



ĐỀ CƯƠNG BÀI GIẢNG

Bài 3: Lập kế hoạch dựa trên phương pháp linh hoạt

NỘI DUNG BÀI HỌC

| | | |
|----|---|----|
| 1. | Xác định tầm nhìn và phạm vi dự án..... | 1 |
| 1. | Tầm nhìn của dự án..... | 1 |
| 2. | Phạm vi của dự án | 3 |
| 3. | Các quy trình quản lý phạm vi dự án | 3 |
| 2. | Hình thành nhóm dự án | 10 |
| 3. | Lập danh sách công việc (Product backlog)..... | 14 |
| 4. | Lập kế hoạch sprint | 16 |
| 5. | Thực hiện sprint..... | 19 |
| 6. | Kiểm tra và điều chỉnh | 22 |
| | Lập kế hoạch kiểm tra (Test Planning) | 22 |
| 7. | Chuyển giao sản phẩm và cải tiến liên tục | 25 |
| 8. | Tái lập kế hoạch và điều chỉnh | 27 |

1. Xác định tầm nhìn và phạm vi dự án

1. Tầm nhìn của dự án

1. Khái niệm tầm nhìn của dự án

Tầm nhìn dự án là một tuyên bố ngắn gọn và rõ ràng về mục tiêu dài hạn của dự án và giá trị mà dự án sẽ mang lại cho người dùng và các bên liên quan. Nó cung cấp một định hướng chiến lược và mục tiêu cuối cùng cho dự án, giúp mọi người trong nhóm và các bên liên quan hiểu rõ về lý do và ý nghĩa của dự án.



2. *Tầm quan trọng của tầm nhìn của dự án*

- Định Hướng Chiến Lược: Tầm nhìn dự án giúp định hướng cho toàn bộ nhóm phát triển, đảm bảo mọi hoạt động đều hướng đến mục tiêu chung.
- Tạo Sự Thống Nhất: Giúp các bên liên quan và thành viên trong nhóm có cùng một hiểu biết về mục tiêu của dự án.
- Đánh Giá Quyết Định: Tầm nhìn dự án là cơ sở để đánh giá các yêu cầu và quyết định phát triển, đảm bảo rằng các hoạt động và tính năng đều phù hợp với mục tiêu tổng thể.
- Tạo Động Lực: Tầm nhìn rõ ràng và hấp dẫn có thể tạo động lực cho nhóm làm việc, giúp họ thấy được giá trị và ý nghĩa của công việc mình đang làm.

3. *Cách xác định tầm nhìn của dự án*

- Phỏng Vấn Các Bên Liên Quan
 - o Mục tiêu: Hiểu rõ mong đợi và yêu cầu của các bên liên quan, bao gồm khách hàng, người dùng cuối, quản lý và các bộ phận liên quan khác.
 - o Phương pháp: Thực hiện các buổi phỏng vấn, khảo sát hoặc hội thảo để thu thập thông tin.
- Phân Tích Thị Trường và Khách Hàng
 - o Mục tiêu: Xác định nhu cầu và mong muốn của khách hàng, hiểu rõ thị trường mục tiêu và bối cảnh cạnh tranh.
 - o Phương pháp: Nghiên cứu thị trường, phân tích đối thủ cạnh tranh và thu thập phản hồi từ khách hàng hiện tại.
- Xác Định Giá Trị Cốt Lõi
 - o Mục tiêu: Tập trung vào những lợi ích chính mà dự án mang lại, xác định các giá trị cốt lõi mà dự án sẽ cung cấp cho người dùng và các bên liên quan.



- Phương pháp: Thảo luận nhóm, sử dụng các công cụ như SWOT analysis (phân tích điểm mạnh, điểm yếu, cơ hội và thách thức) để xác định các giá trị cốt lõi.

2. Phạm vi của dự án

- Phạm vi (Scope) là một danh sách tất cả những gì dự án phải làm (và cũng có thể là một danh sách tất cả những điều mà dự án không phải làm). Dự án phải có một phạm vi được viết ra rõ ràng, nếu không dự án sẽ không bao giờ kết thúc.
- Các kết quả chuyển giao (Deliverables) là những kết quả của dự án mà sẽ chuyển giao như phần cứng, phần mềm (mua hoặc đặt làm), bảo hành, tài liệu, đào tạo và phương thức chuyển giao.
- Những dự án và các bên liên quan (Stakeholders) phải cùng hiểu những sản phẩm nào được tạo ra như là kết quả của dự án và chúng được tạo ra như thế nào.

3. Các quy trình quản lý phạm vi dự án

- Khởi động: bắt đầu một dự án hoặc chuyển tiếp sang giai đoạn tiếp theo.
- Lập kế hoạch phạm vi: phát triển các tài liệu nhằm cung cấp nền tảng cho các quyết định về dự án trong tương lai.
- Xác định phạm vi: chia nhỏ các sản phẩm trung gian của dự án thành các thành phần nhỏ hơn, dễ quản lý hơn.
- Ước lượng và phân bổ công việc dựa trên cấu trúc phân rã công việc (Work Breakdown Structure – WBS) là một phần rất quan trọng, ảnh hưởng đến sự thành công cho dự án.
- Kiểm tra phạm vi: hợp thức hóa việc chấp nhận phạm vi của dự án.
- Điều khiển thay đổi phạm vi: điều khiển những thay đổi của phạm vi dự án.



1. Khởi động

Quá trình này là bước khởi đầu quan trọng của một dự án và không thể được thiếu trong toàn bộ chu trình phát triển dự án. Mục đích của bước này là phát triển và hoàn thiện các mục tiêu của dự án xuất phát từ những ý tưởng sơ bộ ban đầu để đưa ra những mục tiêu cụ thể và tạo ra một bản phác thảo những ý chính của dự án và mang tính quyết định cao.

Thành phần nhân sự tham gia quá trình này gồm:

- Giám đốc dự án và khách hàng: thường xuyên trao đổi để xác định được mục tiêu của dự án và phát biểu được bài toán cụ thể từ các ý tưởng ban đầu.
- Đội phân tích nghiệp vụ: tham gia tích cực vào quá trình này để xác định đầy đủ và rõ ràng những yêu cầu của bài toán.

Các công cụ và kỹ thuật cần thiết cho quá trình này chủ yếu dựa trên giao tiếp và truyền thông: giao tiếp giữa đội dự án (giám đốc dự án và nhóm phân tích nghiệp vụ) với khách hàng để phát biểu bài toán một cách rõ ràng, mạch lạc, để xác định các yêu cầu.

Ngoài ra, giám đốc của dự án và công ty cần thực hiện các nghiên cứu để chuẩn bị cho dự án, ví dụ phân tích nghiên cứu các yếu tố ảnh hưởng tới dự án, những vấn đề kỹ thuật, con người,...

2. Lập kế hoạch phạm vi

Một trong những việc quan trọng nhất của quản lý dự án là việc lập kế hoạch. Vì ta luôn cần có kế hoạch cho việc phát triển một dự án để tạo ra một sản phẩm hoặc dịch vụ duy nhất, do đó mỗi dự án đều có những đặc thù riêng.

Bản kế hoạch cho dự án là một bản hướng dẫn việc thực thi dự án để đảm bảo việc thực hiện có hiệu quả tốt theo mong muốn. Công việc cần thiết của lập kế hoạch là thiết lập và lưu lại các giả thiết cho việc lập kế hoạch dự án, đồng thời lập tài liệu lưu các quyết định về kế hoạch liên quan tới các phương án thay thế được chọn khi thực thi dự án. Lý do cuối cùng của việc cần có một bản kế hoạch



đó là để tạo môi trường giao tiếp thuận lợi giữa những người tham gia dự án, cung cấp bản kế hoạch gốc cho việc đo tiến độ và kiểm soát việc thực thi dự án.

