



FPT POLYTECHNIC

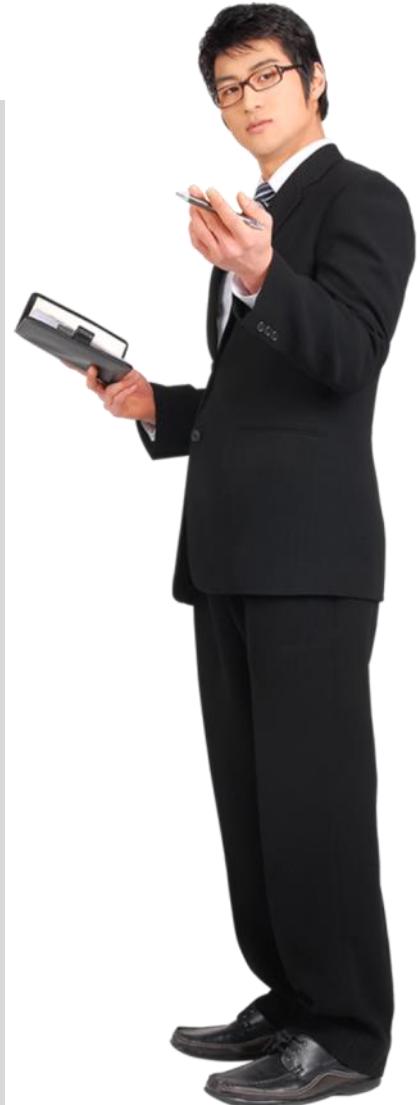


KIỂM THỬ CƠ BẢN(P1)

BÀI 2: KIỂM THỬ TRONG VÒNG ĐỜI PHÁT TRIỂN PHẦN MỀM

Nội dung bài học

- Mô hình phát triển tuần tự
- Mô hình phát triển lặp lại-gia tăng
- Mô hình Scrum





MÔ HÌNH PHÁT TRIỂN PHẦN MỀM

MÔ HÌNH PHÁT TRIỂN TUẦN TỰ

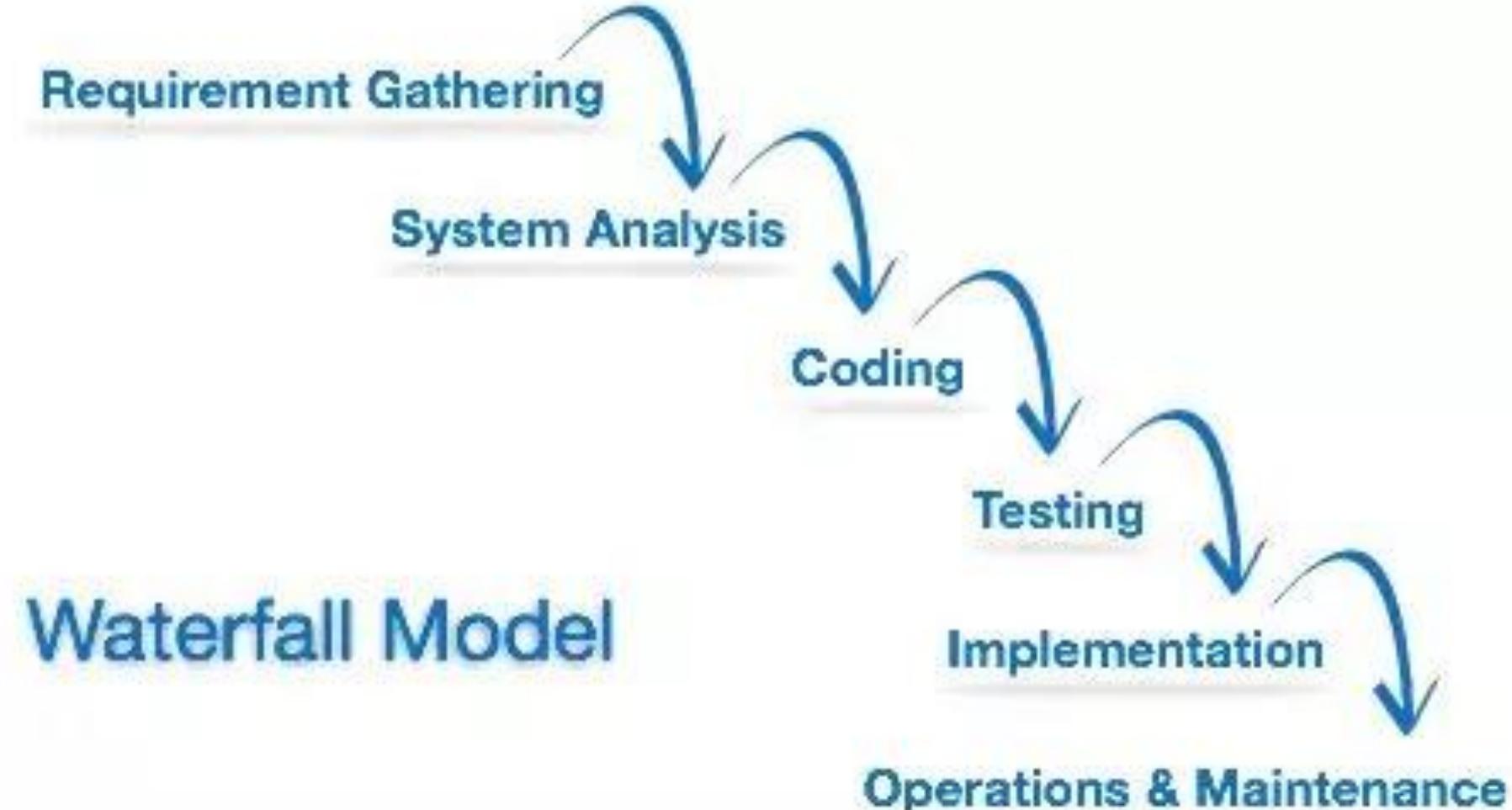
❑ Mô hình phát triển tuần tự

- ❖ Mô hình được phát triển theo tuyến của nhiều giai đoạn rời rạc và kế tiếp nhau, mà không có sự chồng chéo.
- ❖ Có 2 loại mô hình tiêu biểu đại diện cho mô hình phát triển tuần tự : Mô hình thác nước , Mô hình chữ V.

❑ Mô hình phát triển thác nước

- ❖ Trong mô hình này, hoạt động phát triển phần mềm được chia thành các giai đoạn khác nhau và từng giai đoạn bao gồm hàng loạt các nhiệm vụ và có các mục tiêu khác nhau.
- ❖ Trong mô hình thác nước, sự phát triển của một pha chỉ bắt đầu khi giai đoạn trước hoàn thành. Do tính chất này, mỗi giai đoạn của mô hình thác nước phải được xác định khá chính xác. Các giai đoạn chuyển từ mức cao xuống mức thấp hơn, giống như một thác nước nên mô hình này được đặt tên là mô hình thác nước.

