

Quản lý dự án

1. Tại sao cần quản lý dự án:

- Phát triển phần mềm hiện đại làm theo teamworks.
- Các dự án phần mềm đòi hỏi nhiều nguồn nhân lực với chuyên môn khác nhau.
- Đảm bảo tích chuyên nghiệp trong phát triển dự án phần mềm:
 - o Bảo đảm lịch trình của dự án.
 - o Điều phối và khai thác tối đa nguồn nhân lực hiện có.
 - o Bảo đảm chất lượng sản phẩm.
 - o Khả năng khắc phục các sự cố xảy ra khách quan.
- Dự án càng lớn càng cần có sự quản lý chặt chẽ và đồng bộ.

2. Quản lý dự án:

- Dự đoán quy mô và độ phức tạp của dự án.
- Xác định các team cần thiết cho hiện thực dự án.
- Xác định kế hoạch dự đoán thời gian hoàn thành dự án.
- Xác định các tài nguyên cần thiết cho dự án bao gồm phần mềm, hệ thống...
- Tính toán chi phí xây dựng dự án.
- Xây dựng lộ trình thực hiện dự án (milestone).
- Thực hiện các công việc quản lý trong thời gian thực hiện dự án để bảo đảm đúng kế hoạch đã đề ra.

3. Các công việc của quản lý dự án:

- Quản trị nhân sự: điều phối, quản lý công việc...
- Phân bổ các tài nguyên của hệ thống theo kế hoạch.
- Điều phối nhân sự: trong công ty và bên ngoài.
- Xử lý các phát sinh và thời gian biểu.
- Quản lý các thay đổi yêu cầu của dự án.
- Giải quyết các sự cố ngoài kế hoạch: máy móc hư hỏng, nhân sự thay đổi...
- Báo cáo cho lãnh đạo về hiện trạng của dự án.
- Giao tiếp với khách hàng.
- Huấn luyện nhân sự.

4. Ước lượng các dự án phần mềm:

- Dự đoán quy mô của sản phẩm sẽ được phát triển.
- Các yêu cầu, resource cần thiết để có thể phát triển sản phẩm thành công.
- Vấn đề ước lượng rất khó, nó ảnh hưởng rất lớn đến uy tín, thời gian, tiền bạc của công ty.
- Có rất nhiều phương thức để tính toán cho vấn đề ước lượng:

- Các gói phần mềm hỗ trợ ước lượng.
- Lấy số liệu từ những dự án có công việc và công nghệ tương tự.
- Ước lượng từ kinh nghiệm đã qua của những người phát triển.
- Nếu một dự án ước lượng nhiều thời gian hơn lượng thời gian sẵn có thì giải pháp duy nhất là giảm kích thước hệ thống hoặc tăng chi phí bằng cách bổ sung thêm người.

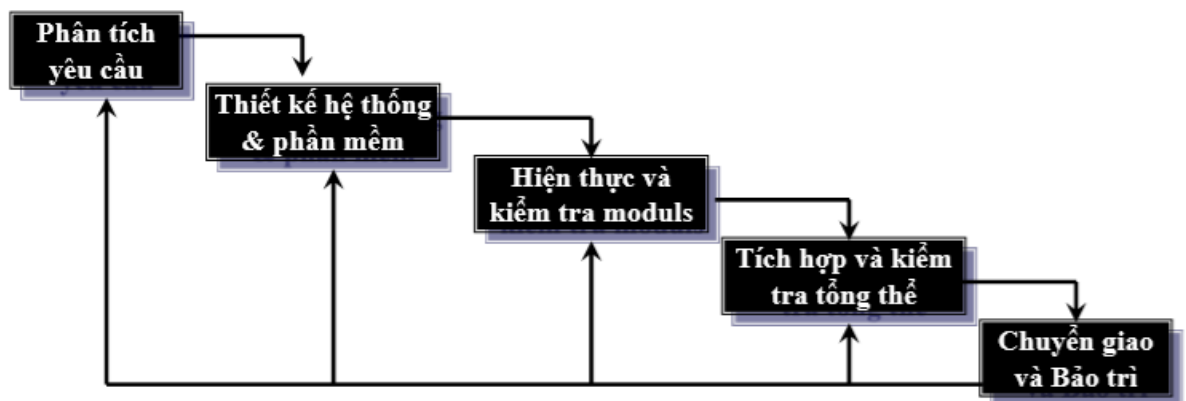
5. Lập lịch dự án:

- Ước lượng thời gian có thể thực hiện bằng công cụ
 - Bao gồm:
 - Xác định các mốc tiến trình thực hiện dự án.
 - Phân bổ resource cho các tiến trình của dự án.
 - Quản trị và thực hiện các điều chỉnh cần thiết cho các tiến trình khi có sự thay đổi ngoài kế hoạch.
 - Phân bổ nguồn resource (thêm người).
 - Phân bổ lại các cột mốc của từng tiến trình.
- ⇒ Mục tiêu là đảm bảo deadline không bị thay đổi.

Các mô hình thiết kế phần mềm

1. Waterfall mode (mô hình thác nước).

- Mô hình phát triển phần mềm đầu tiên.
- Các công việc tiếp nối nhau một cách tuần tự.
- Đặt nền móng cho các phương pháp phân tích, thiết kế, kiểm tra...
- Đầu ra của giai đoạn trước là đầu vào của giai đoạn sau, giai đoạn sau chỉ thực hiện khi giai đoạn trước kết thúc.



- **Ứng dụng:**
 - Các dự án nhỏ ngắn hạn.
 - Các dự án 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.
 - Không thích hợp với những dự án dài, phức tạp.
 - Bản chất của phát triển phần mềm là quá trình lặp đi lặp lại chứ k phải tuần tự.
 - Các bước thực chất không tách biệt hoàn toàn mà có chồng lấn và tham khảo lại.
 - Bắt buộc khách hàng đặc tả tất cả yêu cầu một cách chính xác và đầy đủ ngay từ ban đầu.
 - Khách hàng thường phải chờ đợi rất lâu để thấy được phiên bản đầu tiên của sản phẩm.
 - Tồn tại “delay” tích lũy trong nhóm làm việc -> dự án thường bị trễ.