Tài liệu kế hoạch dự án là sản phẩm của quá trình lập kế hoạch và là một tài liệu chính thức được thông qua bởi các nhà quản lý của dự án và công ty. Ta lưu ý là một kế hoạch dự án không chỉ là một lịch trình thực hiện mà còn bao gồm cách tiếp cận quản lý dự án.

Các tài liệu của dự án được chia làm hai loại:

- Loại tài liệu cho sản phẩm**
 - Phát biểu nhu cầu: là tài liệu mô tả cụ thể các yêu cầu chức năng của hệ thống cần xây dựng.
 - Mô tả giao diện hệ thống: mô tả giao diện với người sử dụng của hệ thống.
 - Mô tả yêu cầu phần mềm.
 - Mô tả thiết kế phần mềm: bản thiết kế các chức năng.
 - Kế hoạch xác thực phần mềm: tài liệu kế hoạch kiểm tra tính đúng đắn của phần mềm.
 - Tài liệu người dùng: tài liệu hướng dẫn người sử dụng phần mềm.
 - Kế hoạch hỗ trợ: mô tả các kế hoạch huấn luyện và hỗ trợ người sử dụng dùng phần mềm sau khi nó được cài đặt.
 - Tài liệu bảo dưỡng: hướng dẫn bảo dưỡng và sửa chữa những lỗi thường gặp.
- Loại tài liệu cho quản lý dự án**
 - Bản kế hoạch phát triển phần mềm (SDP - Software Development Plan).
 - Bản kế hoạch đảm bảo chất lượng phần mềm (SQAP - Software Quality Assure Plan).
 - Bản kế hoạch quản lý cấu hình phần mềm (SCMP - Software Configuration Management Plan).
 - Bản kế hoạch quản lý rủi ro của dự án.
 - Bản kế hoạch cải thiện tiến trình làm phần mềm.



- Bản kế hoạch quản lý truyền thông và giao tiếp.
- Bản kế hoạch chuyển đổi hệ thống từ hệ thống cũ sang hệ thống mới.
- Bản kế hoạch vận hành hệ thống mới.

Trên đây ta chỉ liệt kê các tài liệu kế hoạch nói chung, giám đốc dự án không nhất thiết phải tạo hết những tài liệu này mà cần lựa chọn các tài liệu thích hợp cho mỗi dự án khác nhau, mỗi tài liệu không cần quá dài nhưng cần đầy đủ thông tin và được viết rõ ràng, nếu tuân thủ theo một mẫu chung thì tốt hơn.

Tài liệu kế hoạch quản lý dự án phần mềm (Software Development Plan – SDP/Software Project Management Plan – SPMP) được coi là tài liệu quan trọng nhất trong dự án cùng với bản mô tả yêu cầu của dự án phần mềm (Software Requirements Specification – SRS) vì nó có thể được xem như một bản tích hợp các tài liệu cơ bản khác và nó được phát triển dần qua thời gian bằng cách thêm từng phần nhỏ vào.

3. Cấu trúc phân rã công việc (WBS)

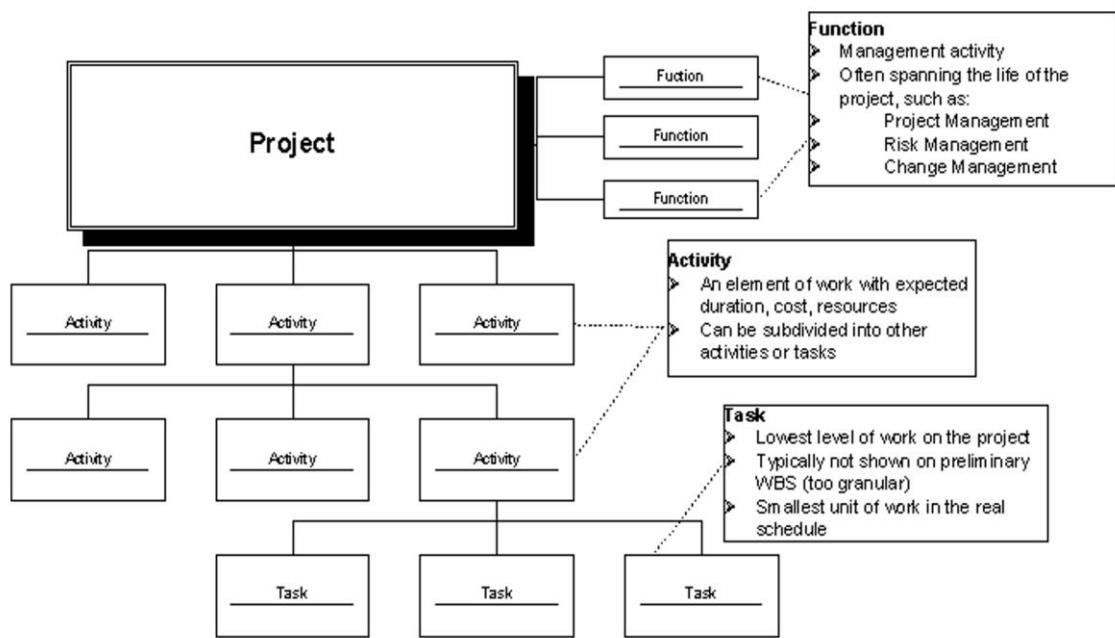
Sau khi hoàn tất kế hoạch về phạm vi, bước tiếp theo là xác định chi tiết công việc bằng cách chia thành các công việc nhỏ hơn có thể quản lý được. Để làm điều này thì người ta thường dùng cấu trúc phân rã công việc (Work Breakdown Structure – WBS).

Cấu trúc phân rã công việc (WBS) là một phần rất quan trọng, ảnh hưởng đến tiến trình lập kế hoạch, dự toán và lập lịch cho dự án. Việc ước lượng cho toàn bộ dự án phụ thuộc rất nhiều vào độ chính xác của WBS. Việc xây dựng được WBS tốt sẽ góp phần tăng thêm sự thành công cho dự án.

Để thiết lập WBS, một trong các cách tiếp cận là biện pháp phân nhỏ dự án thành nhiều phần để có thể quản lý được. Tất cả dự án đều cần bước này, vì đây là một cách dựa trên nguyên lý “Chia để trị”. Nguyên nhân chính dẫn đến sự thất bại của một dự án là do (i) quên một số thứ thiết yếu cho dự án và (ii) cố gắng nhắm tới những con số ước lượng và muốn biến những con số đó thành hiện thực.



Các nhân tố của một dự án bao gồm các chức năng, hoạt động và công việc được mô tả ví dụ trong hình sau đây.



Định nghĩa về WBS: “Một nhóm phân cấp theo định hướng các sản phẩm phân phối của các nhân tố dự án tổ chức và định nghĩa toàn bộ phạm vi của dự án. Mỗi mức bên dưới thể hiện một định nghĩa dự án ở mức chi tiết hơn.”

Cấu trúc phân rã công việc là danh sách phân rã các hoạt động cần thực hiện của dự án. Cấu trúc này gồm hai định dạng:

- Định dạng theo dạng các đầu mục (outline) (hình a)
- Định dạng theo cây đồ họa (biểu đồ tổ chức) (hình b)

Trong đó thường sử dụng một hệ thống đánh số thập phân, ví dụ như 2.1.1, trong đó 0 thường biểu diễn mức độ cao nhất.

Cấu trúc thường thể hiện các mối quan hệ trong của dự án, mối quan hệ giữa các công việc cụ thể. Tuy nhiên, cấu trúc này thường không thể hiện sự phụ thuộc về thời gian và khoảng thời gian cần thực hiện mỗi công việc.



