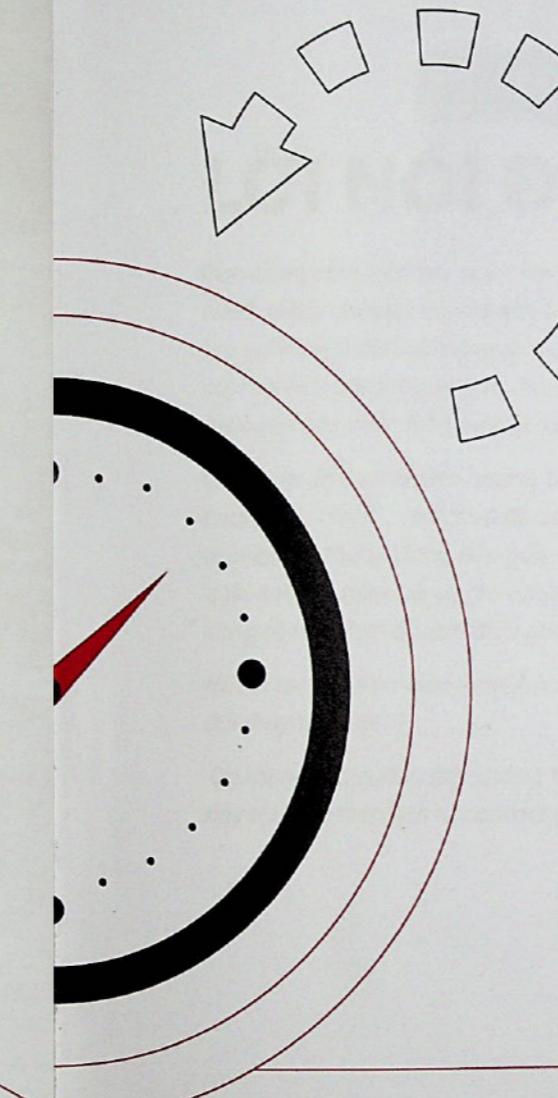




# CẨM NANG SCRUM

► LÀM CHỦ PHƯƠNG PHÁP NĂNG SUẤT VÀ SÁNG TẠO GẤP ĐÔI ◀

Dương Trọng Tấn  
Nguyễn Việt Khoa  
Phạm Anh Đới  
Nguyễn Khắc Nhật  
(Học viện Agile)



# CẨM NANG SCRUM

► LÀM CHỦ PHƯƠNG PHÁP NĂNG SUẤT VÀ SÁNG TẠO GẤP ĐÔI ◀

Dương Trọng Tấn  
Nguyễn Việt Khoa  
Phạm Anh Đới  
Nguyễn Khắc Nhật

**CẨM NANG SCRUM**  
(Làm chủ phương pháp năng suất và sáng tạo gấp đôi)  
Bản quyền © Học viện Agile, 2018

Sách được xuất bản và phát hành theo hợp đồng hợp tác xuất bản độc quyền  
ký kết giữa tác giả và Công ty Cổ phần Văn hóa Truyền thông Sông

Không phần nào trong xuất bản phẩm này được phép sao chép hay phát hành  
dưới bất kỳ hình thức hoặc phương tiện nào mà không có sự cho phép trước  
bằng văn bản của Công ty Cổ phần Văn hóa Truyền thông Sông.

Chúng tôi luôn mong muốn nhận được những ý kiến đóng góp của quý vị độc giả  
để sách ngày càng hoàn thiện hơn.

Góp ý về sách, liên hệ về bản thảo và bản dịch: publication@iamsong.vn  
Liên hệ ebooks, hợp tác xuất bản & truyền thông: hoptac@iamsong.vn  
Liên hệ dịch vụ tư vấn, đại diện & giao dịch bản quyền: copyright@iamsong.vn

Biên mục trên xuất bản phẩm của Thư viện Quốc gia Việt Nam

Cẩm nang Scrum : Làm chủ phương pháp năng suất và sáng tạo gấp đôi / Dương Trọng Tấn,  
Nguyễn Việt Khoa, Phạm Anh Đới, Nguyễn Khắc Nhật. - H. : Thế giới ; Công ty Văn hóa Truyền  
thông Sông, 2018. - 264tr. ; 24cm  
ISBN 9786047753482

1. Scrum 2. Phát triển 3. Phần mềm máy tính  
005.1 - dc23

TGM0156p-CIP

## LỜI NÓI ĐẦU

Bạn đang cầm trên tay cuốn sách về Scrum đầu tiên do người Việt biên soạn. Đây là cuốn sách dành cho đông đảo bạn đọc, từ lập trình viên, chuyên viên phát triển sản phẩm, nhà quản trị dự án, ScrumMaster, Product Owner, doanh nhân khởi nghiệp trong lĩnh vực đổi mới sáng tạo, cho đến giảng viên công nghệ thông tin và nhà đào tạo Agile. Chúng tôi mong muốn bạn có được một tài liệu hành động giúp bạn nhanh chóng chiếm lĩnh được một trong những phương thức tổ chức công việc và quản trị dự án hiệu quả bậc nhất hiện nay: Scrum.

Chúng ta đã chứng kiến những thay đổi lớn lao của công nghệ nói riêng và thế giới nói chung. Những Google, iPhone, iPad, Amazon, Facebook, Uber... ra đời và đào lộn trật tự thế giới. Hiện nay sự thay đổi còn mạnh mẽ và sâu sắc hơn nữa với cuộc cách mạng công nghiệp lần thứ tư đang đến gần. Việc trang bị tư duy và phương pháp tổ chức công việc kiểu mới có tính sáng tạo, có khả năng thích nghi cao với biến động, gia tăng được sức mạnh cạnh tranh bền vững là điều tối quan trọng. Scrum chắc chắn là một trong những công cụ tuyệt vời để làm điều đó ở những cấp độ khác nhau, từ các cá nhân, nhóm cộng tác, cho đến các tổ chức quy mô lớn.

Nhóm tác giả Học viện Agile hân hạnh đồng hành cùng các bạn học hỏi, thực hành Scrum để kiến tạo những giá trị mới, tạo những đổi thay tích cực.

Dù rất nỗ lực, cuốn sách không tránh khỏi những thiếu sót. Chúng tôi luôn mong chờ phản hồi của độc giả để cải tiến liên tục tài liệu này tại hòm thư điện tử contact@hocvienagile.com.

**Học viện Agile**

## LỜI CẢM ƠN

Cuốn sách này được viết dựa trên tài liệu giảng dạy Scrum do chúng tôi khởi thảo cuối năm 2011, và liên tục cải tiến để giúp cho việc học Scrum ngày càng dễ dàng hơn. Phần nhiều những cải tiến đó đến từ người học, mà chúng tôi gọi là những "đối tác phát triển" – những người giúp nhau cùng tiến bộ. Chúng tôi vô cùng biết ơn những "đồng tác giả" mà chúng tôi không thể kể tên ở đây.

Trong khi tổ chức lại thành một cuốn cẩm nang có chất lượng cao như phiên bản bạn cầm trên tay, rất nhiều người đã góp ý trực tiếp về nội dung, bố cục cũng như cách thức trình bày. Chúng tôi xin chân thành cảm ơn bạn bè khắp nơi đã cung cấp những lời khuyên quý báu. Trong đó xin đặc biệt cảm ơn chị Nguyễn Phương Anh – Đại học FPT, anh Nguyễn Tuân – Giám đốc đào tạo FPT Aptech Hà Nội, anh Phạm Mạnh Lân – Co-founder NAL Group, anh Hán Văn Thắng – CEO Deha Software, anh Nguyễn Ngọc Tú và Nguyễn Minh Tâm đến từ NAL Vietnam, anh Nguyễn Văn Hiển đến từ Agile Vietnam, anh Tô Hải Sơn – Giám đốc vận hành NTQ Solution và những người bạn mà ở đây chúng tôi chưa thể kể hết.

Xin cảm ơn anh Đoàn Xuân Trường đã thiết kế bìa; cảm ơn chị Nguyễn Thị Phương – đồng nghiệp đã lao tâm khổ tứ để cuốn sách này đến được với bạn đọc. Cảm ơn Kim Cúc đã hiện thực hóa những ý tưởng trình bày khác thường của nhóm tác giả.

Cuối cùng, chúng tôi xin vô cùng cảm ơn bạn – người đang đọc cuốn sách này, vì chính bạn là người cùng chúng tôi tạo ra giá trị cho cuốn sách và lan tỏa những giá trị Scrum.

# SCRUM

## Cùng bạn dùng sách

Cuốn sách này được viết ra dành cho những nhà phát triển và quản lý bước đầu sử dụng Scrum. Vì thế nó được thiết kế tối ưu cho việc học và tra cứu. Chúng tôi nghĩ rằng 80% giá trị của cuốn sách nằm ở cách bạn đọc vận dụng nó. Do đó, chúng tôi mong muốn thảo luận đôi chút về cấu trúc của cuốn sách cũng như cách đọc, cách dùng cuốn sách này.

Chương đầu tiên rất lý thuyết nhưng là nền tảng. Có thể là chương khó đọc nhất, nhưng lại quan trọng nhất vì nó giải thích cặn kẽ triết lý đứng đằng sau Scrum. Bạn không thể thực hành tốt Scrum mà không nắm rõ những nguyên lý Agile có tính dẫn đường. Do vậy hãy đọc qua Chương 1 trước khi đọc các chương tiếp theo, và thỉnh thoảng đọc lại chương này để suy ngẫm thêm. Nếu bạn nóng ruột muốn tìm hiểu Scrum thì có thể chỉ cần lướt qua Chương 1, nhưng hãy quay trở lại sau khi đã đọc xong cả cuốn sách.

Chương 2, 3, 4, 5 mô tả chi tiết khung Scrum và cách vận hành trong thực tiễn. Cuốn sách này hướng đến việc thực hành “đúng” Scrum. Vì thế hãy đọc kỹ, và tự trả lời các câu hỏi cuối chương. Việc trả lời các câu hỏi này giúp bạn đưa kiến thức vào thực tiễn, bối cảnh đặc thù của bạn. Ngoài ra, việc trả lời các câu hỏi này giúp cho bạn bồi đắp khuyết được một khía cạnh không được đề cập trong khuôn khổ một cuốn cẩm nang ngắn gọn: vận dụng Scrum “hợp lý”.

Chương 6 thảo luận về các tình huống sử dụng Scrum và những điều ghi nhớ khi lần đầu tiên đưa Scrum vào tổ chức. Bạn có thể nhảy cóc đến chương này nếu tò mò về các bước cụ thể áp dụng Scrum.

Các chương 7, 8, 9 có thể coi là phần mở rộng của cuốn sách nhằm cung cấp một bức tranh lớn hơn khi vận dụng Scrum. Người mới bắt đầu với Scrum, đối tượng chính của cuốn sách này, có vẻ ít khi vận dụng những nội dung ở các chương này, nhưng bạn cần nắm được một bức tranh lớn hơn để hình dung ra một lộ trình dài hơi cho việc biến Scrum thành một đường lối làm việc nhất quán, cải tiến liên tục, hiệu quả và thành công bền vững.

Riêng Chương 9 có đề cập nhiều kỹ thuật đặc thù Agile mà nhiều lập trình viên thấy lạ lẫm và rất thích thú. Bạn hãy lướt qua, tìm hiểu kỹ thêm (tham khảo danh mục đọc thêm ở cuối sách) nếu muốn đi xa hơn.

Nếu bạn để ý thì đây là cuốn sách kỹ thuật bằng tiếng Việt hiếm hoi có Bảng chỉ mục. Hãy tận dụng nó cho việc học và tra cứu. Hãy dùng Bảng chỉ mục (Index) ở cuối sách, Mục lục chi tiết, và phần Phụ lục 1 (Thuật ngữ Scrum) để tra cứu, chuyển qua chuyển lại các phần của cuốn sách, hoặc để tra cứu thêm thông tin trên Internet về một khái niệm được đề cập trong sách.

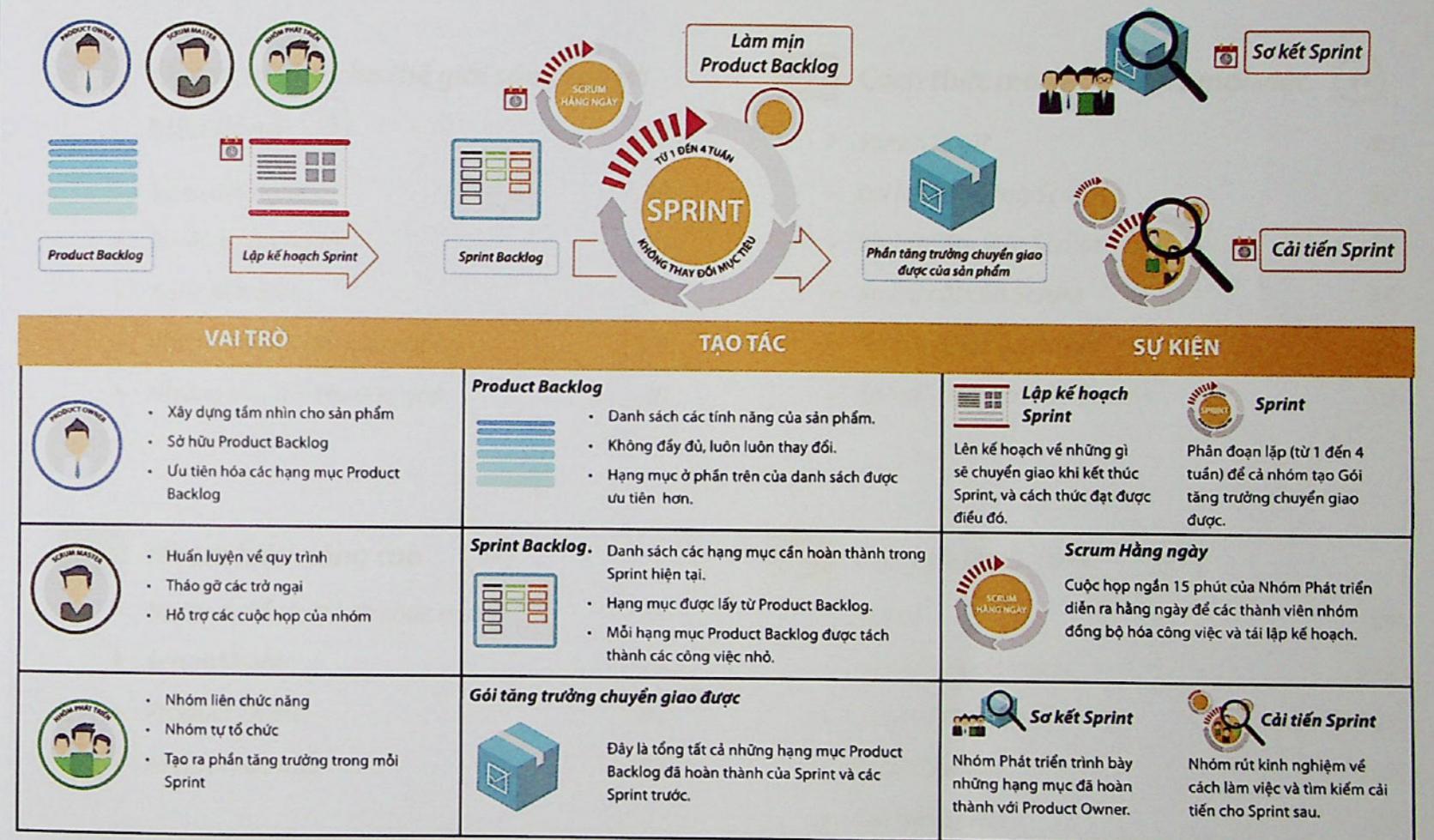
Trong sách, thỉnh thoảng có những ô “Dừng và nghĩ” với một số câu hỏi hướng dẫn, như những nốt lặng trong các bản nhạc. Bạn hãy dừng lại một chút, động não trong phút chốc để tiêu hóa nội dung của cuốn sách. Ngoài ra, các Danh mục Kiểm tra được đặt vào một số phần quan trọng cũng có thể giúp bạn trong việc tự đánh giá được mức độ đúng đắn và thành thực trong việc vận dụng Scrum trong nhóm của mình.

