# Scrum Master

Khóa học giúp người học hiểu vai trò của Scrum Master trong Scaled Agile Framework và có thể lập kế hoạch và thực hiện Program Increment (PI). Như một liên kết trong Scaled Agile Framework, Scrum Masters có cơ hội học cách xây dựng các nhóm Agile hiệu quả.

## Phát triển phần mềm

### Software Development Life Cycle – SDLC

#### Vòng đời phát triển phần mềm (SDLC) là gì?

Vòng đời phát triển phần mềm (Software Development Life Cycle – SDLC) là quy trình toàn diện từ khi bắt đầu phát triển phần mềm cho đến khi hoàn thành và triển khai nó. SDLC có mục tiêu giảm thiểu rủi ro dự án thông qua việc lập kế hoạch trước nhằm đáp ứng nhu cầu của khách hàng không chỉ trong giai đoạn sản xuất mà còn hơn thế nữa.

Một số mô hình SDLC phổ biến như mô hình thác nước, xoắn ốc, hay Agile,… SDLC thường được sử dụng trong quá trình phát triển một dự án công nghệ thông tin với các giai đoạn khác nhau.



**Hình 1:** Vòng đời phát triển phần mềm 2023

#### Các giai đoạn trong vòng đời phát triển phần mềm

Mỗi công ty sẽ có mô hình và hình thức hoạt động riêng nên các giai đoạn trong vòng đời phát triển phần mềm có thể tách hoặc gộp vào theo nhu cầu thực tế. Tuy nhiên, để tạo ra một sản phẩm phần mềm nhìn chung sẽ bao gồm các giai đoạn:

* Planning – Lập kế hoạch
* Analysis – Phân tích
* System Design – Thiết kế hệ thống
* Implementation – Triển khai
* Maintenance – Bảo trì



**Hình 2:** Các giai đoạn trong vòng đời phát triển phần mềm

**1. Plan – Xây dựng plan**

* Trao đổi lại với khách hàng, làm rõ yêu cầu và các kỳ vọng.
* Xác định vấn đề nếu như vấn đề trong hệ thống hiện tại của khách hàng.
* Lên lịch trình và  kế hoạch chi tiết cho hệ thống mới để đạt được mục tiêu chung.

**2. Analysis – Phân tích và thu thập dữ liệu**

* Thu thập, phân tích và xác thực thông tin/tham vấn từ các chuyên gia.
* Đánh giá các lựa chọn thay thế và thứ tự ưu tiên các yêu cầu.
* Tìm hiểu nhu cầu của người dùng cuối của sản phẩm để nâng cao hệ thống.
* Chuẩn bị tài liệu đặc tả yêu cầu phần mềm (SRS).

**3. Thiết kế hệ thống**

* Thiết kế ứng dụng, mạng, cơ sở dữ liệu, giao diện người dùng (UI/UX) và giao diện hệ thống.
* Chuyển đổi tài liệu SRS thành một kiến trúc hệ thống, bao gồm bộ thông số kỹ thuật chi tiết và đầy đủ.
* Rà soát để đảm bảo rằng thiết kế cuối cùng phải đáp ứng các yêu cầu nêu trong tài liệu SRS.

**4. Triển khai dự án**

* Tạo các trường hợp kiểm thử để thử nghiệm các chức năng và tính năng khác nhau của hệ thống. Khi thực hiện kiểm thử, tất cả các lỗi và vấn đề gặp phải đều được ghi lại trong báo cáo thử nghiệm.
* Tích hợp hệ thống thông tin vào môi trường của nó bằng cách cài đặt hệ thống mới và đảm bảo rằng nó hoạt động một cách tương thích với môi trường xung quanh.

**5. Bảo trì dự án**

* Hỗ trợ khách hàng sau khi hệ thống được cài đặt
* Nâng cấp hoặc cập nhật phiên bản phần mềm mới nhất
* Xử lý các lỗi còn sót lại và giải quyết mọi vấn đề có thể tồn tại trong hệ thống sau giai đoạn thử nghiệm.

#### Lợi ích của vòng đời phát triển phần mềm



**Hình 3:** Lợi ích của vòng đời phát triển phần mềm

* **Cải thiện hiệu suất hàng đầu:** Các yếu tố cơ bản như tạo chiến lược và quy trình kinh doanh dẫn đến một cách sâu sắc hơn nhiều để điều tra và tăng doanh thu hàng đầu. Nói tóm lại, việc phát triển hệ thống đảm bảo bạn đáp ứng được kỳ vọng của khách hàng và cải thiện thương hiệu của mình, đây là chìa khóa để phát triển một doanh nghiệp lành mạnh.
* **Đáp ứng mong đợi của khách hàng:** Bạn có thể dễ dàng nhận được thông tin quan trọng về các lĩnh vực kinh doanh cần được cải thiện. Ngoài ra, một số nhu cầu chưa được đáp ứng có thể được phát hiện nếu bạn áp dụng đúng phương pháp phát triển hệ thống.
* **Kết quả nhất quán:** Cho dù bạn đang xem xét mức độ an toàn, chất lượng hay chi phí phát triển, hệ thống kinh doanh được thiết kế để mang lại cho bạn kết quả hiệu quả và có thể lặp lại.
* **Sự gắn kết của nhân viên:** Mục tiêu của hệ thống là tạo cơ hội cho tất cả nhân viên để họ có thể hoàn thành công việc của mình một cách hiệu quả nhất. Ngoài ra, việc có sẵn một hệ thống cho phép công ty của bạn nhanh chóng hòa nhập với nhân viên mới và giúp họ dễ dàng nhận thấy vai trò của mình trong tổ chức.
* **Giảm chi phí và tăng lợi nhuận:** Người ta đã chứng minh rằng việc phát triển hệ thống giúp giảm chi phí mãi mãi. Phát triển một hệ thống kinh doanh thực sự là một cách để cắt giảm chi phí mà không đi đường tắt thường dẫn đến chất lượng sản phẩm hoặc mức độ dịch vụ thấp hơn.

Với vòng đời phát triển phần mềm, các nhà phát triển có một khuôn khổ rõ ràng và có cấu trúc để tiến hành quy trình phát triển phần mềm một cách hợp lý và chính xác. Việc tuân thủ vòng đời phát triển phần mềm giúp đảm bảo phần mềm được cập nhật đúng thời gian, đáp ứng yêu cầu của khách hàng và đạt được mục tiêu dự án một cách hiệu quả.

### Quản lý dự án Agile

#### Quản lý dự án Agile là gì ?

Quản lý dự án Agile là một phương pháp quản lý dự án tập trung vào tính linh hoạt, khả năng phản hồi nhanh chóng với sự thay đổi và sự tham gia liên tục của các bên liên quan. Agile không chỉ là một bộ công cụ hoặc quy trình cụ thể, mà là một triết lý hoặc tư duy (mindset) quản lý dự án, được xác định bởi các giá trị và nguyên tắc trong Tuyên ngôn Agile (Agile Manifesto).

#### Đặc điểm của quản lý dự án Agile

Đặc điểm chính của quản lý dự án Agile:

1. ***Tính linh hoạt cao:***
   * Dự án được quản lý theo các chu kỳ ngắn hạn gọi là "iteration" hoặc "sprint" (trong Scrum). Mỗi chu kỳ này thường kéo dài từ một đến bốn tuần.
   * Kế hoạch dự án có thể được điều chỉnh sau mỗi chu kỳ dựa trên phản hồi từ khách hàng và sự thay đổi trong yêu cầu.
2. ***Phát triển sản phẩm theo từng phần:***
   * Thay vì cố gắng xây dựng toàn bộ sản phẩm từ đầu đến cuối, nhóm sẽ phát triển sản phẩm theo từng phần nhỏ, có thể chuyển giao được. Mỗi phần hoàn thành phải hoạt động và có thể được đưa vào sử dụng.
3. ***Liên tục cải tiến:***
   * Sau mỗi chu kỳ, nhóm sẽ họp lại để đánh giá những gì đã làm tốt và những gì có thể cải thiện (thường được gọi là "Sprint Retrospective" trong Scrum).
   * Phản hồi từ khách hàng và người dùng cuối cũng được sử dụng để điều chỉnh và cải tiến sản phẩm và quy trình làm việc.
4. ***Tương tác liên tục với khách hàng:***
   * Khách hàng hoặc các bên liên quan tham gia liên tục vào quá trình phát triển. Điều này đảm bảo rằng sản phẩm luôn đáp ứng được nhu cầu của khách hàng ngay cả khi những yêu cầu này thay đổi.
5. ***Tập trung vào con người và sự tương tác:***
   * Agile nhấn mạnh giá trị của con người, khuyến khích sự tương tác thường xuyên giữa các thành viên trong nhóm và với khách hàng.

#### Các khung làm việc Agile

Các khung làm việc Agile phổ biến:

* **SAFe (Scaled Agile Framework):** Đây là một khung làm việc Agile mở rộng, được sử dụng để áp dụng các phương pháp Agile trong các tổ chức lớn, nơi mà nhiều nhóm làm việc đồng bộ để phát triển một sản phẩm hoặc hệ thống phức tạp.
* **Scrum:** Tập trung vào việc chia dự án thành các chu kỳ ngắn hạn gọi là "Sprint". Scrum Master giúp nhóm phát triển duy trì và cải thiện quy trình Scrum.
* **Kanban:** Sử dụng bảng Kanban để quản lý luồng công việc, giúp nhóm nhìn thấy và điều chỉnh tiến độ công việc một cách trực quan và liên tục.
* **Extreme Programming (XP):** Nhấn mạnh vào chất lượng mã và việc phát triển phần mềm linh hoạt thông qua các thực hành như Test-Driven Development (TDD), pair programming, và continuous integration.
* **Lean:** Tập trung vào việc loại bỏ lãng phí, tối ưu hóa quy trình và tăng cường giá trị sản phẩm.

#### Lợi ích của quản lý dự án Agile

Lợi ích của quản lý dự án Agile:

* Thích nghi tốt hơn với sự thay đổi: Agile cho phép đội ngũ điều chỉnh nhanh chóng theo những thay đổi trong yêu cầu hoặc môi trường kinh doanh.
* Cải thiện sự cộng tác: Agile khuyến khích sự giao tiếp liên tục giữa các bên liên quan, dẫn đến sự hiểu biết tốt hơn và tăng cường hợp tác.
* Tăng chất lượng sản phẩm: Thông qua việc kiểm thử và phản hồi liên tục, chất lượng sản phẩm được nâng cao theo thời gian.
* Giảm thiểu rủi ro: Bằng cách cung cấp các phiên bản sản phẩm nhỏ và hoạt động, Agile giúp phát hiện và giải quyết các vấn đề sớm hơn trong quy trình phát triển.