✓ Các kỹ thuật tạo WBS

- Kỹ thuật “từ trên xuống”: Bắt đầu ở mức cao nhất sau đó phát triển chi tiết các việc ở mức thấp hơn theo hệ thống. Kỹ thuật này tốt nhất trong trường hợp:

- o Bài toán được hiểu thấu đáo.
- o Kỹ thuật và phương pháp luận không mới.
- o Giống với một dự án hay một bài toán trước đó.



- o Tuy vậy nó được áp dụng trong hầu hết các trường hợp.
 - Kỹ thuật “từ dưới lên”: thường bắt đầu từ mức thấp nhất sau đó kết hợp các công việc ở mức thấp thành các công việc ở mức cao hơn và tổng quát hơn. Nhược điểm của kỹ thuật này là:
 - o Tốn thời gian.
 - o Cần tìm hiểu nhiều yêu cầu để hoàn thành.
 - o Nhưng cũng có ưu điểm là rất chi tiết.
 - Kỹ thuật “tương tự”: là kỹ thuật dựa trên WBS của một dự án tương tự, sử dụng một mẫu sẵn có. Tương tự cũng có thể là tiền đề cho việc ước lượng. Kỹ thuật này có ưu điểm là dựa trên kinh nghiệm thực tế trong quá khứ và nhược điểm là cần có dự án tương tự.
 - Kỹ thuật “cuộn sóng”
 - o Lượt thứ nhất: đi sâu xuống mức 1 – 3.
 - o Thu thập thêm yêu cầu hoặc dữ liệu.
 - o Đưa thêm vào nhiều chi tiết hơn sau đó.
 - Kỹ thuật “tổng hợp ý tưởng”
 - o Tạo ra tất cả các hoạt động cần thực hiện cho dự án mà bạn có thể nghĩ đến
 - o Nhóm chúng vào thành các loại khác nhau
- Cả hai loại kỹ thuật trên xuống và tổng hợp ý tưởng có thể được sử dụng trên cùng một WBS. Ta cần chú ý rằng nên kéo tất cả những người đang thực hiện công việc liên quan tham gia kỹ thuật tổng hợp các ý tưởng. WBS quan trọng vì nó là nền tảng của nhiều quá trình: Lập lịch theo dạng mạng, tính chi phí, phân tích rủi ro, cấu trúc tổ chức, kiểm soát, đo đạc tính toán.
- Ngoài ra, trong thực tế, việc xây dựng WBS thường được triển khai theo các cách như sau:
- Cách 1. Chia việc theo chức năng, trong đó:



- o Chia nhỏ công việc theo hướng chức năng.
 - o Dừng việc chia ở mức tổng quát thấp nhất.
 - o Xác định thời gian làm các việc mức thấp nhất đó.
- Cách 2. Chia việc theo đối tượng thực hiện, bao gồm:
- o Chia nhỏ công việc theo hướng chức năng ở một mức thấp hơn.
 - o Xác định các đối tượng màn hình, công cụ, đối tượng lập trình (textbox, button,
 - ...).
- o Xác định thời gian làm các đối tượng đó.
- Cách 3. Chia việc theo mức kỹ thuật thực hiện, với các công việc:
- o Chia nhỏ công việc theo hướng chức năng ở một mức thấp hơn.
 - o Xác định các mức kỹ thuật cần thực hiện cho một chức năng.
 - o Xác định thời gian làm từng mức kỹ thuật đó.

4. Kiểm tra phạm vi và điều khiển thay đổi phạm vi

Trên thực tế, rất khó để tạo ra được tuyên bố phạm vi tốt và WBS tốt cho một dự án. Cũng như việc kiểm tra phạm vi dự án và giảm thiểu thay đổi phạm vi là điều khó hơn. Rất nhiều dự án CNTT gặp phải tình trạng “phạm vi phình ra” (scope creep) và kiểm tra phạm vi tồi. Do đó giám đốc dự án cần phải nghiên cứu kỹ phạm vi dự án.

2. Hình thành nhóm dự án

Các nhóm Scrum bao gồm:

- (1) Product Owner (Chủ Sản phẩm);
- (2) Nhóm phát triển (Development Team);
- (3) Scrum Master.

Các Nhóm Scrum là các nhóm tự quản (self-organizing) và liên chức năng (cross-functional). Các nhóm tự quản tự mình chọn cách thức tốt nhất để hoàn thành công việc của họ, chứ không bị chỉ đạo bởi ai đó bên ngoài nhóm. Các



nhóm liên chức năng có đủ kỹ năng cần thiết để hoàn thành công việc mà không phụ thuộc vào bất kỳ người ngoài nào khác. Mô hình nhóm trong Scrum được thiết kế để tối ưu hóa sự linh hoạt, sáng tạo và năng suất.

Các Nhóm Scrum chuyển giao sản phẩm theo phân đoạn lặp đi lặp lại và tăng dần, tối đa hóa cơ hội cho các phản hồi.

1. Product owner

Product Owner chịu trách nhiệm tối đa hóa giá trị của sản phẩm và công việc của Nhóm Phát triển. Cách thức để đạt được điều đó có thể rất khác nhau giữa các tổ chức, các Nhóm Scrum và các cá nhân. Product Owner là một người chủ yếu chịu trách nhiệm về việc quản lý Product Backlog. Đây là công cụ quản lý chứa:

- Mô tả các hạng mục (Product Backlog);
- Trình tự của các hạng mục trong Product Backlog để đạt được mục đích và hoàn thành các nhiệm vụ;
- Sự đảm bảo giá trị của các công việc của Nhóm Phát triển;
- Sự đảm bảo cho Product Backlog là luôn luôn hiện hữu, thông suốt và rõ ràng với tất cả mọi người và chỉ ra những gì mà Nhóm Scrum sẽ làm việc;
- Sự đảm bảo cho Nhóm Phát triển hiểu rõ các hạng mục trong Product Backlog với các mức độ cần thiết;
- Product Owner có thể tự mình thực hiện công việc trên hoặc để Nhóm Phát triển làm. Tuy nhiên, Product Owner vẫn phải chịu trách nhiệm chính;
- Product Owner là một người, không phải là một nhóm. Product Owner có thể cần tới một ủy ban tham gia vào Product Backlog, nhưng những người trong ủy ban muốn thay đổi trình tự các hạng mục trong Product Backlog phải thuyết phục được Product Owner;
- Để Product Owner thành công, cả tổ chức phải tôn trọng các quyết định của người này. Các quyết định đó được hiển thị trong nội dung và thứ tự trong Product Backlog. Không ai ngoài Product Owner được phép yêu cầu Nhóm Phát triển làm gì khác và Nhóm Phát triển cũng không được phép làm gì theo lời bất cứ người nào khác.