2. Spiral model (mô hình xoắn ốc).

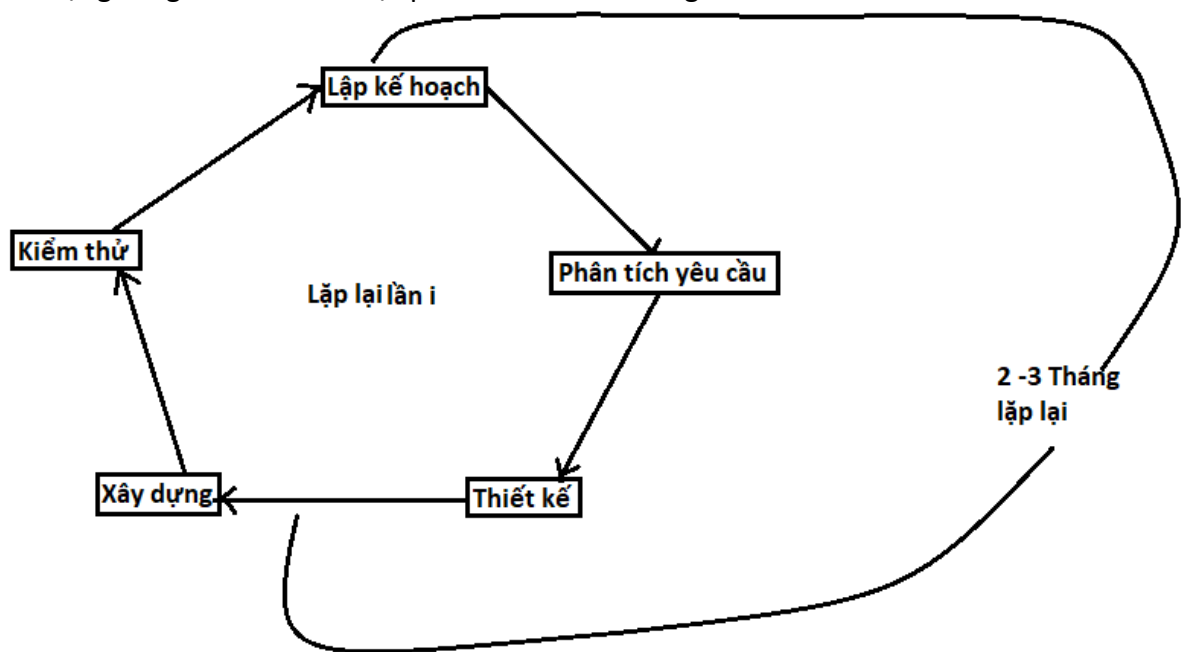


- Là mô hình kết hợp giữa các tính năng của mô hình prototyping và mô hình thác nước.
- Mô hình xoắn ốc được ưu chuộng cho các dự án lớn, đắt tiền và phức tạp.
- Mô hình này sử dụng những giai đoạn tương tự như mô hình thác nước.

- **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 phần mềm quy mô lớn.
 - Dễ kiểm soát các mạo hiểm ở từng mức tiến hóa.
 - Đánh giá thực tế hơn như là một quy trình làm việc, bởi vì những vấn đề quan trọng đã được thực 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 thường xuyên thay đổi dẫn đến lặp vô hạn.
 - Chưa được dùng rộng rãi.

3. Mô hình Agile.

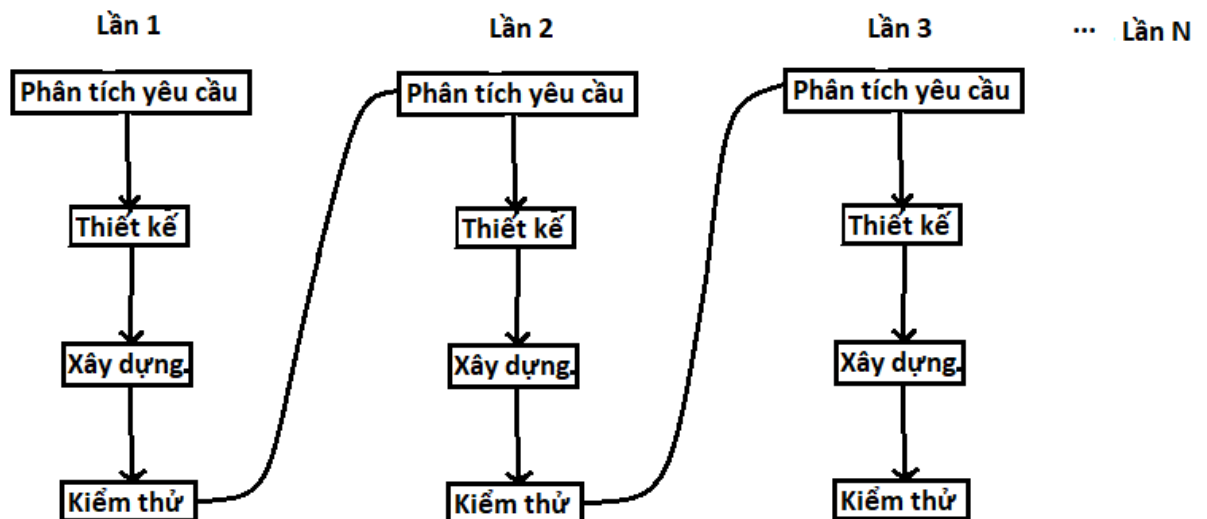
- Agile là một phương pháp phát triển phần mềm linh hoạt để làm sao đưa sản phẩm đến tay người dùng càng nhanh càng tốt và được xem như là sự cải tiến so với những mô hình cũ như mô hình thác nước.
- Phương thức phát triển phần mềm Agile là một tập hợp các phương thức phát triển lặp và tăng dần trong 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.



- **Ứng dụng:**
 - Có thể sử dụng với bất kỳ loại 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 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:**
 - 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 và 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.