☐ Sơ đồ mô hình thác nước



□ Phân tích mô hình thác nước

- ❖ 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.

❑ Ứng dụng của mô hình phát triển thác nước

- ❖ 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 của mô hình phát triển thác nước

- ❖ 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 của mô hình phát triển thác nước

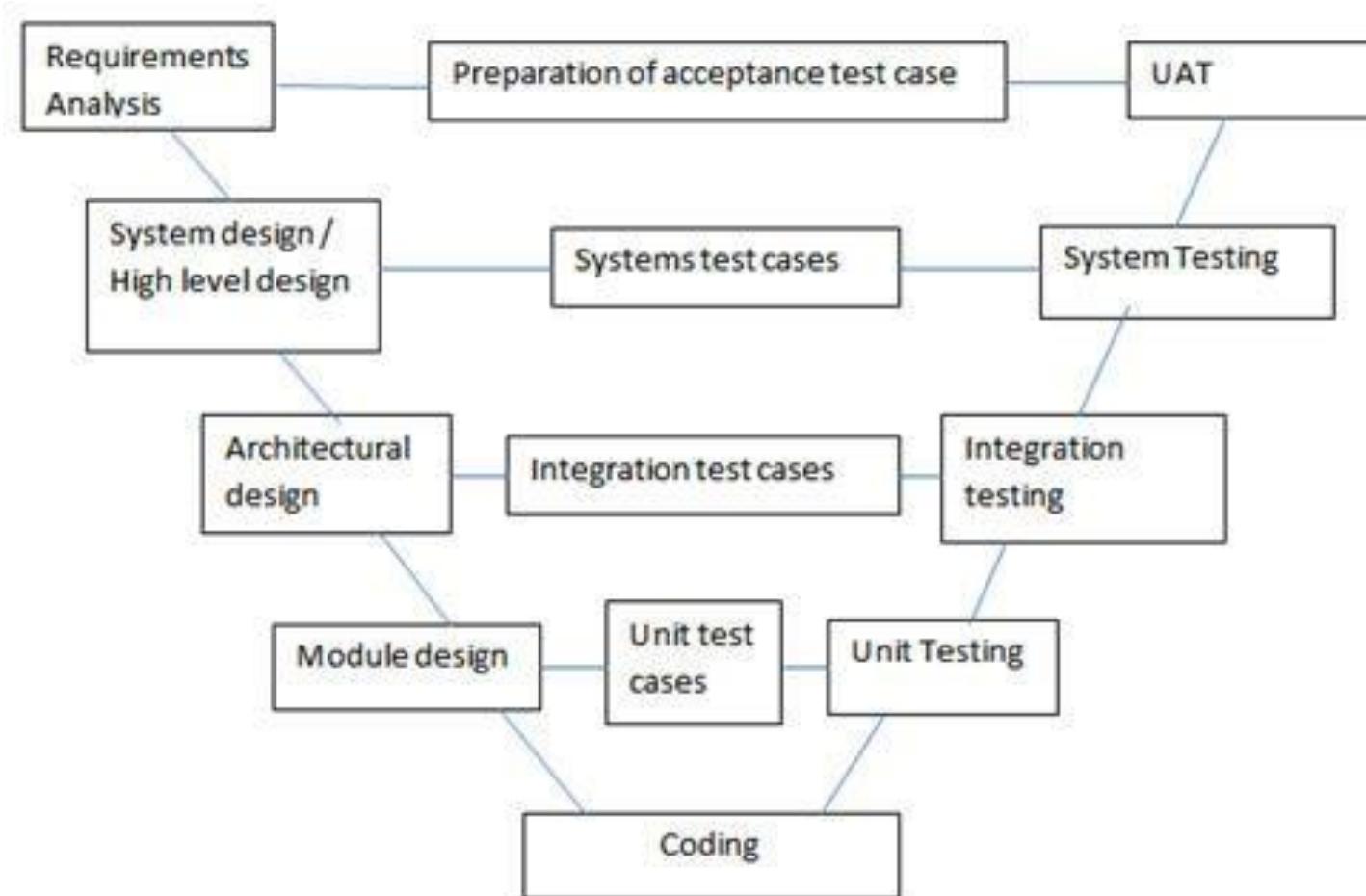
- ❖ Ít linh hoạt, phạm vi điều chỉnh hạn chế.
- ❖ Rất khó để đo lường sự phát triển trong từng giai đoạn.
- ❖ Mô hình không thích hợp với những dự án dài, đang diễn ra, hay những dự án phức tạp, có nhiều thay đổi về yêu cầu trong vòng đời phát triển.
- ❖ Khó quay lại khi giai đoạn nào đó đã kết thúc.

MÔ HÌNH PHÁT TRIỂN CHỮ V

❑ Mô hình chữ V

- ❖ Một trong những hạn chế lớn nhất của mô hình phát triển phần mềm tuần tự là: Các khuyết điểm khuyết được tìm thấy rất chậm trong quá trình phát triển vì kiểm thử được thực hiện vào cuối chu kỳ phát triển. Để khắc phục vấn đề này, một mô hình phát triển mới được giới thiệu gọi là "Mô hình V".
- ❖ Trong mô hình V, các hoạt động phát triển và đảm bảo chất lượng được thực hiện đồng thời. Không có pha rời rạc và kiểm thử được bắt đầu ngay từ giai đoạn lấy yêu cầu. Các hoạt động xác minh và xác nhận đi liền với nhau. Để hiểu mô hình V, chúng ta hãy nhìn vào hình dưới đây:

Sơ đồ mô hình chữ V



□ Phân tích bên trái mô hình chữ V

❖ Phân tích yêu cầu / Requirement Analysis: Trong giai đoạn này các yêu cầu được thu thập, phân tích và nghiên cứu. Ở giai đoạn này việc hệ thống chạy như thế nào không quan trọng, quan trọng là hệ thống có những chức năng gì. Thảo luận, đánh giá và xem xét cần được thực hiện để có mục tiêu rõ ràng.

- Hoạt động xác minh: Đánh giá yêu cầu (Requirements review).
- Hoạt động xác nhận: Tạo test case UAT (User acceptance test- kiểm thử chấp nhận) = Đầu ra cần có:

❖ Yêu cầu hệ thống / High level design: Trong giai đoạn này, high level design của phần mềm được xây dựng. Nhóm sẽ nghiên cứu và điều tra về các yêu cầu có thể được thực hiện như thế nào. Tính khả thi về mặt kỹ thuật của yêu cầu cũng được tìm hiểu. Nhóm cũng tìm hiểu về các mô-đun sẽ được tạo / phụ thuộc, nhu cầu phần cứng / phần mềm.