2. Development Team

Nhóm Phát triển (Development Team) gồm các chuyên gia làm việc để cho ra các phần tăng trưởng có thể phát hành được (potentially releasable) cuối mỗi Sprint. Chỉ các thành viên của Nhóm Phát triển mới tạo ra các phần tăng trưởng này(Increment). Nhóm Phát triển được cấu trúc và trao quyền để tổ chức và quản lý công việc của họ. Sự hợp lực sẽ tối ưu hóa nỗ lực và hiệu quả tổng thể của Nhóm Phát triển.Nhóm Phát triển có các đặc trưng sau:

- Đó là nhóm tự tổ chức. Không ai (kể cả Scrum Master) có quyền yêu cầu Nhóm Phát triển làm thế nào để chuyển Product Backlog thành các phần tăng trưởng có thể chuyển giao được;
- Đó là nhóm liên chức năng, với tất cả các kỹ năng cần thiết để tạo ra phần tăng trưởng của sản phẩm;
- Scrum không ghi nhận một chức danh nào trong Nhóm Phát triển ngoài Nhà phát triển (Developer), theo tính chất công việc của người này, không có ngoại lệ cho quy tắc này;
- Các thành viên Nhóm phát triển có thể có các kỹ năng chuyên biệt và các chuyên môn đặc thù, nhưng họ phải chịu trách nhiệm dưới một thể thống nhất là Nhóm Phát triển;
- Nhóm Phát triển không chứa các nhóm con nào khác với các chức năng đặc thù như ‘nhóm kiểm thử’ hay ‘phân tích nghiệp vụ’;
- Độ lớn tối ưu của Nhóm Phát triển là đủ nhỏ để giữ được sự linh hoạt và đủ lớn để hoàn thành công việc. Ít hơn ba người có thể làm giảm sự tương tác và dẫn đến năng suất thấp. Các Nhóm Phát triển nhỏ hơn có thể phải đối mặt với các ràng buộc kỹ năng trong suốt Sprint, dẫn đến Nhóm Phát triển khó có thể chuyển giao gói tăng trưởng phát hành được. Một nhóm nhiều hơn chín người cần nhiều sự điều phối hơn. Các Nhóm Phát triển lớn phát sinh quá nhiều phức tạp để thực hiện việc kiểm soát tiến trình thực nghiệm. Các vai trò Product Owner và Scrum Master không được tính vào kích thước của Nhóm Phát triển, trừ khi họ cũng kiêm luôn vai trò là thành viên của Nhóm Phát triển.

3. Scrum master



Scrum Master chịu trách nhiệm đảm bảo mọi người hiểu và dùng được Scrum. Scrum Master thực hiện việc này bằng cách đảm bảo Nhóm Scrum tuân thủ lý thuyết, các kĩ thuật thực hành và các quy tắc của Scrum. Scrum Master là một lãnh đạo, nhưng cũng là người phục vụ Nhóm Scrum. Scrum Master giúp đỡ những người ngoài Nhóm Scrum hiểu cách phải tương tác với nhóm sao cho hiệu quả nhất. Scrum Master giúp đỡ tất cả mọi người cải tiến các mối tương tác để tối đa hóa giá trị mà Nhóm Scrum tạo ra.

Scrum Master phục vụ Product Owner theo nhiều cách, bao gồm:

- Tìm kiếm các kĩ thuật để quản lý hiệu quả Product Backlog;
- Giao tiếp tích cực với Nhóm Phát triển về tầm nhìn, mục đích và các hạng mục của Product Backlog;
- Huấn luyện cho Nhóm Phát triển biết cách tạo ra các hạng mục Product Backlog thật rõ ràng và đơn giản;
- Hiểu rõ việc lập kế hoạch dài hạn sản phẩm trong một môi trường thực nghiệm;
- Hiểu rõ và thực hành sự linh hoạt (agility);
- Thúc đẩy các sự kiện Scrum theo yêu cầu hoặc khi cần thiết; Scrum Master phục vụ Nhóm Phát triển theo nhiều cách, bao gồm:

- Huấn luyện Nhóm Phát triển cách tự tổ chức và làm việc liên chức năng;
- Giúp đỡ Nhóm Phát triển để tạo ra các sản phẩm có giá trị cao;
- Loại bỏ các trở lực trong quá trình tác nghiệp của Nhóm Phát triển;
- Thúc đẩy các sự kiện Scrum theo yêu cầu hoặc khi cần thiết;
- Huấn luyện Nhóm Phát triển trong trường hợp tổ chức chưa có hiểu biết và ứng dụng đầy đủ về Scrum;

Scrum Master phục vụ Tổ chức theo nhiều cách, bao gồm:

- Lãnh đạo và huấn luyện tổ chức trong việc áp dụng Scrum;
- Lập kế hoạch triển khai Scrum trong phạm vi tổ chức;
- Giúp đỡ nhân viên và các bên hữu quan hiểu và sử dụng được Scrum cũng như quá trình phát triển sản phẩm thực nghiệm (empirical product development);



- Tạo ra sự thay đổi làm tăng năng suất của Nhóm Scrum;
Làm việc với các Scrum Master khác để gia tăng hiệu quả của việc áp dụng Scrum trong tổ chức của mình.

3. Lập danh sách công việc (Product backlog)

Trong phát triển phần mềm, Product Backlog là danh sách các công việc cần phải thực hiện để hoàn thành một dự án. Những công việc trong Product Backlog có thể là các yêu cầu, tính năng hoặc các lỗi đầu vào cho Sprint Backlog.

Product Backlog các Product Owner hay Product Manager trực tiếp quản lý. Nội dung công việc trong Product Backlog sẽ được cập nhật liên tục theo sự thay đổi của khách hàng hay nhu cầu thị trường.

Về bản chất, Product Backlog tương tự như Sprint Backlog. Cả hai đều giúp nhóm phát triển quản lý công việc tồn đọng trong dự án. Tuy nhiên, Product Backlog có phạm vi công việc rộng hơn. Dưới đây là một số khía cạnh khác nhau giữa Product Backlog và Sprint Backlog:

Product backlog là nơi lưu trữ tất cả các câu chuyện của người dùng. Trong khi đó Sprint Backlog chỉ chứa các câu chuyện của người dùng đã được chọn trong một Sprint. Hiểu một cách đơn giản, Sprint Backlog là một tập hợp con của Product Backlog.

Sprint Backlog phân chia các nhiệm vụ cụ thể, còn Product Backlog mô tả tổng quan về công việc. Nếu Product Backlog là một chiến lược, thì Sprint Backlog được thiết kế để hiện thực hóa chiến lược đó.

Đặc điểm cơ bản của một product backlog: Product Backlog có 4 đặc điểm cơ bản là Detailed appropriately, Estimated, Emergent và Prioritized.

- **Detailed appropriately (Chi tiết một cách hợp lý)**

Trong Product Backlog, không phải bất cứ hạng mục công việc nào cũng cần thể hiện một cách chi tiết. Thông thường, những việc quan trọng phải làm trước sẽ được sắp xếp ở phía trên cùng của Product Backlog. Những công việc này cần phải chi tiết để có thể đưa vào Sprint gần nhất.



Mức độ chi tiết thường giảm dần theo độ ưu tiên và cần thiết của hạng mục công việc đó. Công việc có độ ưu tiên thấp hoặc phụ thuộc vào những hạng mục khác nên để ở cuối cùng. Chúng có thể được phân tích ít chi tiết hơn những hạng mục công việc ở phía trên Product Backlog.

- **Estimated (Tính ước lượng)**

Không chỉ là một danh sách các công việc phải làm, Product Backlog còn là một công cụ lập kế hoạch hữu ích. Trong Product Backlog, các hạng mục dành cho bản phát hành mới cần phải được ước lượng. Chúng có thể do nhóm phát triển hoặc khách hàng cung cấp.

Nhóm Phát triển sẽ cung cấp cho Product Owner khối lượng công việc ước lượng của từng hạng mục. Product Owner và các bên liên quan sẽ cung cấp thông tin về giá trị của sản phẩm. Đó có thể là lợi nhuận, chi phí, rủi ro trong kinh doanh và nhiều hạng mục khác.

- **Emergent (Sự tiến hóa)**

Product Backlog không phải là một thực thể tĩnh mà nó luôn thay đổi theo thời gian. Các câu chuyện của người dùng trong Product Backlog sẽ được thêm, xóa hoặc đánh giá lại. Product Backlog liên tục được Product Owner cập nhật trong suốt Sprint.

Ví dụ: Nhóm Phát triển cung cấp cho Product Owner khối lượng công việc của từng hạng mục. Nhưng trong quá trình thực hiện có một số rủi ro kỹ thuật dẫn đến sự thay đổi các hạng mục. Lúc này, Product Owner sẽ phải xem xét việc thêm bớt, hay sắp xếp lại mức độ ưu tiên trong Product Backlog.