### SAFe Scrum Master

#### Khái niệm

"SAFe Scrum Master" không phải là một cụm từ viết tắt, mà là một vai trò cụ thể trong khung quản lý Agile của **Scaled Agile Framework (SAFe)**. Trong SAFe, Scrum Master đóng vai trò tương tự như trong Scrum thông thường, nhưng với trách nhiệm mở rộng hơn để phù hợp với quy mô lớn của các dự án và tổ chức áp dụng SAFe.

Tuy nhiên, bạn có thể nghĩ về "SAFe" như một từ viết tắt của "Scaled Agile Framework," một khung làm việc giúp mở rộng các phương pháp Agile để áp dụng cho các tổ chức lớn và phức tạp, vượt ra ngoài phạm vi của một đội Scrum đơn lẻ.

#### Vai trò của SAFe Scrum Master

1. **Hỗ trợ các nhóm Agile:** SAFe Scrum Master giúp các đội Agile hoạt động hiệu quả, đảm bảo rằng các quy trình Agile được thực hiện đúng cách và hỗ trợ nhóm loại bỏ các trở ngại.
2. **Điều phối với các nhóm khác:** Trong SAFe, các Scrum Master không chỉ làm việc với một nhóm mà còn phải điều phối và hợp tác với các nhóm khác trong "Agile Release Train" (ART), một đơn vị tổ chức bao gồm nhiều nhóm Agile làm việc cùng nhau để cung cấp sản phẩm hoặc hệ thống lớn.
3. **Tham gia vào các sự kiện SAFe:** SAFe Scrum Master cũng có trách nhiệm tham gia vào các sự kiện SAFe cấp cao hơn, như "Program Increment (PI) Planning," nơi các nhóm lên kế hoạch cho chu kỳ làm việc dài hơn (thường từ 8-12 tuần) và xác định các mục tiêu chung.
4. **Hỗ trợ cải tiến liên tục:** Giống như trong Scrum truyền thống, SAFe Scrum Master cũng tập trung vào việc cải tiến liên tục, nhưng ở mức độ toàn tổ chức, giúp tối ưu hóa quy trình và hợp tác giữa các nhóm.

Tóm lại, "SAFe Scrum Master" không phải là viết tắt mà là một vai trò trong khung làm việc Scaled Agile Framework, nơi Scrum Master đóng vai trò hỗ trợ, điều phối và cải tiến quy trình trong một tổ chức lớn áp dụng phương pháp Agile.

1. Bài Tập 1
2. Bài Tập 2
3. PDCA Cycle

**1. Individuals and interactions over processes and tools**

Cá nhân và sự tương tác quan trọng hơn quy trình và công cụ.

Giải thích:

* Trong quản lý dự án Agile, con người và cách họ tương tác với nhau được coi là yếu tố then chốt để đạt được thành công. Mặc dù quy trình và công cụ là quan trọng, nhưng chúng chỉ là phương tiện hỗ trợ.
* Các thành viên trong nhóm cần hợp tác chặt chẽ, giao tiếp thường xuyên và hiệu quả để giải quyết vấn đề và tạo ra giá trị thực sự. Sự tương tác này giúp nhóm nhanh chóng điều chỉnh khi có thay đổi, phát hiện sớm các vấn đề và tìm giải pháp hiệu quả hơn.
* Ví dụ, nếu một nhóm có các công cụ mạnh mẽ nhưng thiếu sự giao tiếp và hợp tác, dự án vẫn có thể gặp trục trặc. Ngược lại, nếu nhóm có sự tương tác tốt, họ có thể vượt qua các hạn chế về công cụ hoặc quy trình để đạt được kết quả mong muốn.

**2. Working software over comprehensive documentation**

Phần mềm hoạt động quan trọng hơn tài liệu đầy đủ.

Giải thích:

* Agile tập trung vào việc cung cấp phần mềm có giá trị thực sự cho người dùng. Mục tiêu là sản xuất phần mềm hoạt động tốt và có thể được triển khai nhanh chóng để tạo ra giá trị, thay vì đầu tư quá nhiều thời gian vào việc viết tài liệu chi tiết và toàn diện.
* Điều này không có nghĩa là tài liệu không quan trọng, mà là ưu tiên cao hơn được đặt vào việc tạo ra phần mềm có thể sử dụng ngay lập tức. Tài liệu chỉ cần đủ để phục vụ cho việc phát triển và bảo trì sản phẩm, thay vì viết tài liệu chỉ để tuân thủ quy trình.
* Ví dụ, trong một dự án Agile, nhóm sẽ tập trung vào việc tạo ra các tính năng hoạt động tốt và đưa chúng vào sử dụng nhanh chóng, sau đó dựa vào phản hồi từ người dùng để cải thiện, thay vì viết một bản tài liệu dài và chi tiết ngay từ đầu mà có thể không phản ánh đúng nhu cầu thực tế.

**3. Customer collaboration over contract negotiations**

Hợp tác với khách hàng quan trọng hơn đàm phán hợp đồng.

Giải thích:

* Trong Agile, khách hàng được xem là đối tác tích cực, tham gia liên tục trong suốt quá trình phát triển. Thay vì chỉ đàm phán hợp đồng một lần rồi làm theo hợp đồng đó, Agile khuyến khích việc liên tục hợp tác với khách hàng để đảm bảo sản phẩm cuối cùng đáp ứng đúng nhu cầu và mong đợi của họ.
* Hợp tác này có thể bao gồm việc nhận phản hồi thường xuyên, điều chỉnh sản phẩm theo nhu cầu mới của khách hàng, và đảm bảo rằng sự thay đổi được quản lý hiệu quả. Việc này giúp giảm thiểu rủi ro về việc phát triển một sản phẩm không phù hợp với nhu cầu thực tế.
* Ví dụ, thay vì chỉ dựa vào hợp đồng để xác định mọi yêu cầu và kỳ vọng từ khách hàng ngay từ đầu, nhóm Agile sẽ làm việc gần gũi với khách hàng trong suốt quá trình phát triển, điều chỉnh và cải tiến sản phẩm dựa trên phản hồi liên tục.

**4. Responding to change over following a plan**

Phản hồi trước thay đổi quan trọng hơn việc theo kế hoạch.

Giải thích:

* Thế giới kinh doanh và công nghệ luôn thay đổi, và một kế hoạch cố định từ đầu có thể nhanh chóng trở nên lỗi thời. Agile khuyến khích sự linh hoạt và khả năng thích ứng trước những thay đổi này.
* Thay vì cứng nhắc tuân theo một kế hoạch đã đặt ra từ đầu, nhóm Agile liên tục điều chỉnh và cải tiến kế hoạch dựa trên thông tin mới, phản hồi từ khách hàng, và những thay đổi trong môi trường kinh doanh. Điều này giúp đảm bảo rằng sản phẩm cuối cùng phù hợp với nhu cầu thực tế, ngay cả khi những nhu cầu này thay đổi trong quá trình phát triển.
* Ví dụ, nếu trong quá trình phát triển, thị trường thay đổi hoặc khách hàng có những yêu cầu mới, nhóm Agile sẽ ưu tiên điều chỉnh sản phẩm để phù hợp với tình hình mới thay vì tiếp tục thực hiện theo kế hoạch ban đầu mà có thể không còn phù hợp.

**1. Three Pillars of Scrum (Ba trụ cột của Scrum)**

* **Transparency (Tính minh bạch):**
  + Mọi khía cạnh quan trọng của quy trình phải được nhìn thấy rõ ràng bởi tất cả những người liên quan. Tính minh bạch giúp đảm bảo rằng mọi người đều có cùng một nhận thức về công việc đang diễn ra, điều này rất quan trọng để đưa ra các quyết định đúng đắn.
* **Inspection (Kiểm tra):**
  + Quy trình và sản phẩm được thường xuyên kiểm tra để phát hiện các biến động không mong muốn hoặc những sai lệch so với mục tiêu dự định. Các buổi kiểm tra này giúp nhóm phát hiện sớm các vấn đề và thực hiện các điều chỉnh cần thiết.
* **Adaptation (Thích ứng):**
  + Khi phát hiện ra các sai lệch thông qua quá trình kiểm tra, nhóm phải nhanh chóng thích ứng bằng cách thực hiện các điều chỉnh cần thiết. Khả năng thích ứng là yếu tố quan trọng để duy trì hiệu quả và chất lượng của dự án trong suốt quá trình phát triển.

**2. Scrum Values (Giá trị của Scrum)**

* **Courage (Dũng cảm):**
  + Các thành viên trong nhóm cần có dũng cảm để thực hiện những điều đúng đắn và vượt qua các thách thức trong công việc. Điều này bao gồm việc chấp nhận thử thách, đối mặt với sự không chắc chắn, và sẵn sàng thay đổi khi cần thiết.
* **Commitment (Cam kết):**
  + Các thành viên trong nhóm cần cam kết với mục tiêu của Sprint và với nhóm của họ. Cam kết mạnh mẽ giúp nhóm tập trung và đạt được các mục tiêu đã đặt ra, đồng thời thúc đẩy sự đoàn kết và hợp tác trong nhóm.
* **Focus (Tập trung):**
  + Nhóm Scrum tập trung vào công việc cần làm để đạt được các mục tiêu của Sprint và mục tiêu tổng thể của dự án. Sự tập trung này giúp nhóm tránh được sự phân tán và đảm bảo rằng nỗ lực của họ mang lại kết quả cụ thể.
* **Respect (Tôn trọng):**
  + Các thành viên trong nhóm tôn trọng nhau như những chuyên gia độc lập và có giá trị. Sự tôn trọng giúp tạo ra một môi trường làm việc tích cực, nơi mọi người đều cảm thấy được đánh giá cao và có thể đóng góp hết mình.
* **Openness (Cởi mở):**
  + Nhóm Scrum và các bên liên quan cần cởi mở về công việc, thách thức, và tiến độ. Sự cởi mở giúp tạo ra một môi trường tin cậy, nơi thông tin được chia sẻ một cách minh bạch và mọi người đều được lắng nghe.