- Hoạt động xác minh: Đánh giá thiết kế (Design reviews)
- Hoạt động xác nhận: Tạo test plan và test case, tạo ma trận truy vết (traceability metric)
- Đầu ra cần có: System test cases, Feasibility reports, System test plan, tài liệu về yêu cầu phần cứng....

❖ Thiết kế kiến trúc/ Architectural Design: Trong giai đoạn này, dựa trên thiết kế mức cao, kiến trúc phần mềm được tạo ra. Các mô-đun, mối quan hệ và sự phụ thuộc của họ, sơ đồ kiến trúc, bảng cơ sở dữ liệu, chi tiết về công nghệ đều được hoàn tất trong giai đoạn này.

- Hoạt động xác minh: Đánh giá thiết kế
- Hoạt động xác nhận: Kế hoạch thử nghiệm tích hợp và các trường hợp thử nghiệm.
- Đầu ra cần có: Tài liệu thiết kế, Kế hoạch kiểm thử tích hợp và các trường hợp thử nghiệm, Thiết kế bảng cơ sở dữ liệu,...

❖ Thiết kế mô đun/ Module Design: Trong giai đoạn này mỗi mô-đun hoặc các thành phần phần mềm đều được thiết kế riêng. Các method, class, giao diện, các kiểu dữ liệu vv đều được hoàn tất trong giai đoạn này.

- Hoạt động xác minh: Đánh giá thiết kế
- Hoạt động xác nhận: Tạo và xem xét các trường hợp kiểm tra đơn vị.
- Đầu ra cần có: Các đơn vị kiểm tra đơn vị

- ❖ Code: Trong giai đoạn này cần thực hiện.
 - Hoạt động xác minh: Xem xét mã, kiểm tra các trường hợp kiểm tra
 - Hoạt động xác nhận: Tạo các trường hợp kiểm tra chức năng.
 - Đầu ra cần có: các trường hợp thử nghiệm, danh sách kiểm tra xem lại.

❑ Phân tích bên phải mô hình chữ V

- ❖ Unit Testing: Trong giai đoạn này tất cả unit test case đã được tạo ra trong giai đoạn thiết kế cấp thấp sẽ được thực hiện.
- ❖ Integration Testing: Trong giai đoạn này các trường hợp kiểm thử tích hợp được thực hiện đã được tạo ra trong giai đoạn thiết kế kiến trúc. Trong trường hợp có bất kỳ dị thường, bug được ghi lại và theo dõi.

❖ System Testing: Trong giai đoạn này, kiểm thử chức năng và phi chức năng được thực hiện. Nói cách khác, việc kiểm tra thực tế hoạt động của ứng dụng diễn ra ở đây. Lỗi được phát hiện và theo dõi để sửa. Báo cáo tiến độ cũng là một phần quan trọng trong giai đoạn này. Ma trận truy vết (traceability metric) được cập nhật để kiểm tra mức độ bao phủ và rủi ro được giảm bớt.

❖ User Acceptance Testing: Thử nghiệm chấp nhận về cơ bản liên quan đến việc kiểm tra các yêu cầu business. Ở đây kiểm tra được thực hiện để xác nhận rằng các yêu cầu kinh doanh được đáp ứng trong môi trường người dùng. Thử nghiệm khả năng tương thích và đôi khi thử nghiệm phi chức năng (Load, Stress và Volume) cũng được thực hiện trong giai đoạn này.

❑ Ứng dụng của mô hình phát triển chữ V

- ❖ Yêu cầu được xác định rõ ràng và không mơ hồ
- ❖ Tiêu chí chấp nhận được xác định rõ ràng.
- ❖ Dự án có quy mô vừa và nhỏ.
- ❖ Công nghệ và công cụ được sử dụng không thường xuyên thay đổi.

❑ Ưu điểm của mô hình phát triển chữ V

- ❖ Quá trình phát triển và quy trình quản lý có tính tổ chức và hệ thống
- ❖ Hoạt động tốt cho các dự án có quy mô vừa và nhỏ.
- ❖ Kiểm tra bắt đầu từ khi bắt đầu phát triển vì vậy sự mơ hồ được xác định ngay từ đầu.
- ❖ Dễ dàng quản lý vì mỗi giai đoạn có các mục tiêu và mục tiêu được xác định rõ ràng.

❑ Nhược điểm của mô hình phát triển chữ V

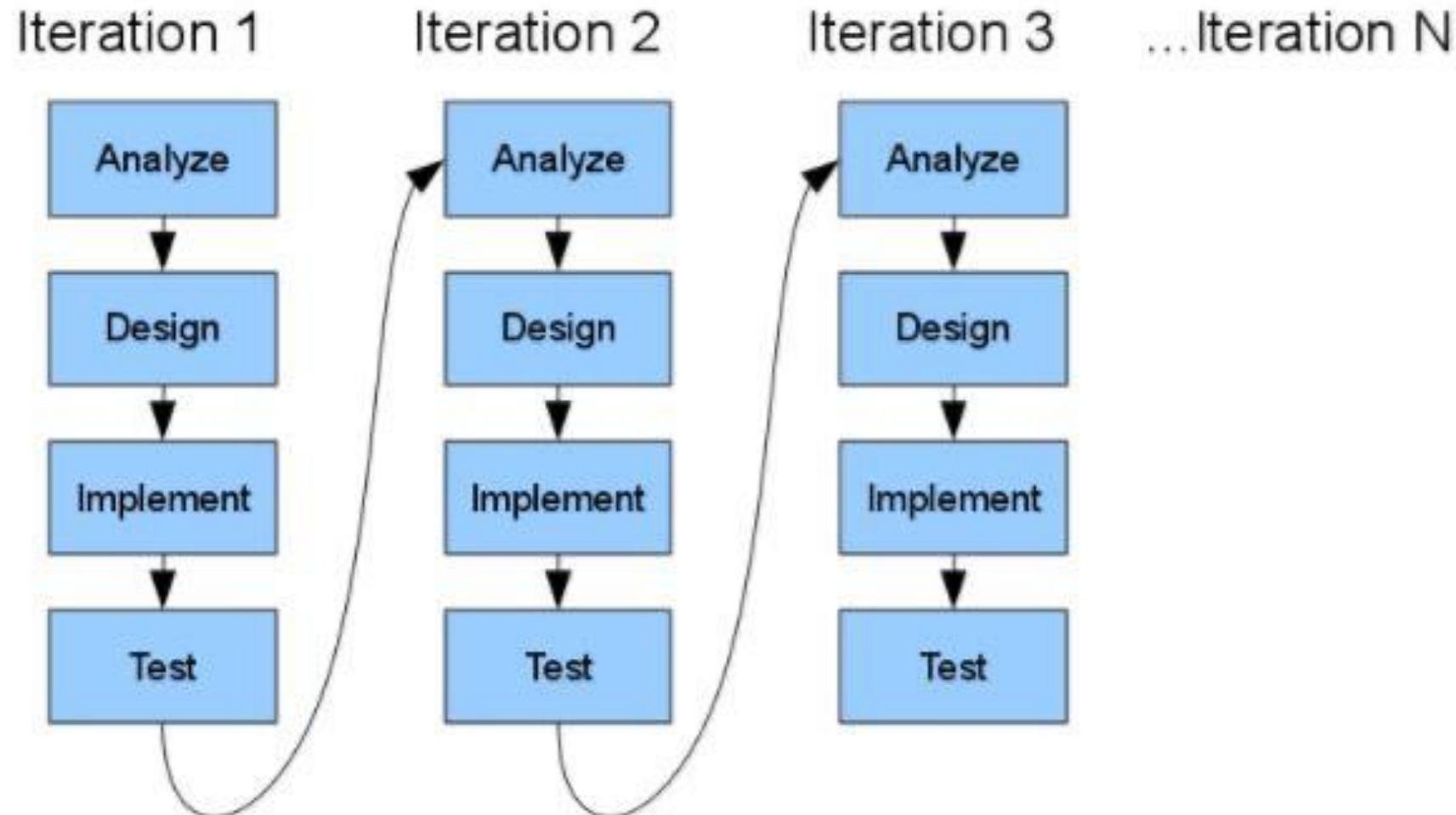
- ❖ Không thích hợp cho các dự án lớn và phức tạp
- ❖ Không phù hợp nếu các yêu cầu thường xuyên thay đổi.
- ❖ Không có phần mềm làm việc được sản xuất ở giai đoạn trung gian.
- ❖ Không có điều khoản cho việc phân tích rủi ro nên có sự không chắc chắn và có tính rủi ro.

