**CHƯƠNG I: MỘT SỐ MÔ HÌNH TRUYỀN THỐNG**

**I. MÔ HÌNH AGILE**

**1. Khái niệm:**

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. Agile là cách thức làm phần mềm linh hoạt để làm sao đưa sản phẩm đến tay người dùng càng nhanh càng tốt càng sớm càng tốt và được xem như là sự cải tiến so với những mô hình cũ như mô hình “Thác nước (waterfall)” hay “CMMI”.

**2. Bốn tôn chỉ của Agile:**

**Agile hoạt động dựa trên những tôn chỉ sau:**

**1 - Cá nhân và sự tương hỗ quan trọng hơn quy trình và công cụ:**

**2 - Phần mềm sử dụng được quan trọng hơn tài liệu về sản phẩm**

**3 - Cộng tác với khách hàng quan trọng hơn đàm phán hợp đồng**

**4 - Phản hồi với sự thay đổi quan trọng hơn bám theo kế hoạch**

->Khách hàng là thượng đế.

-> Đề cao vai trò của khách hàng và các thành viên trong nhóm.

**3. 12 nguyên tắc của Agile**

-- **Nguyên tắc 1:** Thỏa mãn yêu cầu của khách hàng thông qua việc giao hàng sớm và liên tục

-- **Nguyên tắc 2:** Giao phần mềm chạy được cho khách hàng một cách thường xuyên (giao hàng tuần hơn là hàng tháng)

--> Mục đích của dự án phát triển phần mềm là phát triển phần mềm và làm khách hàng hài lòng và không có gì làm khách hàng hài lòng hơn việc cho khách hàng thấy được sản phẩm của mình thường xuyên và chạy được. Trong Agile, sản phẩm sẽ được demo cho khách hàng thường xuyên (thường là hàng tuần) để cho khách hàng thấy được sản phẩm của mình như thế nào. Nếu có chỗ nào không ổn hay cần cải tiến thì sẽ phản hồi với đội dự án ngay lập tức.

-- **Nguyên tắc 3:** Chào đón việc thay đổi yêu cầu, thậm chí là những thay đổi yêu cầu muộn

--> Dù bạn thích hay không thì việc thay đổi yêu cầu từ khách hàng là dường như không thể tránh khỏi và nhiệm vụ của bạn là phải thích ứng với sự thay đổi đó. Thích ứng không có nghĩa là khách hàng yêu cầu gì mình làm cái đó. Khi khách hàng thay đổi yêu cầu, chắc chắn phải có lý do của họ và nhiệm vụ của đội dự án là phải hiểu được lí do đó để có thể điều chỉnh sự thay đổi, tư vấn hay đề nghị giải pháp cho khách hàng tương ứng

-- **Nguyên tắc 4:** Khách hàng của dự án và đội phát triển phải làm việc cùng nhau hàng ngày trong suốt dự án

--> Đội phát triển phải làm việc thường xuyên và gần gũi với khách hàng để hiểu được nhu cầu của họ cũng như cho phép khách hàng hiểu về công việc của đội phát triển. Đó là một trong những lí do mà Agile dường như khó triển khai trong các dự án offshore trong đó rào cản về địa lý, thời gian, ngôn ngữ là một trở ngại lớn.

-- **Nguyên tắc 5:** Các dự án được xây dựng xung quanh những cá nhân có động lực. Cung cấp cho họ môi trường và sự hỗ trợ cần thiết, và tin tưởng họ để hoàn thành công việc

--> Như đã chia sẻ ở trên, Agile đặt trọng tâm là con người. Con người ở đây chỉ những cá nhân có động lực làm việc với tinh thần cộng tác, chia sẻ và giúp đỡ lẫn nhau. Song song đó, đội dự án sẽ được hỗ trợ những công cụ, môi trường, sự tin tưởng và những đãi ngộ cần thiết để hoàn thành công việc.

-- **Nguyên tắc 6:** Trao đổi trực tiếp mặt đối mặt là phương pháp hiệu quả nhất để truyền đạt thông tin

-- **Nguyên tắc 7:** Phần mềm chạy được là thước đo chính của tiến độ

-- **Nguyên tắc 8:** Phát triển bền vững và duy trì được nhịp độ phát triển liên tục

-- **Nguyên tắc 9:**Liên tục quan tâm đến kĩ thuật và thiết kế để cải tiến sự linh hoạt

-- **Nguyên tắc 10:**Sự đơn giản là cần thiết – nghệ thuật tối đa hóa lượng công việc chưa hoàn thành

-- **Nguyên tắc 11:**Nhóm tự tổ chức

-- **Nguyên tắc 12:**Thích ứng thường xuyên với sự thay đổi

**CÁC GIAI ĐOẠN:**

**1. Tiền-Phân-đoạn (Pre-iteration):####**

Yêu cầu được phân tích chi tiết bởi BA (Business Analyst – chuyên viên phân tích nghiệp vụ) và các tiêu chí chấp nhận (acceptance criteria). Và QA sử dụng các yêu cầu này ngay từ đầu, ta cần phải xác minh (verify) những yêu cầu đó từ sớm và thường xuyên.

**2. Xác minh Yêu cầu:**

Kiểm thử Agile thiên về việc đưa ra phản hồi sớm, và phải bắt đầu bằng việc kiểm tra các yêu cầu từ sớm bởi QA hoặc tester để làm sáng rõ ý nghĩa và tính khả-kiểm (testability). Việc này sẽ đảm bảo các yêu cầu luôn rõ ràng và có thể kiểm thử được.

* Yêu cầu cần đủ nhỏ để có ý nghĩa trong bối cảnh xác định
* Tiêu chí chấp nhận (acceptance criteria những story thường được sử dụng cho các tiêu chí chấp nhận) không nên bị trùng lặp, chồng chéo từ những story khác nhau
* Các giai đoạn:
  + user story: là một tóm tắt đơn giản, ngắn gọn về chức năng mà khách hàng mong muốn.
  + Tiêu chí chấp nhận (Acceptance Criteria): là những tiêu chí dùng để đánh giá sản phẩm, chức năng đã thực hiện đúng yêu cầu hay chưa? Có thể coi đó là các tiêu chí xác nhận hoàn thành story.