1. Definition (Định nghĩa):
2. Duration (Thời lượng):
3. Goal (Mục tiêu):
   1. PDCA Cycle (Chu trình PDCA)
      1. Khái niệm PDCA
      2. Iteration Events (Các sự kiện trong Iteration):

* Iteration là một chu kỳ phát triển đơn lẻ trong đó mỗi nhóm Agile sẽ **định nghĩa, xây dựng, tích hợp và kiểm thử** các câu chuyện người dùng (Stories) từ danh sách công việc của họ (Iteration Backlog).
* **Giải thích:** Trong Agile, Iteration là một đơn vị thời gian cố định trong đó một nhóm sẽ hoàn thành một phần công việc cụ thể. Mỗi Iteration kết thúc với một phần mềm hoạt động được kiểm thử và sẵn sàng được trình bày hoặc chuyển giao. Nhóm sẽ thực hiện việc xác định yêu cầu (define), xây dựng tính năng (build), tích hợp (integrate), và kiểm thử (test) trong mỗi Iteration.
* Mỗi Iteration có thời lượng cố định và các Iteration chạy liên tiếp nhau. SAFe khuyến nghị các Iteration kéo dài hai tuần.
* **Giải thích:** Việc giữ thời lượng cố định cho các Iteration giúp nhóm có thể lập kế hoạch và quản lý công việc một cách hiệu quả hơn. SAFe thường khuyến nghị sử dụng chu kỳ hai tuần cho mỗi Iteration để đảm bảo nhịp điệu công việc ổn định và nhanh chóng đáp ứng các thay đổi.
* Mục tiêu của Iteration là cung cấp giá trị mới cho khách hàng vào cuối mỗi Iteration.
* **Giải thích:** Mỗi Iteration kết thúc với một sản phẩm hoặc tính năng có thể sử dụng được và mang lại giá trị cho khách hàng. Điều này giúp đảm bảo rằng dự án liên tục tiến về phía trước và luôn đáp ứng nhu cầu của khách hàng.

**Lưu ý:**Avoid adding scope once the Iteration has begun (Tránh thêm phạm vi công việc sau khi Iteration đã bắt đầu):

* **Giải thích:** Một trong những nguyên tắc quan trọng của Iteration là không thay đổi phạm vi công việc (scope) trong khi Iteration đang diễn ra. Điều này giúp nhóm duy trì sự tập trung và hoàn thành các công việc đã được lên kế hoạch.

**PDCA** là viết tắt của **Plan (Lập kế hoạch), Do (Thực hiện), Check (Kiểm tra), Adjust (Điều chỉnh)**. Đây là một quy trình lặp đi lặp lại được áp dụng trong các Iteration để đảm bảo liên tục cải tiến và điều chỉnh công việc dựa trên kết quả thực tế.

* **Giải thích:** Chu trình PDCA được áp dụng trong từng Iteration để giúp nhóm Agile lập kế hoạch công việc, thực hiện kế hoạch đó, kiểm tra kết quả, và điều chỉnh nếu cần thiết. Quy trình này đảm bảo nhóm luôn cải thiện và nâng cao hiệu quả công việc.
* **Iteration Planning (Lập kế hoạch Iteration):** Xác định những công việc sẽ được thực hiện trong Iteration.
* **Iteration Execution (Thực hiện Iteration):** Nhóm thực hiện các công việc đã được lên kế hoạch.
* **Iteration Review (Đánh giá Iteration):** Xem xét lại công việc đã hoàn thành trong Iteration.
* **System Demo (Trình bày hệ thống):** Trình bày sản phẩm hoặc tính năng đã hoàn thành cho các bên liên quan.
* **Iteration Retrospective (Tổng kết Iteration):** Nhìn lại Iteration để rút ra các bài học và cải tiến quy trình.
  1. Team Backlog

Bức ảnh này giải thích về vai trò của **Team Backlog** trong việc tổ chức công việc của nhóm trong môi trường Agile, đặc biệt là trong bối cảnh của Scaled Agile Framework (SAFe). Dưới đây là phần giải thích chi tiết:

1. The Team Backlog is everything.
2. Work sizes may be estimated, but estimates do not imply committed delivery.
3. The Team Backlog:

* **Giải thích:** Team Backlog là nơi chứa tất cả các công việc mà nhóm cần thực hiện. Nếu một phần công việc nào đó có trong backlog, **nó có thể được hoàn thành**. Nếu không có trong backlog, **sẽ không có cơ hội** để nó được thực hiện. Điều này nhấn mạnh tầm quan trọng của việc quản lý và duy trì backlog một cách hiệu quả.
* **Giải thích:** Kích thước hoặc khối lượng công việc có thể được ước tính, nhưng các ước tính này **không phải là cam kết** về thời gian hoàn thành cụ thể. Điều này có nghĩa là trong Agile, nhóm đưa ra ước tính về khối lượng công việc để lập kế hoạch, nhưng điều này không đảm bảo rằng công việc sẽ được hoàn thành đúng thời hạn dựa trên ước tính đó. Sự linh hoạt và thích ứng là yếu tố quan trọng trong Agile.
* **Is created by the Agile Team:**
  + **Giải thích:** Backlog được **tạo ra bởi nhóm Agile**. Điều này có nghĩa là chính các thành viên trong nhóm, những người hiểu rõ công việc, sẽ tham gia vào việc xác định và ghi lại các công việc cần làm.
* **Is owned and prioritized by the Product Owner:**
  + **Giải thích:** Backlog thuộc quyền sở hữu và **được ưu tiên bởi Product Owner**. Product Owner là người chịu trách nhiệm quyết định thứ tự ưu tiên của các mục trong backlog dựa trên giá trị mang lại cho khách hàng và mục tiêu kinh doanh.
* **Represents opportunities, not commitments:**
  + **Giải thích:** Team Backlog **đại diện cho các cơ hội**, không phải là cam kết. Điều này có nghĩa là các mục trong backlog là các cơ hội để nhóm tạo ra giá trị, nhưng không nhất thiết phải hoàn thành tất cả trong một khoảng thời gian nhất định. Nhóm sẽ chọn và thực hiện những mục có giá trị nhất trong mỗi Iteration, nhưng không phải cam kết với tất cả các mục trong backlog.

1. Agile Teams are cross-functional:

[11:21 PM] Hao Nguyen Phi

1. Agile Teams are cross-functional:
2. Optimized for communication and delivery of value:
3. Typically include 10 team members or less:
4. Contain two specialty roles:
5. Hình ảnh minh họa:

* **Giải thích:** Nhóm Agile được cấu trúc theo mô hình đa chức năng, nghĩa là nhóm bao gồm các thành viên với các kỹ năng đa dạng cần thiết để thực hiện mọi khía cạnh của quá trình phát triển sản phẩm. Điều này bao gồm các kỹ năng liên quan đến việc **định nghĩa (define), xây dựng (build), kiểm thử (test), và (khi cần thiết) triển khai (deploy)** các thành phần của sản phẩm.
* **Tự tổ chức (self-organizing):** Các nhóm Agile tự quản lý và tự tổ chức công việc của mình mà không cần sự can thiệp chi tiết từ bên ngoài. Họ có khả năng tự quyết định cách thức tốt nhất để đạt được các mục tiêu đã đặt ra.
* **Giải thích:** Các nhóm Agile được tối ưu hóa cho việc **giao tiếp** và **cung cấp giá trị**. Do tính chất đa chức năng và tự tổ chức, nhóm có khả năng giao tiếp nội bộ hiệu quả, giảm thiểu các rào cản thông tin và làm việc cùng nhau để đảm bảo rằng giá trị được cung cấp cho khách hàng một cách liên tục và nhanh chóng.
* **Giải thích:** Thường thì một nhóm Agile bao gồm **10 thành viên hoặc ít hơn**. Kích thước nhóm nhỏ giúp đảm bảo rằng giao tiếp giữa các thành viên hiệu quả hơn, nhóm dễ dàng tổ chức và linh hoạt trong việc xử lý các thay đổi hoặc sự cố bất ngờ.
* **Giải thích:** Trong một nhóm Agile, có hai vai trò chuyên biệt rất quan trọng:
  + **Scrum Master:** Đảm bảo rằng nhóm hiểu và tuân thủ các quy trình Agile/Scrum, giúp loại bỏ các trở ngại và hỗ trợ nhóm duy trì hiệu suất cao nhất.
  + **Product Owner:** Là người đại diện cho các bên liên quan và khách hàng, chịu trách nhiệm quản lý backlog sản phẩm, ưu tiên các công việc cần làm để tối đa hóa giá trị của sản phẩm.
* Hình ảnh minh họa cho thấy rằng mỗi nhóm (Team 1, Team N) có thể bao gồm các thành viên có khả năng thực hiện các chức năng **định nghĩa (Define), xây dựng (Build), kiểm thử (Test), và triển khai (Deploy)**. Đây là sự phản ánh của tính đa chức năng, cho phép nhóm hoàn thành mọi công việc cần thiết trong quy trình phát triển mà không cần phụ thuộc vào các nhóm bên ngoài.



Bức ảnh này mô tả các **trách nhiệm của một nhóm Agile** trong việc phát triển và cung cấp sản phẩm. Các trách nhiệm này đều hướng tới mục tiêu chính là **Delivering Great Products** (Cung cấp sản phẩm tuyệt vời). Dưới đây là phần giải thích chi tiết về từng trách nhiệm:

1. Improving Relentlessly (Cải tiến không ngừng)
2. Connecting with the Customer (Kết nối với khách hàng)
3. Planning the Work (Lên kế hoạch công việc)
4. Delivering Value (Cung cấp giá trị)
5. Getting Feedback (Nhận phản hồi)

* **Giải thích:** Nhóm Agile có trách nhiệm liên tục tìm kiếm các cơ hội để cải thiện quy trình, sản phẩm, và hiệu suất làm việc của mình. Việc cải tiến liên tục giúp nhóm nâng cao chất lượng sản phẩm, đáp ứng tốt hơn các nhu cầu của khách hàng và thích ứng nhanh chóng với các thay đổi trong môi trường làm việc.
* **Giải thích:** Nhóm Agile phải duy trì mối liên hệ chặt chẽ với khách hàng để hiểu rõ nhu cầu, thu thập phản hồi, và đảm bảo rằng sản phẩm phát triển đúng với mong đợi của khách hàng. Kết nối thường xuyên với khách hàng giúp nhóm điều chỉnh kịp thời và cung cấp giá trị thực sự.
* **Giải thích:** Lập kế hoạch là một phần không thể thiếu trong trách nhiệm của nhóm Agile. Nhóm cần xác định những công việc cần làm, ưu tiên các nhiệm vụ quan trọng và phân bổ nguồn lực phù hợp để đảm bảo tiến độ công việc được duy trì hiệu quả. Kế hoạch được điều chỉnh linh hoạt dựa trên tiến độ thực tế và phản hồi từ khách hàng.
* **Giải thích:** Mục tiêu chính của nhóm Agile là cung cấp giá trị liên tục cho khách hàng. Mỗi Iteration hoặc Sprint cần kết thúc với một phần sản phẩm có giá trị thực sự, sẵn sàng được sử dụng hoặc thử nghiệm. Việc liên tục cung cấp giá trị giúp duy trì sự tin tưởng của khách hàng và đảm bảo rằng dự án luôn tiến triển theo hướng có lợi nhất.
* **Giải thích:** Nhóm Agile phải chủ động thu thập phản hồi từ khách hàng, người dùng và các bên liên quan khác. Phản hồi này rất quan trọng để nhóm hiểu được những gì đang làm tốt, những gì cần cải thiện, và cách thức sản phẩm có thể được điều chỉnh để phù hợp hơn với nhu cầu thực tế.