Một số tài liệu tham khảo cho cuốn sách được chúng tôi cập nhật tại [www.hocvienagile.com/cam-nang-scrum](http://www.hocvienagile.com/cam-nang-scrum).



Quét mã QR tại các mục để tìm hiểu thêm về các chủ đề tương ứng

Chúc bạn dùng sách thật hiệu quả!



**BA CHÂN: MINH BẠCH - THANH TRA - THÍCH NGHI**  
**GIÁ TRỊ: DŨNG CẨM - TẬP TRUNG - CAM KẾT - CỜI MỞ - TÔN TRỌNG**

# MỤC LỤC

## 1 Khung tư duy cho thế giới sáng tạo và biến động 13

- ▶ *Sự ra đời Agile* 16
- ▶ *Tuyên ngôn Agile* 22
- ▶ *Agile Mindset* 37
- ▶ *Khi nào Agile, khi nào không* 38
- ▶ *Những câu hỏi thường gặp* 40

## 2 Cách thức mới để tổ chức mọi việc 45

- ▶ *Scrum là gì?* 46
- ▶ *Lợi ích khi dùng Scrum* 52
- ▶ *Khung làm việc Scrum* 54
- ▶ *Ba trụ cột của Scrum* 57
- ▶ *Năm giá trị của Scrum* 58
- ▶ *Sprint – trái tim của Scrum* 59

## 3 Nhóm hiệu năng cao 69

- ▶ *Nhóm tự tổ chức liên chức năng* 71
- ▶ *ScrumMaster* 75
- ▶ *Product Owner* 84
- ▶ *Nhóm Phát triển* 88

## 4 Các sự kiện Scrum 107

- ▶ *Sprint* 109
- ▶ *Lập kế hoạch Sprint* 112
- ▶ *Scrum Hàng ngày* 121
- ▶ *Sơ kết Sprint* 124
- ▶ *Cải tiến Sprint* 127

## 5 Các tạo tác và công cụ 138

- ▶ *Product Backlog* 140
- ▶ *Sprint Backlog* 150
- ▶ *Phản tăng trưởng chuyển giao được* 155
- ▶ *Định nghĩa Hoàn thành* 158
- ▶ *Biểu đồ Sprint Burndown* 166
- ▶ *Biểu đồ Release Burndown* 171

## 7 Scrum phân tán 198

- ▶ *Làm việc phân tán* 199
- ▶ *Thử thách đối với nhóm phân tán* 200
- ▶ *Giao tiếp hiệu quả hơn* 202
- ▶ *Xây dựng niềm tin* 203
- ▶ *Hợp Tiễn-kế-hoạch-hóa* 204
- ▶ *Lưu ý công tác hậu cần* 205
- ▶ *Công cụ cộng tác phân tán* 206

## 6 Scrum trong tổ chức 179

- ▶ *Các vấn đề tổ chức cần lưu tâm* 180
- ▶ *Các tình huống và mức độ sử dụng Scrum* 181
- ▶ *Quản lý sự thay đổi* 188
- ▶ *Thay đổi tư duy* 190
- ▶ *Phát triển năng lực Scrum* 192
- ▶ *Đo lường và đánh giá hiệu suất* 195

## 8 Scrum với quy mô lớn 210

- ▶ *Thử thách chính đối với quy mô lớn* 211
- ▶ *Khung Nexus cho Scrum quy mô lớn* 212
- ▶ *Cơ chế cộng tác trong Nexus* 215
- ▶ *Công cụ và quy trình* 216
- ▶ *Scrum quy mô lớn với LeSS* 217

## 9 Các chủ đề nâng cao 225

- ▶ *Quản lý yêu cầu người dùng với User Story* 227
- ▶ *Ước tính linh hoạt* 232
- ▶ *Planning Poker* 233
- ▶ *Phát triển hướng kiểm thử* 235
- ▶ *Tái cấu trúc mã nguồn* 238
- ▶ *ATDD và BDD* 240
- ▶ *Lập trình cặp* 241
- ▶ *Làm bản mẫu* 242
- ▶ *Design thinking* 243
- ▶ *DevOps* 244
- ▶ *Tích hợp liên tục* 245

## Phụ lục 1: Thuật ngữ 251

## Phụ lục 2: Danh mục kiểm tra Scrum 256

## Bảng chỉ mục 261

# 1

## KHUNG TƯ DUY CHO THẾ GIỚI SÁNG TẠO VÀ BIẾN ĐỘNG

Trong chương này...

- Thế giới biến động, đặc trưng VUCA và sự cần thiết của tư duy mới
- Những vấn đề nổi cộm trong phát triển sản phẩm
- Sự ra đời của Tuyên ngôn Agile
- Cách tiếp cận tăng trưởng lặp
- Mô thức Agile là gì?
- Khi nào Agile, khi nào không

“

*May mắn là những gì xảy ra khi sự chuẩn bị gặp được cơ hội.*

*Seneca*

# Thế giới biến động và sự cần thiết của lối tư duy mới

Thế giới đang chứng kiến những biến động không ngừng. Làn sóng khởi nghiệp, cách mạng công nghiệp lần thứ 4 đang phả hơi nóng vào mọi lĩnh vực của đời sống. Với sự nở rộ của các ứng dụng thông minh, công nghệ nền tảng Blockchain tác động mạnh vào hạ tầng của ngành công nghiệp truyền thông, hay sự kết nối sâu rộng giữa mọi thứ xung quanh chúng ta thông qua Internet of Things, thế giới đang trở nên công nghệ hóa, số hóa, thông minh hóa từng ngày. Chúng vừa tạo ra rất nhiều cơ hội nhưng cũng để lộ rất nhiều thách thức đòi hỏi chúng ta phải trang bị những công cụ tư duy và làm việc mới để thích ứng nhanh chóng.

Bối cảnh ngày nay có nhiều tính chất được thể hiện trong thuật ngữ ngày càng được nghiên cứu kỹ hơn mang tên VUCA. Đó là thuật ngữ viết tắt của bốn chữ: Volatility (Thế giới trở nên mong manh, dễ thay đổi, cả về cường độ và tốc độ), Uncertainty (Thế giới trở nên không chắc chắn, khó đoán và quá nhiều sự kiện bất ngờ xảy đến), Complexity (Thế giới ngày càng phức tạp, nhiều yếu tố đan xen tác động qua lại, không dễ để nhận diện mối quan hệ nguyên nhân - kết quả, việc ra quyết định trở nên khó khăn hơn), Ambiguity (Thế giới trở nên mập mờ hơn, khó nhận diện bản chất hơn, dễ hiểu nhầm hơn và cũng khó ra quyết định hơn). Khi các dấu hiệu VUCA càng rõ nét, là khi chúng ta cần một lối tư duy mới, linh hoạt hơn, có khả năng thích ứng cao độ và giúp chúng ta có thể học hỏi với tốc độ nhanh để sáng tạo và kiến lập giá trị vượt trội. Đây không chỉ là điều quan trọng đối với những nhà lãnh đạo doanh nghiệp, mà còn là điều thiết yếu của từng nhóm làm việc, từng cá nhân đang hoạt động trong lĩnh vực tri thức và sáng tạo.

## Câu chuyện ngành phần mềm

Ngành phát triển phần mềm đã có hơn nửa thế kỷ hình thành và phát triển, tuy nhiên việc phát triển phần mềm và quản trị các dự án phần mềm chưa bao giờ hết thử thách. Rất nhiều những vấn đề không ngừng lặp đi lặp lại, khiến đội ngũ phát triển và các nhà quản trị phải đau đầu. Dưới đây là một số vấn đề tiêu biểu.

"Theo cách thức phát triển truyền thống, dự án được lập kế hoạch cẩn thận, được tiến hành với rất nhiều khâu trung gian, và khách hàng ngồi chờ. Các bên liên quan hầu như chỉ nhận được báo cáo, tài liệu chứ không nhận được phần mềm để dùng. Thông tin về phần mềm rất mù mờ. Lâu tới ngày phát hành và không minh bạch về tiến độ khiến khách hàng và các bên liên quan sốt ruột, lo lắng và không làm cách nào để cung cấp các phản hồi có ý nghĩa, khó hợp tác với đội phát triển cho ra sản phẩm tốt nhất."

"Kế hoạch đã định, nhưng nhiều rủi ro xuất hiện: Lỗi xuất hiện ngày càng nhiều, hiểu sai yêu cầu, những khó khăn về mặt công nghệ, một vài thành viên có vấn đề, không làm việc như kỳ vọng và sản phẩm chậm ngày phát hành."

"Mọi thứ đã được tích hợp, nhưng sản phẩm thiếu ổn định do không kiểm soát được chất lượng. Ở rất nhiều dự án, mọi công việc đã hoàn thành nhưng giai đoạn tích hợp và ổn định hệ thống thật là thảm họa, không biết khi nào kết thúc."

"Kế hoạch thường xuyên sai sót đòi hỏi chúng ta phải đầu tư nhiều thời gian hơn vào việc xây dựng kế hoạch. Nhưng dù cố gắng đến mấy thì kế hoạch vẫn không đâu vào đâu và chúng ta lại bị phàn nàn là xây dựng kế hoạch chưa kỹ. Lập kế hoạch thật là một cơn ác mộng, một nhiệm vụ không thể hoàn thành."

"Khi đã thực hiện xong phân tích yêu cầu, thiết kế và bắt đầu lập trình, khách hàng đổi yêu cầu. Vì đó là yêu cầu rất quan trọng, bạn không thể từ chối. Nhóm rất khó xử. Họ không biết làm thế nào bởi việc thay đổi rất khó."

"Ban đầu sản phẩm còn chạy tốt, nhưng lượng mã nguồn ngày một tăng lên, áp lực thời gian, v.v. nên chất lượng sản phẩm cứ giảm sút, kế hoạch thất bại, chất lượng ngày một giảm, tất cả mọi người phải làm thêm giờ với áp lực rất cao, thành viên không hạnh phúc."

"Để đương đầu với các vấn đề trên, rất nhiều nỗ lực theo nhiều hướng khác nhau đã được thực hiện qua hàng chục năm. Kết quả của chúng là những phương pháp, kỹ thuật hoặc các quy trình kỹ thuật mà Agile chỉ là một trong những sáng kiến quan trọng nhất."

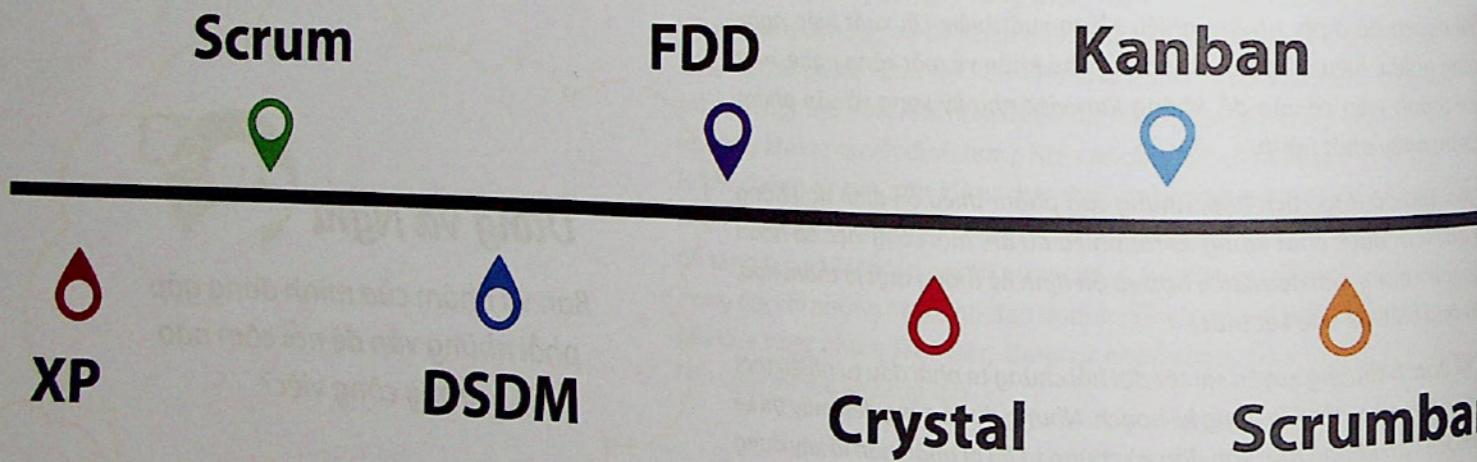
### Dừng và Nghỉ

Bạn và nhóm của mình đang gặp phải những vấn đề nổi cộm nào trong công việc?

# KHỦNG HOẢNG PHƯƠNG PHÁP LUẬN, VÀ SỰ RA ĐỜI CỦA AGILE

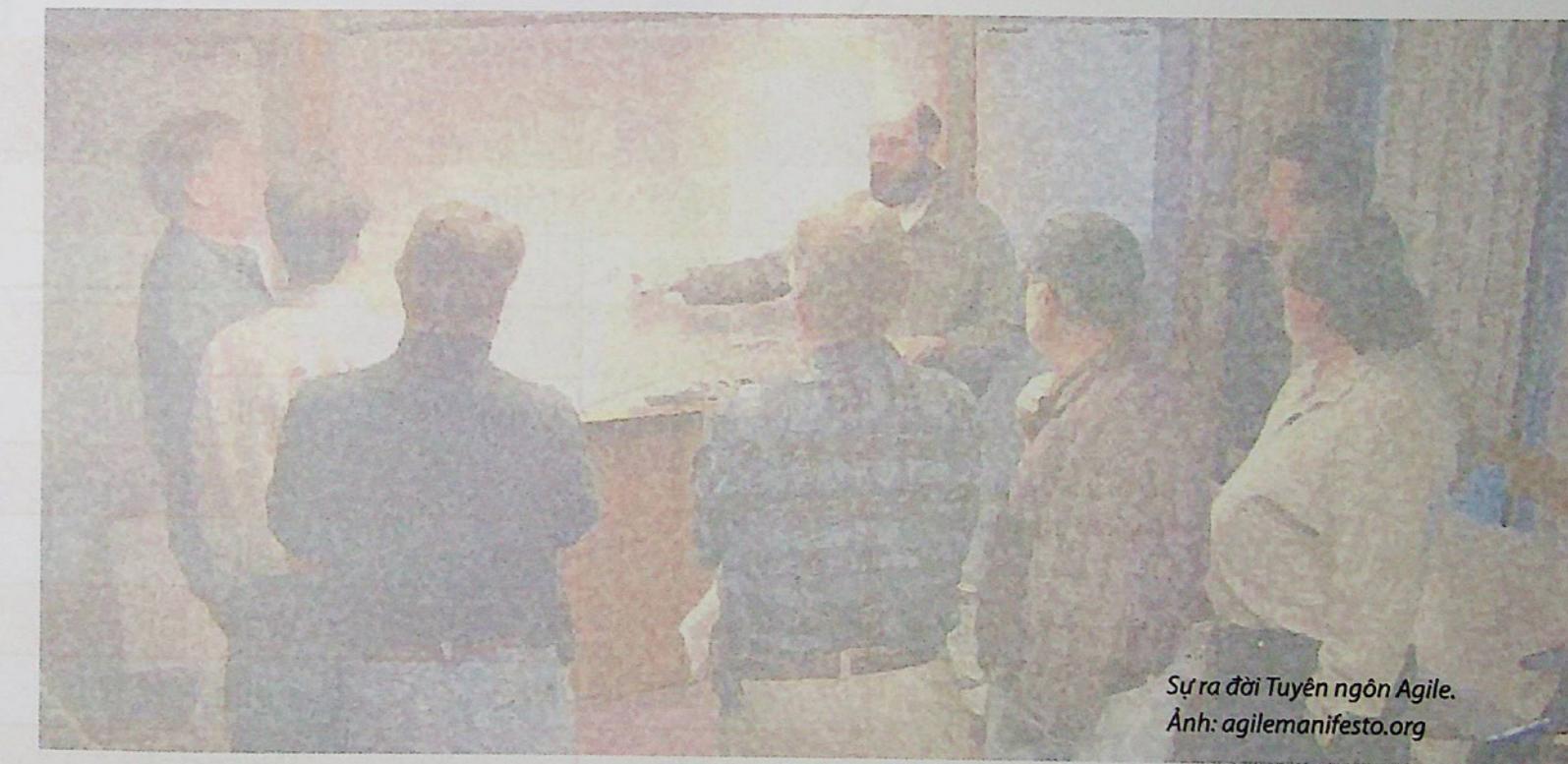
Thập kỉ 80 của thế kỉ XX chứng kiến một thời kỳ khủng hoảng phương pháp luận phát triển phần mềm, do tỷ lệ thất bại của các dự án phần mềm quá cao. Hàng loạt nỗ lực của các nhà thực hành cũng như giới hàn lâm đã cố gắng tìm ra phương pháp hữu hiệu để đảm bảo tính hiệu quả trong phát triển phần mềm.