MÔ HÌNH PHÁT TRIỂN LẶP

❑ Mô hình phát triển lắp

- ❖ Mô hình được lắp đi lắp lại từ khi start cho đến khi đầy đủ Spec.
- ❖ Quá trình được lắp lại, tạo ra một phiên bản mới của phần mềm vào cuối mỗi lần lắp của mô hình.
- ❖ Thay vì phát triển phần 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.

☐ Sơ đồ mô hình phát triển lắp



❑ Phân tích mô hình phát triển lắp

- ❖ Mỗi một Iteration là một quy trình tuần tự được chia nhỏ. Quá trình này sau đó được lắp lại, tạo ra một phiên bản mới của phần mềm vào cuối mỗi lần lắp của mô hình.
- ❖ Thay vì đợi toàn bộ đặc tả yêu cầu rồi phát triển phần mềm thì mô hình này có thể thực hiện từng Iteration mỗi Iteration có tất cả các giai đoạn của mô hình thác nước.

❑ Ứng dụng của mô hình phát triển lắp

- ❖ 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 của mô hình phát triển lắp

- ❖ Xây dựng và hoàn thiện các bước sản phẩm theo từng giai đoạn lắp.
- ❖ 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 của mô hình phát triển lắp

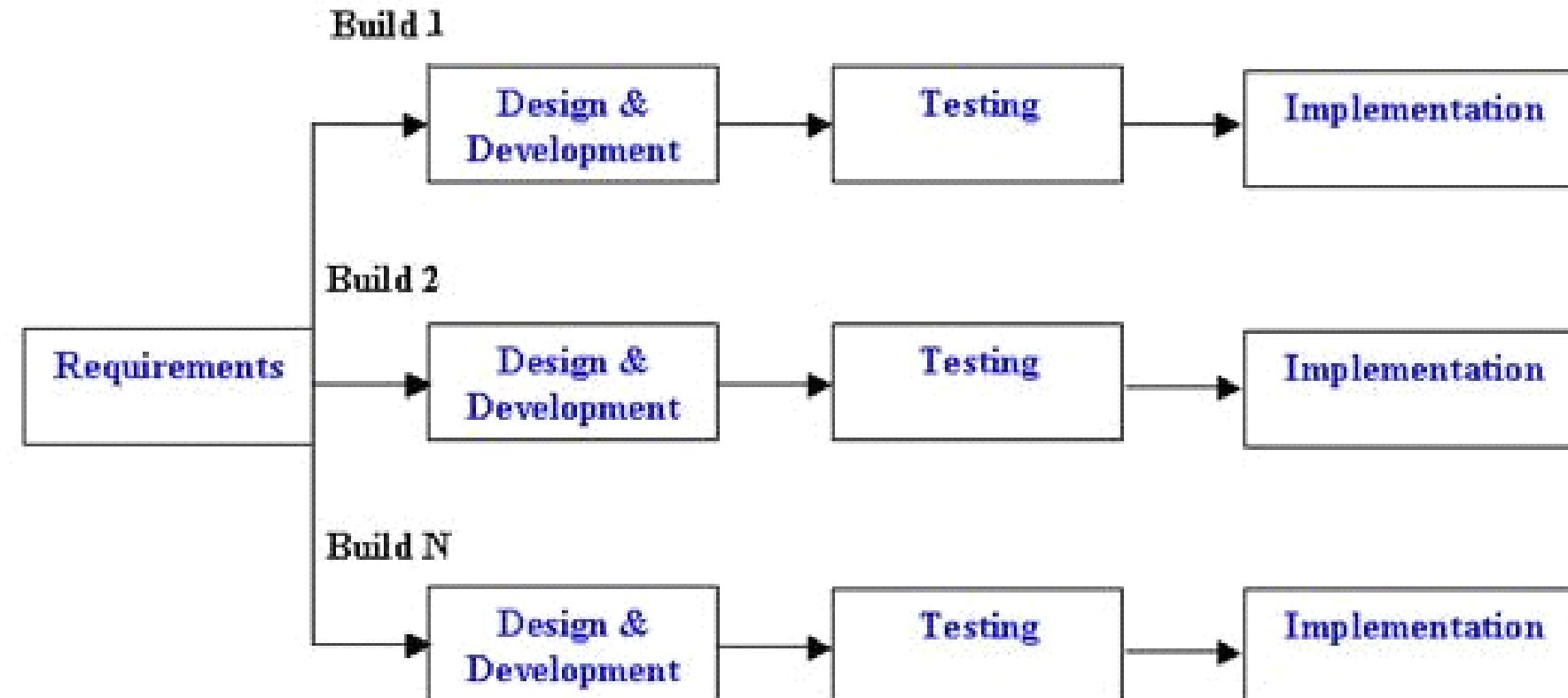
- ❖ Yêu cầu tài nguyên nhiều.
- ❖ Các vấn đề về thiết kế hoặc kiến trúc hệ thống có thể phát sinh bất cứ lúc nào.
- ❖ Yêu cầu quản lý phức tạp hơn.
- ❖ Tiến độ của dự án phụ thuộc nhiều vào giai đoạn phân tích rủi ro.

