Lab 2 Kiểm Thử Cơ Bản

**Phần 1**

**Bài 1:**

**1.1 Đây là mô hình phát triển nào ?**

**1.2 Những dự án như thế nào nên áp dụng mô hình này ? Nêu ưu nhược điểm của mô hình.**

**A picture containing sketch, screenshot, text, design

Description automatically generated**

1.1 Đây là mô hình phát triển Scrum

1.2 \*Những dự án nên áp dụng mô hình Scrum: những dự án đòi hỏi sự quay vòng, có thể thay đổi linh hoạt, không bó buộc trong những quy chuẩn nhất định.

\*Ưu điểm:

- Một người có thể thực hiện nhiều việc ví dụ như DEV có thể TEST.

- Phát hiện lỗi sớm.

\*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.

**Bài 2: Dựa vào hình các nhóm hãy nêu các khái niệm của :**

**+ Product Backlog**

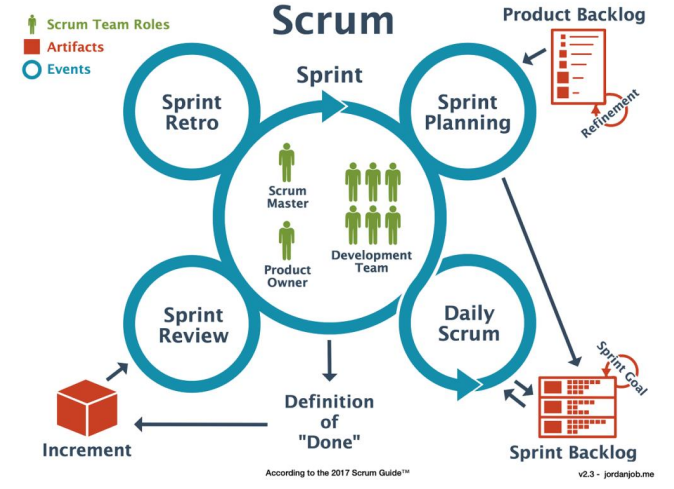
**+ Sprint Backlog**

**+ Sprint Planning**

**+ Sprint Review**

**+ Sprint Retro**

**+ Daily Meeting**

****

+ Product Backlog: một danh mục các công việc cần hoàn thành, được quản lý bởi Product Owner hoặc Product Manager.

+ Sprint Backlog: Một danh sách các đầu việc, hoặc user story, bug được lựa chọn bởi development team (đội phát triển), mang vào để triển khai trong 1 Sprint. Trước mỗi Sprint, đội phát triển sẽ có buổi họp Sprint Planning để lựa chọn sẽ thực hiện các đầu việc nào từ Product Backlog.

+ Sprint Planning : Cuộc họp lên kế hoạch, đặt mục tiêu cho Sprint của đội phát triển. Các user story cụ thể sẽ được thêm vào Sprint backlog từ Product backlog. Các user story cần được các thành viên đồng thuận với nhau rằng là khả thi để thực hiện trong Sprint này.

+ Sprint Review: Cuối mỗi Sprint, team sẽ tụ họp với nhau trong một buổi để demo increment - phần tăng trưởng sản phẩm. Team cũng sẽ chỉ ra những hạng mục công việc đã hoàn thành, và đón nhận góp ý từ product owner.

+ Sprint Retro: Quá trình hoạch định các cách thức để cải thiện chất lượng và hiệu quả.

+ Daily Meeting: Các buổi họp cực ngắn, tổ chức vào một khung giờ cố định, hàng ngày. Các thành viên tham gia trả lời 3 câu hỏi: Hôm qua làm gì? Hôm nay sẽ làm gì? Khó khăn, trở ngại đang gặp phải là gì?

**Phần 2**

**Bài 4:**

**4.1 Đây là mô hình nào ?**

**4.2 Những dự án như thế nào nên áp dụng mô hình này ? Nêu ưu nhược điểm của mô hình.**

**4.1 Đây là mô hình phát triển KanBan.**

**4.2 \* Những dự án nên áp dụng mô hình này : dự án làm một ứng dụng phần mềm, những dự án lớn.**

**A picture containing text, post-it note, handwriting, paper product

Description automatically generated**

\* Ưu điểm:

- Dễ dàng nhìn thấy công việc.

- Minh bạch.

- Cập nhật nhanh quá trình , nhiệm vụ.

- Thời gian rút ngắn.

- Hỗ trợ bổ sung giúp đỡ nhau.

- Ít tắc nghẽn.