- **Prioritized (Tính ưu tiên)**



Một Product Backlog cần sắp xếp với các mặt hàng có giá trị nhất ở trên cùng và ít giá trị nhất ở dưới cùng. Cách làm này giúp nhóm có thể tối đa hóa giá trị của sản phẩm.

Chúng ta có thể đặt mức độ ưu tiên cao cho các hạng mục quan trọng cần được đưa vào Sprint. Tiếp đến sẽ là các hạng mục dự định phát hành trong đợt 1. Khi xếp ưu tiên vượt quá giai đoạn phát hành đợt 1, chúng ta có thể sẽ mất nhiều thời gian hơn.

4. Lập kế hoạch sprint

Sprint planning là sự kiện khởi đầu ở mỗi Sprint, nhằm mục đích xác định mục tiêu và lên kế hoạch chi tiết cho những công việc cần thực hiện. Buổi Sprint Planning sẽ lần lượt giải quyết những vấn đề sau:

- Xác định mục tiêu của Sprint (Sprint goal)
- Lựa chọn các công việc cần làm để hoàn tất mục tiêu Sprint đặt ra
- Chia nhỏ hạng mục công việc (Sprint backlog), ước lượng và thống nhất mục tiêu.

Toàn bộ Scrum Team bắt buộc cần tham gia phần thứ nhất của sự kiện – tức xác định mục tiêu của Sprint, riêng Product Owner (PO) có thể vắng mặt ở phần sau.

Cụ thể:

- Scrum Master: là người bắt buộc phải tham dự buổi Sprint Planning để đưa ra mục tiêu và kế hoạch để hoàn thành các nhiệm vụ có trong Sprint.
- Development Team (Nhóm phát triển): tham dự để chọn lượng công việc sẽ làm trong Sprint dựa vào lượng công việc đã hoàn thành trong các Sprint trước đó.



- Product Owner: bắt buộc tham gia phần thứ nhất và có thể vắng mặt ở phần 2, nhưng phải luôn ở trạng thái sẵn sàng support, giải đáp thắc mắc cho nhóm ở những hạng mục quan trọng.
- Trưởng bộ phận có thể không tham gia hoặc có mặt để giám sát quy trình của Sprint.

Thời lượng tối đa của Sprint 1 tháng là 8 tiếng và ngắn hơn đối với các sprint 3 tuần, 2 tuần, 1 tuần.

Nhìn chung, thời lượng không quá 2 giờ mỗi tuần được xem là khởi điểm tốt cho các nhóm mới thành lập và đang tích lũy kinh nghiệm với Scrum.

Diễn biến của một buổi Sprint planning:

- Phần 1: Xác định mục tiêu sprint

Product Owner sẽ đề xuất làm thế nào để tăng giá trị và tính hữu dụng của sản phẩm trong Sprint. Cả Scrum Team cùng nhau xác định Sprint Goal (hoặc nhắc lại mục tiêu đã được thống nhất từ đầu dự án), qua đó truyền đạt lý do Sprint sẽ mang lại giá trị đến các bên liên quan. Mục tiêu Sprint phải được thông qua trước khi kết thúc Sprint Planning.

- Phần 2: Lựa chọn các công việc cần làm để hoàn tất mục tiêu Sprint đặt ra Thông qua thảo luận với Product Owner, Nhóm phát triển sẽ chọn những hạng mục trong Product Backlog để đưa vào Sprint hiện tại. Scrum Team có thể tinh chỉnh những hạng mục đó trong quá trình này để tăng sự hiểu biết và độ chắc chắn.

Việc xác định số lượng vừa đủ các hạng mục công việc để hoàn thành trong Sprint đôi khi khá thử thách. Tuy nhiên, một khi các Developers đã có thông tin về hiệu suất làm việc trong quá khứ, năng suất hiện tại và định nghĩa về sự hoàn tất, họ sẽ chắc chắn hơn về việc tính toán số lượng công việc của mình.

- Phần 3: Chia nhỏ hạng mục công việc (Sprint backlog), ước lượng và thống nhất mục tiêu



Ở bước này, nhóm phát triển sẽ phân tách các hạng mục trong danh sách lựa chọn thành những hạng mục nhỏ để có thể hoàn thành trong một ngày hoặc ít hơn. Đồng thời, Nhóm phát triển cũng cần ước tính thời gian và lượng nỗ lực cần bỏ ra để hoàn thành từng hạng mục công việc.

Sau khi đã tính toán khối lượng công việc, các Developer có thể trao đổi với PO để điều chỉnh thêm hoặc cắt bớt một vài hạng mục nếu cần thiết.

Các tips để triển khai Sprint Planning hiệu quả

1. Thông nhất thời điểm Sprint Planning

Thời điểm Sprint Planning nên được thống nhất sao cho phù hợp với tính chất công việc của cả nhóm. Ví dụ, đối với Sprint 1 tuần, thời điểm thực hiện Sprint Planning thường là sáng thứ Hai đầu tuần hoặc chiều ngày thứ Sáu.

2. Xác định thời gian nhóm có trong Sprint

Cần xác định thời gian cả nhóm có trong Sprint, vì qua đó mới có thể tính toán đúng khối lượng công việc mà nhóm sẽ hoàn thành. Nếu tính toán không chính xác, sẽ dẫn đến khả năng nhóm nhận nhiều hoặc ít việc hơn, ảnh hưởng đến hiệu suất.

Theo đó, mỗi thành viên sẽ đưa ra thời gian mình có thể dành cho công việc phát triển trong sprint tới như sau:

Thông báo số ngày nghỉ trong sprint tới

Tính số giờ có trong sprint = số ngày đi làm * 6.4. (Một ngày làm việc có 8 giờ nhưng thực chất thời gian tập trung vào công việc mỗi người cao nhất khoảng 6.4 giờ, sau khi trừ những khoảng thời gian phát sinh như họp hành đột xuất, trả lời email, cà phê...).

Sau khi có thời gian mỗi thành viên thì cộng tất cả lại sẽ được tổng thời gian nhóm có trong Sprint tới.

3. Các thành viên chủ động rà soát & chuẩn bị các task của bản thân

Một trong những đặc điểm của Scrum Team là Self-managed (tính tự chủ). Mỗi thành viên có sự chủ động trong việc rà soát và chuẩn bị các task của bản thân



trước khi bước vào cuộc họp thì buổi Sprint planning sẽ được thực hiện một cách nhanh chóng, hiệu quả hơn.

4. Thực hiện Grooming (Làm mịn) thường xuyên

Làm mịn là một kỹ thuật trong việc quản lý Product Backlog. Các hạng mục trong Product Backlog có kích thước khác nhau, mức độ chi tiết khác nhau và thường chúng cần được làm mịn trước khi đi vào sản xuất.

Scrum Team có thể dành ra 10% thời gian trong Sprint để làm mịn Product Backlog cho các phiên tiếp theo. Việc duy công việc từ trước để có sự chuẩn bị và không bị động là bí kíp để các buổi Sprint Planning được hiệu quả cao.

5. Sự cộng tác giữa các thành viên trong nhóm

Như tuyên ngôn của Agile: “Cá nhân và tương tác hơn là quy trình và công cụ”, bản kế hoạch Sprint chính là kết quả đến từ sự cộng tác của toàn bộ thành viên trong nhóm Scrum. Và bí kíp cho một story thành công là nên có sự nhìn nhận từ nhiều phía. Bởi, trong một nhóm thường là tập hợp của các thành viên có những kỹ năng khác nhau, mỗi vị trí khác nhau sẽ nhìn nhận được những khó khăn mà vị trí còn lại không lường trước được.