Các tiêu chí đặt ra phải đáp ứng các đặc tính sau:

* + - Tính khả dụng (usability): là tiêu chí trả lời cho câu hỏi: Có dễ sử dụng hay không?
    - Tính chức năng (Functionality)
    - Xử lý lỗi (error handing) : Liệt kê ra những lỗi có thể gặp phải trong quá trình sử dụng chương trình và phương thức để xử lý. Ví dụ người dùng có thể thực hiện sai thứ tự các bước và khi đó chương trình sẽ xử lý như thế nào?
    - Hiệu suất( Performance)
    - Stress tests: Là tiêu chí trả lời cho câu hỏi: Hệ thống sẽ hoạt động như thế nào dưới những áp lực như có nhiều người truy cập tại cùng 1 thời điểm, có quá nhiều request được gửi đến hệ thống…

Để đạt được mục tiêu của giai đoạn này cần có sự giao tiếp chặt chẽ giữa các bên Đội Phát triển / Nhà phân tích nghiệp vụ/ Đảm bảo chất lượng.

* + - Khả kiểm (Testable): Các khía cạnh có thể kiểm thử được phải được xem xét chi tiết. Những yếu tố này thường là:
      * Tìm kiếm các yêu cầu ẩn
      * Môi trường
      * Dữ liệu kiểm thử (test data)
      * Sự phụ thuộc vào các yêu cầu khác

### **5. Đặc trưng của agile:**

* **Tính lặp**(Iterative): Dự án sẽ được thực hiện trong các phân đoạn lặp đi lặp lại (Iteration hoặc Sprint), thường có khung thời gian ngắn (từ 1-4 tuần). Trong mỗi phân đoạn, nhóm phát triển thực hiện đầy đủ các công việc cần thiết như lập kế hoạch, phân tích yêu cầu, thiết kế, triển khai, kiểm thử để cho ra các phần nhỏ của sản phẩm.
* **Tính tăng trưởng và tiến hóa** (Incremental & Evolutionary): Cuối các phân đoạn, nhóm cho ra các phần nhỏ của sản phẩm cuối cùng, thường là đầy đủ, có khả năng chạy tốt, được kiểm thử cẩn thận và có thể sử dụng. Theo thời gian, phân đoạn này tiếp nối phân đoạn kia, các phần chạy được này sẽ được tích lũy, lớn dần lên cho tới khi toàn bộ yêu cầu của khách hàng được thỏa mãn.
* **Tính thích nghi**(adaptive): Do các phân đoạn chỉ kéo dài trong một khoảng thời gian ngắn và việc lập kế hoạch cũng được điều chỉnh liên tục, nên các thay đổi trong quá trình phát triển (yêu cầu thay đổi, thay đổi công nghệ, thay đổi định hướng về mục tiêu v.v.) đều có thể được đáp ứng theo cách thích hợp.
* **Nhóm tự tổ chức và liên chức năng**: Các cấu trúc nhóm này tự phân công công việc mà không dựa trên các mô tả cứng về chức danh hay làm việc dựa trên một sự phân cấp rõ ràng trong tổ chức. Nhóm tự tổ chức đã đủ các kĩ năng cần thiết để có thể được trao quyền tự ra quyết định, tự quản lí và tổ chức lấy công việc của chính mình để đạt được hiệu quả cao nhất.
* **Quản lý tiến trình thực nghiệm** (Empirical Process Control): Các nhóm Agile ra các quyết định dựa trên các dữ liệu thực tiễn thay vì tính toán lý thuyết hay các tiền giả định. Agile rút ngắn vòng đời phản hồi để dễ dàng thích nghi và gia tăng tính linh hoạt nhờ đó có thể kiểm soát được tiến trình, và nâng cao năng suất lao động.
* **Giao tiếp trực diện**(face-to-face communication): Agile không phản đối việc tài liệu hóa, nhưng đánh giá cao hơn việc giao tiếp trực diện thay vì thông qua giấy tờ. Agile khuyến khích nhóm phát triển trực tiếp nói chuyện để hiểu rõ hơn về cái khách hàng thực sự cần. Trong giao tiếp giữa nội bộ nhóm, Agile khuyến khích trực tiếp trao đổi và thống nhất với nhau về thiết kế của hệ thống và cùng nhau triển khai thành các chức năng theo yêu cầu.
* **Phát triển dựa trên giá trị** (value-based development): Một trong các nguyên tắc cơ bản của agile là “sản phẩm chạy tốt chính là thước đo của tiến độ”. Nhóm Agile thường cộng tác trực tiếp và thường xuyên với khách hàng để biết yêu cầu nào có độ ưu tiên cao hơn, mang lại giá trị hơn sớm nhất có thể cho dự án.

### **6. Ưu điểm:**

### 

* **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.

### **7. Nhược điểm:**

* **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.

## **III. MÔ HÌNH THÁC NƯỚC**

**1. Khái niệm**

**Mô hình thác nước** trong tiếng Anh là **Waterfall Model.**

*Mô hình thác nước*là một trong những mô hình quản lí dự án dễ hiểu và dễ quản lí nhất hiện nay.*Mô hình thác nước*là một phương pháp quản lí dự án dựa trên qui trình thiết kế tuần tự và liên tiếp.

Trong *mô hình thác nước*, các giai đoạn của dự án được thực hiện lần lượt và nối tiếp nhau, giai đoạn mới chỉ được bắt đầu khi giai đoạn trước nó đã được hoàn thành.

Phương pháp thác nước vốn được tạo với mục đích là một mô hình chu kì vòng đời phát triển phần mềm, nhưng cũng được ứng dụng trong các dự án có qui mô nhỏ và có mọi tiêu chuẩn được xác định rõ ràng từ ban đầu.

### Các giai đoạn trong mô hình thác nước

Một**mô hình thác nước** đơn giản có 6 giai đoạn: yêu cầu, thiết kế, thực hiện (xây dựng), kiểm chứng, triển khai và bảo trì.

*Giai đoạn yêu cầu*: Nhóm thực hiện tìm kiếm các yêu cầu liên quan đến dự án, ví dụ như xác định dự án sẽ giải quyết nhu cầu kinh doanh nào, yêu cầu của người dùng đối với sản phẩm được phát triển bởi dự án, các ràng buộc và rủi ro đi kèm.

*Giai đoạn thiết kế*: Nhóm tạo ra thiết kế cho sản phẩm để giải quyết mọi yêu cầu, ràng buộc và mục tiêu thiết kế.