4. Iterative model (mô hình lặp).

- Một mô hình lặp đi lặp lại từ khi start cho đến khi làm đầy đủ giai đoạn. Quá trình này sau đó được lặp lại, tạo ra một phiên bản mới của phần mềm vào cuối mỗi lần lặp mô hình.
- Thay vì phát triển phần mềm từ giai đoạn 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 cho các dự án:**
 - 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 giai đoạn.
 - 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, phần 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.
 - Tiến độ của dự án phụ thuộc nhiều vào giai đoạn phân tích rủi ro.

5. V model.

- V model là một phần 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 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 cho các dự án:**
 - 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 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.

6. Scrum.

- Chia các yêu cầu ra làm theo từng giai đoạn. Mỗi giai đoạn chỉ làm một số lượng yêu cầu nhất định.
- Mỗi giai đoạn thường kéo dài 1 – 4 tuần.
- Đầu giai đoạn sẽ lên kế hoạch làm những yêu cầu nào. Sau đó sẽ thực hiện code và test. Cuối giai đoạn là 1 sản phẩm hoàn thiện cả code lẫn test có thể demo và chạy được.
- Hoàn thành giai đoạn 1, tiếp tục giai đoạn 2... cho đến khi hoàn thành hết các yêu cầu.
- Trong mỗi giai đoạn sẽ có họp hằng ngày từ 15 – 20 phút để báo cáo tiến độ và khó khăn.
- Scrum là mô hình hướng khách hàng.
- **Các yếu tố quan trọng tạo nên scrum:**
 - Tổ chức: tổ chức dự án và vai trò, khách hàng, người điều phối, nhóm phát triển...
 - Tài liệu: kết quả đầu ra, danh sách chức năng cần phát triển của hệ thống và của từng giai đoạn, kết quả..
 - Quy trình: quy định cách vận hành của scrum, hoạch định, tổng kết mỗi giai đoạn, review hằng ngày.
- **Ưu điểm:**
 - Một người có thể thực hiện nhiều việc.
 - 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 Product Owner (người sở hữu sản phẩm, người quyết định sản phẩm có những vai trò nào thường là khách hàng hoặc người đại diện khách hàng) rất quan trọng, nếu Product Owner làm không tốt sẽ ảnh hưởng đến kết quả chung.

7. Mô hình Prototype.

- Prototype như là một cơ chế để nhận diện chính xác yêu cầu của khách hàng.
- Bản thân khách hàng chưa hiểu rõ yêu cầu của mình, cũng như các quy trình chưa được xác lập rõ ràng.
- Khách hàng chưa hiểu rõ khả năng hỗ trợ của hệ thống máy tính.

- **Ứng dụng cho các dự án:**

- Dùng cho hệ thống nhỏ. Có chi phí khi thay đổi hệ thống không quá lớn khi cần phải thay đổi sau khi thực hiện prototype.
- Cần sự cấp bách về thời gian triển khai gấp. Hệ thống cần được đưa vào ứng dụng từng phần trong khoảng thời gian nhất định.
- Trong trường hợp những hệ thống mà việc đặc tả các yêu cầu là rất khó và không rõ ràng ngay từ đầu.

- **Ưu điểm:**

- Kích thích sự thích thú của người dùng đối với dự án

- **Nhược điểm:**

- Prototype có thể bị “throw-away” -> lãng phí.
- Các process không được phân định rõ ràng.
- Hệ thống thông thường có cấu trúc lồng lèo.
- Cần có kỹ năng đặc biệt trong quản lý và phát triển.
- Khách hàng hối thúc nhà phát triển hoàn thành sản phẩm một khi thấy được các prototype đầu tiên.

8. Incremental model (mô hình tăng trưởng).

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

- **Ứng dụng:**
 - Áp dụng cho những dự án có yêu cầu 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:**
 - Cần lập kế hoạch và thiết kế tốt.
 - Tổng chi phí cao hơn so với mô hình thác nước.

9. Mô hình RAD (mô hình phát triển ứng dụng nhanh).



- 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 module 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 cho các dự án:**
 - Module hóa rõ ràng. Nếu dự án không thể chia thành các module RAD có thể không thành công.
 - RAD nên được sử dụng khi có nhu cầu 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 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.