5. Thực hiện sprint

Daily Stand-Up (còn được gọi là Daily Scrum) là một cuộc họp ngắn gọn diễn ra hàng ngày, thường kéo dài khoảng 15 phút, nơi các thành viên trong nhóm phát triển cùng nhau cập nhật tiến độ công việc và lập kế hoạch ngắn hạn cho ngày làm việc tiếp theo. Cuộc họp này giúp đảm bảo sự phối hợp hiệu quả và nhận diện sớm các vấn đề có thể cản trở tiến độ.

Mục tiêu của Daily Stand-Up:

- Cập nhật tiến độ: Giúp tất cả các thành viên trong nhóm biết được trạng thái hiện tại của dự án và tiến độ công việc.



- Lập kế hoạch ngắn hạn: Xác định các công việc cần thực hiện trong ngày tiếp theo.
- Nhận diện vấn đề: Phát hiện sớm các rào cản và vấn đề ảnh hưởng đến công việc để tìm cách giải quyết kịp thời.
- Tăng cường giao tiếp: Tạo cơ hội cho các thành viên trong nhóm trao đổi và hiểu rõ công việc của nhau, từ đó tăng cường sự phối hợp và cộng tác.

Daily Stand-Up là một phần quan trọng của quy trình Agile, giúp nhóm phát triển duy trì sự liên lạc, phối hợp hiệu quả và phản ứng nhanh chóng với các vấn đề phát sinh. Việc thực hiện đúng cách và duy trì kỷ luật trong Daily Stand-Up sẽ giúp nhóm làm việc hiệu quả hơn và đạt được mục tiêu của Sprint.

Development Work là quá trình phát triển và hoàn thiện các tính năng, chức năng, và sản phẩm trong một dự án. Nó bao gồm các hoạt động lập trình, kiểm thử, thiết kế và tích hợp để tạo ra một sản phẩm có thể chuyển giao được vào cuối mỗi Sprint.

Các Hoạt Động Chính trong Development Work:

- Lập Trình (Coding)

Mục tiêu: Viết mã nguồn để triển khai các yêu cầu và tính năng đã được xác định trong Sprint Planning.

Hoạt động cụ thể:

Viết mã nguồn cho các tính năng mới.

Cải tiến mã nguồn hiện tại để tăng hiệu suất hoặc bổ sung tính năng.

Thực hiện các thay đổi nhỏ dựa trên phản hồi từ khách hàng hoặc người dùng.

- Thiết Kế (Design)

Mục tiêu: Đảm bảo rằng các giải pháp phát triển tuân theo nguyên tắc thiết kế tốt và đáp ứng yêu cầu của sản phẩm.

Hoạt động cụ thể:

Thiết kế giao diện người dùng (UI/UX).

Thiết kế kiến trúc phần mềm.



Tạo các sơ đồ thiết kế như UML, wireframe, và mockup.

- Kiểm Thử (Testing)

Mục tiêu: Đảm bảo chất lượng sản phẩm bằng cách phát hiện và sửa lỗi trước khi sản phẩm được phát hành.

Hoạt động cụ thể:

Kiểm thử đơn vị (Unit Testing): Kiểm tra các thành phần nhỏ nhất của mã nguồn.

Kiểm thử tích hợp (Integration Testing): Đảm bảo rằng các thành phần hoạt động tốt khi kết hợp với nhau.

Kiểm thử hệ thống (System Testing): Kiểm tra toàn bộ hệ thống để đảm bảo rằng nó đáp ứng yêu cầu đã định.

Kiểm thử chấp nhận (Acceptance Testing): Đảm bảo rằng sản phẩm đáp ứng được yêu cầu của khách hàng.

- Tích Hợp và Triển Khai (Integration and Deployment)

Mục tiêu: Đảm bảo rằng sản phẩm phát triển được tích hợp và triển khai một cách hiệu quả và an toàn.

Hoạt động cụ thể:

Tích hợp mã nguồn từ các thành viên trong nhóm vào một repository chung.

Sử dụng công cụ CI/CD (Continuous Integration/Continuous Deployment) để tự động hóa quá trình build và deploy.

Triển khai sản phẩm lên môi trường kiểm thử và sản xuất.

- Tài Liệu (Documentation)

Mục tiêu: Tạo và duy trì tài liệu cho sản phẩm và quá trình phát triển.

Hoạt động cụ thể:

Viết tài liệu hướng dẫn sử dụng.

Cập nhật tài liệu kiến trúc và thiết kế.

Ghi chép các quyết định quan trọng và lý do đưa ra các thay đổi.

Vai trò và trách nhiệm trong development work



- Nhóm Phát Triển (Development Team)

Trách nhiệm chính: Thực hiện các hoạt động phát triển để hoàn thành các mục tiêu đã đề ra trong Sprint.

Thành phần: Các lập trình viên, kiểm thử viên, chuyên gia thiết kế, và các chuyên gia khác cần thiết cho việc phát triển sản phẩm.

- Scrum Master

Trách nhiệm chính: Hỗ trợ nhóm phát triển bằng cách loại bỏ các rào cản và đảm bảo rằng nhóm tuân theo quy trình Agile.

Hỗ trợ: Đảm bảo rằng nhóm có môi trường làm việc thuận lợi và giải quyết các vấn đề phát sinh.

- Product Owner

Trách nhiệm chính: Quản lý Product Backlog và đảm bảo rằng các yêu cầu phát triển phù hợp với mục tiêu kinh doanh và nhu cầu của khách hàng.

Tham gia: Thường xuyên làm việc với nhóm phát triển để làm rõ các yêu cầu và ưu tiên công việc.

6. Kiểm tra và điều chỉnh

Trong phương pháp Agile, lập kế hoạch kiểm tra và điều chỉnh là một phần quan trọng của quy trình phát triển, giúp đảm bảo rằng sản phẩm luôn đáp ứng được yêu cầu của người dùng và các bên liên quan, cũng như thích ứng kịp thời với các thay đổi. Quá trình này thường bao gồm việc lập kế hoạch kiểm tra (test planning), thực hiện kiểm tra (testing), và điều chỉnh dựa trên kết quả kiểm tra (adjustment).

Lập kế hoạch kiểm tra (Test Planning)

Xác định phạm vi kiểm tra:

- Phạm vi kiểm tra: Xác định những phần của sản phẩm sẽ được kiểm tra, bao gồm các tính năng, chức năng, hiệu suất và bảo mật.



- Mục tiêu kiểm tra: Đặt ra các mục tiêu cụ thể cho quá trình kiểm tra, chẳng hạn như xác nhận tính đúng đắn của các tính năng mới, đảm bảo không có lỗi nghiêm trọng, hoặc kiểm tra hiệu suất dưới tải cao.

Chuẩn bị tài nguyên và công cụ:

- Tài nguyên kiểm tra: Xác định các tài nguyên cần thiết cho quá trình kiểm tra, bao gồm nhân lực, phần cứng, phần mềm và môi trường kiểm tra.
- Công cụ kiểm tra: Chọn các công cụ phù hợp để tự động hóa và quản lý quá trình kiểm tra, chẳng hạn như JUnit cho kiểm tra đơn vị, Selenium cho kiểm tra giao diện người dùng, và JIRA cho quản lý lỗi.

Lập kế hoạch kiểm tra chi tiết:

- Kế hoạch kiểm tra: Tạo một kế hoạch kiểm tra chi tiết, bao gồm các bước kiểm tra, kịch bản kiểm tra (test cases), và tiêu chí chấp nhận (acceptance criteria).
- Lịch trình kiểm tra: Đặt ra một lịch trình kiểm tra cụ thể, xác định thời gian và thứ tự thực hiện các bài kiểm tra.