*Giai đoạn thực hiện* hay *giai đoạn xây dựng*: Sản phẩm được chế tạo để hỗ trợ thiết kế. Đôi khi, sản phẩm được xây dựng trong các đơn vị dùng để thí nghiệm và tích hợp trong giai đoạn tiếp theo.

*Giai đoạn kiểm chứng*: các bộ phận của sản phẩm được kiểm tra và nếu cần, sẽ được tích hợp lại với nhau để thử nghiệm. Toàn bộ hệ thống được kiểm tra để tìm ra lỗi và để đảm bảo các mục tiêu thiết kế và chức năng người dùng yêu cầu đã được đáp ứng, và nhu cầu kinh doanh được giải quyết.

*Giai đoạn triển khai*: sản phẩm được thử nghiệm thực sự đi vào hoạt động. Đối với các dự án thuộc lĩnh vực công nghệ thông tin, sản phẩm được triển khai vào môi trường để người dùng có thể bắt đầu sử dụng nó. Đối với một dự án xây dựng, giai đoạn triển khai là khi tòa nhà hoàn toàn sẵn sàng cho người ở.

*Giai đoạn bảo trì*: là một khoảng thời gian giám sát ngắn trong đó nhóm dự án giải quyết các vấn đề của khách hàng. Đối với các dự án phần mềm, điều này thường có nghĩa phát hành các bản vá và cập nhật để sửa vấn đề. Trong các dự án khác, các điều chỉnh về môi trường được thực hiện để giải quyết vấn đề, chẳng hạn như tối ưu hóa điều hòa không khí trong một tòa nhà mới.

#### Ưu nhược điểm của mô hình thác nước

**Ưu điểm:**

- Thích hợp cho các dự án đơn giản hoặc nhỏ

- Nhóm dự án hiểu rõ các yêu cầu

- Dễ hiểu

- Dễ quản lí

- Không cần đề ra các mốc thực hiện

- Có nhiều tài liệu cung cấp thông tin cho khách hàng

**Nhược điểm:**

- Không cho phép thay đổi [phạm vi dự án](https://vietnambiz.vn/pham-vi-du-an-project-scope-la-gi-thong-nhat-pham-vi-du-an-20191017141616478.htm)

- Không cho phép thay đổi yêu cầu của dự án

- Không có sản phẩm hoạt động được cho đến khi dự án gần hoàn thành

- Không dễ xử lí rủi ro bất ngờ

**VI. MÔ HÌNH CHỮ V**

# I. Mô hình phát triển phần mềm V là gì?

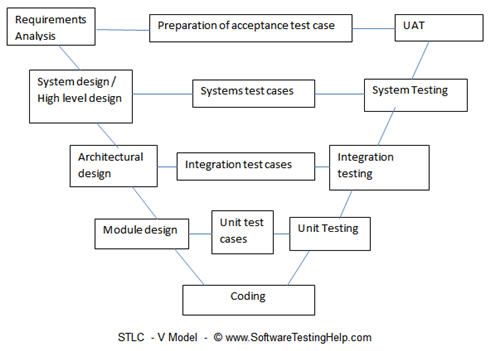
Mô hình V hiện nay là một trong những quy trình phát triển phần mềm được sử dụng rộng rãi nhất. Trong mô hình V việc thực hiện kiểm tra được diễn ra ngay từ giai đoạn lấy yêu cầu. V mô hình cũng được gọi là mô hình xác minh (verification) và mô hình xác nhận (validation).

Để hiểu được mô hình V, trước hết chúng ta hãy hiểu xác minh (verification) và xác nhận hợp lệ (validation) trong phần mềm là gì.

**Xác minh (verification) :** Xác minh là một kỹ thuật phân tích tĩnh. Trong kiểm thử, kỹ thuật này được thực hiện mà không phải chạy code. Nó bao gồm một số hoạt đông như xem lại (review), kiểm tra (inspection) và kiểm tra từ đầu tới cuối (walkthrough).

**Xác nhận (validation):** Xác nhận là một kỹ thuật phân tích động, trong đó việc kiểm thử được thực hiện bằng cách thực hiện code. Ví dụ bao gồm kỹ thuật kiểm tra chức năng (function) và phi chức năng (non-function).

Trong mô hình V, các hoạt động phát triển và đảm bảo chất lượng được thực hiện đồng thời. Không có pha rời rạc được gọi là kiểm thử, thay vào đó 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:



Trong một quá trình phát triển theo mô hình V điển hình, bên trái là các hoạt động phát triển và bên tay phải là các hoạt động kiểm thử. Trong giai đoạn phát triển cả việc xác minh và xác nhận được thực hiện cùng với các hoạt động phát triển thực tế. Bây giờ chúng ta hãy tìm hiểu bức tranh phía trên rõ hơn:

## Phía bên tay trái:

Như đã nói ở trên, hoạt động phía bên trái là hoạt động phát triển. Thông thường, chúng ta khó có thể thực hiện thử nghiệm trong giai đoạn phát triển, nhưng đây là ưu điểm của mô hình này, nó chứng tỏ rằng thử nghiệm cũng có thể được thực hiện ở tất cả các giai đoạn của các hoạt động phát triển.

### 1. Phân tích yêu cầu:

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. Brain storming/walkthrough, interviews 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: Tài liệu hiểu về yêu cầu, UAT test case.

### 2. 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 trân truy vết (traceability metrics)
* Đầu ra cần có: System test cases, Feasibility reports, System test plan, tài liệu về yêu cầu phần cứng và các mô đun, ...

### 3. Thiết kế kiến trúc:

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,...

### 4. Thiết kế mô đun / Thiết kế cấp thấp:

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

### 5. Code:

Trong giai đoạn này, code được 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.

## Bên tay phải:

Phía bên phải thể hiện các hoạt động kiểm thử hoặc Giai đoạn Xác nhận. Chúng ta sẽ bắt đầu đi từ phía dưới.

### 1. Kiểm tra đơn vị (unit test):

Trong giai đoạn này tất cả unit test case đã được tạo ra trong giai đoạn thiết kế cấp thấp sẽ được thực hiện.

* Unit test là một kỹ thuật kiểm tra hộp trắng, nơi một đoạn code được viết, nó sẽ tích hợp một phương pháp (hoặc một đoạn code khác) để kiểm tra xem đoạn code có cho kết quả mong muốn hay không. Thử nghiệm này về cơ bản được thực hiện bởi nhóm phát triển. Trong trường hợp có bất thường, bug được ghi lại và theo dõi.
* Đầu ra cần có: Kết quả thực hiện unit test

### 2. Kiểm tra tích hợp:

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.