MÔ HÌNH PHÁT TRIỂN GIA TĂNG

❑ Mô hình phát triển gia tăng

❖ Mô hình với đặc tả yêu cầu được chia thành nhiều phần, chủ kỳ được chia thành các module nhỏ, dễ quản lý và mỗi module sẽ đi qua các yêu cầu về thiết kế, thực hiện, ... như 1 vòng đời phát triển thông thường.

☐ Sơ đồ mô hình phát gia tăng



Incremental Life Cycle Model

□ Phân tích mô hình phát triển gia tăng

- ❖ Đặc tả yêu cầu 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 thiết kế, thực hiện, ... như 1 vòng đời phát triển thông thường.

❑ Ứng dụng mô hình phát triển gia tăng

- ❖ 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 mô hình phát triển gia tăng

- ❖ 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 mô hình phát triển gia tăng

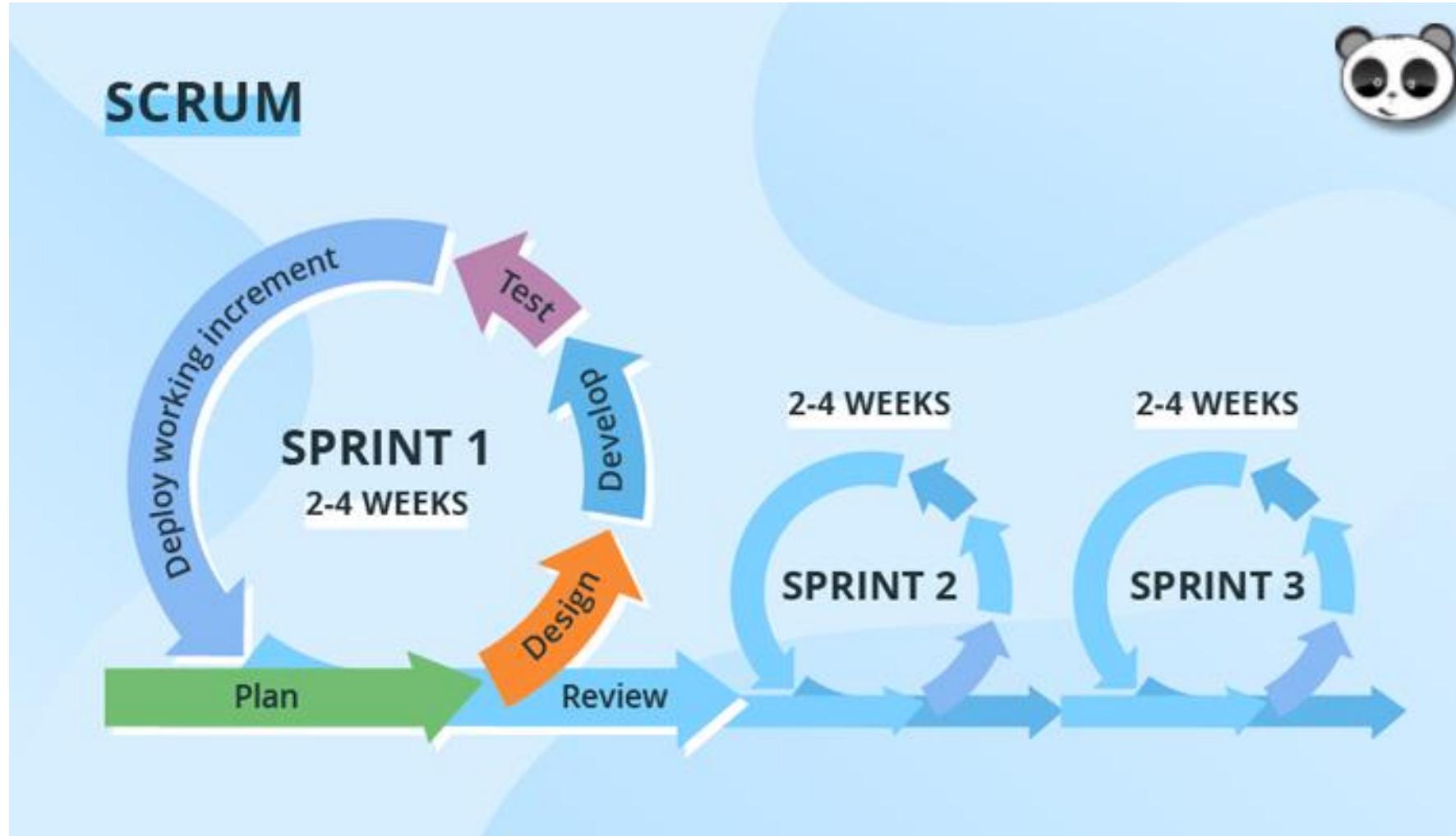
- ❖ Cần lập plan và thiết kế tốt.
- ❖ Tổng chi phí là cao hơn so với mô hình thác nước.

MÔ HÌNH PHÁT TRIỂN SCRUM

Mô hình phát triển Scrum

- ❖ Scrum là khung quy trình lắp lại và gia tăng giành cho một nhóm hợp tác hiệu quả, thường được áp dụng trong sự phát triển Agile – phương pháp lắp lại phổ biến nhất.
- ❖ Nguyên tắc chủ đạo trong mô hình này là chia nhỏ module cần phát triển. Lấy ý kiến của khách hàng, thay đổi cho phù hợp với quá trình phát triển. Đảm bảo sản phẩm release có thể đáp ứng mọi vấn đề khách hàng mong muốn.

☐ Sơ đồ mô hình phát triển Scrum



❑ Phân tích mô hình phát triển Scrum

- ❖ 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

❖ Tổ chức:

✓ Product Owner: Người sở hữu sản phẩm.

✓ ScrumMaster: Người điều phối.

✓ Development Team: Nhóm phát triển.

❖ Tài liệu: 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.

- ❖ Qui trình: 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(PO)

✓ 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.

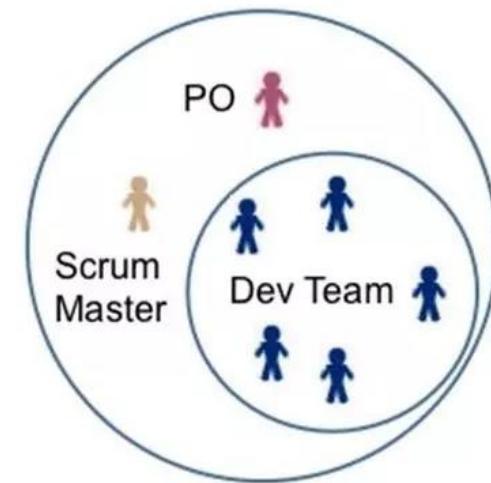
✓ Thông thường Role này được khách hàng hoặc người đại diện cho khách hàng đảm nhận.

