



Use cases for the Waterfall SDLC model

Introduction to Software Engineering (FPT University)

Authentication & Authorization	Register a course			Change the course has been registered
<div>Sign up</div> <div>Log in Log out</div>	View course list	View course information	Register for a course	<div>Add new courses</div> <div>Drop courses</div>
Using email to reset the password in case Forgot the password	<div>Course will have max of 10 min of 3 students.</div> <div>A course fewer than 3 students will be canceled.</div> <div>Student will be notified when the course be canceled.</div> <div>Show the bill to the student once the registration process is completed</div>	<div>Student will be able to choose 4 course for the coming semester</div> <div>Student will indicate two alternative choices in case cannot be assigned to a primary selection.</div>	<div>Students must be able to access the system during the time to edit</div> <div>If a course added during the registration process, the student must be notified of the change before submitting the schedule for processing.</div>	

Use cases for the Waterfall SDLC model:

- The requirements are precisely documented
- Product definition is stable
- The technologies stack is predefined which makes it not dynamic
- No ambiguous requirements
- The project is short

Use cases for the Iteration model:

- The requirements to the final product are strictly predefined
- Applied to the large-scale projects
- The main task is predefined, but the details may advance with the time

Use cases for the Spiral model

- Customer isn't sure about the requirements
- Major edits are expected during the development cycle

- The projects with mid or high-level risk, where it is important to prevent these risks
- The new product that should be released in a few stages to have enough of clients feedback

Use cases for the V-shaped model:

- For the projects where an accurate product testing is required
- For the small and mid-sized projects, where requirements are strictly predefined
- The engineers of the required qualification, especially testers, are within easy reach.

Use cases for the Agile model:

- The users' needs change dynamically
- Less price for the changes implemented because of the many iterations
- Unlike the Waterfall model, it requires only initial planning to start the project

1. Định nghĩa

Mô hình phát triển phần mềm hay quy trình phát triển phần mềm xác định các pha/ giai đoạn trong xây dựng phần mềm. Có nhiều loại mô hình phát triển phần mềm khác nhau ví dụ như:

- 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)

Sau đây mình sẽ đi vào phân tích chi tiết từng mô hình.

1.1 Mô hình thác nước (Waterfall model)

Mô tả

- Đây được coi như là mô hình phát triển phân mềm đầu tiên được sử dụng.
- Mô hình này áp dụng tuân tự các giai đoạn của phát triển phân 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.

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 phân mềm, xác định kiến trúc hệ thống tổng thể của phân 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 phân mềm. Công việc chính của giai đoạn này là kiểm tra và sửa tất cả những lỗi tìm được sao cho phân 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.

Ứng dụng

Mô hình thường được áp dụng cho các dự án phân 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.

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.

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.

1.2 Mô hình xoắn ốc

Mô tả

- Là mô hình kết hợp giữa các tính năng của mô hình prototyping và mô hình thác nước.
- Mô hình xoắn ốc được ưa chuộng cho các dự án lớn, đặt tiên và phức tạp.
- Mô hình này sử dụng những giai đoạn tương tự như mô hình thác nước, vẽ thứ tự, plan, đánh giá rủi ro, ...

Phân tích mô hình

Các pha trong quy trình phát triển xoắn ốc bao gồm:

- **Objective identification- Thiết lập mục tiêu:** xác định mục tiêu, đối tượng cho từng pha của dự án.
- **Alternate evaluation- Đánh giá và giảm thiểu rủi ro:** đánh giá rủi ro và thực hiện các hành động để giảm thiểu rủi ro.

- **Product development- Phát triển sản phẩm:** Lựa chọn mô hình phù hợp để phát triển hệ thống.
- **Next phase planning- Lập kế hoạch:** đánh giá dự án và lập kế hoạch cho pha tiếp theo.

Ứng dụng

Mô hình này thường được sử dụng cho các ứng dụng lớn và các hệ thống được xây dựng theo các giai đoạn nhỏ hoặc theo các phân đoạn.