* Kiểm thử tích hợp (Integration Testing): Kiểm thử tích hợp là một kỹ thuật mà đơn vị kiểm tra là mô-đun được tích hợp và kiểm tra xem các mô-đun tích hợp vào có cho kết quả mong đợi không. Nói một cách đơn giản, nó xác nhận xem các thành phần của ứng dụng có hoạt động với nhau như mong đợi hay không.
* Đầu ra cần có: Các kết quả kiểm tra tích hợp.

### 3. Kiểm tra hệ thống (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 metrics) được cập nhật để kiểm tra mức độ bao phủ và rủi ro được giảm bớt.

* Đầu ra cần có: Kết quả kiểm tra, Các bản ghi kiểm tra, báo cáo lỗi, báo cáo tóm tắt kiểm tra và các ma trận truy xuất cập nhật.

### 4. Thử nghiệm chấp nhận của người dùng:

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.

* Đầu ra cần có: Kết quả UAT, Ma trận độ bao phủ business được cập nhật.

# II. Khi nào sử dụng V Model?

V được áp dụng khi:

* 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.

# III. Ưu điểm và nhựơc điểm khi sử dụng mô hình V:

## 1. Ưu điểm:

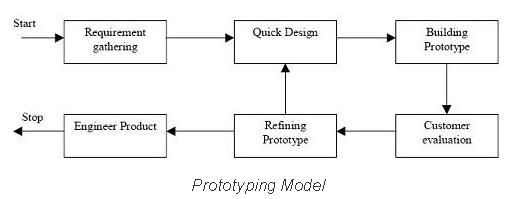
* Quá trình phát triển và quy trình quản lý có tính tổ chức và hệ thống
* Hoạt động tốt cho các dự án có quy mô vừa và nhỏ.
* Kiểm tra bắt đầu từ khi bắt đầu phát triển vì vậy sự mơ hồ được xác định ngay từ đầu.
* Dễ dàng quản lý vì mỗi giai đoạn có các mục tiêu và mục tiêu được xác định rõ ràng.

## 2. Nhược điểm:

* 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.

# 1. Prototype model là gì?

* Prototype là một mô hình phát triển phần mềm được phát triển dựa trên các yêu cầu hệ thống. Dựa vào bản prototype mà khách hàng có cái nhìn tổng quan về hệ thống thực tế.
* Prototype là một ý tưởng hay cho các hệ thống phức tạp và lớn mà không có quy trình thủ công để giúp xác định các yêu cầu.
* Prototype thường không phải là hệ thống hoàn chỉnh và nhiều chi tiết không được xây dựng trong bản prototype. Mục tiêu là cung cấp một hệ thống với chức năng tổng thể.

Sơ đồ mô hình prototype:

# 2. Ưu điểm của prototype model

* Người dùng tích cực tham gia vào quá trình phát triển phần mềm từ đó sẽ hiểu rõ hơn về hệ thống đang được phát triển
* Lỗi có thể được phát hiện sớm và được fix sớm hơn
* Người dùng sẽ dễ dàng nhận thấy những điểm bất cập và đưa ra các giải pháp để giải quyết sớm hơn
* Tránh tình trạng bị thiếu mất chức năng của hệ thống
* Những chức năng phức tập gây khó hiểu được xác định và giải thích sớm hơn
* Xác thực được yêu cầu của hệ thống từ đó thực hiện nhanh hơn nhưng không đầy đủ chức năng của ứng dụng

# 3. Nhược điểm của prototype

* Trên thực tế bản prototype có thể làm tăng sự phức tạp của hệ thống vì phạm vi của hệ thống có thể mở rộng hơn các kế hoạch ban đầu
* Bản mẫu này được chính khách hàng nhận xét đánh giá, nếu bản prototype mà khách hàng không hài lòng phải làm lại gây tốn thời gian và công sức.

# 4. Khi nào sử dụng prototype

* Mô hình prototype nên được sử dụng khi hệ thống cần có nhiều tương tác với người dùng cuối
* Thông thường, các hệ thống trực tuyến, giao diện web có lượng tương tác rất cao với người dùng cuối, phù hợp nhất với mô hình Prototype.
* Bản prototype phải đảm bảo rằng người dùng cuối liên tục tương tác với hệ thống và cung cấp phản hồi cần cải tiến về bản prototype đó để tạo ra một hệ thống có thể sử dụng được.

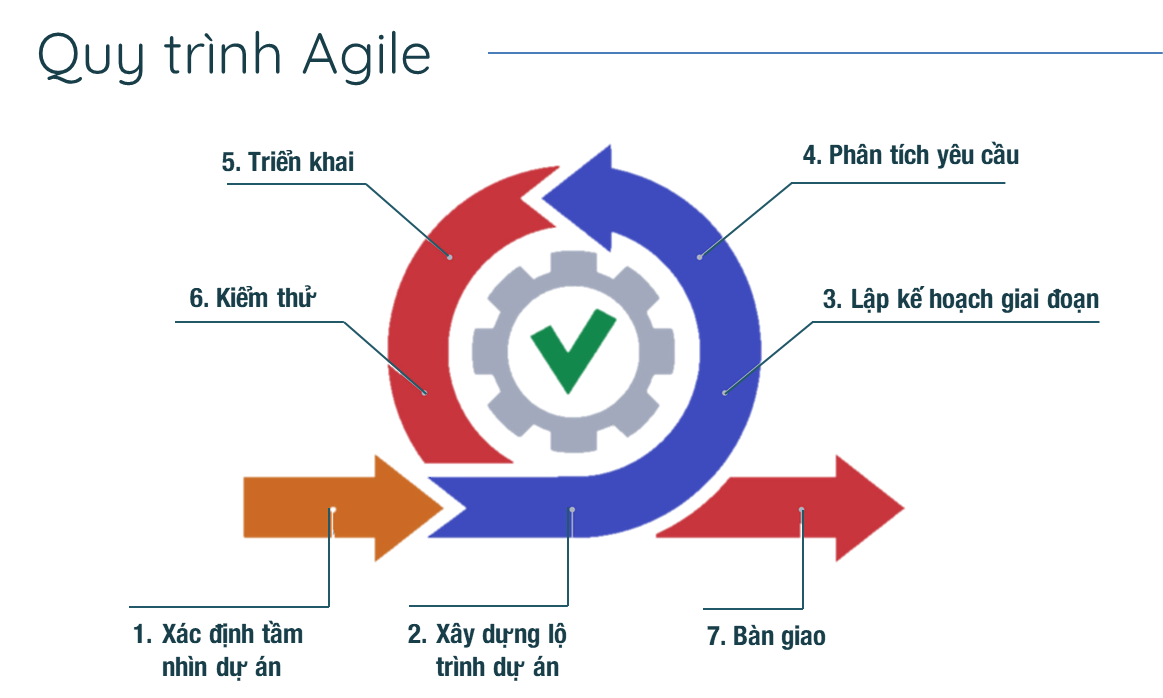
CHƯƠNG II: ÁP DỤNG SCRUM VÀO VIỆC QUẢN LÝ DỰ ÁN

