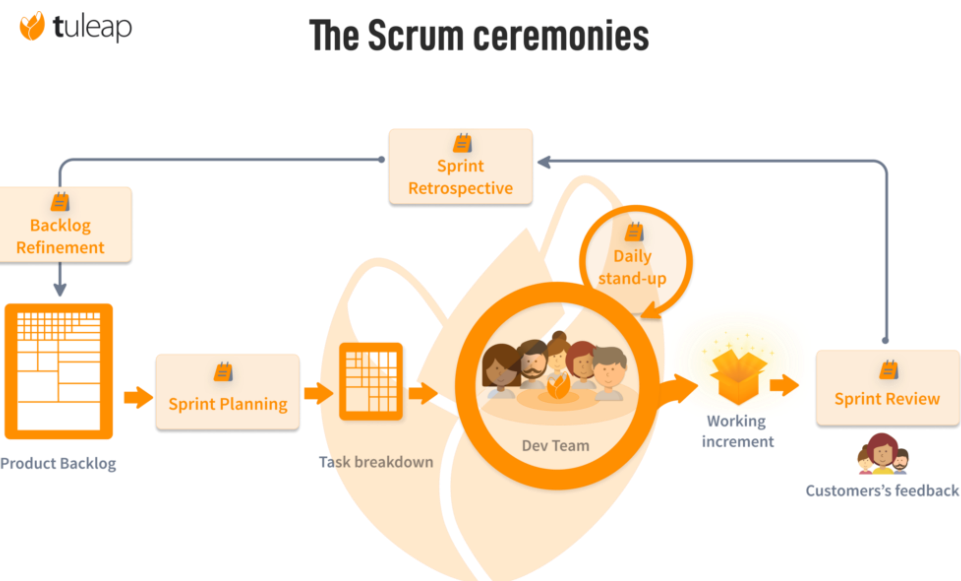
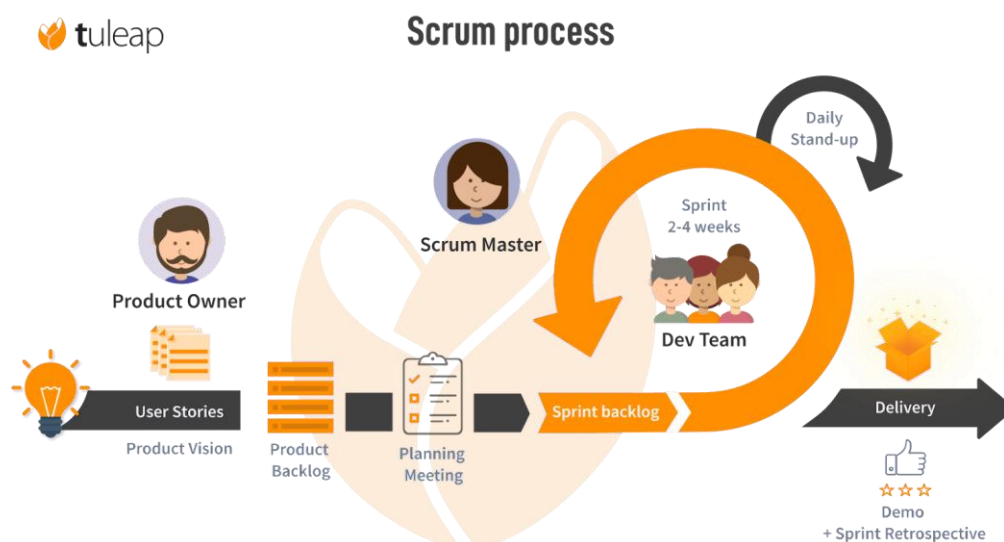
Prototyping

V

RAD

RUP

Scrum-Agile





7. Mô hình Scrum-Agile

Các bước thực hiện trong mô hình thác nước bao gồm:

1. Sprint Planning (Lập kế hoạch Sprint): Trong giai đoạn này, nhóm Scrum sẽ tổ chức cuộc họp Sprint Planning để xác định và cam kết cho mục tiêu của Sprint tiếp theo.

2. Sprint: Sprint là khoảng thời gian ngắn, thường từ 1 đến 4 tuần, trong đó nhóm Scrum thực hiện các công việc để hoàn thành các backlog items đã cam kết trong Sprint Planning.