Thực hiện kiểm tra (Testing)

Kiểm tra đơn vị (Unit Testing):

- Viết và thực hiện kiểm tra đơn vị: Các nhà phát triển viết các bài kiểm tra đơn vị để kiểm tra từng phần nhỏ của mã nguồn (unit) một cách độc lập.
- Tự động hóa kiểm tra đơn vị: Sử dụng các công cụ tự động hóa kiểm tra đơn vị để đảm bảo các bài kiểm tra được thực hiện thường xuyên và liên tục.

Kiểm tra tích hợp (Integration Testing):

- Kiểm tra tích hợp: Thực hiện kiểm tra tích hợp để đảm bảo rằng các module hoặc thành phần khác nhau của hệ thống hoạt động cùng nhau một cách đúng đắn.
- Tự động hóa kiểm tra tích hợp: Sử dụng các công cụ tự động hóa để kiểm tra tích hợp liên tục.

Kiểm tra hệ thống (System Testing):



- Kiểm tra hệ thống: Kiểm tra toàn bộ hệ thống để đảm bảo rằng nó hoạt động như mong đợi trong các điều kiện thực tế.
- Kiểm tra chức năng: Đảm bảo rằng các tính năng của hệ thống hoạt động đúng theo yêu cầu.
- Kiểm tra phi chức năng: Kiểm tra các yếu tố như hiệu suất, bảo mật, và khả năng mở rộng.

Kiểm tra chấp nhận (Acceptance Testing):

- Kiểm tra chấp nhận bởi người dùng: Người dùng hoặc các bên liên quan thực hiện kiểm tra chấp nhận để đảm bảo rằng hệ thống đáp ứng được các yêu cầu và mong đợi của họ.
- Kiểm tra chấp nhận tự động: Sử dụng các kịch bản kiểm tra chấp nhận tự động để đảm bảo rằng các tiêu chí chấp nhận được đáp ứng.

Điều chỉnh (Adjustment)

Phân tích kết quả kiểm tra:

- Xem xét kết quả kiểm tra: Phân tích kết quả của các bài kiểm tra để xác định các lỗi, vấn đề hoặc khu vực cần cải tiến.
- Báo cáo lỗi: Ghi lại các lỗi hoặc vấn đề phát hiện được trong một hệ thống quản lý lỗi (bug tracking system) để theo dõi và giải quyết.

Điều chỉnh kế hoạch và phát triển:

- Điều chỉnh kế hoạch phát triển: Dựa trên kết quả kiểm tra và phản hồi từ người dùng, điều chỉnh kế hoạch phát triển để ưu tiên các công việc cần thiết.
- Cải thiện mã nguồn: Sửa chữa các lỗi và thực hiện các cải tiến cần thiết để đảm bảo rằng sản phẩm luôn đạt được chất lượng cao nhất.

Lập kế hoạch kiểm tra tiếp theo:

- Cập nhật kế hoạch kiểm tra: Dựa trên các điều chỉnh và cải tiến, cập nhật kế hoạch kiểm tra để đảm bảo rằng các bài kiểm tra tiếp theo sẽ phù hợp và hiệu quả.



Lập kế hoạch kiểm tra và điều chỉnh trong Agile là một quá trình liên tục và lặp lại, giúp đảm bảo rằng sản phẩm luôn đáp ứng được yêu cầu của người dùng và các bên liên quan. Bằng cách lập kế hoạch kiểm tra chi tiết, thực hiện kiểm tra kỹ lưỡng và điều chỉnh dựa trên kết quả kiểm tra, nhóm phát triển có thể cải thiện chất lượng sản phẩm và đáp ứng tốt các yêu cầu thay đổi.

7. Chuyển giao sản phẩm và cải tiến liên tục

Lập kế hoạch chuyển giao sản phẩm và cải tiến liên tục trong Agile là một quy trình quan trọng giúp đảm bảo rằng sản phẩm được phát triển và phát hành hiệu quả, đồng thời liên tục cải tiến dựa trên phản hồi từ khách hàng và người dùng. Dưới đây là các bước chi tiết để lập kế hoạch chuyển giao sản phẩm và cải tiến liên tục trong Agile:

Lập Kế Hoạch Chuyển Giao Sản Phẩm (Release Planning)

Xác định mục tiêu và phạm vi phiên bản:

- Mục tiêu phiên bản: Xác định rõ mục tiêu của phiên bản sắp tới, bao gồm những tính năng và giá trị mà phiên bản mang lại cho người dùng.
- Phạm vi phiên bản: Định rõ các tính năng, cải tiến, và sửa lỗi sẽ được bao gồm trong phiên bản này.

Lập kế hoạch phiên bản chi tiết:

- Thời gian phiên bản: Xác định thời gian cụ thể cho phiên bản, bao gồm các Sprint cần thiết để hoàn thành các tính năng trong phạm vi.
- Ưu tiên công việc: Sắp xếp các mục trong Product Backlog theo thứ tự ưu tiên để đảm bảo rằng các tính năng có giá trị cao nhất được phát triển trước.

Tạo kế hoạch chuyển giao (Release Plan):

- Phân phối Sprint: Chia các tính năng và công việc cần thực hiện vào các Sprint cụ thể, đảm bảo rằng mỗi Sprint đều có mục tiêu rõ ràng và có thể đạt được.
- Đánh giá rủi ro: Xác định và đánh giá các rủi ro có thể ảnh hưởng đến quá trình phát triển và chuyển giao sản phẩm.



Thực Hiện Chuyển Giao Sản Phẩm

Phát triển và kiểm thử:

- Phát triển: Thực hiện phát triển các tính năng và yêu cầu theo kế hoạch đã định trong mỗi Sprint.
- Kiểm thử liên tục: Thực hiện kiểm thử tự động và thủ công liên tục trong suốt quá trình phát triển để đảm bảo chất lượng sản phẩm.

Kiểm tra tích hợp liên tục (Continuous Integration):

- Tích hợp mã nguồn: Liên tục tích hợp mã nguồn mới vào kho mã chính, sử dụng các công cụ CI/CD để tự động hóa quá trình build và kiểm thử.
- Phát hiện và sửa lỗi sớm: Phát hiện và sửa lỗi sớm trong quá trình phát triển, giúp giảm thiểu rủi ro và tăng cường chất lượng.

Chuẩn bị chuyển giao (Release Preparation):

- Tài liệu chuyển giao: Chuẩn bị các tài liệu hướng dẫn, tài liệu người dùng, và các thông tin cần thiết cho việc chuyển giao.
- Kiểm tra chấp nhận (Acceptance Testing): Thực hiện kiểm tra chấp nhận với người dùng hoặc khách hàng để đảm bảo rằng phiên bản đáp ứng các yêu cầu và mong đợi.

Chuyển giao sản phẩm:

- Triển khai sản phẩm: Triển khai phiên bản sản phẩm vào môi trường sản xuất.
- Đào tạo và hỗ trợ: Cung cấp đào tạo và hỗ trợ cho người dùng cuối và khách hàng để đảm bảo rằng họ có thể sử dụng sản phẩm hiệu quả.

Cải Tiết Liên Tục (Continuous Improvement)

Thu thập phản hồi:

- Phản hồi từ người dùng: Thu thập phản hồi từ người dùng và khách hàng về phiên bản mới nhất của sản phẩm.
- Phân tích phản hồi: Phân tích phản hồi để xác định các vấn đề, yêu cầu mới và cơ hội cải tiến.

Đánh giá và điều chỉnh Product Backlog:



- Cập nhật Product Backlog: Cập nhật Product Backlog dựa trên phản hồi và phân tích, thêm các tính năng mới, cải tiến và sửa lỗi cần thiết.
- Ưu tiên lại các mục Backlog: Điều chỉnh thứ tự ưu tiên của các mục trong Product Backlog để phản ánh đúng nhu cầu và giá trị hiện tại.