1.QUY TRÌNH AGILE:

### **Một quy trình agile hoàn chỉnh:**

Các giai đoạn phát triển của sản phẩm sẽ được chia nhỏ ra thành những phần tăng trưởng cụ thể mà người dùng có thể tương tác được. Nhờ đó sản phẩm sẽ có được phản hồi cần thiết để tránh khỏi những vấn đề nghiêm trọng và được cải tiến tốt hơn.

Thêm vào đó, quy trình quản lý sản phẩm có tính chất lặp lại này còn giúp cho cả nhóm có thể chuyển sang một phần tăng trưởng khác trong khi những vấn đề của phần tăng trưởng hiện tại đang được giải quyết.



2.SCRUM

**MÔ HÌNH SCRUM**

**1. Khái niệm:**

**Scrum** là một “bộ khung làm việc” cơ bản để tiếp cận những công việc phức tạp. Dựa trên bộ khung này, nhóm làm việc có thể áp dụng những quy trình, kỹ thuật khác nhau cho công việc của mình… Nó là một thành viên của họ [Agile](https://topdev.vn/blog/agile-la-gi/).

***Nhắc đến Scrum là:***

* Nhẹ nhàng
* Dễ hiểu
* Khó để quản lý và làm chủ

## **2. Ba giá trị cốt lõi của Scrum**

### **a. Minh bạch**

Muốn áp dụng thành công Scrum, các thông tin liên quan đến quá trình phải mình bạch và thông suốt. Các thông tin có thể là tầm nhìn của sản phẩm, yêu cầu của khách hàng, tiến độ công việc, các rào cản khác…

Từ đó mọi thành viên ở vai trò khác nhau có đầy đủ thông tin cần có để tiến hành quyết định trong việc nâng cao hiệu quả công việc.

### **b. Thanh tra**

Phải thường xuyên thanh tra các hoạt động trong Scrum và tiến độ đến đích để phát hiện các bất thường không theo ý muốn. Tần suất thanh tra không nên quá dày để khỏi ảnh hưởng đến công việc. Công tác thanh tra khi được thực hiện bởi người có kĩ năng tại các điểm quan trọng của công việc sẽ giúp cải tiến liên tục trong Scrum.

### **c. Thích nghi**

Scrum mang lợi thế là tính linh hoạt rất cao, nhờ đó mang lại tính thích nghi cao. Dựa vào thông tin liên tục và minh bạch từ quá trình thanh tra và làm việc, Scrum có thể cho lại các thay đổi tích cực, nhờ đó mang lại thành công cho dự án.

## **3. LỢI ÍCH MÀ SCRUM MANG LẠI**

Tính minh bạch, kiểm tra, và thích nghi là 3 nền tảng cơ bản của Scrum. Và dưới đây là những lý do tại sao nên dùng Scrum.

1. Cải thiện chất lượng phần mềm, dễ học và dễ sử dụng.
2. Rút ngắn thời gian phát hành phần mềm, cho phép khách hàng sử dụng sản phẩm sớm hơn.
3. Nâng cao tinh thần đồng đội, tối ưu hóa hiệu quả và nỗ lực của đội phát triển.
4. Gia tăng tỷ suất hoàn vốn đầu tư (ROI)
5. Tăng mức độ hài lòng của khách hàng
6. Kiểm soát dự án tốt, cải tiến liên tục
7. Giảm thiểu rủi ro khi xây dựng sản phẩm

## **4. Các khái niệm cơ bản của Scrum**

### **1. Scrum Team**

Scrum team chia làm 3 vai trò bao gồm những thành phần sau:

* **Product Owner**: Nhiệm vụ của Product Owner là đảm bảo việc quản lý những công việc còn tồn đọng (**Product backlog**) của việc phát triển sản phẩm phần mềm. Product Owner phải liên tục cập nhật thông tin cho các thành viên trong team để họ hiểu về yêu cầu hay các tính năng cần có của sản phẩm ngay cả khi họ không trực tiếp phát triển tính năng đó.
* **Development Team**: là những [lập trình viên](https://topdev.vn/blog/lap-trinh-vien/) sẽ tham gia vào việc phát triển từng tính năng cụ thể. Các lập trình viên này có thể sẽ có kỹ năng khác nhau và một số sẽ giỏi về những kỹ năng nhất định. Tuy nhiên khi sử dụng Scrum thì tất cả các thành viên của Development Team yêu cầu phải có khả năng làm việc thay thế vị trí của nhau và không ai chỉ chịu trách nhiệm phát triển một (hoặc một số) tính năng nhất định.
* **Scrum Master**: sẽ chịu trách nhiệm cho việc lên kế hoạch để phân công công việc, sắp xếp thứ tự ưu tiên giải quyết những công việc tồn đọng nào có trong **Backlog** trước, tổ chức các buổi họp với Product Owner để theo dõi tình hình và nắm thông tin cần thiết.

### **2. Sprint**

**Sprint**là mộ phân đoạn lặp đi lặp lại trong quy trình phát triển phần mềm, có khung thời gian thường là 1 tháng (từ 1 – 4 tuần) mà theo đó sản phẩm sẽ được release phiên bản mới. Khi một Sprint kết thúc thì Scrum Master cần phải chuyển trạng thái của nó sang Done.

Khi bắt đầu một Sprint thì Scrum Master cần đưa ra mục tiêu của Sprint đó và mục tiêu này không được phép thay đổi cho tới khi Sprint hoàn thành. Tuy nhiên Product Owner vẫn có quyền huỷ một Sprint trước thời hạn kết thúc của nó.