- Hệ thống đơn giản dễ hiểu.

- Hệ thống phản ứng nhanh chóng.

- Tập trung vao nhiệm vụ đến khi hoàn thành.

\* Nhược điểm:

- Nhiệm vụ hoàn thành nhưng kết quả chưa chắc đạt được.

- Không có khung thời gian cho cái giai đoạn.

- Phải cập nhật bảng.

- Khó áp dụng đối với những mô hình thay đổi thường xuyên.

- Kết quả không đảm bảo chất lượng.

**Bài 5: Dựa vào hình hãy so sánh giữa phát triển Agile và phát triển Waterfall**

**A picture containing text, diagram, line, font

Description automatically generated**

Giống nhau: đều là mô hình phát triển phần mềm.

Khác nhau

|  |  |
| --- | --- |
| Mô hình Agile | Mô hình Waterfall |
| Nó tách vòng đời phát triển dự án thành chạy nước rút. | Quá trình phát triển phần mềm được chia thành các giai đoạn riêng biệt. |
| Nó theo một cách tiếp cận gia tăng | Phương pháp thác nước là một quá trình thiết kế tuần tự. |
| Phương pháp nhanh được biết đến với tính linh hoạt của nó. | Thác là một phương pháp phát triển phần mềm có cấu trúc nên hầu hết thời gian nó có thể khá cứng nhắc. |
| Agile có thể được coi là một bộ sưu tập của nhiều dự án khác nhau. | Phát triển phần mềm sẽ được hoàn thành như một dự án duy nhất. |
| Agile là một phương pháp khá linh hoạt cho phép thay đổi được thực hiện trong các yêu cầu phát triển dự án ngay cả khi kế hoạch ban đầu đã được hoàn thành. | Không có phạm vi thay đổi các yêu cầu khi phát triển dự án bắt đầu. |
| Phương pháp nhanh, theo một cách tiếp cận phát triển lặp lại vì quy hoạch, phát triển, tạo mẫu và các giai đoạn phát triển phần mềm khác có thể xuất hiện nhiều lần. | Tất cả các giai đoạn phát triển dự án như thiết kế, phát triển, thử nghiệm, được hoàn thành một lần trong mô hình Thác. |
| Kế hoạch kiểm tra được xem xét sau mỗi lần chạy nước rút. | Kế hoạch kiểm tra hiếm khi được thảo luận trong giai đoạn thử nghiệm. |
| Phát triển nhanh là một quá trình trong đó các yêu cầu được dự kiến sẽ thay đổi và phát triển. | Phương pháp này là lý tưởng cho các dự án có yêu cầu nhất định và thay đổi không được mong đợi. |
| Agile giới thiệu tư duy sản phẩm , nơi sản phẩm phần mềm đáp ứng nhu cầu của khách hàng cuối cùng và thay đổi chính nó theo nhu cầu của khách hàng. | Mô hình này cho thấy một tư duy dự án và đặt trọng tâm của nó hoàn toàn vào việc hoàn thành dự án. |
| Thích các nhóm nhỏ nhưng chuyên dụng với mức độ phối hợp và đồng bộ cao. | Phối hợp/ đồng bộ hóa rất hạn chế. |
| Chủ sở hữu dsanr phẩm với nhóm chuẩn bị các yêu cầu chỉ là về mỗi ngày trong một dự án. | Phân tích kinh doanh chuẩn bị các yêu cầu trước khi bắt đầu dự án. |
| Đội kiểm tra có thể tham gia vào các yêu cầu thay đổi mà không có vấn đề gì. | Thật khó để thử nghiệm bắt đầu bất kỳ thay đổi nào về yêu cầu |
| Mô tả chi tiết dự án có thể được thay đổi bất cứ lúc nào trong quá trình SDLC | Mô tả chi tiết cần thực hiện phương pháp tiếp cận phát triển phần mềm thác nước. |
| Các thành viên của nhóm Agile có thể hoán đổi cho nhau, do đó , chúng hoạt động nhanh hơn. | Trong phương pháp thác nước, quy trình luôn đơn giản như vậy, người quản lý dự án đống một vai trò thiết yếu trong mọi giai đoạn của SDLC. |
| Có nhiều thay đổi yêu cầu thường xuyên hơn. | Cho các dự án xác định được yêu cầu và không có thay đổi nào trong quá trình phát triển. |
| Rất linh hoạt và có thể thay đổi trong bất kì giai đoạn nào. | Dễ quản lý, tuần tự, cứng nhắc. |