❖ ScrumMaster

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

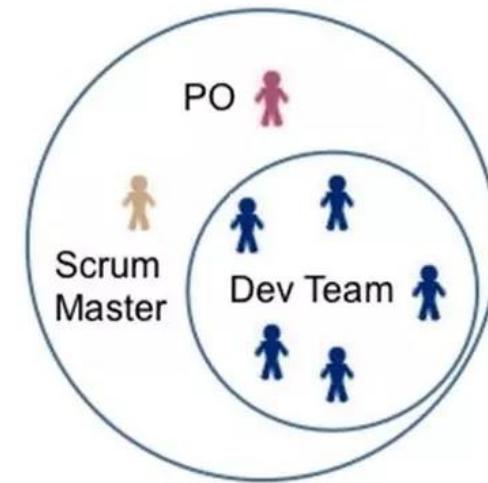
❖ Development Team(Dev Team)

- ✓ 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
- ✓ 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 mô hình phát triển Scrum

- ❖ 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 mô hình phát triển Scrum

- ❖ 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.

Tóm tắt bài học

- Mô hình phát triển tuần tự
- Mô hình phát triển lặp lại-gia tăng
- Mô hình Scrum



Nội dung tiếp theo

- Mô hình Kanban
- Mô hình phát triển Agile
- Mối liên quan giữa các hoạt động phát triển phần mềm với các hoạt động kiểm thử trong vòng đời phát triển phần mềm





FPT POLYTECHNIC

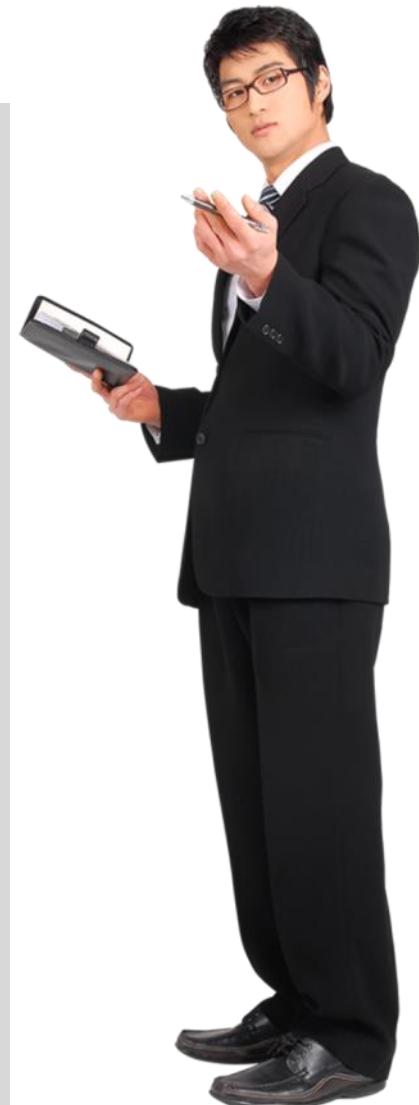


KIỂM THỬ CƠ BẢN(P2)

BÀI 2: KIỂM THỬ TRONG VÒNG ĐỜI PHÁT TRIỂN PHẦN MỀM

Nội dung bài học

- Mô hình Kanban
- Mô hình phát triển Agile
- Mối liên quan giữa các hoạt động phát triển phần mềm với các hoạt động kiểm thử trong vòng đời phát triển phần mềm