Bức ảnh này giải thích về hai vai trò chuyên biệt trong các nhóm Agile: **Scrum Master** và **Product Owner**. Dưới đây là phần giải thích chi tiết về trách nhiệm của từng vai trò:

1. Scrum Master
2. Product Owner
3. Tóm lại:

* **Facilitates PI Planning (Hỗ trợ lập kế hoạch PI):**
  + Scrum Master đóng vai trò quan trọng trong việc hỗ trợ và điều phối quá trình lập kế hoạch Program Increment (PI Planning) - một sự kiện quan trọng trong SAFe nơi các mục tiêu và kế hoạch cho một khoảng thời gian nhất định được thiết lập.
* **Supports Iteration Execution (Hỗ trợ thực hiện Iteration):**
  + Scrum Master hỗ trợ nhóm trong việc thực hiện các Iteration. Điều này bao gồm việc loại bỏ các trở ngại, đảm bảo rằng nhóm tuân theo quy trình Agile, và hỗ trợ nhóm đạt được các mục tiêu của mỗi Iteration.
* **Improves Flow (Cải thiện luồng công việc):**
  + Scrum Master liên tục tìm cách tối ưu hóa quy trình làm việc của nhóm, đảm bảo rằng các công việc di chuyển mượt mà qua các giai đoạn của quy trình phát triển, từ lập kế hoạch đến hoàn thành.
* **Builds a high-performing team (Xây dựng một nhóm hiệu suất cao):**
  + Scrum Master có trách nhiệm xây dựng và duy trì một nhóm có hiệu suất làm việc cao. Điều này bao gồm việc hỗ trợ phát triển kỹ năng của các thành viên, thúc đẩy tinh thần làm việc nhóm, và tạo ra một môi trường làm việc tích cực.
* **Improves ART performance (Cải thiện hiệu suất của ART):**
  + ART (Agile Release Train) là một nhóm các nhóm Agile hoạt động cùng nhau. Scrum Master làm việc để cải thiện hiệu suất tổng thể của ART, đảm bảo rằng tất cả các nhóm phối hợp hiệu quả để đạt được mục tiêu chung.
* **Connects with the Customer (Kết nối với khách hàng):**
  + Product Owner duy trì mối quan hệ chặt chẽ với khách hàng để hiểu rõ nhu cầu, thu thập phản hồi, và đảm bảo rằng sản phẩm phát triển đúng với mong đợi của khách hàng.
* **Contributes to the Vision and Roadmap (Đóng góp vào tầm nhìn và lộ trình):**
  + Product Owner đóng vai trò quan trọng trong việc phát triển và duy trì tầm nhìn sản phẩm và lộ trình phát triển dài hạn. Họ đảm bảo rằng các mục tiêu ngắn hạn của nhóm phù hợp với chiến lược dài hạn của sản phẩm.
* **Manages and prioritizes the Team Backlog (Quản lý và ưu tiên Team Backlog):**
  + Product Owner chịu trách nhiệm quản lý và ưu tiên các công việc trong Team Backlog, đảm bảo rằng nhóm tập trung vào các nhiệm vụ quan trọng nhất để tối đa hóa giá trị mà sản phẩm mang lại cho khách hàng.
* **Supports the team in delivering value (Hỗ trợ nhóm trong việc cung cấp giá trị):**
  + Product Owner làm việc cùng nhóm để đảm bảo rằng công việc được hoàn thành đúng cách và giá trị thực sự được cung cấp cho khách hàng qua mỗi Iteration.
* **Gets and applies fast feedback (Thu nhận và áp dụng phản hồi nhanh chóng):**
  + Product Owner thu nhận phản hồi từ khách hàng và các bên liên quan khác một cách nhanh chóng và điều chỉnh chiến lược sản phẩm hoặc công việc nhóm dựa trên phản hồi đó.
* **Scrum Master** tập trung vào việc tối ưu hóa quy trình, hỗ trợ nhóm và cải thiện hiệu suất tổng thể.
* **Product Owner** tập trung vào việc quản lý sản phẩm, kết nối với khách hàng, và đảm bảo rằng nhóm đang làm việc đúng hướng để cung cấp giá trị cao nhất.

Cả hai vai trò này đều quan trọng để đảm bảo rằng một nhóm Agile hoạt động hiệu quả và sản phẩm cuối cùng đáp ứng được các nhu cầu của khách hàng và mục tiêu của tổ chức.



Bức ảnh này trình bày về khái niệm **Agile Release Trains (ARTs)** trong Scaled Agile Framework (SAFe). Dưới đây là phần giải thích chi tiết về các thành phần và khái niệm liên quan:

1. Agile Release Trains (ARTs):
2. A virtual organization of 5–12 teams (50–125+ individuals):
3. Synchronized on a common cadence—a PI:
4. Aligned to a common mission via a single ART Backlog:
5. Continuous Delivery Pipeline:
6. Tóm lại:

* **Giải thích:** ART là một tổ chức ảo bao gồm nhiều nhóm Agile làm việc cùng nhau để phát triển và cung cấp các giá trị thông qua một chuỗi phát hành liên tục. Mỗi ART hoạt động như một "đoàn tàu" với nhiều nhóm tham gia, phối hợp nhịp nhàng để đạt được mục tiêu chung.
* **Giải thích:** Một ART thường bao gồm từ **5 đến 12 nhóm**, tương đương với khoảng **50 đến 125+ thành viên**. Điều này cho thấy ART có thể bao gồm một số lượng lớn các nhóm và cá nhân, tất cả đều phối hợp chặt chẽ để cùng phát triển và cung cấp sản phẩm.
* **Giải thích:** Các nhóm trong ART hoạt động theo một nhịp điệu chung gọi là **Program Increment (PI)**. PI là một chu kỳ thời gian định kỳ, thường kéo dài từ 8 đến 12 tuần, trong đó các nhóm phối hợp thực hiện công việc để đạt được các mục tiêu của ART. Nhịp điệu chung này giúp đảm bảo rằng tất cả các nhóm đều đồng bộ và làm việc hướng tới cùng một mục tiêu.
* **Giải thích:** Tất cả các nhóm trong ART đều hướng tới một **sứ mệnh chung**, và công việc của họ được quản lý thông qua một **ART Backlog** duy nhất. ART Backlog là danh sách các công việc cần làm để đạt được mục tiêu của ART. Điều này giúp đảm bảo rằng tất cả các nhóm đều tập trung vào việc hoàn thành những nhiệm vụ quan trọng nhất.
* **Giải thích:** ART là một phần của quy trình **Continuous Delivery Pipeline**, nơi các sản phẩm được phát triển, tích hợp, kiểm thử, và triển khai liên tục. Quy trình này bao gồm:
  + **Continuous Exploration (Khám phá liên tục):** Tìm hiểu và định nghĩa các yêu cầu mới, cũng như xác định các cơ hội để phát triển thêm giá trị.
  + **Continuous Integration (Tích hợp liên tục):** Liên tục tích hợp mã nguồn mới vào hệ thống, đảm bảo rằng tất cả các thay đổi đều được kiểm thử và hợp nhất một cách nhất quán.
  + **Continuous Deployment (Triển khai liên tục):** Liên tục triển khai các tính năng mới vào môi trường sản xuất hoặc môi trường thử nghiệm để sẵn sàng cung cấp cho khách hàng.
  + **Release on Demand (Phát hành theo yêu cầu):** Các tính năng hoặc sản phẩm mới được phát hành cho khách hàng khi cần thiết, không phải đợi đến cuối một chu kỳ phát triển lớn.

**Agile Release Trains (ARTs)** là tổ chức ảo trong SAFe, bao gồm nhiều nhóm Agile phối hợp làm việc theo một nhịp điệu chung và hướng tới một mục tiêu chung. ART được quản lý thông qua một ART Backlog duy nhất và hoạt động trong quy trình **Continuous Delivery Pipeline** để đảm bảo sản phẩm được phát triển và phát hành liên tục, đáp ứng nhanh chóng nhu cầu của khách hàng.