Thực hiện cải tiến liên tục:

- Refactoring: Liên tục cải tiến mã nguồn và kiến trúc hệ thống để tăng cường hiệu suất, bảo trì và khả năng mở rộng.
- Cải tiến quy trình: Xem xét và cải tiến quy trình làm việc, công cụ và phương pháp để nâng cao hiệu quả và chất lượng công việc.

Sprint Retrospective:

- Cuộc họp hồi cứu Sprint: Sau mỗi Sprint, tổ chức cuộc họp hồi cứu (Sprint Retrospective) để xem xét những gì đã làm tốt, những gì cần cải thiện và lập kế hoạch cho các hành động cải tiến cụ thể.
- Hành động cải tiến: Thực hiện các hành động cải tiến được xác định trong cuộc họp hồi cứu để nâng cao hiệu quả và chất lượng của các Sprint tiếp theo.

Lập kế hoạch chuyển giao sản phẩm và cải tiến liên tục trong Agile là một quy trình linh hoạt và lặp lại, giúp nhóm phát triển không chỉ tạo ra sản phẩm có chất lượng cao mà còn đảm bảo rằng sản phẩm luôn đáp ứng được nhu cầu và mong đợi của người dùng và các bên liên quan. Bằng cách thực hiện các bước này một cách hiệu quả, nhóm phát triển có thể duy trì sự linh hoạt, tối ưu hóa quy trình làm việc và không ngừng cải thiện sản phẩm.

8. Tái lập kế hoạch và điều chỉnh

Tái lập kế hoạch và điều chỉnh là những yếu tố quan trọng trong phương pháp Agile giúp đảm bảo dự án luôn đi đúng hướng và đáp ứng được các yêu cầu thay đổi của khách hàng.

Tái lập kế hoạch (Replanning)



Sprint Planning (Lập kế hoạch Sprint):

- Đánh giá Sprint trước: Trước khi bắt đầu Sprint mới, nhóm sẽ đánh giá kết quả của Sprint trước đó để xác định những gì đã hoàn thành, những gì chưa hoàn thành và lý do tại sao.
- Điều chỉnh Sprint Backlog: Dựa trên đánh giá và phản hồi, nhóm sẽ điều chỉnh Sprint Backlog để đảm bảo các mục tiêu cho Sprint mới là khả thi và ưu tiên đúng đắn.
- Xác định mục tiêu Sprint: Mục tiêu của Sprint mới được xác định dựa trên các yêu cầu hiện tại và khả năng của nhóm, đảm bảo tính linh hoạt và khả năng đáp ứng các thay đổi.

Product Backlog Refinement (Tinh chỉnh Product Backlog):

- Liên tục xem xét và cập nhật: Product Owner và nhóm phát triển liên tục xem xét và cập nhật Product Backlog để đảm bảo các mục công việc được xác định rõ ràng, chi tiết và có độ ưu tiên phù hợp.
- Collaboration: Tương tác thường xuyên giữa Product Owner và nhóm phát triển để đảm bảo Backlog phản ánh đúng yêu cầu thực tế và các thay đổi từ khách hàng.

Điều chỉnh (Adjustments)

Daily Stand-ups (Cuộc họp hàng ngày):

- Cập nhật tiến độ: Mỗi ngày, nhóm sẽ cập nhật về công việc đã hoàn thành, công việc sẽ làm và các trở ngại đang gặp phải. Điều này giúp nhóm nhanh chóng nhận ra các vấn đề và điều chỉnh kế hoạch nếu cần.
- Xác định trở ngại: Các trở ngại được nêu ra trong cuộc họp hàng ngày sẽ được giải quyết nhanh chóng để không làm gián đoạn tiến độ của nhóm.

Sprint Review (Đánh giá Sprint):

- Trình bày increment: Trong cuộc họp Sprint Review, nhóm trình bày increment (phần tăng trưởng) của sản phẩm cho các bên liên quan và nhận phản hồi từ họ.



- Điều chỉnh dựa trên phản hồi: Dựa trên phản hồi từ Sprint Review, nhóm có thể điều chỉnh Product Backlog và kế hoạch cho các Sprint tiếp theo để đảm bảo sản phẩm đáp ứng được nhu cầu của khách hàng.

Sprint Retrospective (Cuộc họp hồi cứu):

- Xem xét quy trình làm việc: Trong cuộc họp hồi cứu, nhóm xem xét lại quy trình làm việc và kết quả đạt được trong Sprint vừa qua.
- Xác định cải tiến: Nhóm sẽ xác định những điểm cần cải thiện và đưa ra các giải pháp cụ thể để nâng cao hiệu quả làm việc trong Sprint tới.
- Action Items: Các hành động cải tiến được xác định và theo dõi để đảm bảo chúng được thực hiện trong các Sprint tiếp theo.

Adaptation (Điều chỉnh liên tục):

- Tự động hóa và công cụ: Sử dụng các công cụ tự động hóa và kỹ thuật để liên tục kiểm tra và điều chỉnh sản phẩm, giảm thiểu lỗi và tăng tốc độ phát hành.
- Feedback Loop: Thiết lập vòng phản hồi liên tục với khách hàng và các bên liên quan để nhanh chóng nhận diện và điều chỉnh các vấn đề phát sinh.

Tại Sao Phải Tái Lập Kế Hoạch và Điều Chỉnh trong Agile?

- Thích Ứng VỚI THAY ĐỔI:

Thay đổi yêu cầu: Yêu cầu của khách hàng có thể thay đổi trong quá trình phát triển, việc tái lập kế hoạch giúp nhóm phát triển thích ứng nhanh chóng với các thay đổi này.

Thị trường biến động: Thị trường và môi trường kinh doanh có thể thay đổi, yêu cầu nhóm phải điều chỉnh kế hoạch để phù hợp với tình hình mới.

- TỐI ƯU HÓA NGUỒN LỰC:

Quản lý tài nguyên: Tái lập kế hoạch giúp đảm bảo rằng tài nguyên (nhân lực, thời gian, công cụ) được sử dụng một cách hiệu quả và tối ưu.

Giải quyết trở ngại: Nhận diện và giải quyết các vấn đề kịp thời để duy trì tiến độ và hiệu suất công việc.



- Nâng Cao Chất Lượng Sản Phẩm:

Phản hồi liên tục: Thu thập và áp dụng phản hồi từ khách hàng và các bên liên quan để cải thiện chất lượng sản phẩm.

Cải tiến quy trình: Đánh giá và cải tiến quy trình làm việc để nâng cao hiệu quả và chất lượng công việc.

- Đảm Bảo Sự Minh Bạch và Trách Nhiệm:

Cập nhật liên tục: Các cuộc họp hàng ngày và hồi cứu Sprint giúp duy trì sự minh bạch trong nhóm và đảm bảo rằng mọi người đều biết về tiến độ và các vấn đề cần giải quyết.

Trách nhiệm: Tăng cường trách nhiệm của từng thành viên trong nhóm đối với công việc của mình và đối với nhóm.

- Đạt Được Mục Tiêu Kinh Doanh:

Tối ưu giá trị: Đảm bảo rằng các tính năng và công việc được ưu tiên và thực hiện dựa trên giá trị mang lại cho khách hàng và doanh nghiệp.

Định hướng mục tiêu: Giúp nhóm phát triển luôn tập trung vào các mục tiêu kinh doanh quan trọng và điều chỉnh hướng đi khi cần thiết.

Tái lập kế hoạch và điều chỉnh là những quy trình liên tục và linh hoạt trong phương pháp Agile, giúp đảm bảo dự án luôn đáp ứng được các yêu cầu thay đổi và duy trì chất lượng cao. Bằng cách sử dụng các cuộc họp hàng ngày, đánh giá Sprint, hồi cứu và tinh chỉnh Product Backlog, nhóm có thể nhanh chóng nhận diện và giải quyết các vấn đề, đảm bảo tiến độ và hiệu quả của dự án.