Mặc dù để làm điều này thì Product Owner cần sự đồng thuận của Development Team cũng như Scrum Master. Sau khi một Sprint kết thúc thì các bên sẽ dựa trên kết quả của Sprint đó để lên kế hoạch cho Sprint tiếp theo.

### **3. Sprint Planning**

Đây là bước đầu tiên cần phải thực hiện trước khi một Sprint bắt đầu. Development team họp với Product Owner để lên kế hoạch cho một sprint. Những công việc nào cần phải được hoàn thành trong Sprint này và làm sao để có thể hoàn thành những công việc này.

Sau khi thống nhất được số lượng công việc, thời gian hoàn thành thì chúng ta có thể bắt đầu Sprint. Trong khi thực hiện một Sprint chúng ta sẽ phải có những buổi họp được gọi là Daily Sprint hay Daily Meeting.

### **4. Daily Sprint**

Các buổi họp Daily Sprint thường kéo dài khoản 15 phút, trong buổi họp này tất cả các thành viên sẽ lần lượt báo cáo lại:

* Những gì họ đã làm được ngày hôm qua
* Những gì họ cần làm ngày hôm nay
* Những khó khăn mà họ gặp phải

Mỗi buổi họp này sẽ giúp việc dự kiến được kế hoạch đưa ra trong Sprint đang làm sẽ tiến triển ra sao và liệu có cần phải cập nhật lại bản kế hoạch đã đưa ra hay không. Tất nhiên cần nhớ rằng việc thay đổi kế hoạch này không bao gồm thay đổi mục tiêu đã đưa ra của Sprint.

Ví dụ bạn có thể tăng thêm thời gian để hoàn thành một chức năng và qua đó khiến Sprint phải kéo dài hơn dự kiến. Tuy nhiên mục tiêu của Sprint là cho phát hành một phiên bản mới cần được giữ nguyên.

### **5. Sprint Review**

Là công việc được thực hiện bởi nhóm phát triển và product owner ở cuối mối Sprint nhằm đánh giá lại kết quả thực hiện được. Từ lúc Sprint mới hoàn thành và qua đó đưa ra những chỉnh sửa, thay đổi cần thiết ở Sprint sau.

### **6. Sprint Restrospective**

Dưới sự trợ giúp của Scrum master, team phát triển sẽ tổng kết những kiến nghị và đánh giá từ bước Sprint Review ở trên để đưa ra những cải tiến nhằm nâng cao hiệu quả làm việc cũng như sản phẩm.

### **7. Các công cụ (artifacts) Scrum**

*Scrum sử dụng các công cụ rất đơn giản nhưng hiệu quả để trợ giúp công việc.*

* **Product backlog**

Đây là danh sách ưu tiên các tính năng (feature) hoặc đầu ra khác của dự án. Có thể hiểu như là danh sách yêu cầu (requirement) của dự án.

Product Owner chịu trách nhiệm sắp xếp độ ưu tiên cho từng hạng mục (Product Backlog Item) trong Product Backlog dựa trên các giá trị do Product Owner định nghĩa (thường là giá trị thương mại – business value).

* **Sprint backlog**

Đây là bản kế hoạch cho một Sprint; là kết quả của buổi họp lập kế hoạch (Sprint Planning).

Với sự kết hợp của Product Owner, nhóm sẽ phân tích các yêu cầu theo độ ưu tiên từ cao xuống thấp để hiện thực hóa các hạng mục trong Product Backlog dưới dạng danh sách công việc (TODO list).

* **Burndown Chart**

Đây là biểu đồ hiển thị xu hướng của dự án dựa trên lượng thời gian cần thiết còn lại để hoàn tất công việc.

Burndown Chart có thể được dùng để theo dõi tiến độ của Sprint (được gọi là Sprint Burndown Chart) hoặc của cả dự án (Project Burndown Chart).

Biểu đồ burndown không phải là một thành tố tiêu chuẩn của Scrum theo định nghĩa mới, nhưng vẫn được sử dụng rộng rãi do tính hữu ích của nó.

**SỰ KIỆN**

## Lập kế hoạch sprint

Khi một sprint bắt đầu, thì buổi Planing Meeting sẽ được thực hiện để xác định mục tiêu và chọn ra các task cần làm. Sự kiện này chia làm hai phần, phần một tất nhiên là chọn task và phần 2 là quyết định cách thức hoàn thành các task đó. Trong buổi planing meeting chúng ta sẽ lần lượt trả lời các câu hỏi sau:

* Mục tiêu của Sprint này là gì ?
* Sprint này phải chuyển giao cái gì ?
* Làm sao để đạt được điều đó

Toàn bộ nhóm Scrum bắt buộc phải tham gia phần thứ nhất của sự kiện. ở phần thứ 2 PO có thể vắng mặt nhưng phải luôn sẵn sàng hỗ trợ nhóm phát triển làm rõ các hạng mục khi cần thiết. Buổi planing meeting kéo dài thường 2 giờ đồng hồ. Thời gian dành cho 2 này được chia đều cho nhau, mỗi phần chiếm một nửa.

### Phần 1: Làm gì trong Sprint này?

Đáp án của câu hỏi này chính là mục tiêu và danh sách các backlog được lựa chọn cho sprint. Phần này bắt đầu bằng việc PO trình bày mục tiêu momg muốn đạt được trong Sprint này. Sau đó PO làm rõ thêm các hạng mục ở phần trên của Product backlog để cả nhóm hiểu kỹ về các mục đó. Trong trường hợp, số lượng backlog ở sprint này so với sprint trước khác nhau, PO phải có trách nhiệm làm rõ. Căn cứ vào mục tiêu sprint và năng lực hiện có, nhóm phát triển sẽ lựa chọn những mục mà họ tin có thể hoàn thành trong Sprint này, bắt đầu từ những hạng mục đứng đầu trong Product Backlog - nói cách khách, họ bắt đầu với những hạng mục có độ ưu tiên cao do PO sắp xếp. Đây là cách làm chủ chốt trong Scrum: nhóm phát triển quyết định có bao nhiêu hạng mục sẽ được hoàn thành, thay vì được giao bởi PO. Điều này sẽ giúp nhóm phát triển đưa racacs dự báo tin cậy hơn do họ làm việc đó căn cứ vào những phân tích và lập kế hoạch của chính bản thân họ.