Thập kỉ 90, nhiều phương pháp phát triển phần mềm với quy trình nhẹ (light weight) và linh hoạt ra đời như: XP, Scrum, FDD, Crystal, và nhanh chóng được lan rộng.



Từ ngày 11 - 13 tháng 2 năm 2001, 17 nhà phát minh và nhà thực hành đã họp với nhau tại bang Utah, Hoa Kỳ, để thảo luận về hướng đi mới trong phương pháp luận phát triển phần mềm. Họ đã đi đến thống nhất và cho ra đời bản Tuyên ngôn Agile (The Manifesto for Agile Software Development, gọi tắt là Tuyên ngôn Agile) và đánh

dấu một xu thế mới trong phát triển phần mềm. Ngày nay chúng ta gọi là Agile, tức là chỉ chung một họ phương pháp phát triển phần mềm có chung sự chia sẻ các giá trị và nguyên tắc được phát biểu trong Tuyên ngôn Agile. Biện pháp thực hành có thể rất khác nhau, nhưng triết lý chung thì giống nhau.



Sự ra đời Tuyên ngôn Agile.  
Ảnh: agilemanifesto.org

Sau một thập kỷ rưỡi ra đời, Agile đã cải thiện đáng kể khả năng thành công của các dự án.

Báo cáo CHAOS<sup>1</sup> của Standish Group năm 2015 cho thấy, các dự án Agile có tỷ lệ thành công cao hơn 3 lần so với dự án truyền thống.

Dự án lớn áp dụng Agile có tỷ lệ thành công cao gấp 6 lần

Dự án vừa áp dụng Agile có tỷ lệ thành công cao gấp gần 4 lần

Dự án nhỏ áp dụng Agile cũng cho tỷ lệ thành công cao hơn

Quy mô Dự án	Phương pháp	Thành công	Thử thách	Thất bại
Tổng kết	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Lớn	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Vừa	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Nhỏ	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

<sup>1</sup> Báo cáo CHAOS, giai đoạn 2011-2015. Số liệu các dự án được phân tích: 10.000+

**Dự án thành công:** đúng thời gian, đúng ngân sách với mọi tính năng và kết quả thỏa mãn.

**Dự án thử thách:** hoàn thành nhưng không đạt một trong các tiêu chí đúng ngân sách, đúng thời gian hoặc kết quả không thỏa mãn.

**Dự án thất bại:** dự án bị cắt ở một thời điểm nào đó trong quá trình phát triển.

Những nghiên cứu chi tiết hơn định lượng được tác động của những phương pháp Agile lên năng suất lao động.

Như trong nghiên cứu của Sutherland và đồng nghiệp năm 2008, năng suất của nhóm Scrum cao hơn tới gần 9 lần so với nhóm sử dụng phương pháp truyền thống.

Loại nhóm	Nhóm Scrum tập trung	Waterfall	Nhóm Scrum phân tán Sirsi Dynix	Nhóm Scrum phân tán Xebia
Chỉ số				
Người * tháng	54	540	827	125 (-77%) *
Số dòng mã Java	51.000	58.000	671.668	100.000 (-42%) *
Function points (FP)	959	900	12.673	1.887 (+210%) *
FP cho mỗi KLOC **	18,80	15,52	18,86	18,87 (+22%) *
Lượng FP trên người * tháng	17,8	1,7	15,3	15,1 (+888%) *

\*Theo Sutherland, J., Schoonheim, G., Rusten- burg, E., & Rijk, M. (2008, August). Fully distributed scrum: The secret sauce for hyperproductive offshored development teams. In Agile, 2008. AGILE'08. Conference (pp. 339-344). IEEE.

Trong nghiên cứu này, kích thước phần mềm được đo bằng hai thước đo phổ biến trong phát triển phần mềm là số dòng mã và điểm chức năng (function point).

Năng suất tính theo điểm chức năng của nhóm Scrum cao hơn nhóm truyền thống  $15,3/1,7 = 8,88$  lần.

## Câu chuyện linh hoạt hóa Microsoft



\* So sánh với phương pháp truyền thống waterfall

\*\* KLOC: 1000 dòng mã

Sách "Software in 30 days" (Làm phần mềm trong vòng 30 ngày) của Jeff Sutherland và Ken Schwaber có ghi vào khoảng 2008, Microsoft bắt đầu đưa Scrum vào để đổi cách làm việc thuộc khuôn khổ dự án Visual Studio Online để hòng gia tăng tốc độ chuyển giao sản phẩm ra thị trường. Trước đó, Microsoft luôn gặp vấn đề với chất lượng của các phần mềm cũng như tốc độ phát hành sản phẩm mới. Việc thí điểm thành công triển khai phương thức phát triển sản phẩm mới Scrum giúp Microsoft giải quyết rõ ráo những vấn đề tồn tại hàng thập kỉ này. Từ chỗ thí điểm, Microsoft đưa Scrum thành cách tổ chức công việc chính thức (khi quyết định mở rộng phạm vi áp dụng vào 2011), rồi thành quy trình chủ đạo như hiện nay. Theo báo cáo của Steve Denning trên tạp chí Forbes<sup>1</sup>, thì quá trình chuyển đổi này có thể tính là đã hoàn tất vào năm 2014. Khoảng 4000 lập trình viên của Microsoft được biên chế trong hàng trăm nhóm nhỏ có quy mô 10-12 người, làm việc trong các chu trình ngắn (Sprint) 3 tuần để liên tục chuyển giao sản phẩm có chất lượng tới khách hàng. Nếu trước kia Microsoft có thể mất vài năm để cho ra một phiên bản sản phẩm mới, thì nay thời gian đó rút ngắn xuống còn từ vài tuần đến vài tháng, chất lượng thì khác hẳn.

Vào khoảng cuối những năm 2000, nhiều người nhận rõ sự khác biệt về chất lượng phần mềm của Microsoft viết ra so với các đối thủ chính của nó. Ngày nay thì ít còn thấy những so sánh kiểu như vậy nữa.

Quá trình chuyển đổi sang Agile của Microsoft cũng phản ánh sự biến chuyển theo bể sâu của ngành công nghệ đầu thế kỷ XXI. Ngày nay, Agile đã trở thành thiết yếu của các tổ chức sáng tạo.

Phản sau chúng ta cùng khảo sát một văn bản lịch sử quan trọng đánh dấu một cách thức tư duy mới mang tên Agile.

<sup>1</sup> Ngạc nhiên chưa, Microsoft đã Agile" (<https://www.forbes.com/sites/stevedenning/2015/10/27/surprise-microsoft-is-agile>)

# TUYÊN NGÔN PHÁT TRIỂN PHẦN MỀM LINH HOẠT

Chúng tôi đã tìm ra cách tốt hơn để phát triển phần mềm thông qua việc thực hành và giúp đỡ người khác thực hiện. Qua đó, chúng tôi đánh giá cao:

**Cá nhân và tương tác** *hơn là quy trình và công cụ;*

**Phần mềm chạy tốt** *hơn là tài liệu đầy đủ;*

**Cộng tác với khách hàng** *hơn là đàm phán hợp đồng;*

**Phản hồi với thay đổi** *hơn là bám sát kế hoạch.*

Mặc dù các thứ bên phải vẫn còn giá trị, chúng tôi đánh giá cao hơn các mục ở bên trái.

Theo AgileManifesto.org

# 12 nguyên lý phía sau tuyên ngôn Agile

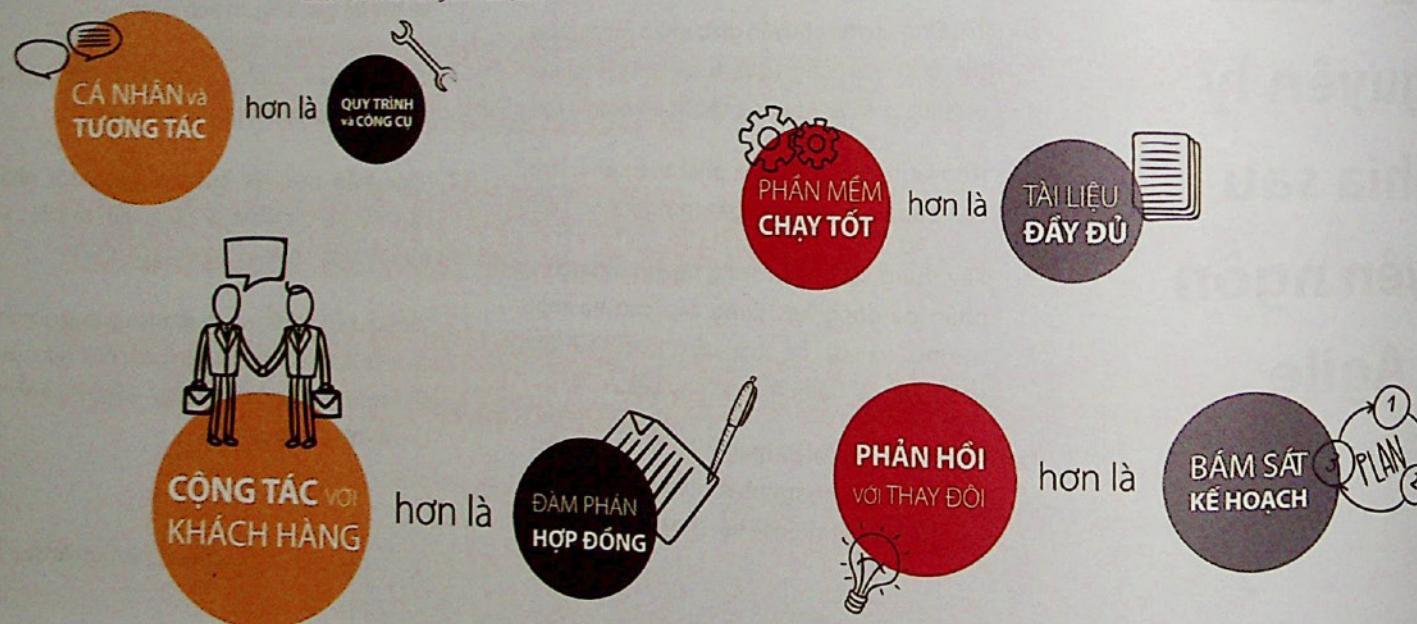
1. Ưu tiên cao nhất của chúng tôi là thỏa mãn khách hàng thông qua việc chuyển giao sớm và liên tục các phần mềm có giá trị.
2. Chào đón việc thay đổi yêu cầu, thậm chí rất muộn trong quá trình phát triển. Các quy trình linh hoạt tận dụng sự thay đổi cho các lợi thế cạnh tranh của khách hàng.
3. Thường xuyên chuyển giao phần mềm chạy tốt tới khách hàng, từ vài tuần đến vài tháng, ưu tiên cho các khoảng thời gian ngắn hơn.
4. Nhà kinh doanh và nhà phát triển phải làm việc cùng nhau hằng ngày trong suốt dự án.
5. Xây dựng các dự án 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.
6. Phương pháp hiệu quả nhất để truyền đạt thông tin tới nhóm phát triển và trong nội bộ nhóm phát triển là hội thoại trực tiếp.
7. Phần mềm chạy tốt là thước đo chính của tiến độ.
8. Các quy trình linh hoạt thúc đẩy phát triển bền vững. Các nhà tài trợ, nhà phát triển và người dùng có thể duy trì một nhịp độ liên tục không giới hạn.
9. Liên tục quan tâm đến các kỹ thuật và thiết kế tốt để gia tăng sự linh hoạt.
10. Sự đơn giản – nghệ thuật tối đa hóa lượng công việc chưa xong – là căn bản.
11. Các kiến trúc tốt nhất, yêu cầu tốt nhất và thiết kế tốt nhất sẽ được làm ra bởi các nhóm tự tổ chức.
12. Nhóm phát triển sẽ thường xuyên suy nghĩ về việc làm sao để trở nên hiệu quả hơn, sau đó họ sẽ điều chỉnh và thay đổi các hành vi của mình cho phù hợp.

Theo AgileManifesto.org

# Ý NGHĨA CỦA TUYÊN NGÔN AGILE

theo Jeff Sutherland<sup>1</sup>

<sup>1</sup> Lược trích từ bài "Agile Principles and Values" của Jeff Sutherland đăng trên MSDN, 2013.



Những giá trị đó không chỉ là thứ mà các tác giả của Tuyên ngôn Agile dự định cung cấp ra để phục vụ cho tuyên ngôn và rồi sau đó đi vào quên lãng. Chúng là các giá trị căn cứ vào để làm việc. Bản thân mỗi phương pháp linh hoạt đều tiếp cận các giá trị trên theo các cách thức khác nhau, nhưng tất cả các phương pháp này đều có tiến trình và biện pháp thực hành cụ thể để nuôi dưỡng một hoặc nhiều trong số các giá trị đó.

## Cá nhân và tương tác

Cá nhân và sự tương tác giữa họ là cốt yếu để nhóm đạt được hiệu suất cao. Các nghiên cứu về sự "bão hòa giao tiếp" (communication saturation) trong dự án cho thấy rằng, khi không có vấn đề trong giao tiếp, nhóm có thể thực hiện công việc tốt hơn 50 lần so với mức trung bình trong lĩnh vực của mình. Để tạo điều kiện cho giao tiếp, các phương pháp linh hoạt thường xuyên sử dụng chu trình thanh-trả-và-thích-nghi. Chu trình này có thể diễn ra hàng phút với lập trình cặp (pair-programming), hàng giờ với tích hợp liên tục (continuous integration), hàng ngày với họp đứng (standup meeting), hàng phân đoạn với các buổi họp sơ kết và cải tiến.

Tuy nhiên, chỉ tăng tần suất phản hồi và giao tiếp là không đủ để loại bỏ các vấn đề về giao tiếp. Chu kỳ thanh-trả-và-thích-nghi hoạt động tốt chỉ khi các thành viên trong nhóm thể hiện những hành vi quan trọng sau:

- Tôn trọng giá trị của mỗi cá nhân;
- Trung thực trong giao tiếp;
- Minh bạch về dữ liệu, hoạt động và quyết định;
- Tin tưởng vào sự hỗ trợ của mỗi cá nhân với nhóm;
- Cam kết với nhóm và các mục tiêu của nhóm.



hơn là



Để thúc đẩy các hành vi này, nhà quản lý linh hoạt phải cung cấp một môi trường hỗ trợ, các nhà huấn luyện phải tạo điều kiện thuận lợi và các thành viên của nhóm phải thể hiện chúng. Chỉ khi đó nhóm mới có thể phát huy được hết tiềm năng của mình.