[11:22 PM] Hao Nguyen Phi



Bức ảnh này mô tả các sự kiện chính (ART events) trong một **Program Increment (PI)**, một chu kỳ thời gian trong quy trình làm việc của **Agile Release Train (ART)**. Dưới đây là giải thích chi tiết về từng sự kiện, thời gian giới hạn (timebox) và giá trị mà chúng mang lại:

1. PI Planning (Lập kế hoạch PI):
2. Coach Sync (Đồng bộ hóa huấn luyện viên):
3. System Demo (Trình diễn hệ thống):
4. Inspect and Adapt (Kiểm tra và Thích ứng):
5. Tóm lại:

* **Timebox: 2 ngày**
* **Giá trị:**
  + Trong sự kiện này, các nhóm Agile tham gia vào ART sẽ **cam kết với một tập hợp các mục tiêu cần hoàn thành trong suốt PI**. Đây là một sự kiện quan trọng để thiết lập kế hoạch tổng thể và xác định rõ ràng các mục tiêu mà ART sẽ theo đuổi trong chu kỳ PI.
* **Giải thích:** Lập kế hoạch PI là sự kiện quan trọng giúp các nhóm đồng bộ với nhau, hiểu rõ nhiệm vụ của mình và đảm bảo rằng tất cả đều hướng tới cùng một mục tiêu. Điều này giúp giảm thiểu rủi ro về xung đột mục tiêu và tăng cường sự phối hợp trong quá trình làm việc.
* **Timebox: 1 giờ**
* **Giá trị:**
  + Đây là sự kiện mà các nhóm trong ART **đồng bộ với nhau về tiến độ của PI**. Coach Sync giúp các nhóm đảm bảo rằng tất cả đều đi đúng hướng và bất kỳ vấn đề nào cũng được nhận diện và xử lý kịp thời.
* **Giải thích:** Sự kiện này giúp duy trì sự liên kết chặt chẽ giữa các nhóm trong ART, đảm bảo rằng mọi người đều hiểu rõ về tiến độ và có cơ hội chia sẻ thông tin, giải quyết khó khăn.
* **Timebox: 1 giờ**
* **Giá trị:**
  + Các sản phẩm hoàn thành được **trình bày cho các bên liên quan** để nhận phản hồi. Đây là cơ hội để đánh giá xem liệu sản phẩm có đáp ứng được mong đợi hay không và xác định những cải tiến cần thiết.
* **Giải thích:** System Demo là cơ hội để thể hiện kết quả của công việc, thu thập phản hồi từ các bên liên quan và điều chỉnh kế hoạch nếu cần. Điều này giúp đảm bảo rằng sản phẩm cuối cùng sẽ phù hợp với yêu cầu của khách hàng.
* **Timebox: ½ ngày**
* **Giá trị:**
  + ART **xem xét và cải thiện quy trình** của mình trước khi bắt đầu PI tiếp theo. Đây là bước quan trọng để đánh giá hiệu quả của các phương pháp đã sử dụng và thực hiện các điều chỉnh cần thiết nhằm cải thiện hiệu suất.
* **Giải thích:** Sự kiện này giúp ART không ngừng cải tiến quy trình làm việc của mình, học hỏi từ những gì đã diễn ra trong PI trước đó và sẵn sàng cho các thử thách mới. Đây là một phần quan trọng của quy trình Agile, đảm bảo rằng ART luôn tiến bộ và không ngừng nâng cao chất lượng công việc.

Các sự kiện của ART trong một Program Increment (PI) đều nhằm mục tiêu đảm bảo sự phối hợp nhịp nhàng giữa các nhóm, liên tục cải tiến quy trình và sản phẩm, và tối đa hóa giá trị được cung cấp cho khách hàng. Những sự kiện này giúp ART giữ vững nhịp điệu làm việc, phản hồi nhanh chóng với thay đổi, và duy trì sự nhất quán trong việc đạt được các mục tiêu kinh doanh.



1. **Chuẩn bị cho PI Planning:**

Trước khi sự kiện bắt đầu, SM/TC đảm bảo rằng nhóm đã được chuẩn bị đầy đủ về các tính năng và công việc sắp tới. Họ phối hợp với Product Owners, Business Owners, và các bên liên quan để hiểu rõ yêu cầu và xác định các công việc quan trọng.

* **Lập kế hoạch PI:** SM/TC hướng dẫn nhóm tạo ra bản kế hoạch PI ban đầu, bao gồm việc xác định các mục tiêu PI, viết các mục tiêu SMART (Specific, Measurable, Achievable, Realistic, Time-bound), và đánh giá các rủi ro liên quan đến ART.
* **Điều phối với các nhóm khác:** SM/TC đảm bảo rằng các nhóm trong ART hợp tác hiệu quả, xử lý các phụ thuộc giữa các nhóm, và duy trì liên lạc với các chuyên gia chủ đề (SMEs) và các bên liên quan trong suốt quá trình lập kế hoạch.
  1. Một Scrum Master cần có gì để làm tốt công việc của mình

**2. Supporting Iteration Execution (Hỗ trợ thực hiện Iteration)**

* **Nội dung chính:** SM/TC hỗ trợ các nhóm Agile trong suốt quá trình thực hiện **Iteration**, giúp nhóm đạt được các mục tiêu đã đề ra cho mỗi chu kỳ phát triển ngắn hạn.
* **Giải thích chi tiết:**
  + **Tổ chức các sự kiện nhóm:** SM/TC đảm bảo rằng các sự kiện nhóm như họp lập kế hoạch Iteration, họp đồng bộ (Daily Stand-up), họp đánh giá (Iteration Review), và họp tổng kết (Retrospective) diễn ra hiệu quả và đúng thời điểm.
  + **Duy trì nhịp điệu của ART:** SM/TC giúp các nhóm duy trì nhịp điệu phát triển nhất quán với yêu cầu đồng bộ hóa của ART, đảm bảo rằng các nhóm hoạt động nhịp nhàng cùng nhau để đạt được mục tiêu chung của PI.
  + **Hỗ trợ Product Owner:** SM/TC làm việc chặt chẽ với Product Owner để đảm bảo rằng backlog của nhóm được quản lý rõ ràng và ưu tiên đúng cách, đồng thời hỗ trợ nhóm trong việc áp dụng các phương pháp như Design Thinking để phát triển sản phẩm một cách tập trung vào khách hàng.

**3. Improving Flow (Cải thiện luồng công việc)**

* **Nội dung chính:** SM/TC chịu trách nhiệm tối ưu hóa **luồng công việc** của nhóm để loại bỏ các trở ngại, giảm thiểu lãng phí, và tăng cường hiệu suất.
* **Giải thích chi tiết:**
  + **Thiết lập Kanban Board:** SM/TC giúp nhóm thiết lập và duy trì bảng Kanban để trực quan hóa công việc và tối ưu hóa luồng công việc. Việc này bao gồm việc xác định quy trình làm việc, giới hạn công việc đang thực hiện (WIP), và phát hiện các nút thắt trong quy trình.
  + **Đo lường và tối ưu hóa:** SM/TC giúp nhóm xây dựng các chỉ số để đánh giá và cải thiện hiệu suất tổng thể, như các chỉ số về luồng công việc, năng lực đội ngũ, và kết quả đạt được. Họ liên tục điều chỉnh quy trình để đảm bảo rằng nhóm hoạt động hiệu quả nhất.

**4. Building High-Performing Teams (Xây dựng các nhóm hiệu suất cao)**

* **Nội dung chính:** SM/TC đóng vai trò quan trọng trong việc xây dựng và duy trì **nhóm Agile có hiệu suất cao**, giúp nhóm đạt được các mục tiêu phát triển sản phẩm một cách hiệu quả và liên tục cải tiến.
* **Giải thích chi tiết:**
  + **Khuyến khích và hỗ trợ:** SM/TC tạo điều kiện cho nhóm tự quản lý, khuyến khích sự hợp tác và trách nhiệm giữa các thành viên. Họ thúc đẩy tinh thần đồng đội và tạo ra môi trường làm việc tin cậy và minh bạch.
  + **Giải quyết xung đột:** SM/TC đóng vai trò là người hòa giải, giúp nhóm giải quyết các xung đột nội bộ và cải thiện các động lực làm việc nhóm. Họ cũng hỗ trợ nhóm trong việc thiết lập và đạt được các mục tiêu rõ ràng và có giá trị cao trong mỗi Iteration.

**5. Improving ART Performance (Cải thiện hiệu suất của ART)**

* **Nội dung chính:** SM/TC không chỉ hỗ trợ từng nhóm riêng lẻ mà còn đóng góp vào việc cải thiện **hiệu suất tổng thể của toàn bộ Agile Release Train (ART)**.
* **Giải thích chi tiết:**
  + **Thúc đẩy hợp tác giữa các nhóm:** SM/TC khuyến khích sự hợp tác giữa các nhóm trong ART, đảm bảo rằng mọi người đều làm việc cùng nhau một cách hiệu quả để đạt được các mục tiêu chung. Họ tham gia vào các sự kiện quan trọng của ART như Coach Sync, System Demo, và Inspect & Adapt.
  + **Xây dựng niềm tin với các bên liên quan:** SM/TC giúp nhóm xây dựng niềm tin với các bên liên quan bằng cách đảm bảo rằng các mục tiêu của PI được lên kế hoạch và thực hiện một cách đáng tin cậy. Họ đóng vai trò quan trọng trong việc duy trì sự cân bằng giữa tính linh hoạt và tính dự đoán trong phát triển sản phẩm.

**Tóm tắt:**

**Scrum Master/Team Coach (SM/TC)** trong **Scaled Agile Framework (SAFe)** đóng một vai trò chiến lược và quan trọng trong việc đảm bảo các nhóm Agile hoạt động hiệu quả và liên tục cải tiến. SM/TC không chỉ hỗ trợ từng nhóm trong suốt quá trình phát triển mà còn góp phần nâng cao hiệu suất của toàn bộ **Agile Release Train (ART)**. ART là một tổ chức ảo bao gồm nhiều nhóm Agile, hoạt động đồng bộ và phối hợp với nhau để phát triển và cung cấp sản phẩm hoặc tính năng thông qua các chuỗi phát hành liên tục.

Một trong những trách nhiệm chính của SM/TC là **Facilitating Program Increment (PI) Planning**. PI là một chu kỳ thời gian định kỳ, thường kéo dài từ 8-12 tuần, trong đó các nhóm lập kế hoạch và thực hiện công việc để đạt được mục tiêu của ART. Trong quá trình này, SM/TC hợp tác chặt chẽ với **Release Train Engineer (RTE)** và các **Subject Matter Experts (SMEs)** để chuẩn bị và điều phối sự kiện PI Planning. Họ hỗ trợ các nhóm lập kế hoạch cho các **Iteration** (chu kỳ phát triển ngắn hạn), xác định các mục tiêu PI và xử lý các phụ thuộc giữa các nhóm.

Trong suốt quá trình **Iteration Execution**, SM/TC hỗ trợ các nhóm Agile tổ chức các sự kiện như họp lập kế hoạch Iteration, họp đồng bộ hàng ngày, họp đánh giá và họp tổng kết. SM/TC cũng làm việc cùng **Product Owner (PO)** để quản lý **backlog** của nhóm, đảm bảo rằng các công việc được ưu tiên đúng cách và hỗ trợ nhóm trong việc áp dụng các phương pháp như **Design Thinking** để phát triển sản phẩm tập trung vào khách hàng.

Một nhiệm vụ khác của SM/TC là **Improving Flow**, tức là tối ưu hóa luồng công việc của nhóm. Họ thiết lập và duy trì bảng **Kanban** để trực quan hóa công việc và loại bỏ các trở ngại, giúp nhóm duy trì hiệu suất cao. SM/TC sử dụng các chỉ số để đánh giá và cải thiện quy trình, như các chỉ số về luồng công việc và hiệu suất nhóm.