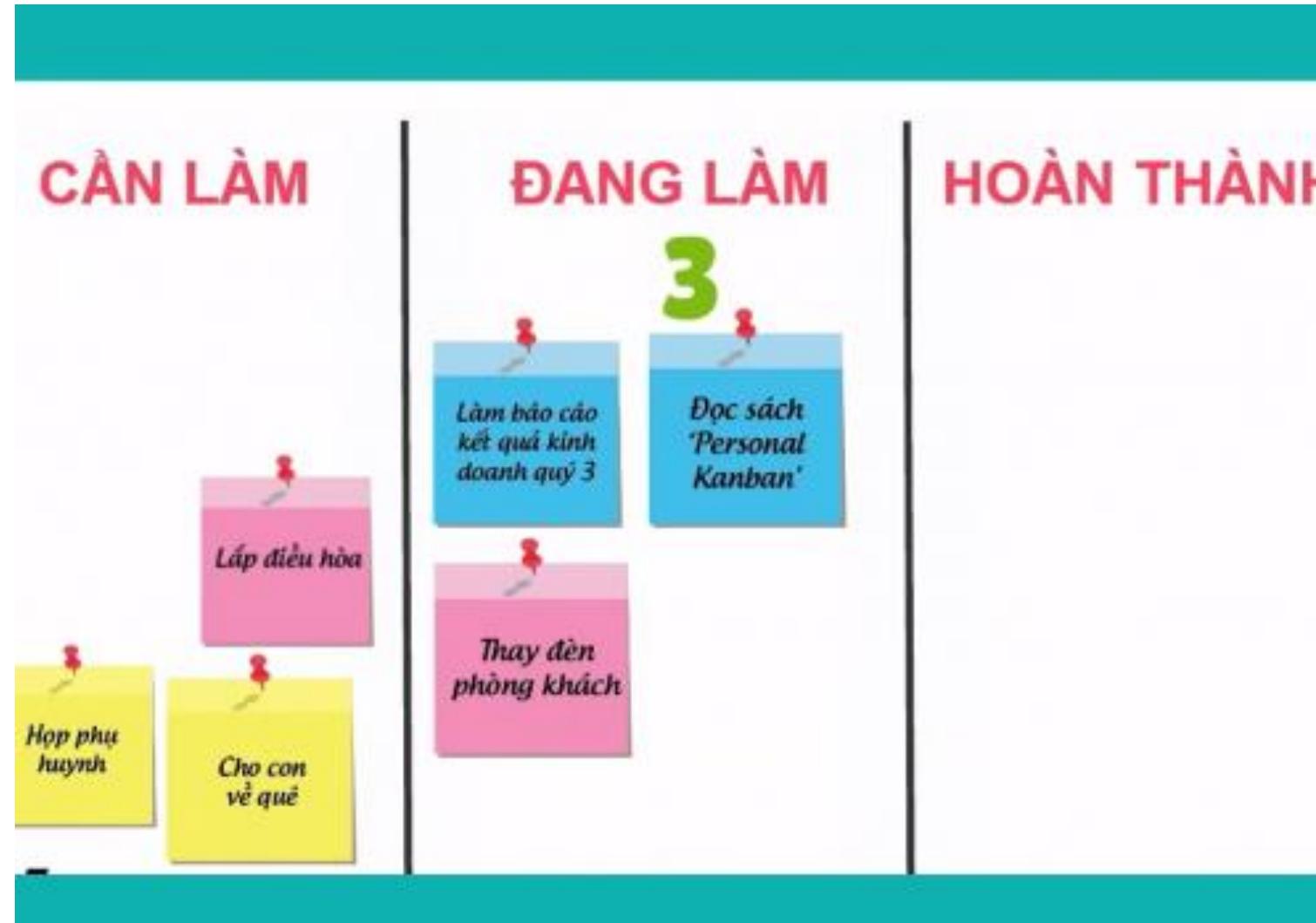
MÔ HÌNH PHÁT TRIỂN KANBAN

Mô hình phát triển Kanban

- ❖ Kanban là một phương pháp Agile nhưng không nhất thiết cần có tính lặp. Các quy trình như Scrum có các lần lặp ngắn (Sprint) là vòng đời của dự án trên quy mô nhỏ, có điểm bắt đầu và kết thúc riêng biệt cho mỗi lần lặp. Kanban cho phép phần mềm được phát triển trong một chu kỳ phát triển lớn.
- ❖ Kanban là một ví dụ về một phương pháp Agile vì nó đáp ứng tất cả mười hai nguyên tắc đằng sau tuyên ngôn Agile, bởi vì mặc dù nó không có tính lặp, nhưng vẫn có tính tăng trưởng.

❖ Khác với mô hình Scrum là giới hạn thời gian cho phép để hoàn thành một công việc cụ thể (sprint) thì Kanban giới hạn số lượng công việc cho phép trong một điều kiện nhất định (bao gồm nhiều task trên một thẻ Kanban).

☐ Bảng Kanban



❑ Ưu điểm mô hình phát triển Kanban

❖ Mọi người đều ở trên cùng một mặt phẳng: Tất cả các nhiệm vụ đều dễ dàng nhìn thấy, điều này mang lại sự minh bạch cho toàn bộ quá trình làm việc. Mỗi thành viên có thể cập nhật nhanh về trạng thái của mọi dự án hoặc nhiệm vụ.

❖ Chu kì thời gian làm việc được rút ngắn lại:
Trong Kanban, không phải mỗi người nắm giữ một kỹ năng, vì như vậy nếu người đó không hoàn thành tốt công việc thì sẽ sở thành điểm tắc nghẽn trong quy trình làm việc. Vì vậy, nhóm Kanban luôn hỗ trợ và bổ sung kỹ năng cho nhau, đảm bảo các thành viên luôn được học hỏi và không chỉ tập trung vào kỹ năng nào.

- ❖ Ít tắc nghẽn hơn: Kanban là giới hạn số lượng công việc đang thực hiện. Giới hạn công việc đang tiến hành giúp ít tắc nghẽn và tăng dự phòng trong quy trình của nhóm do thiếu tập trung, con người hoặc kỹ năng.

- ❖ Hệ thống đơn giản, dễ hiểu: Kanban đặc biệt linh hoạt ở chỗ có thể dễ dàng sử dụng ở bất kỳ đội nhóm trong ngành nghề hay quy mô nào vì nó dễ sử dụng.
- ❖ Hệ thống phản ứng nhanh nhạy: Kanban giúp chúng ta dễ dàng đáp ứng các yêu cầu thay đổi liên tục của khách hàng. Nó cho phép thay đổi các ưu tiên, tổ chức lại hoặc chuyển nhiệm vụ trọng tâm thực sự nhanh chóng.

❖ Yêu cầu tập trung vào các nhiệm vụ hiện tại cho đến khi hoàn thành: Điều này nhờ vào khái niệm giới hạn công việc đang xử lý. Giới hạn công việc đang xử lý giúp thúc đẩy các nhóm cộng tác để hoàn thành các mục công việc nhanh hơn, mặt khác giúp loại bỏ các phiền nhiễu như đa nhiệm.

❑ Nhược điểm mô hình phát triển Kanban

❖ Kanban thường tập trung vào các tác vụ hàng ngày, nó thúc đẩy việc thực thi nhiệm vụ, nhưng có thể gây rủi ro cho bức tranh lớn hơn là chiến lược và những kết quả quan trọng nhất. Các nhiệm vụ nhỏ được hoàn thành, nhưng kết quả cuối cùng chưa chắc đã đạt được.

- ❖ Không có khung thời gian của từng giai đoạn: Vì các cột chỉ được gắn nhãn với các giai đoạn (phải thực hiện, đang thực hiện, hoàn thành), nên có thể khó thấy khi nào mọi việc sẽ được thực hiện.
- ❖ Phải cập nhật bảng: Các nhóm phải nhấn mạnh tầm quan trọng của việc cập nhật bảng, nếu không, họ có nguy cơ làm việc với thông tin không chính xác.

❖ Kanban sẽ trở nên rất khó áp dụng vì có quá nhiều hoạt động hoặc nhiệm vụ liên quan đến nhau trong một hệ thống. Việc tăng cường chuyển giao tính năng và chuyên môn giữa các nhiệm vụ khác nhau quá thường xuyên và làm tăng khó khăn để theo kịp tất cả các hoạt động.

❖ Đầu ra có thể không đảm bảo chất lượng:
Kanban hoạt động giống như một cấu trúc giám sát giúp cho các luồng công việc trôi chảy hơn. Nếu bất kỳ công việc nào được thực hiện là không thỏa đáng, yêu cầu làm lại có thể làm tình hình tồi tệ hơn vì đòi hỏi nhiều thời gian và nguồn lực hơn để hoàn thành.