Đạt tới các hành vi đó khó khăn hơn rất nhiều so với việc hình thành chúng. Hầu hết các nhóm tránh né sự thật, sự minh bạch và tin tưởng vào các chuẩn mực văn hóa hoặc có những kinh nghiệm tiêu cực từ các xung đột xuất phát từ truyền thống trước đây. Để chống lại những khuynh hướng này, ban lãnh đạo và thành viên nhóm phải tạo điều kiện cho những xung đột tích cực. Làm như vậy không chỉ giúp tạo ra hành vi sinh lợi mà còn đem lại các lợi ích khác:

- Cải tiến quy trình phụ thuộc vào nhóm trong việc tạo ra danh sách các trở ngại hoặc vấn đề trong tổ chức, đổi mới với chúng một cách trung thực và sau đó loại bỏ chúng một cách có hệ thống tùy theo độ ưu tiên.
- Đổi mới chỉ xảy ra với việc tự do trao đổi các ý tưởng mâu thuẫn nhau, như Takeuchi<sup>1</sup> và Nonaka đã từng nghiên cứu và chỉ ra.
- Việc hướng các nhóm tới mục tiêu chung đòi hỏi nhóm phải đổi mới và giải quyết các vấn đề về xung đột.
- Cam kết làm việc cùng nhau sẽ xảy ra chỉ khi mọi người đồng ý với mục tiêu chung và sau đó đấu tranh cho sự tiến bộ của bản thân cũng như của nhóm.

Ý cuối cùng ở trên nói về cam kết là đặc biệt quan trọng. Đó là khi mà các cá nhân và các nhóm được cam kết và cảm thấy có trách nhiệm với việc cung cấp các giá trị cao, đó là điểm mấu chốt đối với các nhóm phát triển phần mềm. Các phương pháp linh hoạt tạo điều kiện cho việc cam kết bằng cách khuyến khích nhóm đưa ra một danh sách công việc được ưu tiên hóa, từ đó họ tự quản lý công việc của mình, và tập trung vào cải tiến về cách thực hiện các công việc đó.

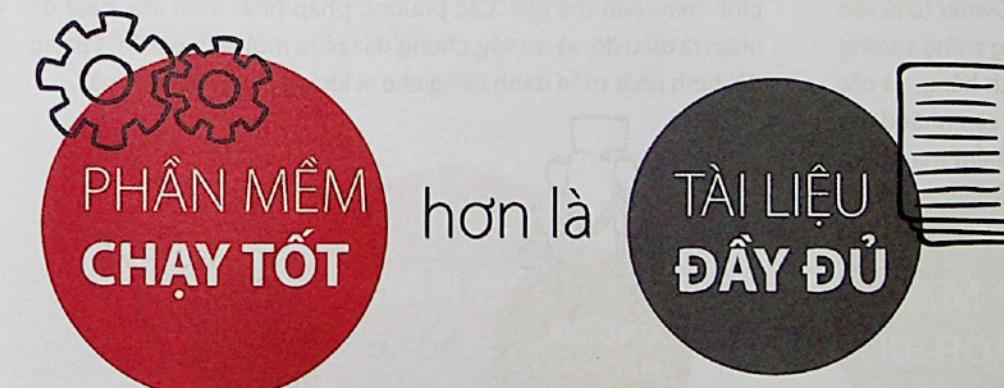
## Phần mềm chạy tốt

Phần mềm chạy tốt là một trong những khác biệt lớn mà phát triển linh hoạt mang lại. Tất cả các phương pháp linh hoạt thể hiện Tuyên ngôn Agile bằng cách nhấn mạnh việc cung cấp một phần nhỏ của phần mềm chạy tốt tới khách hàng sau một khoảng thời gian nhất định.

Tất cả các nhóm Agile phải định nghĩa “phần mềm chạy tốt” là như thế nào, Scrum gọi đó là “Định nghĩa Hoàn thành”. Nói chung, một phần sản phẩm được coi là hoàn thành chỉ khi các tính năng của chúng vượt qua tất cả các kiểm thử và có thể được vận hành bởi người dùng cuối. Ở mức thấp nhất, các nhóm phải vượt qua được kiểm thử đơn vị (unit test) và kiểm thử hệ thống. Các nhóm tốt nhất còn thực hiện việc kiểm thử tích hợp, kiểm thử hiệu năng

và kiểm thử chấp nhận của khách hàng trong định nghĩa hoàn thành đối với một phần chức năng. Thông qua nguồn dữ liệu phong phú từ các dự án, một công ty CMMI<sup>1</sup> cấp độ 5 cho thấy việc xác định kiểm thử chấp nhận cùng với các tính năng, triển khai một loạt các tính năng và theo độ ưu tiên, ngay lập tức chạy các kiểm thử chấp nhận với mỗi tính năng, và sửa bất cứ một lỗi nào có độ ưu tiên cao nhất sẽ tăng gấp đôi tốc độ sản xuất và giảm các sai sót đến 40%. Điều này có được từ một công ty có tỷ lệ sai sót thấp nhất thế giới.

Tuyên ngôn Agile khuyến nghị các nhóm cung cấp phần mềm chạy tốt sau một khoảng thời gian nhất định. Đồng thuận với Định nghĩa Hoàn thành là một trong những cách thực tế để nhóm Agile mang lại hiệu suất và chất lượng cao, cái cần thiết để hoàn thành mục tiêu này.



<sup>1</sup> Hirotaka Takeuchi: Giáo sư Đại học Harvard, Ikujiro Nonaka – Giáo sư tại Hitotsubashi University, Nhật Bản. Hai ông là đồng tác giả của bài báo đăng trên tạp chí Harvard Business Review năm 1986 “The New New Product Development Game”, được coi là có ảnh hưởng lớn tới sự ra đời của Scrum.

<sup>1</sup> CMMI: Viết tắt của (Capability Maturity Model Integration) được phát triển tại Viện CMMI (<https://cmmiinstitute.com>) thuộc Đại học Carnegie Mellon ở Pittsburgh. CMMI là mô hình nâng lực trưởng thành tích hợp cung cấp một định nghĩa về những hành động cần được doanh nghiệp xác định để nâng cao năng suất hoạt động. CMMI có 5 cấp độ, xác lập mức độ trưởng thành của doanh nghiệp. Mức 5 là mức cao nhất.

## Cộng tác với khách hàng

Trong hơn hai thập kỷ qua, tỷ lệ thành công của các dự án tăng hơn hai lần trên toàn thế giới. Điều này được cho là vì các dự án nhỏ hơn và mức độ chuyên giao thường xuyên đã cho phép khách hàng cung cấp các thông tin phản hồi về phần mềm hoạt động một cách đều đặn hơn. Các tác giả của bản Tuyên ngôn Agile đã làm sáng tỏ điều này khi họ nhấn mạnh rằng việc khách hàng tham gia vào quá trình phát triển phần mềm là hết sức cần thiết để dẫn tới thành công.

Các phương pháp phát triển linh hoạt đã thúc đẩy giá trị này bằng cách đưa vào một đồng minh tích cực của khách hàng làm việc sát cánh với đội phát triển. Ví dụ, một nhóm Scrum đầu tiên có hàng ngàn khách hàng. Sẽ là không khả thi nếu cho phép tất cả khách hàng tham gia vào quá trình phát triển sản phẩm, vì vậy họ chọn ra một vị đại sứ của khách hàng, được gọi là Product Owner (chủ sản phẩm), để đại diện cho không chỉ tất cả khách hàng trong trường hợp này mà còn bao gồm cả quản lý, dịch vụ khách hàng, và các bên liên quan khác. Product Owner có trách nhiệm cập nhật danh sách yêu cầu về sản phẩm sau mỗi bốn tuần thời điểm mà nhóm Scrum phát hành phiên bản sản phẩm chạy tốt, có tính đến yếu tố thực tế cùng phản hồi của khách hàng và các bên liên quan. Điều này cho phép khách hàng có thể định hình sản phẩm phần mềm đang được tạo ra.

Một nhóm XP<sup>1</sup> đầu tiên đã bắt đầu với một dự án IT nội bộ. Họ có thể có sẵn người dùng cuối trong nhóm làm việc với họ hằng ngày. Sử dụng khoảng 10% thời gian, các tư vấn viên và nhóm nội bộ có thể tìm được một người dùng cuối có thể làm việc với nhóm từng ngày. 90% thời gian còn lại, họ phải cùi ra người đại diện cho khách hàng. Người này, được nhóm XP gọi là Customer (khách hàng), làm việc trực tiếp với người dùng cuối để cung cấp một danh sách các tính năng rõ ràng cùng độ ưu tiên cho phép đội phát triển có thể thực hiện.

Cộng tác với khách hàng (hoặc đại diện của khách hàng) hằng ngày là một trong những lý do lý giải tại sao các dữ liệu trong ngành công nghiệp cho thấy rằng các dự án linh hoạt có tỷ lệ thành công cao gấp đôi so với các dự án truyền thống tính trung bình trên toàn thế giới. Các phương pháp phát triển linh hoạt đã nhận ra điều đó, và do vậy, chúng đã tạo ra một vị trí đặc biệt trong đội hình phát triển dành riêng cho vị khách hàng đại diện này.



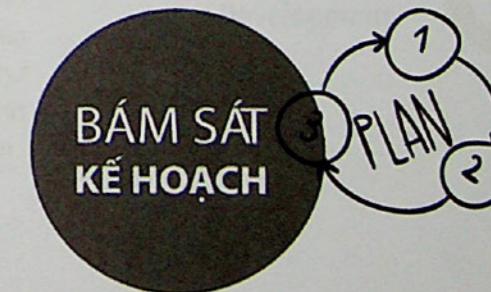
<sup>1</sup> eXtreme Programming – một trong các phương pháp Agile phổ biến nhất.

## Phản hồi với thay đổi

Phản hồi với thay đổi là điều cần thiết cho việc tạo ra một sản phẩm làm hài lòng khách hàng cũng như mang lại những giá trị kinh doanh. Dữ liệu ngành công nghiệp cho thấy hơn 60% các yêu cầu về sản phẩm hay dự án bị thay đổi suốt quá trình phát triển phần mềm. Ngay cả khi các dự án truyền thống kết thúc đúng thời gian, trong giới hạn kinh phí, với tất cả các tính năng theo kế hoạch, nhưng khách hàng thường không hài lòng vì những gì họ thấy không thật sự đúng như những gì họ muốn. *Luật Humphrey* nói rằng *khách hàng không bao giờ biết những gì họ muốn cho đến khi họ thấy phần mềm hoạt động*. Nếu khách hàng không nhìn thấy phần mềm hoạt động cho đến khi kết thúc dự án, sẽ là quá muộn cho việc kết hợp các thông tin phản hồi của họ ở thời điểm này. Các phương pháp phát triển linh hoạt tìm kiếm sự phản hồi của khách hàng trong suốt dự án để có thể kết hợp thông tin phản hồi và thông tin mới ngay khi sản phẩm đang được phát triển.

Tất cả các phương pháp phát triển linh hoạt đều được tích hợp sẵn những tiến trình thay đổi các kế hoạch trong một khoảng thời gian đều đặn dựa trên những thông tin phản hồi từ phía khách hàng cũng như bên đại diện của khách hàng. Các kế hoạch được thiết kế sao cho luôn cung cấp giá trị kinh doanh cao nhất trước hết. Bởi vì 80% giá trị nằm trong 20% các tính năng, một dự án phát triển linh hoạt chạy tốt có xu hướng kết thúc sớm, trong khi hầu hết các dự án truyền thống thường kết thúc trễ. Kết quả là, khách hàng thì vui vẻ hơn, và các nhà phát triển thì thích thú với công việc của họ hơn. Các phương pháp phát triển linh hoạt dựa trên những hiểu biết đó, để thành công hơn chúng phải có kế hoạch để thay đổi. Đó là lý do tại sao chúng thiết lập các quy trình, chẳng hạn như Sơ kết và Cải tiến, được thiết kế đặc biệt để thay đổi các ưu tiên thường xuyên dựa trên thông tin phản hồi của khách hàng và giá trị kinh doanh.

hơn là



## Dùng và Nghĩ

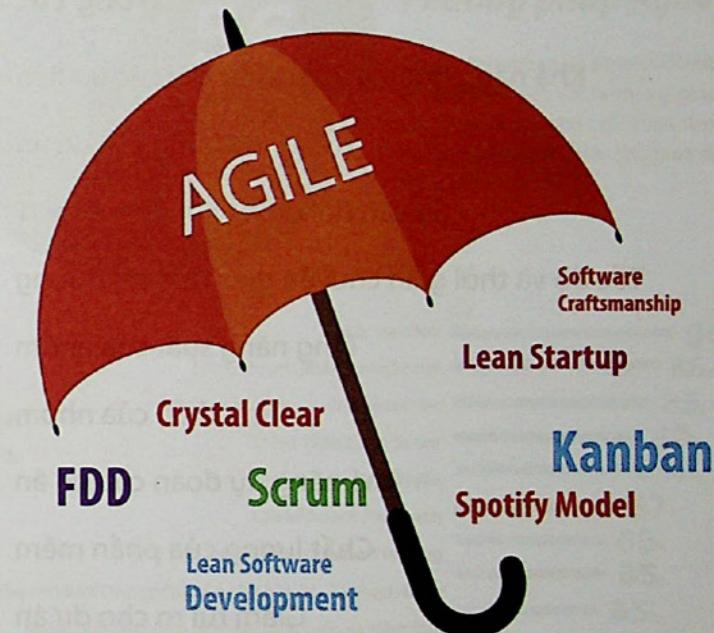


Bạn có đồng tình với các điểm chính của Tuyên ngôn Agile không? Bạn có muốn bổ sung hoặc chỉnh sửa gì thêm không?

Nếu vận dụng vào lĩnh vực của bạn (không phải phần mềm) thì nó nên được viết lại như thế nào?