**Building High-Performing Teams** cũng là một phần quan trọng trong vai trò của SM/TC. Họ xây dựng và duy trì một nhóm Agile có hiệu suất cao bằng cách thúc đẩy tinh thần đồng đội, khuyến khích sự tự quản lý và tạo ra môi trường làm việc tin cậy và minh bạch. SM/TC cũng giúp giải quyết các xung đột nội bộ và thúc đẩy sự phát triển kỹ năng của các thành viên trong nhóm.

Cuối cùng, SM/TC đóng góp vào việc **Improving ART Performance** bằng cách thúc đẩy hợp tác giữa các nhóm trong ART, xây dựng niềm tin với các bên liên quan và hỗ trợ các sự kiện như **Inspect and Adapt (I&A)**. Họ cũng giúp chuẩn bị cho **Innovation and Planning (IP) Iteration**, một khoảng thời gian đặc biệt trong SAFe dành cho đổi mới, lên kế hoạch và cải tiến.

Một **Scrum Master** có vai trò quan trọng trong việc đảm bảo rằng nhóm Agile hoạt động hiệu quả và tuân thủ các nguyên tắc của Scrum. Để làm tốt vai trò của mình, một Scrum Master cần thực hiện các nhiệm vụ chính sau:

1. Hỗ trợ và huấn luyện nhóm Scrum
2. Loại bỏ các trở ngại (Impediments)
3. Tạo điều kiện thuận lợi cho các sự kiện Scrum
4. Làm cầu nối giữa nhóm và các bên liên quan
5. Thúc đẩy cải tiến liên tục
6. Xây dựng và duy trì tinh thần nhóm
7. Liên tục học hỏi và phát triển

* **Giúp nhóm hiểu và tuân thủ quy trình Scrum:** Scrum Master phải đảm bảo rằng nhóm hiểu rõ và áp dụng đúng các giá trị, nguyên tắc, và quy trình của Scrum. Điều này bao gồm việc huấn luyện nhóm về cách thực hiện các sự kiện Scrum như Sprint Planning, Daily Scrum, Sprint Review, và Sprint Retrospective.
* **Huấn luyện kỹ năng tự quản lý:** Scrum Master giúp nhóm phát triển khả năng tự tổ chức và tự quản lý, để nhóm có thể chủ động giải quyết vấn đề mà không cần sự can thiệp từ bên ngoài.
* **Nhận diện và giải quyết các trở ngại:** Scrum Master phải liên tục theo dõi các trở ngại ảnh hưởng đến tiến độ của nhóm và tìm cách giải quyết chúng. Trở ngại có thể bao gồm các vấn đề nội bộ trong nhóm hoặc các vấn đề đến từ bên ngoài, như sự phụ thuộc vào các nhóm khác hoặc sự thiếu hụt tài nguyên.
* **Facilitate các cuộc họp Scrum:** Scrum Master chịu trách nhiệm tổ chức và tạo điều kiện cho các cuộc họp Scrum diễn ra hiệu quả. Họ đảm bảo rằng các cuộc họp được tiến hành đúng thời gian, đúng mục tiêu và tất cả các thành viên đều tham gia tích cực.
* **Hỗ trợ lập kế hoạch Sprint (Sprint Planning):** Scrum Master giúp nhóm lập kế hoạch cho các Sprint một cách rõ ràng và có thể đo lường được. Họ giúp nhóm xác định các mục tiêu Sprint và đảm bảo rằng nhóm có đủ năng lực để hoàn thành công việc đã đặt ra.
* **Đại diện cho nhóm trước các bên liên quan:** Scrum Master đảm bảo rằng nhóm có sự giao tiếp hiệu quả với Product Owner và các bên liên quan khác. Họ giúp bảo vệ nhóm khỏi các yêu cầu không hợp lý và giúp các bên liên quan hiểu rõ tiến độ và kết quả công việc của nhóm.
* **Hỗ trợ Product Owner:** Scrum Master giúp Product Owner quản lý backlog sản phẩm, đảm bảo rằng các mục tiêu của Product Owner được truyền đạt rõ ràng và đúng lúc cho nhóm.
* **Khuyến khích nhóm nhìn lại và cải tiến:** Thông qua các buổi Sprint Retrospective, Scrum Master khuyến khích nhóm xem xét lại những gì đã làm tốt và những gì cần cải thiện. Họ giúp nhóm xác định các thay đổi cần thiết và thực hiện chúng trong các Sprint tiếp theo.
* **Áp dụng phương pháp Lean và Agile:** Scrum Master hướng dẫn nhóm áp dụng các phương pháp Lean và Agile để tối ưu hóa quy trình làm việc, giảm thiểu lãng phí và tăng giá trị mà nhóm mang lại.
* **Thúc đẩy văn hóa hợp tác:** Scrum Master xây dựng một môi trường làm việc tích cực, nơi các thành viên trong nhóm cảm thấy an toàn để chia sẻ ý tưởng, thử nghiệm, và học hỏi từ nhau.
* **Giải quyết xung đột:** Khi có xung đột xảy ra trong nhóm, Scrum Master giúp giải quyết một cách tích cực, đảm bảo rằng mối quan hệ giữa các thành viên không bị tổn hại và nhóm vẫn tập trung vào mục tiêu chung.
* **Cập nhật kiến thức:** Scrum Master cần liên tục cập nhật các xu hướng, phương pháp và công cụ mới trong Agile và Scrum để hỗ trợ nhóm tốt hơn.
* **Tham gia vào cộng đồng Scrum:** Tham gia vào các cộng đồng Scrum hoặc Agile, chia sẻ và học hỏi kinh nghiệm từ các Scrum Master khác là một cách để phát triển kỹ năng và mở rộng kiến thức.



1. Self-organizing (Tự tổ chức)

Nhóm có khả năng tự quản lý công việc của mình mà không cần sự can thiệp chi tiết từ bên ngoài. Các thành viên biết cách phân công nhiệm vụ và tự chịu trách nhiệm về kết quả công việc.

2. Effective decision-making (Ra quyết định hiệu quả)

Nhóm có khả năng đưa ra quyết định một cách nhanh chóng và chính xác, dựa trên thông tin có sẵn và sự hiểu biết chung về mục tiêu của nhóm.

3. Open and clear communication (Giao tiếp cởi mở và rõ ràng)

Các thành viên trong nhóm giao tiếp một cách minh bạch và thẳng thắn, đảm bảo rằng mọi người đều hiểu rõ mục tiêu, tiến độ và bất kỳ vấn đề nào có thể phát sinh.

4. Value diversity (Trân trọng sự đa dạng)

Nhóm đánh giá cao sự khác biệt về kỹ năng, quan điểm và kinh nghiệm của các thành viên, sử dụng sự đa dạng này để đưa ra những giải pháp sáng tạo và toàn diện hơn.

5. Mutual trust (Niềm tin lẫn nhau)

Các thành viên trong nhóm tin tưởng lẫn nhau, tin rằng mọi người đều làm việc vì lợi ích chung của nhóm và tổ chức.

6. Healthy conflict (Xung đột lành mạnh)

Nhóm có khả năng xử lý xung đột một cách tích cực, xem xung đột như một cơ hội để thảo luận và cải thiện, thay vì một điều gì đó tiêu cực.

7. Clear goals and purpose (Mục tiêu và mục đích rõ ràng)

Nhóm có một mục tiêu và mục đích rõ ràng, cụ thể, giúp định hướng công việc và đảm bảo rằng mọi người đều hướng tới cùng một đích đến.

8. Concentration and focus (Tập trung và chú ý)

Nhóm duy trì sự tập trung cao độ vào các nhiệm vụ quan trọng, tránh bị phân tâm bởi những công việc không cần thiết.

9. Ownership and accountability (Chủ động và chịu trách nhiệm)

Mỗi thành viên trong nhóm đều chủ động nắm bắt công việc của mình và chịu trách nhiệm về kết quả, không đùn đẩy trách nhiệm cho người khác.

10. Understand work's impact on organization (Hiểu tác động của công việc đối với tổ chức)

Nhóm hiểu rõ công việc của mình ảnh hưởng như thế nào đến mục tiêu tổng thể của tổ chức, giúp định hướng và động viên họ hoàn thành nhiệm vụ một cách hiệu quả nhất.

11. Aligned and collaborative (Đồng bộ và hợp tác)

Nhóm hoạt động một cách đồng bộ và hợp tác chặt chẽ, đảm bảo rằng mọi người đều làm việc cùng nhau một cách hiệu quả để đạt được các mục tiêu chung.

12. Safe atmosphere to take risks (Môi trường an toàn để chấp nhận rủi ro)

Nhóm tạo ra một môi trường mà các thành viên cảm thấy an toàn khi thử nghiệm và chấp nhận rủi ro mà không sợ bị chỉ trích hay phạt.

13. Effective, timely feedback (Phản hồi hiệu quả và kịp thời)

Nhóm cung cấp và nhận phản hồi một cách hiệu quả và kịp thời, giúp cải thiện quy trình làm việc và sản phẩm.

14. Sufficient resources for local control (Tài nguyên đủ để kiểm soát cục bộ)

Nhóm có đủ tài nguyên và quyền tự quyết để kiểm soát công việc của mình, giúp họ hoạt động hiệu quả mà không phải phụ thuộc quá nhiều vào bên ngoài.

15. Success focus over failure avoidance (Tập trung vào thành công hơn là tránh thất bại)

Nhóm chú trọng vào việc đạt được thành công hơn là lo lắng về việc tránh thất bại, điều này khuyến khích sự sáng tạo và thử nghiệm.

16. Abilities balanced with challenge (Cân bằng khả năng với thách thức)

Nhóm đảm bảo rằng công việc được phân công phù hợp với khả năng của từng thành viên, nhưng cũng đủ thách thức để giữ động lực và phát triển.

17. Engagement (Tham gia tích cực)

Các thành viên trong nhóm tham gia tích cực vào công việc, đóng góp ý tưởng và giải pháp, và cam kết với mục tiêu chung.

18. Have fun with work and each other (Vui vẻ với công việc và với nhau)

Nhóm không chỉ làm việc hiệu quả mà còn tạo ra một môi trường làm việc vui vẻ, nơi các thành viên cảm thấy hứng thú với công việc và tận hưởng sự hợp tác với đồng đội.