Để quyết định những hạng mục nào của Product Backlog sẽ triển khai trong Sprint này. Nhóm phát triển xem xét tốc độ trung bình của mình trong các sprint trước kết hợp với những yếu tố bất ngờ ở sprint này để tính đưa ra số lượng backlog cho sprint này. Ví dụ: Trong sprint trước, nhóm phát triển 7 người đã hoàn thành 80 point. Mà trong sprint này có người xin nghỉ phép 3 ngày. Vậy khả năng của nhóm không thể đạt được 80 point như trước. Do đó chỉ lựa chọn số lượng hạng mục đủ làm với tổng point là nhỏ hơn 80.

Ở đây phần trên chúng ta có nói đến việc tốc độ của sprint (velocity). Tốc độ đó được tính bằng số lượng đơn vị được hoàn thành trong mỗi Sprint. Nếu bạn dùng đơn vị là point, thì tốc độ chính là số điểm mà nhóm hoàn thành sau mỗi Sprint. Qua thời gian, tốc độ của nhóm có thể tương đối ổn định. Đó là tiền đề quan trọng có thể phỏng đoán được khối lượng công việc của nhóm trong mỗi Sprint.

Nếu nhóm phát triển muốn lựa chọn một vài hạng mục có độ ưu tiên thấp ở phía dưới của Product backlog họ cần thảo luận với PO. Điều này thường xảy ra khi có sự phụ thuộc giữa các hạng mục hoặc nhóm cảm thấy mục đó có độ ưu tiên thấp nhưng phù hợp để phát triển cùng với các mục khác đã lựa chọn. Kết thúc phần 1, nhóm phát triển và PO thống nhất lại mục tiêu và danh sách backlog . Mục tiêu của sprint là một đoạn mô tả ngắn về kết quả kỳ vọng đạt được sau khi sprint kết thúc. Mục tiêu của sprint đóng vai trò định hướng nhóm Phát triển trong suốt quá trình diễn ra Sprint và đồng thời giúp nhóm đưa ra được những quyết định hợp lý nhằm đạt được mục tiêu này. Một ví dụ về Mục tiêu Sprint là :"Xây dựng chức năng mua hàng bao gồm: Xem danh sách sản phẩm, thêm sản phẩm vào giỏ hàng, xem giỏ hàng, loại bỏ sản phẩm khỏi gior hàng, hiển thị thanh toán".

### Phần 2: Làm sao để hoàn thành công việc đã chọn

Phần 2 của buổi lập kế hoạch Sprint với mục đích trả lời cho câu hỏi: Làm sao để hoàn thành công việc đã chọn? Kết quả của phần này đó là Sprint Backlog - tức là bằng công việc được Nhóm phát triển sử dụng trong suốt sprint, bao gồm các task trong Backlog đã lựa chọn và danh sách công việc tương tứng với từng hạng mục đó. Nhóm phát triển bắt đầu thiết kế hệ thống và lên danh sách các công việc cần làm. Đối với mỗi hạng mục trong danh sách đã lựa chọn, nhóm sẽ tách thành các công việc cụ thể. Các công việc này phải đảm bảo đủ nhỏ để hoàn thành trong vài giờ. Mộ số loại công việc thường được thấy là: design mockup, viết test, nghiên cứu kĩ thuật, triển khai tính năng v.v...Tất cả các công việc này đều phải được nhóm ước lược thời gian để hoàn thành. Thường được tính qua đơn vị point và được quyết định qua Planing Poker( một loại bài để quyết định số point cho một task). Những point này sẽ được cập nhập trong Sprint Backlog đồng thời cũng được sử dụng để tạo biểu độ Sprint Burndown nhằm theo dõi tiến độ Sprint. Sau mỗi ngày làm việc, các thành viên sẽ cập nhập đồng thời Sprint Backlog và biểu đồ với các giá trị mới. Nếu nhóm thấy rằng lược công việc quá nhiêu hay quá ít so với khả năng của nhóm họ có thể trao đổi với PO để loại bỏ bới các hạng mục.

## Daily Scum

Sau khi một sprint bắt đầu thì cũng có một sự kiện bắt đầu theo đó là daily scrum. Đây là sự kiện quan trọng diễn ra hàng ngày trong suốt quá trình phát triển. Thời lượng là 15p cho mỗi buổi nhằm mục đích trao đổi tiến độ công việc giữa các thành viên trong nhóm, lên kết hoạch cho công việc trong ngày hôm nay. Nó còn có một tên gọi khác quen thuộc hơn đó là daily meeting. Trong buổi này mỗi thành viên sẽ trả lời 3 câu hỏi:

* Tôi đã làm được gì vào hôm qua ?
* Tôi sẽ làm gì trong hôm nay ?
* Tôi đang gặp những khó khăn gì ?

Để đảm bảo không "cháy" time box mỗi thành viên nên chỉ trả lời 3 câu hỏi trên và không lan man thảo luận sâu khi phát hiện ra vấn đề nào đó. Và Daily meeting khuyến nghị người tham gia đứng thay vì ngồi để tăng sự tập trung.

## Tổng kết Sprint

Buổi tổng kết sprint sẽ được tiến hành khi thời gian triển khai sprint đã hết. Đây là hoạt động thanh tra và thích nghi đối với sản phẩm đang được xây dựng. Kết thúc buổi này, lộ trình sản phẩm và Product Backlog có thể được điều chỉnh để phù hợp với tình hình phát triển. Tham sự buổi này gôm có nhóm phát triển, PO, SM. Ngoài ra PO có thể mời thêm khách hàng để có thể đưa đánh giá. Tất cả những người tham dự đều hoàn toàn tự do trong việc đặt câu hỏi và đong góp. Khung thời gian cho sự kiện này thường là 1 giờ - 2 giờ. Nội dung trong buổi này là nhóm phát triển và PO trao đổi với nhau để tìm hiểu về tình hình và ghi nhận các khuyến nhị của nhau. Đây là cơ hội để PO lắm được sản phẩm. Còn với nhóm phát triển đây là cơ hội để tìm hiểu và nắm tình hình của PO và thị trường. Nội dung củ tổng kết sprint sẽ gồm:

* PO trình bày mục tiêu Sprint
* Nhóm phát triển trình bày kết quả đã hoàn thành
* Trực tiếp sử dụng sản phẩm và đưa đóng góp.