3. Daily Stand-up (Họp hàng ngày): Mỗi ngày, các thành viên trong nhóm sẽ tham gia vào cuộc họp hàng ngày để báo cáo tiến độ, cập nhật công việc và giải quyết các vấn đề phát sinh.

4. Sprint Review (Xem xét Sprint): Sau khi hoàn thành một Sprint, nhóm sẽ tổ chức cuộc họp Sprint Review để xem xét và đánh giá kết quả làm việc trong Sprint vừa qua.

5. Sprint Retrospective (Hội hướng Sprint): Cuộc họp Sprint Retrospective diễn ra để nhóm đánh giá quá trình làm việc và đề xuất cải tiến cho các Sprint tiếp theo.

6. Backlog Refinement (Tái cấu trúc backlog): Trong giai đoạn này, các backlog items sẽ được đánh giá, cải thiện và điều chỉnh để chuẩn bị cho Sprint tiếp theo.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





7. Mô hình Scrum-Agile

Ưu nhược điểm:

#	Ưu điểm	Nhược điểm
1	Tính linh hoạt và tương tác	Khó khăn trong quản lý dự án lớn
2	Tăng sự tương tác và tham gia của khách hàng	Yêu cầu sự tham gia chủ động của khách hàng
3	Giảm rủi ro	Khả năng dự đoán kém
4	Phát triển sản phẩm đồng đều	Yêu cầu kỹ năng cao của nhóm phát triển
5	Tăng sự tự chủ của nhóm	Đòi hỏi sự cam kết lớn từ mọi thành viên
6	Tạo động lực cho nhóm	

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





7. Mô hình Scrum-Agile

Dự án áp dụng mô hình này:

#	Tiêu chí	Mô tả
1	Yêu Cầu Dự Án	Dự án có yêu cầu thay đổi thường xuyên và không rõ ràng từ đầu, cần sự linh hoạt trong việc thích ứng với sự biến động của yêu cầu.
2	Thời Gian và Ngân Sách	Dự án có thời gian giới hạn và ngân sách có hạn, cần phát triển sản phẩm nhanh chóng và tiết kiệm chi phí.
3	Phạm Vi Dự Án	Dự án có phạm vi nhỏ hoặc có thể chia nhỏ thành các phần nhỏ để phát triển theo các Sprint, giúp tập trung vào các tính năng quan trọng và ưu tiên cao.
4	Tính Linh Hoạt	Mô hình Agile-Scrum linh hoạt và cho phép điều chỉnh linh hoạt theo phản hồi từ khách hàng và sự thay đổi trong quá trình phát triển.
5	Tương Tác Khách Hàng	Khách hàng có thể tham gia vào các cuộc họp Sprint Planning, Sprint Review để cung cấp phản hồi và điều chỉnh yêu cầu sản phẩm.
6	Quản Lý Rủi Ro	Agile-Scrum cung cấp cơ chế linh hoạt để quản lý và giảm thiểu rủi ro trong quá trình phát triển, đặc biệt là thông qua các cuộc họp hàng ngày và Retrospective.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





7. Mô hình Scrum-Agile

Dự án áp dụng mô hình này:

#	Tiêu chí	Mô tả
7	Tính Phù Hợp	Agile-Scrum thích hợp cho dự án phần mềm với tính chất phức tạp và không chắc chắn từ đầu, nơi yêu cầu thay đổi liên tục và kiểm tra sản phẩm thường xuyên.
8	Chất Lượng và Hiệu Suất	Agile-Scrum giúp tạo ra sản phẩm chất lượng cao thông qua việc kiểm thử liên tục, tái sử dụng mã nguồn và tập trung vào giá trị cốt lõi của sản phẩm.
9	Khả Năng Tái Sử Dụng	Agile-Scrum thúc đẩy việc tái sử dụng mã nguồn và các thành phần đã phát triển, giúp tăng cường hiệu suất và tiết kiệm thời gian trong các dự án phần mềm.
10	Tính Đáng Tin Cậy	Agile-Scrum đảm bảo tính đáng tin cậy của sản phẩm thông qua việc kiểm thử liên tục, kiểm tra và cải thiện liên tục dựa trên phản hồi từ khách hàng và nhóm phát triển.
11	Phản ứng Cộng Đồng	Agile-Scrum thúc đẩy sự hợp tác và phản hồi tích cực từ cộng đồng phát triển, giúp cải thiện sản phẩm và quá trình phát triển.

Waterfall

Spiral

Prototyping

V

RAD

RUP

Scrum-Agile





Trân trọng cảm ơn!