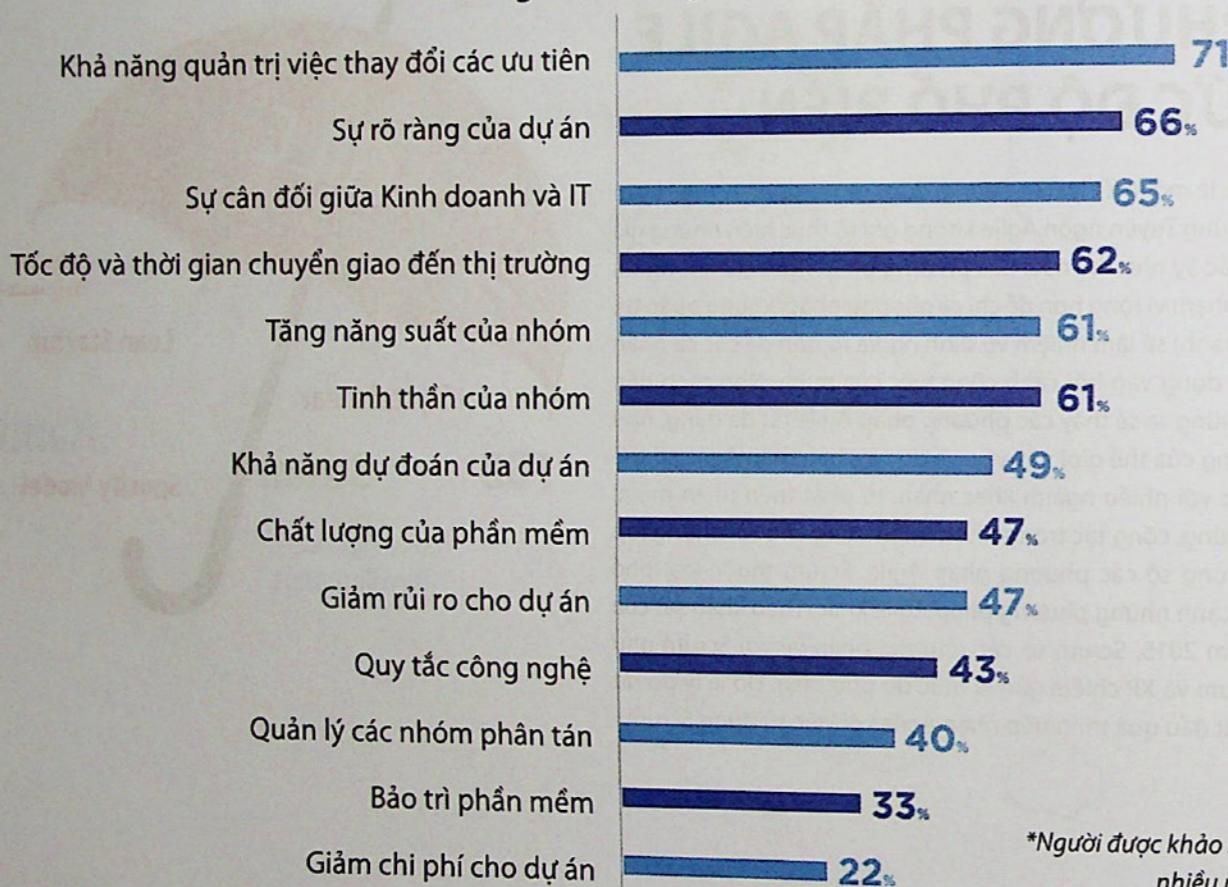
## CÁC PHƯƠNG PHÁP AGILE VÀ MỨC ĐỘ PHỔ BIẾN

Có thể coi Agile là một triết lý dẫn đường, được mô tả sơ bộ trong Tuyên ngôn Agile. Nhưng Tuyên ngôn Agile không ghi rõ thực hiện những giá trị và nguyên tắc ấy như thế nào. Các phương pháp Agile (từ chúng tôi gọi chung với phạm vi rộng hơn để chỉ cả phương pháp, khung quản trị, kỹ thuật thực hành) sẽ làm nhiệm vụ định nghĩa rõ hơn để các cá nhân và tập thể vận dụng vào bối cảnh công việc của mình. Nhờ cách tiếp cận như vậy, chúng ta sẽ thấy các phương pháp Agile rất đa dạng, như chính sự đa dạng của thế giới này vậy. Và do vậy, bản thân Agile có thể thích ứng được với nhiều ngành khác nhau, từ phát triển phần mềm, thiết kế phần cứng, cộng tác trong nhóm marketing cho tới những lớp học tiến bộ. Trong số các phương pháp Agile, Scrum thuộc loại phổ biến nhất bên cạnh những phương pháp Agile khác. Theo khảo sát của VersionOne năm 2015, Scrum và các phương pháp lai với Scrum như Scrumban, Scrum và XP chiếm gần  $\frac{3}{4}$  mức độ phổ biến. Đó là lý do rất nhiều nhóm bắt đầu quá trình tiếp nhận Agile với việc sử dụng Scrum.

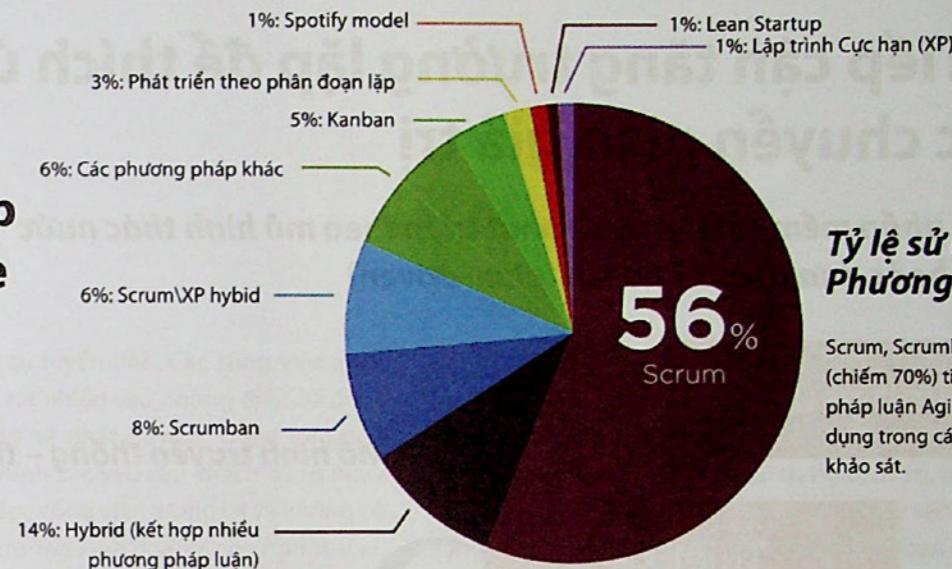


## Các lợi ích khi áp dụng Agile

*Khảo sát những người triển khai Agile cho thấy có sự cải thiện trong các lĩnh vực sau:*



## Các phương pháp và kỹ thuật Agile



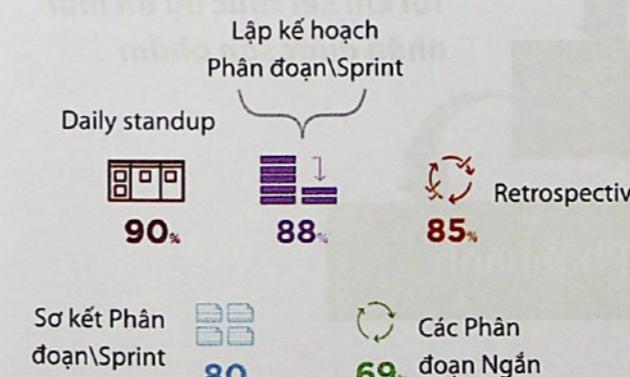
## Tỷ lệ sử dụng các Phương pháp Agile

Scrum, Scrumban và Scrum\XP hybrid (chiếm 70%) tiếp tục là những phương pháp luận Agile phổ biến nhất được sử dụng trong các tổ chức có người được khảo sát.



## Tỷ lệ sử dụng các Kỹ thuật Agile

Từ năm 2016-2017, tỷ lệ sử dụng Kanban đã tăng từ 50% lên 65%; product roadmapping tăng từ 38% lên 46% và portfolio planning từ 25% lên 35%.



\*Người được khảo sát có thể chọn nhiều mục khác nhau

# Agile: Tiếp cận tăng trưởng lặp để thích ứng và liên tục chuyển giao giá trị

Trước Agile, phần mềm cơ bản được phát triển theo mô hình thác nước (Waterfall), hoặc dựa theo kế hoạch (plan-driven)



Đặc trưng của lối tiếp cận này là sự tuyến tính. Các công việc được làm một cách tuần tự phụ thuộc rất nhiều vào những điều đã được tiên lượng trước đó. Mô hình này sẽ phát huy tác dụng nếu như chúng ta có thể xác định được chính xác yêu cầu khách hàng ngay từ đầu, có khả năng tiên lượng được công việc tương lai và không có thay đổi nào về công nghệ, môi trường kinh doanh, con người, v.v..

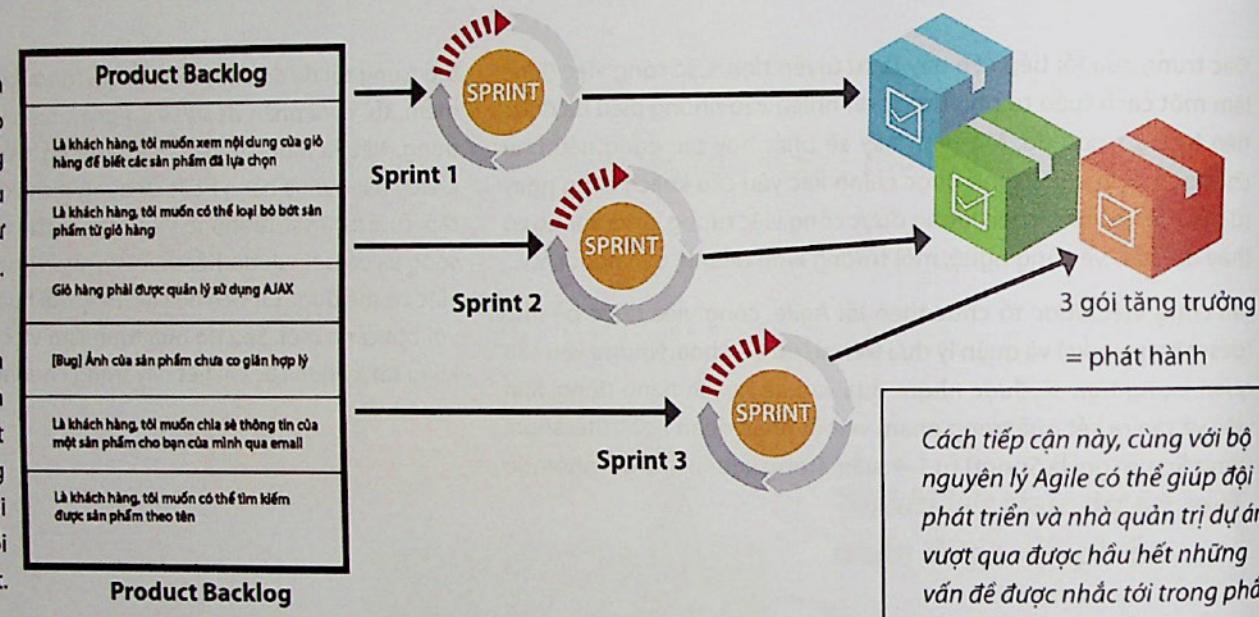
Khi công việc được tổ chức theo lối Agile, công việc được bẻ nhỏ (descaling works) và quản lý dựa trên sự ưu tiên hóa. Những yêu cầu quan trọng hơn sẽ được nhóm đưa vào kế hoạch hành động, làm việc và tạo ra kết quả trong phạm vi một phân đoạn ngắn (Iteration, trong Scrum gọi là Sprint) từ 1-4 tuần. Trong khuôn khổ đó, nhóm sẽ

tập trung tối đa để chuyển giao giá trị tới khách hàng. Nếu là phần mềm, thì cuối phân đoạn sẽ có phần mềm để cho người dùng sử dụng. Nếu là một chiến dịch marketing, thì cuối phân đoạn đã có khách hàng mua hàng hoặc sử dụng dịch vụ. Nếu là một dự án học tập, cuối dự án sẽ là những kiến thức có thể vận dụng ngay vào cuộc sống và đo đếm được. Kết thúc các phân đoạn đó, các yêu cầu công việc có thể được ưu tiên hóa lại, thay đổi hoặc cập nhật để phù hợp với bối cảnh mới. Sau đó quá trình làm việc lại lặp lại. Chúng ta sẽ khảo sát kỹ hơn các chi tiết này trong chương tiếp theo.

Nhóm Phát triển sẽ hoàn thành một tập những tính năng ở phần trên của Product Backlog trong một khoảng thời gian đóng khung (Sprint) tạo ra một phần tăng trưởng cho khách hàng.

Yêu cầu sản phẩm được sắp xếp theo độ ưu tiên. Những hạng mục đem lại nhiều giá trị trên vốn đầu tư (ROI) sẽ ở trên.

Product Backlog chứa các yêu cầu về sản phẩm được cập nhật liên tục để thích ứng với những phản hồi với những thay đổi cần thiết.



Theo Ken Schwaber, đồng tác giả của phương pháp Scrum, cách thức tổ chức công việc theo lối tăng trưởng (incremental) và lặp (iterative) này mang lại nhiều lợi ích rất đáng kể so với cách thức tổ chức công việc truyền thống dựa vào kế hoạch hóa chi tiết. Các lợi ích đó là: Gia tăng độ trực quan (về yêu cầu, vấn đề và tiến độ đạt được mục tiêu), Chuyển giao giá trị sớm và thường xuyên, Kiểm soát rủi ro và đặc biệt là khả năng thay đổi cao để thích ứng với biến động của môi trường (từ phía khách hàng, từ thị trường hoặc từ nhận thức mới của đội ngũ).

Khách hàng sẽ nhận được phần tăng trưởng cuối mỗi Sprint.

Sản phẩm sẽ tăng trưởng dần cùng với thời gian, tuy nhiên tại bất cứ thời điểm nào họ cũng có phần mềm chạy tốt.

## AGILE MINDSET

Ngày nay các chuyên gia thường sử dụng thuật ngữ Agile Mindset (Mô thức Agile) để mô tả thật ngắn gọn cái tinh thần chủ đạo của Agile. Không có một định nghĩa chung nào về Agile Mindset được đưa ra, nhưng chúng ta có thể tóm lại một vài điểm quan trọng theo đề xuất của chuyên gia quản trị Steve Denning, tác giả của cuốn "The Age of Agile" (Thời đại Agile). Một nhóm khi vận dụng các phương pháp hay kỹ thuật Agile cụ thể (ví dụ như Scrum, DSDM hoặc Kanban chẳng hạn), sẽ cần phải vận dụng các nguyên tắc dẫn đường sau để thực sự đạt được hiệu quả cao nhất của phương pháp.

Thứ nhất, mục tiêu của công việc (business theo nghĩa rộng, bao gồm những công việc hằng ngày chúng ta làm, những công việc kinh doanh hoặc những công tác xã hội) là **làm cho khách hàng sung sướng** (delight customers). Mọi hành động nên hướng tới khách hàng, bắt đầu từ khách hàng và xoay quanh khách hàng.

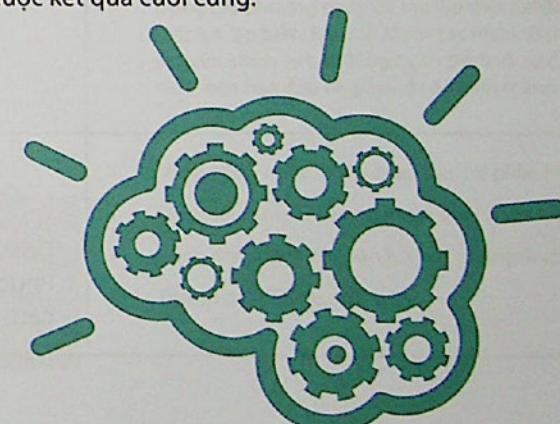
Thứ hai, phải **giảm nhỏ phạm vi công việc** (descaling works), trao quyền nhiều hơn, gia tăng tính tự trị cho các nhóm nhỏ và cá nhân.

Việc này giúp nhóm tập trung giải quyết vấn đề, gia tăng khả năng sáng tạo từ người trực tiếp tạo ra giá trị, cũng như thúc đẩy động lực làm việc mạnh mẽ.

Thứ ba, dùng **dữ liệu thực tiễn** (evidence-based – tức dựa trên bằng chứng) từ các **chu trình phản hồi ngắn** (short-feedback cycle) để ra quyết định và liên tục cải tiến.

Thứ tư, tạo dựng văn hóa nuôi dưỡng và phát triển. Để cá nhân trưởng thành và gia tăng năng lực, nhóm trở thành hiệu năng cao và tổ chức đạt được sự linh hoạt trong vận hành cũng như chiến lược. Một tổ chức Agile sẽ nuôi dưỡng mô thức phát triển (Growth Mindset), để mọi cá nhân luôn có cơ hội được học hỏi qua thực hành, qua các thử thách, qua những thất bại, để trưởng thành từng ngày.

Cuối cùng, luôn **duy trì khả năng thích nghi cao**, chào đón mọi thay đổi và chủ động tạo dựng cũng như dẫn dắt sự thay đổi để đạt được kết quả cuối cùng.