Ưu điểm

- Tốt cho các hệ phân mềm quy mô lớn.
- Dễ kiểm soát các mạo hiểm ở từng mức tiên hóa.
- Đánh giá thực tế hơn như là một quy trình làm việc, bởi vì những vấn đề quan trọng đã được phát hiện sớm hơn.

Nhược điểm

- Manager cần có kỹ năng tốt để quản lý dự án, đánh giá rủi ro kịp thời.
- Chi phí cao và mất nhiều thời gian để hoàn thành dự án.
- Phức tạp và không thích hợp với các dự án nhỏ và ít rủi ro.
- Yêu cầu thay đổi thường xuyên dẫn đến lặp vô hạn.
- Chưa được dùng rộng rãi.

1.3 Mô hình Agile

Agile là một phương pháp phát triển phần mềm linh hoạt để làm sao đưa sản phẩm đến tay người dùng càng nhanh càng tốt và được xem như là sự cải tiến so với những mô hình cũ như mô hình “Thác nước (waterfall)” hay “CMMI”. Phương thức phát triển phần mềm Agile là một tập hợp các phương thức phát triển lặp và tăng dần trong đó các yêu cầu và giải pháp được phát triển thông qua sự liên kết cộng tác giữa các nhóm tự quản và liên chức năng.

Mô tả

- Dựa trên mô hình iterative and incremental.
- Các yêu cầu và giải pháp phát triển dựa trên sự kết hợp của các function.
- Trong Agile, các tác vụ được chia thành các khung thời gian nhỏ để cung cấp các tính năng cụ thể cho bản phát hành cuối.

Ứng dụng

- Có thể được sử dụng với bất kỳ loại hình dự án nào, nhưng cần sự tham gia và tính tương tác của khách hàng.
- Sử dụng khi khách hàng yêu cầu chức năng sẵn sàng trong khoảng thời gian ngắn.

Ưu điểm

- Tăng cường tinh thần làm việc nhóm và trao đổi công việc hiệu quả.
- Các chức năng được xây dựng nhanh chóng và rõ ràng, dễ quản lý.
- Dễ dàng bổ sung, thay đổi yêu cầu.
- Quy tắc tối thiểu, tài liệu dễ hiểu, dễ sử dụng.

Nhược điểm

Mô hình Agile được sử dụng rộng rãi trên thế giới nhưng cũng không đồng nghĩa với phù hợp với tất cả các dự án phần mềm.

- Không thích hợp để xử lý các phụ thuộc phức tạp.
- Có nhiều rủi ro về tính bền vững, khả năng bảo trì và khả năng mở rộng.
- Cần một team có kinh nghiệm.
- Phụ thuộc rất nhiều vào sự tương tác rõ ràng của khách hàng.
- Chuyển giao công nghệ cho các thành viên mới trong nhóm có thể khá khó khăn do thiếu tài liệu.

1.4 Mô hình tiếp cận lặp

Mô tả

- Một mô hình được lập đi lập lại từ khi start cho đến khi làm đầy đủ spec. Quá trình này sau đó được lặp lại, tạo ra một phiên bản mới của phân mềm vào cuối mỗi lần lặp của mô hình.
- Thay vì phát triển phân mềm từ spec đặc tả rồi mới bắt đầu thực thi thì mô hình này có thể review dần dần để đi đến yêu cầu cuối cùng.

Ứng dụng

- Yêu cầu chính phải được xác định; tuy nhiên, một số chức năng hoặc yêu cầu cải tiến có thể phát triển theo thời gian.
- Một công nghệ mới đang được sử dụng và đang được học tập bởi nhóm phát triển trong khi làm việc trong dự án.
- Phù hợp cho các dự án lớn và nhiệm vụ quan trọng.