1. Backlog Refinement (Làm rõ backlog)
2. Iteration Planning (Lập kế hoạch Iteration)
3. Team Sync (Đồng bộ nhóm)
4. Iteration Review (Đánh giá Iteration)
5. Iteration Retrospective (Tổng kết Iteration)

* **Timebox:** 1 giờ
* **Giá trị:** Nhóm chuẩn bị các yêu cầu cho việc lập kế hoạch Iteration.
* **Giải thích:** Trong sự kiện này, nhóm cùng với Product Owner xem xét và làm rõ các mục công việc (user stories, tasks, bugs, etc.) trong backlog, đảm bảo rằng chúng đã sẵn sàng để được đưa vào Iteration sắp tới. Điều này giúp nhóm hiểu rõ hơn về công việc cần thực hiện và chuẩn bị tốt hơn cho Iteration Planning.
* **Timebox:** 2 đến 4 giờ
* **Giá trị:** Nhóm cam kết với một tập hợp các mục tiêu sẽ được hoàn thành trong Iteration.
* **Giải thích:** Trong cuộc họp này, nhóm thảo luận và quyết định các công việc sẽ được thực hiện trong Iteration sắp tới, đặt ra các mục tiêu cụ thể và đảm bảo rằng tất cả các thành viên đều hiểu rõ những gì cần làm. Cuộc họp này rất quan trọng để đảm bảo rằng nhóm có một kế hoạch cụ thể và rõ ràng cho Iteration.
* **Timebox:** 15 phút
* **Giá trị:** Các thành viên trong nhóm đồng bộ với nhau về tiến độ của các mục tiêu trong Iteration.
* **Giải thích:** Đây là cuộc họp hàng ngày, thường được gọi là Daily Scrum hoặc Stand-up, nơi các thành viên trong nhóm cập nhật cho nhau về tiến độ công việc, thảo luận về các trở ngại và lập kế hoạch cho ngày làm việc tiếp theo. Cuộc họp này giúp giữ cho nhóm đồng bộ và nhận diện sớm các vấn đề tiềm ẩn.
* **Timebox:** 1 giờ
* **Giá trị:** Nhóm gặp gỡ các bên liên quan để đánh giá các sản phẩm đã hoàn thành và nhận phản hồi.
* **Giải thích:** Sau khi một Iteration kết thúc, nhóm trình bày các kết quả đã đạt được cho các bên liên quan, nhận phản hồi về công việc đã thực hiện. Điều này giúp đảm bảo rằng sản phẩm phát triển theo đúng hướng và đáp ứng nhu cầu của khách hàng hoặc người dùng cuối.
* **Timebox:** 1 đến 1,5 giờ
* **Giá trị:** Nhóm đánh giá lại và cải thiện quy trình làm việc trước khi bắt đầu Iteration tiếp theo.
* **Giải thích:** Đây là cơ hội để nhóm nhìn lại toàn bộ Iteration, thảo luận về những gì đã làm tốt, những gì cần cải thiện và đề ra các hành động để nâng cao hiệu quả công việc trong các Iteration tiếp theo. Cuộc họp này rất quan trọng để thúc đẩy cải tiến liên tục trong nhóm.
* **Tóm lại:**

Các sự kiện này là nền tảng của quy trình Agile, giúp nhóm liên tục lập kế hoạch, thực hiện, đánh giá và cải thiện công việc của mình. Việc tuân thủ thời gian giới hạn (timebox) đảm bảo rằng các cuộc họp diễn ra hiệu quả và không làm gián đoạn quá trình phát triển. Scrum Master thường chịu trách nhiệm điều phối các sự kiện này, đảm bảo rằng nhóm tuân theo quy trình và đạt được mục tiêu đã đề ra.



Bức ảnh này thảo luận về các thách thức thường gặp trong các cuộc họp, đặc biệt là những vấn đề có thể làm cho các cuộc họp trở nên không hiệu quả và không mang lại giá trị. Dưới đây là mô tả chi tiết về các thách thức này:

Thách **thức trong các cuộc họp:**

1. **The purpose is not clear (Mục đích không rõ ràng):**
   * Khi mục đích của cuộc họp không được xác định rõ ràng, người tham gia có thể không hiểu được lý do tại sao họ cần phải tham gia, dẫn đến sự thiếu tập trung và hiệu quả thấp.
2. **There are no actionable outcomes (Không có kết quả cụ thể có thể thực hiện):**
   * Cuộc họp kết thúc mà không đưa ra được các kết quả cụ thể hoặc các bước hành động rõ ràng, khiến cho thời gian họp bị lãng phí mà không tạo ra giá trị thực tế.
3. **They may result in unproductive conflict (Có thể dẫn đến xung đột không hiệu quả):**
   * Trong cuộc họp, các ý kiến khác nhau có thể dẫn đến xung đột không mang tính xây dựng, khiến cuộc họp trở nên căng thẳng và không đạt được mục tiêu.
4. **They may be boring (Cuộc họp có thể nhàm chán):**
   * Nếu cuộc họp không được tổ chức một cách sinh động và hấp dẫn, người tham gia có thể mất hứng thú, dẫn đến thiếu sự tham gia tích cực và kết quả không hiệu quả.
5. **Conversation may divert from agenda to deep discussion (Cuộc trò chuyện có thể đi lạc khỏi chương trình nghị sự sang thảo luận sâu):**
   * Khi cuộc họp đi lạc khỏi chương trình nghị sự ban đầu và chuyển sang những thảo luận không cần thiết hoặc không liên quan, thời gian họp bị kéo dài mà không đạt được kết quả mong muốn.

**Những cuộc họp như vậy thêm rất ít giá trị (Such meetings add almost no value):**

* Các cuộc họp không rõ ràng, không có kết quả cụ thể, hoặc gặp các vấn đề như đã liệt kê ở trên thường không mang lại giá trị thực sự cho tổ chức hoặc nhóm, và có thể làm lãng phí thời gian của mọi người.

**Cuộc họp không hiệu quả có thể (và nên) được khắc phục (Ineffective meetings can (and should) be fixed):**

* Các cuộc họp không hiệu quả không phải là điều không thể tránh khỏi. Với sự chuẩn bị đúng đắn, xác định mục tiêu rõ ràng, quản lý tốt thời gian và chương trình nghị sự, những vấn đề này có thể được khắc phục để làm cho các cuộc họp trở nên hữu ích và giá trị hơn.

**Tóm lại:**

Cuộc họp là một phần quan trọng trong việc điều hành nhóm và dự án, nhưng nếu không được quản lý tốt, chúng có thể trở thành nguồn lãng phí thời gian và gây căng thẳng. Việc xác định và khắc phục các thách thức nêu trên là cần thiết để đảm bảo rằng các cuộc họp mang lại giá trị thực sự cho tổ chức.

[11:22 PM] Hao Nguyen Phi



Bức ảnh này trình bày một bảng so sánh giữa các hành vi cần thay đổi (Move away from...) và những hành vi mới cần hướng tới (Move toward...) trong quá trình chuyển đổi sang cách làm việc hiệu quả hơn trong một môi trường Agile hoặc Scrum. Dưới đây là phân tích chi tiết:

**Move away from... (Tránh xa các hành vi sau đây):**

1. **Coordinating individual contributions (Phối hợp các đóng góp cá nhân):**
   * Việc tập trung vào phối hợp các đóng góp riêng lẻ của từng cá nhân có thể dẫn đến sự thiếu đồng bộ và không tận dụng được sức mạnh của nhóm.
2. **Acting as a subject matter expert (Đóng vai trò là chuyên gia chủ đề):**
   * Khi người lãnh đạo chỉ tập trung vào việc thể hiện chuyên môn của mình, họ có thể làm giảm sự phát triển và sáng tạo của nhóm.
3. **Driving toward specific outcomes (Chỉ hướng đến các kết quả cụ thể):**
   * Tập trung quá mức vào các kết quả cụ thể có thể khiến nhóm bỏ qua các khía cạnh quan trọng khác của quy trình làm việc, như sự học hỏi và cải tiến liên tục.
4. **Knowing the answer (Biết câu trả lời):**
   * Khi người lãnh đạo cho rằng họ biết tất cả các câu trả lời, điều này có thể làm giảm sự tự chủ và khả năng tìm ra giải pháp sáng tạo của nhóm.
5. **Directing (Chỉ đạo):**
   * Chỉ đạo quá mức có thể tạo ra môi trường làm việc bị động, nơi các thành viên nhóm chỉ làm theo chỉ dẫn mà không phát triển khả năng tự quản lý.
6. **Talking about deadlines and technical options (Chỉ nói về các hạn chót và lựa chọn kỹ thuật):**
   * Việc tập trung quá nhiều vào hạn chót và các lựa chọn kỹ thuật có thể làm mất đi tầm nhìn tổng thể về giá trị kinh doanh mà nhóm cần đạt được.
7. **Driving 'the right' (your) decisions (Ép buộc các quyết định của mình):**
   * Khi người lãnh đạo ép buộc các quyết định của mình, điều này có thể làm giảm tinh thần và sự đóng góp ý kiến của các thành viên khác trong nhóm.
8. **Fixing problems rather than helping others fix them (Tự mình sửa các vấn đề thay vì giúp người khác sửa chữa):**
   * Tự mình sửa các vấn đề có thể tạo ra sự phụ thuộc, trong khi việc giúp nhóm tự giải quyết vấn đề sẽ thúc đẩy sự tự quản và phát triển kỹ năng của nhóm.

**Move toward... (Hướng tới các hành vi sau đây):**

1. **Coaching the whole team to collaborate (Huấn luyện toàn bộ nhóm để hợp tác):**

* Hướng tới việc thúc đẩy sự hợp tác toàn diện trong nhóm, giúp tất cả các thành viên cùng nhau làm việc để đạt được mục tiêu chung.

1. **Being a facilitator (Trở thành người điều phối):**

* Thay vì chỉ là chuyên gia, người lãnh đạo nên đóng vai trò điều phối, giúp nhóm tự giải quyết các vấn đề và phát triển các giải pháp.

1. **Being invested in the team’s overall performance (Đầu tư vào hiệu suất tổng thể của nhóm):**

* Thay vì chỉ tập trung vào các kết quả cụ thể, hãy đầu tư vào sự phát triển và hiệu suất tổng thể của nhóm, giúp nhóm không chỉ đạt được kết quả mà còn học hỏi và cải tiến liên tục.

1. **Letting the team find their own way (Để nhóm tự tìm ra con đường của mình):**

* Khuyến khích nhóm tự tìm ra các giải pháp và cách thức làm việc hiệu quả, tạo ra sự tự chủ và sáng tạo trong nhóm.