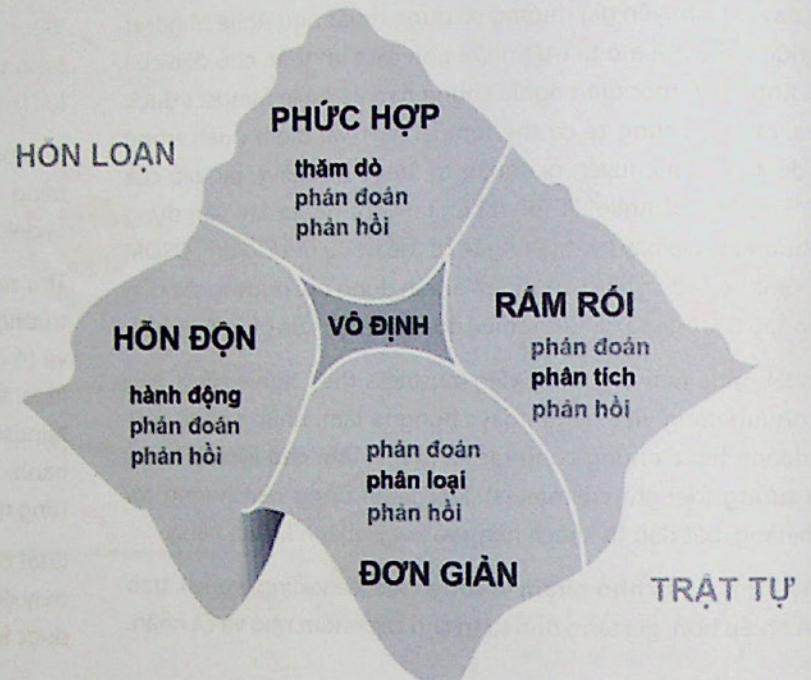
# KHI NÀO AGILE, KHI NÀO KHÔNG

Các dự án có quy mô, đặc thù và độ phức tạp không giống nhau, do vậy chúng ta cần phải có những phương pháp phát triển phù hợp.

Để xác định được khi nào thì ra quyết định kiểu gì, làm việc như thế nào, chúng ta có thể sử dụng Mô hình Cynefin cho việc phân loại tình huống và dự án, rồi ra các quyết định quan trọng.

Theo đó có các vùng chính mô tả các đặc trưng khác nhau bao gồm Hiển nhiên, Rầm rối, Phức hợp, Hỗn độn.

Bảng sau đây liệt kê các đặc tính từng vùng và vai trò của người ra quyết định.



Đối với các dự án phát triển sản phẩm mới, thông thường chúng rơi vào vùng Phức hợp. Đó là khu vực mà phương pháp truyền thống sẽ gặp khó khăn, và cách tiếp cận Agile sẽ phù hợp hơn.

Tình huống	Đặc điểm nhận dạng	Vai trò người ra quyết định
<b>Hiển nhiên</b>	<ul style="list-style-type: none"> <li>Có một câu trả lời đúng đắn</li> <li>Mọi người biết rõ những cái đã biết</li> <li>Quản lý theo thực tế (Fact-based management)</li> </ul>	<ul style="list-style-type: none"> <li>Phán đoán trước, phân loại sau, rồi phản hồi</li> <li>Sử dụng quy trình đúng đắn</li> <li>Ủy quyền</li> <li>Sử dụng kinh nghiệm tốt (best practice)</li> <li>Truyền đạt rõ ràng, trực tiếp</li> <li>Giao tiếp hai chiều có thể không cần thiết</li> </ul>
<b>Rầm rối</b>	<ul style="list-style-type: none"> <li>Cần chuyên gia phân tích</li> <li>Có thể nhận biết quan hệ nhân-quả nhưng không hiển nhiên đối với mọi người</li> <li>Có nhiều hơn một câu trả lời</li> <li>Mọi người biết là có những thứ chưa biết đến</li> <li>Quản lý theo thực tế</li> </ul>	<ul style="list-style-type: none"> <li>Phán đoán trước, phân tích sau, rồi phản hồi</li> <li>Thiết lập đội chuyên gia</li> <li>Lắng nghe các lời khuyên có phần khác biệt</li> </ul>
<b>Phức hợp</b>	<ul style="list-style-type: none"> <li>Thay đổi liên tục và không thể tiên lượng</li> <li>Không có câu trả lời đúng đắn; các khuôn mẫu hoạt động được ló dần ra</li> <li>Mọi người không biết những điều chưa biết</li> <li>Rất nhiều ý tưởng đối nghịch nhau</li> <li>Cần những tiếp cận sáng tạo</li> <li>Lãnh đạo dựa-trên-khuôn-mẫu (pattern-based leadership)</li> </ul>	<ul style="list-style-type: none"> <li>Thử nghiệm trước, phán đoán sau, rồi phản hồi</li> <li>Tạo môi trường thử nghiệm cho phép các khuôn mẫu xuất hiện</li> <li>Tăng mức độ tương tác và giao tiếp hai chiều</li> <li>Dùng các phương pháp để tạo lập thật nhiều ý tưởng: thảo luận mở; tạo lập biên; khuyến khích các nhân tố thu hút (attractor); khuyến khích sự bất đồng quan điểm và sự đa dạng; quản lý các điều kiện ban đầu và giám sát để hình thành các khuôn mẫu</li> </ul>
<b>Hỗn độn</b>	<ul style="list-style-type: none"> <li>Nhiều loạn cao độ</li> <li>Không rõ quan hệ nhân - quả, thậm chí không có điểm nào để tìm kiếm manh mối</li> <li>Không thể biết được điều gì</li> <li>Phải ra nhiều quyết định cấp tập mà không có thời gian để suy nghĩ</li> <li>Lãnh đạo dựa-trên-khuôn-mẫu (pattern-based leadership)</li> </ul>	<ul style="list-style-type: none"> <li>Hành động trước, phán đoán sau, rồi phản hồi</li> <li>Quan sát xem cái nào hiệu quả thay vì tìm kiếm câu hỏi đúng đắn</li> <li>Hành động tấp lục để vãn hồi trật tự (mệnh lệnh và kiểm soát)</li> <li>Truyền đạt rõ ràng, trực tiếp</li> </ul>

► Theo Snowden, D. J., & Boone, M. E. (2007). A leader's framework for decision making. *Harvard Business Review*, 85(11), 68.

# NHỮNG CÂU HỎI THƯỜNG GẶP

## Agile là gì?

Phát triển Linh hoạt (Agile Development, gọi tắt là Agile) là một thuật ngữ có nguồn gốc từ Tuyên Ngôn Phát triển Phần mềm Linh hoạt (The Manifesto for Agile Software Development). Tuyên ngôn này được soạn thảo năm 2001 bởi một nhóm các tác giả, chuyên gia của các phương pháp phát triển phần mềm Scrum, Extreme Programming (XP), DSDM (Dynamic Systems Development Method), Crystal, FDD (Feature-Driven Development) và một vài nhà lãnh đạo khác.

Tuyên ngôn Agile đã tổng kết ra một số giá trị và nguyên tắc chung, phổ quát cho tất cả các phương pháp luận về linh hoạt đang tồn tại độc lập tại thời điểm đó.

Khi dùng chữ "agile" với tư cách là một tính từ, thì nó mang nghĩa là "linh hoạt". Nhưng khi ta dùng chữ Agile (A viết hoa), tức là chỉ triết lý và tất cả những phương pháp phát triển phần mềm, phát triển sản phẩm và quản lý dựa trên triết lý được mô tả trong Tuyên ngôn Agile.

Agile là triết lý đã làm thay đổi diện mạo nền công nghệ thế giới trong những thập kỷ đầu thế kỷ XXI. Các phương pháp Agile đã được thực tế chứng minh là công cụ mạnh của các cá nhân, tổ chức trong việc đổi mới, sáng tạo, làm nên những sản phẩm chất lượng cao, đột phá, cũng như gia tăng hiệu quả hoạt động và năng lực cạnh tranh, phát triển văn hóa tổ chức. Những nghiên cứu thực nghiệm và thống kê từ ngành công nghiệp phần mềm cho thấy Agile có thể giúp cho các doanh nghiệp gia tăng năng suất và hiệu quả lên gấp nhiều lần, nâng tỷ lệ thành công của dự án lên gấp đôi và gia tăng năng lực cạnh tranh bền vững cho các doanh nghiệp. Cùng với Tư duy Tinh gọn (Lean Thinking), Khởi nghiệp Tinh gọn (Lean Startup), và Tư duy Thiết kế (Design Thinking), Agile vẫn đang tiếp tục tạo nên những làn sóng đổi thay lớn trên thế giới.

## Scrum có phải là Agile không?

Scrum là một phương pháp Agile (phổ biến nhất) nhưng không phải là Agile. Agile định nghĩa các giá trị cốt lõi và nguyên tắc định hướng, còn Scrum là một phương pháp cụ thể chia sẻ các nguyên tắc đó.



Agile/Scrum là gì?

## Vậy Scrum là gì?

Tài liệu Hướng dẫn Scrum định nghĩa "Scrum là một khung làm việc để phát triển bền vững các sản phẩm phức tạp". Có thể hiểu đây là khung tổ chức công việc tổng quát hướng đến phát triển các sản phẩm phức tạp, chủ yếu là phần mềm. Ngày nay Scrum có thể được dùng như là nền tảng tổ chức các công việc khác nhau, từ quản trị dự án linh hoạt nói chung, đến phát triển sản phẩm, thực hiện các chiến dịch marketing, tổ chức dạy học, sản xuất ô tô module hóa hoặc những công việc cá nhân khác.

Dựa trên lý thuyết quản lý thực nghiệm (empirical process control), Scrum sử dụng cơ chế lặp (iterative) và tăng trưởng (incremental) để tối ưu hóa tính hiệu quả và kiểm soát rủi ro. Scrum rất đơn giản, dễ học và có khả năng ứng dụng rất rộng. Để có thể dùng Scrum, chúng ta cần hiểu rõ và vận dụng đúng các thành tố tạo nên Scrum bao gồm các giá trị cốt lõi (còn gọi là "ba chân", hay ba trụ cột của Scrum), các vai trò, các sự kiện, các tạo tác (artifact) và những quy tắc đặc thù của Scrum.

Hiện nay, định nghĩa Scrum là gì được ghi trong tài liệu Scrum Guide (Hướng dẫn Scrum) và được cập nhật thường xuyên bởi chính các tác giả tại <http://scrumguides.org>.

Toàn bộ cuốn sách này sẽ nói về Scrum. Hãy đọc tiếp các chương sau để hiểu rõ và làm chủ một phương pháp tổ chức công việc có thể mang đến cho bạn sự thay đổi đột biến về năng suất và khả năng sáng tạo.

## Tại sao chọn Agile?

Trong nền kinh tế sáng tạo nhưng cũng đầy biến động hiện nay, yêu cầu cấp thiết đối với các cá nhân hay tổ chức là phải đạt được năng lực linh hoạt (Agility) cao độ để thích ứng và dẫn đầu trong các cuộc cạnh tranh. Agile là nền tảng để đạt được Agility và duy trì lợi thế cạnh tranh bền vững.

Triết lý Agile cùng với các framework, phương pháp cũng như các công cụ đặc thù có thể giúp các cá nhân, đội nhóm và doanh nghiệp trở nên hiệu quả hơn, năng suất hơn và đổi mới - sáng tạo tốt hơn.

## Muốn triển khai Agile thì bắt đầu từ đâu?

Các cá nhân có thể bắt đầu với việc áp dụng những kỹ thuật cơ bản của Personal Kanban cùng với tư duy Agile để nâng cao năng suất cá nhân, làm việc hiệu quả, kiểm soát tốt stress.

Các lập trình viên có thể bắt tay với những kỹ thuật lập trình Agile cơ bản như TDD, Pair-Programming để nâng cao chất lượng phần mềm trước khi đi xa hơn với đầy đủ bộ kỹ năng để trở thành Nghệ nhân Phần mềm (Software Craftsman) đích thực.

Các nhóm có thể ứng dụng Scrum để tổ chức lại công việc, nâng cao hiệu quả giao tiếp và hiệu suất nhóm.

Các công ty có thể bắt đầu với việc tái tổ chức công việc và quy trình nghiệp vụ với Scrum, Lean để giảm thiểu lãng phí, nâng cao hiệu quả hoạt động, tối ưu hóa nguồn lực và gia tăng năng lực cạnh tranh bền vững.

## Triển khai Agile có tốn kém không?

**Có đòi hỏi điều kiện gì đặc biệt về trình độ không?**

Phụ thuộc vào nguồn lực chúng ta có, việc thực hành Agile có thể đơn giản là sắp xếp lại công việc mà không cần thêm chi phí. Đối với các chương trình chuyển đổi căn bản và toàn diện, sự đầu tư có thể sẽ phải tương xứng với mục tiêu do tổ chức đề ra.

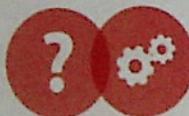
## Có bộ công cụ hay phần mềm nào để thực hành Agile?

Agile quy trình tư duy, trong khi phương pháp sẽ định hình cách triển khai cụ thể và các công cụ sẽ trợ giúp một phần nào đó. Công cụ và quy trình cần được lựa chọn kỹ để phục vụ các cá nhân làm việc hiệu quả và tương tác nhóm tốt hơn, không nên là lựa chọn để "đánh dấu" rằng "tôi đã sử dụng Agile".

Các nhóm Scrum nhỏ, làm việc tập trung có thể chỉ cần các đồ chi phí thấp để có thể vận hành tốt Scrum: Bảng trắng, giấy dán, bút viết bảng.

Chúng ta cũng có thể sử dụng hàng loạt các hệ thống phần mềm, cloud hỗ trợ tương tác nhóm theo triết lý Agile như Trello, Asana, JIRA, TFS. Chúng có thể miễn phí hoặc tính phí tùy nhu cầu sử dụng của mỗi nhóm và tổ chức.

Đối với các tổ chức lớn, việc đầu tư cho các công cụ để gia tăng hiệu quả tổng thể luôn là một trong những ưu tiên trong danh mục đầu tư cho công nghệ để phát triển.



## CÂU HỎI ỨNG DỤNG

1. Tuyên ngôn Agile giúp bạn nhận ra giá trị gì cho mình?
2. Có người diễn dịch ý thứ hai trong Tuyên ngôn Agile là “phần mềm chạy tốt thay vì tài liệu đầy đủ”, theo bạn diễn dịch như vậy có gì không đúng? Tại sao?
3. Nhiều người cho rằng Tuyên ngôn Agile có 5 điều, theo bạn điều thứ 5 của bản tuyên ngôn này là gì?
4. Có phải tất cả các nhóm nhỏ đều nên làm việc theo “kiểu Agile” không?
5. Nếu bạn làm Marketing hay Giáo dục, tuyên ngôn Agile có thể được viết lại như thế nào mà vẫn giữ nguyên tinh thần chủ đạo?
6. Dựa theo mô hình Cynefin, bạn có nhận diện được độ phức tạp của dự án mình đang làm không?
7. Tại sao cách thức phát triển lặp (iterative) và tăng trưởng (incremental) lại giúp các phương pháp Agile thích ứng với những thay đổi từ khách hàng, môi trường kinh doanh, công nghệ, v.v.?
8. Bạn sẽ nói gì với một bạn đồng nghiệp cho rằng làm Agile là không cần có kế hoạch?
9. Những lợi ích chính của việc giảm nhò phạm vi công việc và trao quyền cho các nhóm nhỏ thực thi là gì?
10. Hãy phân tích những yếu tố công cụ, quy trình nào đang cản trở động lực làm việc của cá nhân và sự cộng tác trong nhóm/tổ chức của bạn?

# 2

## CÁCH THỨC MỚI ĐỂ TỔ CHỨC MỌI VIỆC

Trong chương này...