Ưu điểm

- Xây dựng và hoàn thiện các bước sản phẩm theo từng bước.
- Thời gian làm tài liệu sẽ ít hơn so với thời gian thiết kế.
- Một số chức năng làm việc có thể được phát triển nhanh chóng và sớm trong vòng đời.
- Ít tốn kém hơn khi thay đổi phạm vi, yêu cầu.
- Dễ quản lý rủi ro.
- Trong suốt vòng đời, phân mềm được sản xuất sớm để tạo điều kiện cho khách hàng đánh giá và phản hồi.

Nhược điểm

- Yêu cầu tài nguyên nhiều.
- Các vấn đề về thiết kế hoặc kiến trúc hệ thống có thể phát sinh bất cứ lúc nào.

- Yêu cầu quản lý phức tạp hơn.
- Tiên độ của dự án phụ thuộc nhiều vào giai đoạn phân tích rủi ro.

1.5 Mô hình tăng trưởng

Mô tả

- Spec được chia thành nhiều phân.
- Chu kỳ được chia thành các module nhỏ, dễ quản lý.
- Mỗi module sẽ đi qua các yêu cầu về thiết kế, thực hiện, ... như 1 vòng đời phát triển thông thường.

Ứng dụng

- Áp dụng cho những dự án có yêu cầu đã được mô tả, định nghĩa và hiểu một cách rõ ràng.
- Khách hàng có nhu cầu về sản phẩm sớm.

Ưu điểm

- Phát triển nhanh chóng.
- Mô hình này linh hoạt hơn, ít tốn kém hơn khi thay đổi phạm vi và yêu cầu.
- Dễ dàng hơn trong việc kiểm tra và sửa lỗi.

Nhược điểm

- Cân lập plan và thiết kế tốt.
- Tổng chi phí là cao hơn so với mô hình thác nước.

1.6 Mô hình chữ V(V model)

Mô tả

- Mô hình chữ V là một phân mở rộng của mô hình thác nước và được dựa trên sự kết hợp của một giai đoạn thử nghiệm cho từng giai đoạn phát triển tương ứng. Đây là một mô hình có tính kỷ luật cao và giai đoạn tiếp theo chỉ bắt đầu sau khi hoàn thành giai đoạn trước.
- Với V model thì công việc test được tham gia ngay từ đầu.

Ứng dụng

- Yêu cầu được xác định rõ ràng.
- Xác định sản phẩm ổn định.
- Công nghệ không thay đổi và được hiểu rõ bởi nhóm dự án.
- Không có yêu cầu không rõ ràng hoặc không xác định.
- Dự án ngắn.

Ưu điểm

- Đây là một mô hình có tính kỷ luật cao và các giai đoạn được hoàn thành cùng một lúc.
- Hoạt động tốt cho các dự án nhỏ, khi các yêu cầu được hiểu rất rõ.
- Đơn giản và dễ hiểu và dễ sử dụng, dễ quản lý.

Nhược điểm

- Khó quản lý kiểm soát rủi ro, rủi ro cao.
- 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 thích hợp cho các dự án có nguy cơ thay đổi yêu cầu trung bình đến cao.

1.7 Mô hình Scrum

Mô tả

- Chia các yêu cầu ra làm theo từng giai đoạn. Mỗi 1 giai đoạn(sprint) chỉ làm 1 số lượng yêu cầu nhất định.
- Mỗi một sprint thường kéo dài từ 1 tuần đến 4 tuần (ko dài hơn 1 tháng).
- Đầu sprint sẽ lên kế hoạch làm những yêu cầu nào. Sau đó, sẽ thực hiện code và test. Cuối sprint là 1 sản phẩm hoàn thiện cả code lẫn test có thể demo và chạy được.
- Hoàn thành sprint 1, tiếp tục làm sprint 2, sprint... cho đến khi hoàn thành hết các yêu cầu.
- Trong mỗi 1 sprint thì sẽ có họp hàng ngày – daily meeting từ 15 – 20 phút. Mỗi thành viên sẽ báo cáo: Hôm qua tôi đã làm gì? Hôm nay tôi sẽ làm gì? Có gặp khó khăn gì không?
- Scrum là mô hình hướng khách hàng (Customer oriented).