1. **Guiding (Hướng dẫn):**

* Thay vì chỉ đạo, hãy hướng dẫn nhóm, cung cấp hỗ trợ khi cần thiết nhưng để nhóm tự quyết định và học hỏi từ quá trình.

1. **Focusing on business value delivery (Tập trung vào việc cung cấp giá trị kinh doanh):**

* Thay vì chỉ tập trung vào hạn chót và kỹ thuật, hãy đặt mục tiêu cung cấp giá trị thực sự cho khách hàng và tổ chức, giúp định hướng cho công việc của nhóm.

1. **Doing the right thing for the business right now (Làm điều đúng đắn cho doanh nghiệp ngay lúc này):**

* Thay vì ép buộc các quyết định của mình, hãy cân nhắc điều gì là tốt nhất cho doanh nghiệp trong hoàn cảnh hiện tại và đưa ra quyết định phù hợp.

1. **Facilitating team problem-solving (Hỗ trợ nhóm trong việc giải quyết vấn đề):**

* Hướng tới việc hỗ trợ nhóm tự giải quyết các vấn đề, thay vì tự mình sửa chữa, giúp nhóm phát triển kỹ năng và tăng cường sự tự tin.

Tóm **lại:**

Bảng so sánh này nhấn mạnh sự chuyển đổi từ cách tiếp cận lãnh đạo truyền thống sang một phong cách lãnh đạo hỗ trợ và huấn luyện trong môi trường Agile hoặc Scrum. Điều này không chỉ giúp tăng cường sự tự chủ và sáng tạo của nhóm mà còn thúc đẩy sự phát triển cá nhân và hiệu suất chung của cả nhóm.



Bức ảnh này liệt kê các lý do tại sao câu hỏi lại có sức mạnh và quan trọng trong việc giao tiếp, học hỏi và phát triển cá nhân hoặc nhóm. Dưới đây là giải thích chi tiết về từng lý do:

* **1. They are thought-provoking (Câu hỏi kích thích suy nghĩ)**
* Câu hỏi có khả năng khuyến khích người được hỏi suy nghĩ sâu sắc hơn về vấn đề đang được thảo luận. Điều này có thể dẫn đến sự nhận thức mới mẻ hoặc những hiểu biết sâu hơn.
* **2. They generate curiosity in the listener (Câu hỏi tạo ra sự tò mò ở người nghe)**
* Khi được đặt ra câu hỏi, người nghe thường cảm thấy tò mò và muốn tìm hiểu thêm. Điều này giúp duy trì sự chú ý và sự tham gia tích cực trong cuộc trò chuyện.
* **3. They channel focus (Câu hỏi điều hướng sự tập trung)**
* Câu hỏi giúp hướng sự tập trung của người tham gia vào vấn đề chính, giúp tránh bị phân tâm và đảm bảo rằng cuộc thảo luận đi đúng hướng.
* **4. They generate energy and forward movement (Câu hỏi tạo ra năng lượng và thúc đẩy sự tiến bộ)**
* Câu hỏi có thể kích thích năng lượng trong cuộc thảo luận, tạo động lực cho người tham gia để tiếp tục tìm kiếm giải pháp và tiến về phía trước.
* **5. They stimulate reflective conversation (Câu hỏi kích thích cuộc trò chuyện mang tính phản chiếu)**
* Câu hỏi giúp người tham gia tự đánh giá lại suy nghĩ của mình, thúc đẩy họ xem xét lại các quan điểm và cách tiếp cận của mình trong một bối cảnh mới.
* **6. They surface underlying assumptions (Câu hỏi làm rõ các giả định ngầm)**
* Thông qua câu hỏi, các giả định ngầm có thể được bộc lộ và kiểm tra, giúp đảm bảo rằng mọi người đều hiểu rõ các tiền đề và lý do đằng sau những quan điểm hoặc quyết định.
* **7. They invite creativity and new possibilities (Câu hỏi mời gọi sự sáng tạo và các khả năng mới)**
* Câu hỏi có thể mở ra không gian cho sự sáng tạo, khuyến khích người tham gia nghĩ về các giải pháp mới mẻ và độc đáo mà trước đó có thể chưa từng được cân nhắc.
* **8. They inspire more questions (Câu hỏi khuyến khích thêm nhiều câu hỏi khác)**
* Câu hỏi thường dẫn đến những câu hỏi khác, mở rộng cuộc thảo luận và đào sâu vào các khía cạnh khác nhau của vấn đề, tạo ra một môi trường học hỏi liên tục.
* **9. They help reach for deep meaning (Câu hỏi giúp tìm kiếm ý nghĩa sâu sắc)**
* Câu hỏi có thể hướng dẫn người tham gia đến với những hiểu biết sâu sắc và ý nghĩa hơn về một vấn đề, giúp họ thấy được tầm quan trọng và giá trị của những gì đang được thảo luận.

**Tóm lại:**

Câu hỏi không chỉ là công cụ để thu thập thông tin mà còn là phương tiện mạnh mẽ để thúc đẩy sự suy nghĩ, kích thích sáng tạo, và tạo động lực cho sự tiến bộ. Đặt câu hỏi đúng cách có thể dẫn đến những cuộc thảo luận phong phú hơn, giúp người tham gia khám phá những góc nhìn mới và đạt được hiểu biết sâu sắc hơn về vấn đề.



Bức ảnh này đưa ra những hướng dẫn về việc hợp tác với các nhóm khác trong một tổ chức Agile, đặc biệt là trong bối cảnh Agile Release Train (ART). Dưới đây là giải thích chi tiết cho từng điểm:

**1. Integrate their work often with other teams on the ART (at least multiple times per Iteration)**

* **Giải thích:** Nhóm nên tích hợp công việc của mình với các nhóm khác trong ART thường xuyên, ít nhất là nhiều lần trong mỗi Iteration. Điều này giúp đảm bảo rằng tất cả các nhóm đang làm việc đồng bộ, giảm thiểu rủi ro khi phát hiện các vấn đề tích hợp muộn và đảm bảo rằng sản phẩm cuối cùng có thể được triển khai một cách trơn tru.

**2. Work with the System Team on automated system-level tests**

* **Giải thích:** Nhóm nên làm việc với đội System Team để thực hiện các kiểm thử tự động ở cấp độ hệ thống. Điều này giúp đảm bảo rằng các chức năng và tích hợp hệ thống hoạt động như mong đợi, giảm thiểu lỗi và đảm bảo chất lượng của sản phẩm.

**3. Join other team's planning events, demos, or syncs when important issues arise**

* **Giải thích:** Nhóm nên tham gia các sự kiện lập kế hoạch, demo, hoặc các buổi đồng bộ của các nhóm khác khi xuất hiện các vấn đề quan trọng. Việc này giúp giải quyết kịp thời các phụ thuộc giữa các nhóm, đảm bảo rằng tất cả các bên liên quan đều hiểu rõ vấn đề và cùng nhau tìm ra giải pháp.

**4. Work with the System Architect to better manage dependencies with other teams**

* **Giải thích:** Nhóm nên làm việc với System Architect để quản lý tốt hơn các phụ thuộc giữa các nhóm. System Architect có vai trò quan trọng trong việc hiểu và điều phối các yếu tố kỹ thuật phức tạp, đảm bảo rằng các phụ thuộc được giải quyết kịp thời và không gây cản trở cho tiến độ công việc của các nhóm.
* **Tóm lại:**

Sự hợp tác giữa các nhóm là yếu tố quan trọng trong một tổ chức Agile, đặc biệt là trong một môi trường làm việc phức tạp như ART. Những hướng dẫn này giúp đảm bảo rằng các nhóm làm việc đồng bộ, quản lý tốt các phụ thuộc và tích hợp công việc một cách trơn tru, từ đó tạo ra sản phẩm chất lượng cao và đáp ứng đúng yêu cầu của khách hàng.



**Coach Syncs là gì?**

* **Mục tiêu:** Cuộc họp Coach Syncs là nơi các **Scrum Masters** và **Release Train Engineer (RTE)** gặp nhau để có cái nhìn rõ ràng hơn về tiến độ của các nhóm cũng như các trở ngại mà ART đang gặp phải. Mục tiêu của cuộc họp này là để đảm bảo rằng các vấn đề được giải quyết kịp thời và các phụ thuộc giữa các nhóm được quản lý hiệu quả.

**Tần suất tổ chức:**

* **Thời gian tổ chức:** Cuộc họp Coach Syncs thường được tổ chức hàng tuần. Tuy nhiên, nếu cần thiết, có thể tổ chức cuộc họp này thường xuyên hơn để đảm bảo mọi thứ vẫn nằm trong tầm kiểm soát.

**Đặc điểm của cuộc họp:**

* **Timeboxed:** Cuộc họp này có thời gian giới hạn để đảm bảo rằng nó diễn ra hiệu quả và không lãng phí thời gian. Sau cuộc họp chính, có thể có thêm một buổi gặp gỡ để giải quyết các vấn đề cụ thể và tìm ra giải pháp cho các trở ngại đã được xác định.

**Vai trò của Release Train Engineer (RTE):**

* **RTE:** RTE đóng vai trò quan trọng trong việc dẫn dắt và điều phối cuộc họp Coach Syncs. RTE chịu trách nhiệm đảm bảo rằng các Scrum Masters có đầy đủ thông tin cần thiết để hỗ trợ nhóm của họ và giải quyết các vấn đề liên quan đến tiến độ và phụ thuộc.

**Ý nghĩa của Coach Syncs:**

* **Duy trì sự đồng bộ và tiến độ:** Coach Syncs giúp "synchronize and keep the train on the tracks" (đồng bộ hóa và giữ cho con tàu trên đường ray), nghĩa là đảm bảo rằng tất cả các nhóm trong ART đều hoạt động đồng bộ, không có vấn đề nào bị bỏ qua và mọi người đều hướng tới mục tiêu chung.
* **Tóm lại:**

Cuộc họp Coach Syncs là một phần quan trọng của việc quản lý ART trong môi trường Agile. Nó giúp đảm bảo rằng các Scrum Masters và RTE có thể phối hợp chặt chẽ, xử lý các vấn đề nhanh chóng và giữ cho toàn bộ ART di chuyển về phía trước một cách suôn sẻ và hiệu quả.

1. Resolve team conflicts

1. Lesson 3: Experiencing PI Planning
   1. 3.1 PI Planning basics

1. 3.2 Draft PI plans
2. 3.3 Final plans and business value
3. 3.4 Final plan review and PI Objectives
4. 3.5 Facilitating PI Planning