- Định nghĩa Scrum
- Lịch sử Scrum
- Tổng quan khung làm việc Scrum
- Các ứng dụng của Scrum
- Tại sao dùng Scrum?
- Sprint
- Gói tăng trưởng

“

*Năng suất không phải là tất cả, nhưng về lâu dài thì nó hầu như là tất cả.*

*Paul Krugman - Nobel kinh tế 2008*

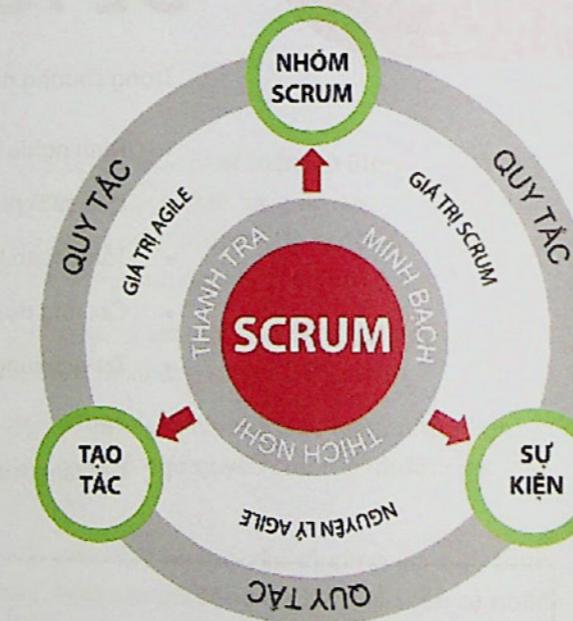
## SCRUM LÀ GÌ ?

Đôi khi được gọi là một phương pháp Agile, Scrum là khung làm việc linh hoạt được sử dụng phổ biến nhất đến nỗi nhiều người lầm tưởng Agile chính là Scrum hay Scrum chính là Agile.

Nhưng Scrum chỉ là một trong số hơn chục phương pháp cụ thể chia sẻ các giá trị được phát biểu trong Tuyên ngôn Agile.

Theo tài liệu Hướng dẫn Scrum (The Scrum Guide), Scrum là khung làm việc (framework) để phát triển, chuyển giao và duy trì các sản phẩm phức tạp theo cách thức lặp (iterative) và tăng trưởng (incremental). Quá trình phát triển được thực hiện thông qua các phân đoạn nối tiếp nhau. Khung làm việc Scrum định nghĩa rõ các giá trị cốt lõi, vai trò, sự kiện, tạo tác và các quy tắc để gắn kết tất cả thành một thể thống nhất giúp các nhóm làm việc đạt được hiệu quả cao. Ngày nay, Scrum còn có thể được quan niệm như là một khung quản trị hiện đại đa mục đích.

Một số người hiểu lầm Scrum là một bộ công cụ hay biện pháp thực hành nên cho rằng mình cứ có Sprint Backlog hay thực hiện Scrum Hàng ngày thì gọi là Scrum. Một hiểu lầm khác, cho rằng Scrum là phép thần để giải quyết tất cả các vấn đề của tổ chức hay dự án. Để sử dụng Scrum cần sự thay đổi sâu sắc trong cách nghĩ, cách làm chứ không dừng lại ở việc có một số công cụ hay thực hiện một số sự kiện. Chương này sẽ tìm hiểu toàn cảnh khung làm việc Scrum. Riêng Chương 6 sẽ thảo luận cách đưa Scrum vào tổ chức sao cho hiệu quả.



## Scrum – một cách diễn giải khác

Trong một bài viết năm 2011, Denning tóm tắt 10 đặc điểm của cách thức tổ chức công việc dựa theo Scrum:

1. Tổ chức công việc theo các chu trình ngắn (gọi là phân đoạn)
2. Khi nhóm làm việc trong các chu trình ngắn này, cấp quản lý không can thiệp (tức nhóm được trao quyền tối đa)
3. Nhóm báo cáo trực tiếp cho khách hàng, không phải cho nhà quản lý
4. Nhóm ước tính thời gian để hoàn thành công việc
5. Nhóm quyết định khối lượng công việc để làm trong phân đoạn
6. Nhóm quyết định cách hoàn thành công việc trong phân đoạn
7. Nhóm tự đánh giá hiệu suất của mình
8. Xác định mục đích của phân đoạn trước khi bắt đầu
9. Xác định mục tiêu thông qua các câu chuyện người dùng (user stories)
10. Loại bỏ các trở ngại cho công việc một cách có hệ thống

Theo: Steve Denning (2011). *Scrum is a major management discovery*, Forbes.

# Lịch sử Scrum

**1986: Takeuchi và Nonaka** là những người có ảnh hưởng rất lớn tới sự ra đời của Scrum với bài viết "The New New Product Development Game" trên Harvard Business Review.

**1993: Jeff Sutherland** lần đầu giới thiệu Scrum tại công ty Easel.

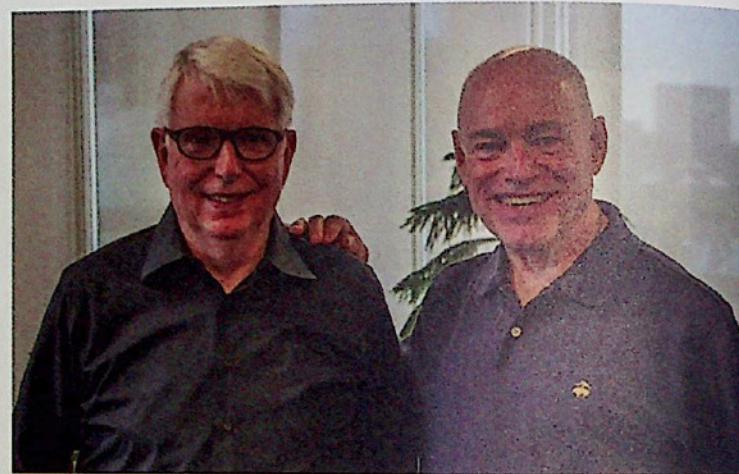
**1995: Ken Schwaber** cùng **Jeff Sutherland** thuyết trình về Scrum tại sự kiện OOPSLA.

**2001: Tuyên ngôn Phát triển Phần mềm Linh hoạt** và Liên minh Agile (Agile Alliance) ra đời.

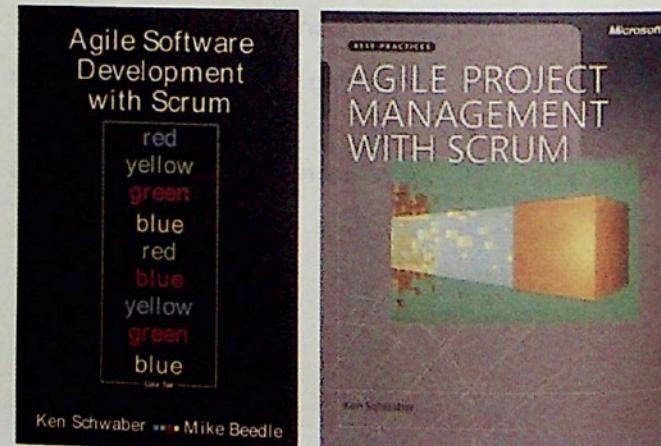
**2002: Liên minh Scrum** (Scrum Alliance) được thành lập.

**2010: Ken Schwaber** và **Jeff Sutherland** cùng xây dựng định nghĩa Scrum tại ScrumGuides.org với phiên bản đầu tiên.

**2017:** Phiên bản Scrum Guide 2017 được phát hành để phù hợp hơn với việc áp dụng Scrum rộng rãi ở những lĩnh vực khác ngoài phát triển phần mềm. Kể từ phiên bản đầu tiên thì Scrum Guide đã được cập nhật nhiều lần với các điều chỉnh nhỏ, qua các năm 2011, 2013, 2016 và 2017.



Jeff Sutherland(trái) và Ken Schwaber(phải)  
<anh Scrum.org>



Ken Schwaber ... Mike Beedle

## Scrum được dùng để làm gì?

Khởi đầu, Scrum được xây dựng để quản lý và phát triển sản phẩm. Kể từ đầu những năm 1990, Scrum đã được sử dụng rộng rãi cho các mục đích:

- Nghiên cứu tính khả dụng của thị trường, công nghệ và sản phẩm
  - Phát triển sản phẩm và cải tiến
  - Phát hành sản phẩm và cải tiến thường xuyên – có thể lặp lại nhiều lần trong ngày
  - Phát triển và duy trì Đám mây (trực tuyến, bảo mật và theo nhu cầu) và các môi trường vận hành cho các sản phẩm
  - Duy trì và làm mới các sản phẩm
- Nhiều báo cáo cho thấy Scrum được sử dụng rộng rãi, không chỉ trong lĩnh vực phát triển phần mềm, mà còn lan ra cả những dự án sản xuất phần cứng, các đội nhóm marketing hay giáo dục.
- Phát triển phần mềm thương mại
  - Phát triển ứng dụng nội bộ
  - Phát triển theo đơn đặt hàng
  - Các dự án mà giá đã được chốt
  - Các ứng dụng tài chính
  - Các ứng dụng tuân thủ chuẩn ISO 9001
  - Các hệ thống nhúng
  - Các hệ thống hoạt động 24x7 với yêu cầu 99,999% thời gian hoạt động
  - Hệ thống điều khiển máy bay Joint Strike Fighter
  - Phát triển video game
  - Phần mềm Điều khiển - Vệ tinh
  - Làm ra các Website
  - Phần mềm cho thiết bị cầm tay
  - Điện thoại di động
  - Các ứng dụng chuyển mạng
  - Các chiến dịch marketing



Một cuộc họp DailyScrum ở Microsoft,  
<anh Wikipedia>



Đội wikispeed sử dụng Scrum để cộng tác và sản xuất ô tô <anh: wikispeed>

- Các dự án học tập
- Quản lý các sự kiện công nghệ
- Sản xuất ô tô
- Quản lý lớp học hiện đại (eduScrum)
- Quản lý công việc cá nhân
- Và nhiều nữa

Tại sao Scrum lại có thể được ứng dụng rộng rãi và đa dạng như vậy? Lý do bởi vì nó chỉ định ra một cái khung (framework) gồm các thành tố cơ bản nhất để tổ chức công việc hiệu quả. Tùy từng tình huống, từng bối cảnh mà các chi tiết được thêm vào cho phù hợp. Do vậy tính tùy biến của Scrum là rất cao, thích hợp với các lĩnh vực hết sức đa dạng. Dưới đây là một số tình huống và lĩnh vực tiêu biểu mà Scrum đã được chứng minh là rất hiệu quả và được chấp nhận rộng rãi.



Một lớp học dùng eduScrum ở Hà Lan,  
<anh eduScrum.nl>

## Scrum trong phát triển sản phẩm

Công ty phần mềm Adobe sử dụng Scrum để phát triển những sản phẩm nổi tiếng cho giới làm đồ họa và nội dung số như Premier Pro, After Effects, Acrobat Connect<sup>1</sup>... Công ty Microsoft sử dụng Scrum để phát triển hầu hết các sản phẩm của mình Visual Studio, Skype, Xbox<sup>2</sup>... Trong khi đó Wikispeed sử dụng Scrum để phát triển xe ô tô<sup>3</sup>. Đó là những ví dụ điển hình về việc áp dụng Scrum thành công trong phát triển sản phẩm. Trong các trường hợp này, Scrum được sử dụng cho một hoặc nhiều nhóm để thực hiện các công việc như thiết lập và chia sẻ tầm nhìn của sản phẩm, tìm hiểu thị trường, lập kế hoạch, phân tích và thiết kế sản phẩm, sản xuất, đảm bảo chất lượng, phát triển khách hàng, điều chỉnh và cải tiến sản phẩm... với mục tiêu đáp ứng được yêu cầu kinh doanh và mang lại được lợi nhuận cho tổ chức hoặc nhà đầu tư.

## Scrum trong quản lý dự án

Openview Venture Partners – một công ty đầu tư mạo hiểm có trụ sở tại Boston, Hoa Kỳ - đã sử dụng Scrum để quản trị các dự án mình đầu tư vào nhằm nâng cao giá trị của các dự án đó<sup>4</sup>, Vodafone Thổ Nhĩ Kỳ sử dụng Scrum trong tất cả các loại dự án Loại-A (Nhiều hơn 50 man-day), Loại-B (ít hơn 50 man-day) và Loại-C (Thực hiện cấu hình theo yêu cầu) của mình<sup>5</sup> và đặc biệt là nhiều các công ty công phần mềm đã sử dụng Scrum cho các dự án của mình<sup>6</sup>, là những ví dụ về việc áp dụng Scrum cho quản lý dự án. Trong các trường hợp này, Scrum giúp lập kế hoạch tốt hơn, theo sát tiến độ của dự án, cải thiện giao tiếp với khách hàng, đảm bảo chất lượng và gia tăng độ hài lòng của khách hàng thông qua việc giảm thiểu rủi ro và cung cấp giá trị sớm cho khách hàng.

## Scrum cho đào tạo

Có hẳn một phiên bản Scrum được xây dựng dành riêng cho giáo dục, được gọi tên là eduScrum<sup>1</sup>, nhiều trường học áp dụng Scrum trong đào tạo các lớp dài hạn, nhiều đơn vị đào tạo áp dụng Scrum trong các khóa học ngắn hạn và cũng có nhiều tổ chức áp dụng Scrum cho các lớp đào tạo nội bộ... đây là thể hiện của việc rất nhiều tổ chức đã và đang thành công trong việc sử dụng Scrum trong các hoạt động đào tạo của mình. Trong các trường hợp này, Scrum rất hữu ích trong việc giúp lập kế hoạch học tập, nâng cao hiệu quả giao tiếp giữa học sinh - học sinh và học sinh - giáo viên, theo dõi tiến độ học tập, đánh giá năng lực của học sinh thường xuyên... và đặc biệt là nâng cao động lực học tập và tính cam kết của học sinh thông qua cơ chế trao quyền.

## Scrum cho quản lý công việc cá nhân

Scrum khởi đầu như là một khung làm việc dành cho các nhóm cộng tác, tuy nhiên đã có nhiều trường hợp áp dụng Scrum thành công cho quản lý công việc cá nhân. Nhiều thuật ngữ khác nhau được sử dụng để chỉ đến việc ứng dụng Scrum cho cá nhân, chẳng hạn như Solo Scrumming, Scrum for Individuals, Personal Scrum, Scrum for One,... Điều này cũng nói lên rằng đã có nhiều thử nghiệm và thực hành Scrum dành cho quản lý công việc cá nhân. Đối với những trường hợp này, Scrum giúp cho các cá nhân làm tốt việc lập kế hoạch trung và ngắn hạn, đặt mục tiêu tốt cho từng giai đoạn, theo dõi và điều chỉnh tiến độ công việc, thường xuyên cải tiến cách thức làm việc và các công cụ hỗ trợ để tăng năng suất và hiệu quả công việc.

1 [https://www.agilealliance.org/wp-content/uploads/2017/06/Adobe\\_HCSS44.pdf](https://www.agilealliance.org/wp-content/uploads/2017/06/Adobe_HCSS44.pdf)

2 <https://arstechnica.com/information-technology/2014/08/how-microsoft-dragged-its-development-practices-into-the-21st-century>

3 <https://www.forbes.com/sites/stevedenning/2012/05/10/wikispeed-how-a-100-mpg-car-was-developed-in-3-months/>

4 *Organizational Transformation with Scrum: How a Venture Capital Group Gets Twice as Much Done With Half the Work* – Jeff Sutherland

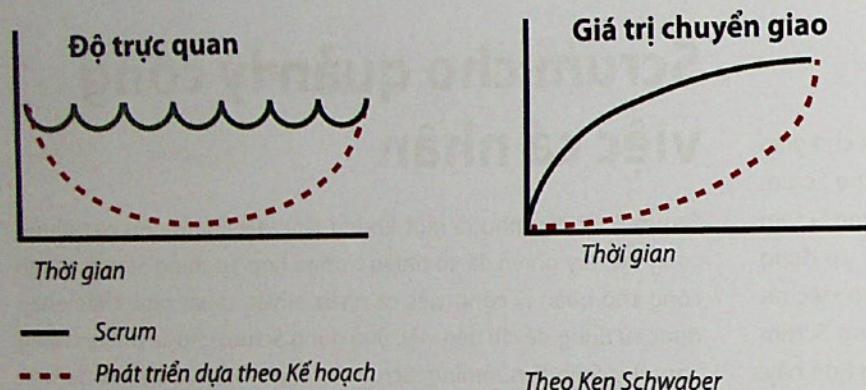
5 <http://www.scrumcasestudies.com/wp-content/uploads/2014/10/AgileTransformationInVodafoneTurkey.pdf>

6 <https://agilebreakfast.vn/chuyen-doi-agile-tai-mot-cong-ty-outsourcing/>

1 <http://eduscrum.nl/>

2 <https://medium.com/seriouslearning/eduscrum-goes-international-b768c6755d6a>

# LỢI ÍCH KHI DÙNG SCRUM



Với phương pháp truyền thống, phải đến giai đoạn cuối mới thấy được sản phẩm, còn trước đó không có sản phẩm thật. Từ khi bắt đầu phát triển tính trực quan ngày càng giảm. Với Scrum, ngay từ đầu chúng ta đã chuyển giao những phần tăng trưởng hoạt động tốt của sản phẩm và việc chuyển giao là liên tục và đều đặn, độ trực quan được duy trì ở mức cao.