Các nhân tố tạo nên quy trình Scrum

Có 3 thành tố quan trọng cấu thành nên SCRUM:

- **Tổ chức (Organization)**
 - Tổ chức nhóm dự án và Roles: Vài trò.
 - Product Owner: Người sở hữu sản phẩm.
 - ScrumMaster: Người điều phối.
 - Development Team: Nhóm phát triển.
- **Tài liệu (Artifacts):** đó chính là các kết quả đầu ra.
 - Product Backlog: Danh sách các chức năng cần phát triển của sản phẩm.
 - Sprint Backlog: Danh sách các chức năng cần phát triển cho mỗi giai đoạn.
 - Estimation: Kết quả ước lượng của team.
- **Quy trình(Process):** Quy định cách thức vận hành của SCRUM.
 - Sprint Planning meeting: Hoạch định cho mỗi giai đoạn.
 - Review: Tổng kết cho mỗi giai đoạn.

- Daily Scrum Meeting: Review hàng ngày.

Tổ chức dự án

- **Product Owner**

- Product Owner là người sở hữu sản phẩm, người quyết định sản phẩm có những chức năng nào và là người quyết định Product Backlog.
- Thông thường Role này được khách hàng hoặc người đại diện cho khách hàng đảm nhận.

- **ScrumMaster**

- Scrum Master là người đảm bảo các qui trình của Scrum được thực hiện đúng và thuận lợi.

- **Development Team**

- Một nhóm từ 4-7 kỹ sư phần mềm chịu trách nhiệm phát triển sản phẩm.
- Nhóm dự án phải làm việc với Product Owner để quyết định những gì sẽ làm trong Sprint (giai đoạn) này và kết quả sẽ ra sao.
- Thảo luận để đưa ra các giải pháp, ước lượng thời gian thực hiện công việc, họp đánh giá kết quả công việc.

- **Product Backlog**

- Product Backlog là danh sách các chức năng cần được phát triển của sản phẩm.
- Danh sách này do Product Owner quyết định.
- Thường xuyên được cập nhật để đáp ứng được nhu cầu thay đổi của khách hàng và dự án.

Ưu điểm

- Một người có thể thực hiện nhiều việc ví dụ như dev có thể test.
- Phát hiện lỗi sớm.

- Có khả năng áp dụng được cho những dự án mà yêu cầu khách hàng không rõ ràng ngay từ đầu.

Nhược điểm

- Trình độ của nhóm cần có một kỹ năng nhất định.
- Phải có sự hiểu biết về mô hình agile.
- Khó khăn trong việc xác định ngân sách và thời gian.
- Luôn nghe ý kiến phản hồi từ khách hàng và thay đổi theo nên thời gian sẽ kéo dài.
- Vai trò của PO rất quan trọng, PO là người định hướng sản phẩm. Nếu PO làm không tốt sẽ ảnh hưởng đến kết quả chung.

1.8 Mô hình RAD

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.

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.

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.

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.

3. **Sắp xếp các nhiệm vụ và nhiệm vụ con theo trình tự:** Sắp xếp các nhiệm vụ nhỏ hơn và nhiệm vụ phụ tuần tự bên dưới các hoạt động của chúng theo thứ tự từ trái sang phải (hoặc bất kỳ cách nào bạn muốn đọc sơ đồ của mình). Điều này có thể hiển thị thứ tự mà khách hàng của bạn hoàn thành các hoạt động và nhiệm vụ.



4. **Ưu tiên các nhiệm vụ và nhiệm vụ phụ:** Cho biết mức độ ưu tiên của các nhiệm vụ / nhiệm vụ phụ bằng cách sắp xếp chúng theo chiều dọc - các nhiệm vụ quan trọng hơn sẽ cao hơn trong bản đồ câu chuyện.