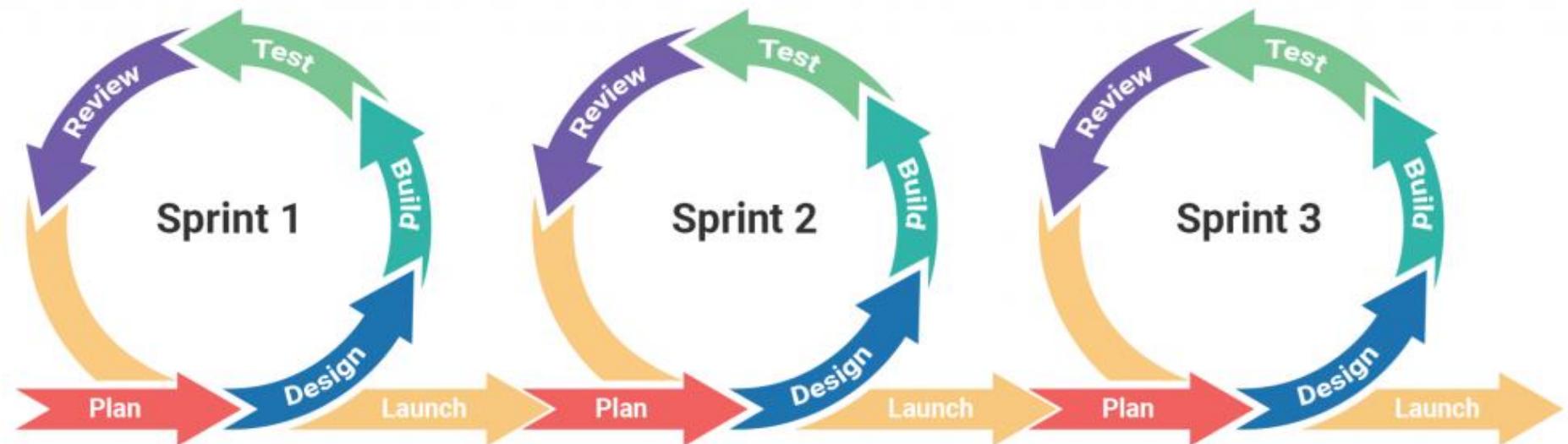
MÔ HÌNH PHÁT TRIỂN AGILE

☐ Mô hình phát triển Agile

- ❖ Agile (Agile Software Development) có nghĩa là phương thức phát triển phần mềm linh hoạt, được ứng dụng trong quy trình phát triển phần mềm với mục tiêu là đưa sản phẩm đến tay người dùng càng nhanh càng tốt.
- ❖ Agile giống như một phương pháp luận, một triết lý dựa trên hơn nguyên tắc phân đoạn vòng lặp (iterative) và tăng trưởng (incremental).

❖ Triết lí Agile đã vượt xa khỏi khu vực truyền thống của mình là phát triển phần mềm để đóng góp sự thay đổi trong cách thức làm việc, quản lý, sản xuất ở các ngành khác như sản xuất, dịch vụ, sales, marketing, giáo dục... và trở thành một phương thức quản lý dự án phổ biến nhất hiện nay với nhiều đại diện được gọi là các phương pháp “họ Agile”

☐ Mô hình phát triển Agile



☐ Ứng dụng mô hình phát triển Agile

- ❖ 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 mô hình phát triển Agile

❖ **Thực hiện thay đổi dễ dàng:** Bởi vì dự án được chia thành các phần nhỏ, riêng biệt, không phụ thuộc lẫn nhau, nên những thay đổi được thực hiện rất dễ dàng, ở bất kỳ giai đoạn nào của dự án.

❖ Không cần phải nắm mọi thông tin ngay từ đầu: Phù hợp với những dự án chưa xác định được mục tiêu cuối cùng rõ ràng, vì việc này không quá cần thiết trong giai đoạn đầu.

- ❖ **Bàn giao nhanh hơn:** Việc chia nhỏ dự án cho phép đội ngũ có thể tiến hành kiểm tra theo từng phần, xác định và sửa chữa vấn đề nhanh hơn, nhờ đó việc bàn giao công việc sẽ nhất quán và thành công hơn.
- ❖ **Chú ý đến phản hồi của khách hàng và người dùng:** Cả khách hàng và người dùng cuối đều có cơ hội để đóng góp các ý kiến và phản hồi, từ đó họ sẽ có ảnh hưởng một cách mạnh mẽ và tích cực tới sản phẩm cuối cùng.

❖ **Cải tiến liên tục:** Agile khuyến khích thành viên trong đội ngũ làm việc và khách hàng cung cấp phản hồi của mình, khi đó các giai đoạn khác nhau của sản phẩm cuối có thể được kiểm tra và cải thiện lại nhiều lần nếu cần.

❑ Nhược điểm mô hình phát triển Agile

❖ **Khó lên kế hoạch dự án:** Khá là khó để xác định rõ ràng thời gian bàn giao sản phẩm cuối cùng, vì dự án được chia nhỏ thành các phần khác nhau và mỗi phần lại có thời gian bàn giao riêng biệt.

❖ **Bắt buộc phải hướng dẫn và đào tạo chi tiết:**

Phương pháp Agile phức tạp hơn nhiều so với phương pháp truyền thống. Họ sẽ cần phải trải qua đào tạo, hướng dẫn thì mới có thể nắm được phương pháp một cách rõ ràng, đặc biệt là thời gian đầu.

❖ **Ít tài liệu hướng dẫn:** Vì Agile thay đổi rất nhiều nên các tài liệu thích hợp cũng thường bị bỏ qua, vì không xác định rõ được kỳ vọng và thành phẩm ngay từ đầu. Mặc dù tài liệu không phải là yếu tố quan trọng nhất, nhưng chúng vẫn rất cần thiết.

- ❖ **Bắt buộc phải hợp tác để dự án thành công:** Điều này đòi hỏi một sự cam kết về thời gian từ cả hai bên trong suốt thời gian của dự án mà các cấu trúc quản lý dự án khác không luôn yêu cầu. Phải có sự tham gia tích cực của người dùng và tiếp tục cộng tác để nó hoạt động.
- ❖ **Chi phí cao:** Chi phí thực hiện theo phương pháp Agile thường hơn một chút so với các phương pháp phát triển khác.

❑ Mối liên quan giữa các hoạt động phát triển phần mềm với các hoạt động kiểm thử trong vòng đời phát triển phần mềm

❖ **Vòng đời phát triển phần mềm:** Đây là chuỗi các hoạt động được các Developers thực hiện để thiết kế và phát triển phần mềm có chất lượng tốt. Vòng đời phát triển phần mềm sử dụng thuật ngữ ‘Development’, không chỉ liên quan đến các nhiệm vụ coding được thực hiện bởi các Developers mà còn kết hợp các nhiệm vụ được đóng góp bởi Tester và các bên liên quan. Trong Vòng đời phát triển phần mềm, các test cases được tạo ra bởi các tester.

❖ **Vòng đời Kiểm thử phần mềm:** Bao gồm một loạt các hoạt động được thực hiện bởi tester, sử dụng các phương pháp để kiểm thử sản phẩm phần mềm. Vòng đời kiểm thử Phần mềm sử dụng thuật ngữ “Testing”, không chỉ liên quan đến tester, trong một số trường hợp cũng liên quan đến các Developers.

Tóm tắt bài học

- Mô hình phát triển tuần tự
- Mô hình phát triển lặp lại-gia tăng
- Mô hình Scrum
- Mô hình Kanban
- Mô hình phát triển Agile
- Mối liên quan giữa các hoạt động phát triển phần mềm với các hoạt động kiểm thử trong vòng đời phát triển phần mềm



Nội dung tiếp theo

- Component Testing - Kiểm thử thành phần
- Integration Testing - Kiểm thử tích hợp
- System Testing - Kiểm thử hệ thống
- Acceptance Testing - Kiểm thử chấp nhận





FPT Education

FPT POLYTECHNIC

Thank you