Ở trong buổi này, chúng ta chú ý không trình bày những tính năng chưa hoàn thành. Những phản hổi nhận được có thể được đánh giá lại độ ưu tiên cho các backlog. Tổng kết không hoàn toàn chỉ là demo sản phẩm, mà demo sản phẩm chỉ là một nội dung trong buổi này. Nếu tập chung cho việc demo sẽ bỏ qua đi những nội dung quan trọng khác đến việc thảo luận và cộng tác giữa các thành viên tham gia. Từ đó gây hiểu nhầm và thực hiên sai mục tiêu thực sự của buổi tổng kết.

# Cải tiến sprint

Và sau khi tiến hành tổng kết xong, chúng ta sẽ có buổi cải tiến sprint. Mục đích của sự kiện này là thanh tra và thi thích quy trình làm việc. Nói cách khác là tìm cách để sprint sau tốt hơn. Nhóm phát triển và SM bắt buộc phải tham gia buổi này. PO có thể có, có thể không. Thời gian đối đa cho buổi này là 2 tiếng. Ở buổi này, cần có một người đứng ra làm vai trò hỗ trợ và đó là SM. Hoạt động chính là:

* Liệt kê những điểm đã làm tốt
* Liệt kê những điểm đã làm chưa tốt
* Đưa ra một vài hành động cải thiện
* Kế hoạch cải thiện cho sprint sau Có rất nhiều kĩ thuật có thể dùng cho buổi này nhưng đơn giản và phổ biến nhất là Glad-Sad-Mad. Glad-Sad-Mad là một kĩ thuật để thực hiện buổi cải tiến dựa trên việc phân loại các ý kiến của thành viên thành 3 nhóm: Glad (vui) - Sad( buồn) - Mad( tức giận). Những hạng mục nào mà thành viên cảm thấy hài lòng thì sẽ phân loại vào Glab. Những task họ cảm thấy chưa hài lòng và có thể cải tiến được thì đưa vào mục Sad. Những hạng mục nào gây cản trở nghiêm trọng và mình móng muốn loại bỏ nó ngay thì đưa vào mục MaD. Cụ thể cách tiến thành nó như sau:

1. Chuẩn bị

* Để bắt đầu cần chuẩn bị một tấm bảng đủ lớn được chia thành 3 cột.
* Những mẩu giấy nhớ và bút

1. Thu nhập dữ liệu Sm yêu cầu thành viên ghi tất cả những quan sát của mình vào mảnh giấy nhớ. Mỗi ý kiên trên từng mảnh riêng biệt, sau đó dán lên các cột tương ứng. Hoạt động này cần sử dụng từ 10- 15p
2. Tổng hợp thông tin Người hỗ trợ gom các ý kiến trùng nhau lại thành một hoặc loại bỏ, sau đó , các thành viên sẽ bỏ phiếu để lựa chọn ra những hạng mục mà mình muốn thảo luận. Cách phổ biến nhất là bỏ phiếu chấm, tức là mỗi thành viên lựa chọn tối đa 3 hạng mục bằng cách ghi một dấu châm vào tờ giấy của mình. Người hỗ trợ sẽ sắp xếp lại trật tư các tờ giấy dựa trên kết quả dấu chấm đó.
3. Thảo luận và đưa ra hành động cải tiến Sau đó, cả nhóm sẽ lần lượt thảo luận từng hạng mục cho đến khi không còn hạng mục nào nữa, hoặc đã hết thời gian đóng khung. Việc thảo luận nên tập trung vào việc đưa ra hành động cải tiển trong thời gian tới.

**3.ÁP DỤNG SCRUM**

**Cơ cấu phát triển nhóm:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Tài khoản** | **Chức vụ** | **Ngày bắt đầu** | **Ngày kết thúc** |
| **1** | **Cao Phước Sơn** | **Soncao** | **Product Owner** | **17/11/2020** | **14/12/2020** |
| **2** | **Nguyễn Văn Tỵ** | **Tynguyen** | **Scrum master** | **17/11/2020** | **14/12/2020** |
| **3** | **Huỳnh Trọng Linh** | **Linhhuynh** | **Dev** | **17/11/2020** | **14/12/2020** |
| **4** | **Hoàng Duy Đạt** | **Dathoang** | **Dev** | **17/11/2020** | **14/12/2020** |
| **5** | **Trần Trân** | **Trantran** | **Tester** | **18/11/2020** | **21/12/2020** |

**SPRINT BACKLOG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STORY ID** | **STORY NAME** | **POINTS** | **NAME OF DEV** | **Note** |
|  |  | **36** |  |  |
| **1** | **Xem trang chủ** | **1** | **Cao Phước Sơn** |  |
| **2** | **Xem danh sách đt** | **1** | **Cao Phước Sơn** |  |
| **3** | **Xem chi tiết đt** | **3** | **Cao Phước Sơn** |  |
| **4** | **Quản lý giỏ hàng** | **3** | **Nguyễn Văn Tỵ** |  |
| **5** | **Đăng ký** | **3** | **Nguyễn Văn Tỵ** |  |
| **6** | **Liên hệ với cửa hàng** | **2** | **Nguyễn Văn Tỵ** |  |
| **7** | **Xem danh sách khách hàng** | **1** | **Huỳnh Trọng Linh** |  |
| **8** | **Xem chi tiết khách hàng** | **2** | **Huỳnh Trọng Linh** |  |
| **9** | **Thêm khách hàng** | **3** | **Huỳnh Trọng Linh** |  |
| **10** | **Xóa khách hàng** | **2** | **Huỳnh Trọng Linh** |  |
| **11** | **Đăng nhập** | **2** | **Hoàng Duy Đạt** |  |
| **12** | **Đăng xuất** | **1** | **Hoàng Duy Đạt** |  |
| **13** | **Thêm đt** | **3** | **Hoàng Duy Đạt** |  |
| **14** | **Xóa đt** | **1** | **Cao Phước Sơn** |  |
| **15** | **Quản lý đt tồn kho** | **3** | **Cao Phước Sơn** |  |
| **16** | **Quản lý đơn hàng** | **3** | **Nguyễn Văn Tỵ** |  |
| **17** | **Thanh toán trực tuyến** | **2** | **Nguyễn Văn Tỵ** |  |

**CHƯƠNG III: SỬ DỤNG CÔNG CỤ HỖ TRỢ QUẢN LÝ DỰ ÁN**

**CHƯƠNG VI: GITHUB**