Với phương pháp truyền thống, khách hàng phải đợi đến cuối mới nhận được giá trị từ sản phẩm bởi việc dành nhiều thời gian để lập kế hoạch và thi công dự án mà không đưa ra được một sản phẩm thực sự. Do đó giá trị họ nhận được ban đầu là rất thấp. Ngược lại, Scrum tập trung chuyển giao các tính năng hoàn chỉnh của sản phẩm ngay từ đầu. Do đó, ngay trong giai đoạn sản xuất, khách hàng đã nhận được các giá trị. Những giá trị này ngày càng được tích lũy và lớn dần lên theo tiến độ dự án.

Ở phương pháp quản lý truyền thống, chúng ta đặt cược vào độ chính xác của kế hoạch ban đầu mặc dù khi đó chúng ta có nhiều giả định về con người, công nghệ, môi trường kinh doanh không thay đổi hoặc chúng ta biết đủ những yếu tố này, do đó rủi ro là rất cao cho tới khi kết thúc.

Đối với Scrum, mức độ rủi ro giảm rất nhanh theo cấp số mũ. Chìa khóa ở đây chính là làm việc lặp và tăng trưởng cùng với sự thanh tra và thích nghi liên tục. Những yếu tố tiềm ẩn rủi ro liên quan đến sản phẩm, công nghệ con người, v.v., đều được phát hiện và điều chỉnh từ sớm.

Các phương pháp truyền thống không sẵn sàng cho việc thay đổi, bởi chúng ta phải chuẩn bị một kế hoạch chi tiết và cụ thể từ giai đoạn đầu, sau đó phải bám sát từng kế hoạch. Chi phí cho sự thay đổi là rất lớn, đặc biệt ở giai đoạn sau.

Ngược lại, Scrum luôn chủ động đón nhận thay đổi. Chúng ta không dành nhiều công sức để lập kế hoạch cho những giả định mà luôn chuẩn bị cho mọi sự phía trước, do đó chi phí cho thay đổi thấp hơn.

## Những lợi ích khác...

Mọi hoạt động đều định hướng giá trị (Value-Oriented) do đó Scrum giúp chúng ta tối ưu hóa giá trị trên vốn đầu tư (ROI).

**Định hướng khách hàng (Customer-Centric):** Tăng độ hài lòng của khách hàng bởi tất cả những gì chúng ta làm là để phục vụ khách hàng.

Giảm thiểu các **"nợ kỹ thuật"** bởi cách làm là luôn hoàn thành mọi thứ với chất lượng đảm bảo và linh hoạt cao để chấp nhận sự thay đổi. Bởi thế khi phát triển, các lập trình viên cần sử dụng các phương pháp để tăng khả năng linh hoạt như mã sạch, tái cấu trúc mã nguồn, kiểm thử tự động, tích hợp liên tục, v.v.

**Chất lượng sản phẩm cao.** Nhóm luôn bàn giao sản phẩm hoàn thành và thực hiện việc thanh tra, nên nếu có những vấn đề chất lượng thì nó sẽ được phát hiện và nhóm sẽ thích nghi để đưa mọi thứ trở về tiêu chuẩn.

Giảm thiểu rủi ro khi ứng dụng gấp vấn đề. Việc bàn giao sản phẩm sớm, khách hàng có thể xem và sử dụng sản phẩm, nên việc phát hiện các rủi ro sớm hơn. Thậm chí khi ứng dụng gấp vấn đề thì nhóm nhanh chóng có thể thích nghi.

**Tăng năng suất và hiệu quả lao động.** Scrum giúp nhóm liên tục phát hiện ra những tính năng đem lại giá trị trên vốn đầu tư cao, hoàn thành những tính năng có chất lượng và tốc độ cao hơn như trình bày ở Chương 1.

**Phát triển bền vững cá nhân và đội nhóm.** Việc nhóm được tự tổ chức giúp họ có cơ hội làm việc cùng với nhau, qua đó mỗi cá nhân trưởng thành hơn ở rất nhiều yếu tố như tính chủ động, cam kết, động lực, tri thức và kỹ năng phát triển sản phẩm. Tri thức trong nhóm được lan tỏa giữa các thành viên, do đó bản thân nhóm cũng phát triển.

## KHUNG LÀM VIỆC SCRUM

Khung làm việc Scrum gồm các vai trò của Nhóm Scrum, các Tạo tác (Artifact) được làm ra, các Sự kiện Scrum, cùng với các quy tắc ràng buộc tạo thành một thể thống nhất. Tất cả các thành phần này được xây dựng dựa trên một hệ giá trị nhất quán với các giá trị Agile và các giá trị cốt lõi của Scrum.

### Vai trò

Định ra trách nhiệm của từng thành viên trong nhóm Scrum tự tổ chức

### Tạo tác

Các công cụ hoặc sản phẩm do nhóm Scrum làm ra

- Product Owner
- ScrumMaster
- Nhóm Phát triển

- Gói tăng trưởng
- Product Backlog
- Sprint Backlog

### Sự kiện

Các cuộc họp và cơ chế cộng tác có cấu trúc để cộng tác hiệu quả hướng đến mục tiêu chung

- Sprint
- Lập kế hoạch Sprint
- Scrum Hằng ngày
- Sơ kết Sprint
- Cải tiến Sprint

**Ba trụ cột: Thanh tra – Minh bạch – Thích nghi**

**Năm giá trị: Dũng cảm, Tập trung, Cam kết, Cởi mở, Tôn trọng**

## SCRUM Một mô tả tổng quát

Giả sử bạn đang là thành viên của một nhóm Scrum. Sáng thứ Hai, bạn sẽ cùng nhóm làm việc ngồi với nhau, họp lập kế hoạch để quyết định cả đội sẽ làm gì trong tuần và làm như thế nào. Sau đó các công việc được chia nhỏ ra thành công việc cụ thể mà ai cũng hiểu được, dán lên bảng công việc chung. Hằng ngày từ thứ Ba đến thứ Năm, bạn sẽ tự lựa chọn việc làm phù hợp rồi bắt tay thực hiện. Vào đầu giờ sáng, trước khi làm việc, bạn cùng cả nhóm đứng trước bảng công việc chung, rà soát lại tiến độ của từng người, thông báo về một số khó khăn gặp phải trong khi làm việc. Cuộc họp đứng đó diễn ra hằng ngày, trong vòng 15 phút. Cuối mỗi ngày, bạn cố gắng hoàn thành thật tốt những việc đã lựa chọn. Vào chiều thứ Sáu, cả đội ngồi rà soát lại công việc xem kế hoạch đề ra từ đầu tuần có được hoàn tất không. Ngay sau đó, cả đội ngồi với nhau để đề ra cải tiến cách làm việc cho tuần sau sao cho hiệu quả và vui vẻ hơn. Tuần sau cả nhóm lập lại nhịp làm

việc như vậy, nhưng với lưu ý về một số cải tiến sẽ được mọi người tuân thủ.

Đó là hình dung đơn giản về nhịp làm việc của một nhóm Scrum nhỏ suốt một tuần. Trong mô tả trên, Nhóm Scrum đóng khung hoạt động của mình trong vòng 1 tuần, Scrum gọi khung thời gian này là **Sprint**. Buổi họp kế hoạch đầu tuần được gọi là **buổi Lập kế hoạch Sprint**. Buổi rà soát thành quả công việc vào thứ Sáu gọi là **hoạt động Sơ kết Sprint**. Hoạt động họp để tìm kiếm cách làm việc tốt hơn được gọi là **hợp Cải tiến Sprint**. Buổi họp đứng kéo dài 15 phút mỗi ngày được gọi là **Scrum Hằng ngày**. Bảng công việc chứa những việc cần làm cho một tuần đó được gọi là **Sprint Backlog**.

Trong Nhóm Scrum kia còn có các thành viên giữ các vai trò khác nhau: một **Product Owner** chịu trách nhiệm quản lý danh sách những yêu cầu công việc (được gọi là Product Backlog), một **ScrumMaster** chịu

trách nhiệm tổ chức cách thức làm việc theo Scrum cho cả nhóm và một **Nhóm Phát triển** gồm những người có đủ chuyên môn để hoàn thành mục tiêu mà cả nhóm đề ra.

Trên đây là cách chúng ta hình dung đơn giản hóa về một nhóm làm việc theo Scrum trong vòng một tuần. Ở mức độ tổng quan, chúng ta thấy nhịp làm việc này rất gần gũi với cách làm việc của những nhóm làm việc gắn kết và chuyên nghiệp.

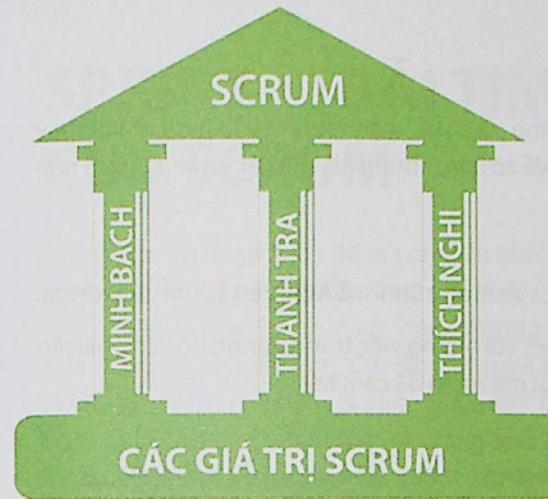
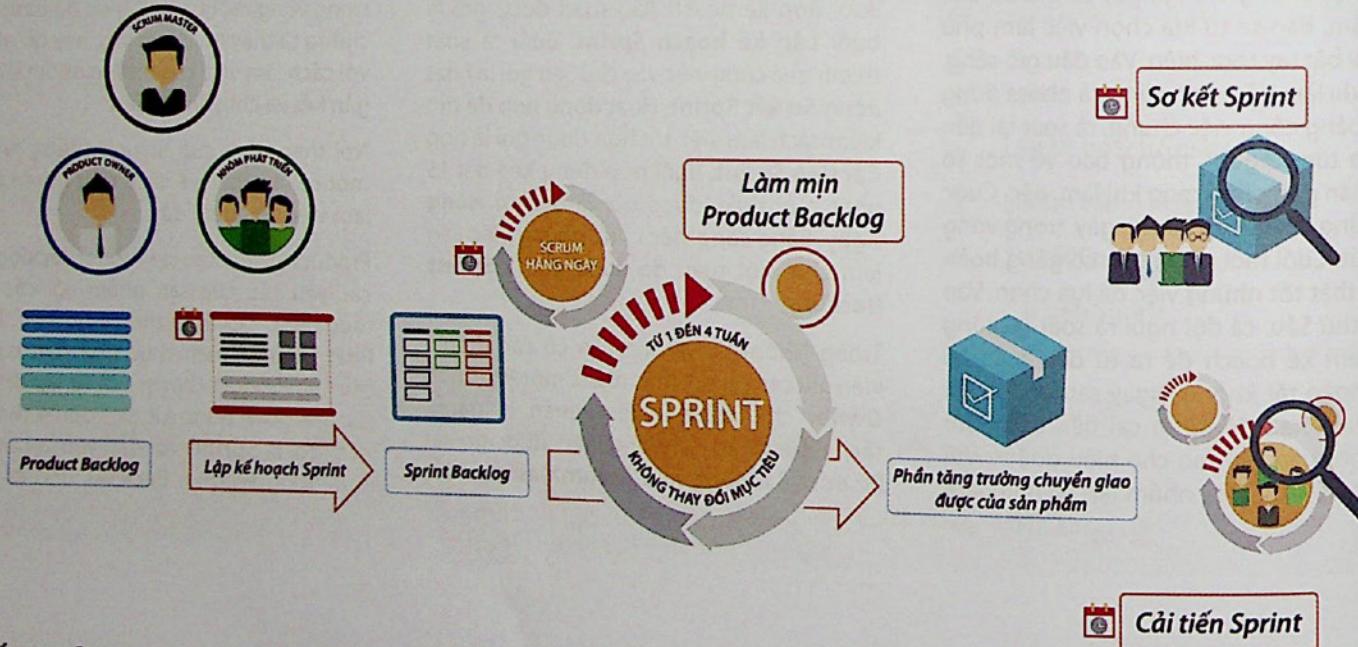
Nói theo cách của Scrum, chúng ta sẽ có một mô tả khác về cách một nhóm Scrum làm việc như dưới đây.

Product Owner tạo ra Product Backlog chứa các yêu cầu của sản phẩm với các hạng mục được sắp theo thứ tự ưu tiên. Nhóm Phát triển sẽ hiện thực hóa dần các yêu cầu của Product Owner với sự lặp đi lặp lại các giai đoạn nước rút từ 1 đến 4 tuần làm việc (gọi là **Sprint**) với đầu vào là các hạng mục trong Product Backlog, đầu ra là các

gói sản phẩm hoàn chỉnh có thể chuyển giao được (Potentially Shippable Product Increment). Trước khi cả nhóm cùng đưa ra quyết định có thể chuyển giao (shippable) cho khách hàng. Nhóm Phát triển cùng họp với Product Owner để lập kế hoạch cho từng Sprint. Kết quả của buổi lập kế hoạch (theo cách làm của Scrum) là **Sprint Backlog** chứa các công việc cần làm trong suốt một Sprint. Trong quá trình phát triển, nhóm sẽ cập nhật Sprint Backlog và họp hằng ngày (Daily Scrum) để chia sẻ tiến độ công việc, tái lập kế hoạch và trao đổi về các vướng mắc trong quá trình làm việc cùng nhau. Nhóm được trao quyền tự quản lý và tổ chức lấy công việc của mình

để hoàn thành công việc trong Sprint. Khi kết thúc Sprint, nhóm tạo ra gói sản phẩm có chức năng hoàn chỉnh, sẵn sàng chuyển giao (shippable) cho khách hàng. Buổi họp Sơ kết Sprint (Sprint Review) ở cuối Sprint sẽ giúp khách hàng thấy được nhóm đã có thể chuyển giao những gì, còn những gì phải làm hoặc còn gì phải thay đổi hay cải tiến. Sau khi kết thúc việc đánh giá Sprint, ScrumMaster và Nhóm Phát triển cùng tổ chức họp Cải tiến Sprint (Sprint Retrospective) để tìm kiếm các cải tiến trước khi Sprint tiếp theo bắt đầu. Điều này sẽ giúp nhóm liên tục học hỏi và trưởng thành qua từng Sprint.

Các Sprint sẽ được lặp đi lặp lại cho tới khi nào các hạng mục trong Product Backlog đều được hoàn tất hoặc khi Product Owner quyết định có thể dừng dự án căn cứ tình hình thực tế. Do sử dụng chiến thuật “cố giá trị hơn làm trước” nên các hạng mục mang lại nhiều giá trị hơn cho chủ sản phẩm luôn được hoàn tất trước. Vì vậy, Scrum luôn mang lại giá trị cao nhất cho người đầu tư cho sản phẩm. Do quy trình luôn luôn được cải tiến, nhóm Scrum thường có năng suất lao động rất cao. Đây là hai lợi ích to lớn mà Scrum mang lại cho tổ chức.



## BA TRỤ CỘT CỦA SCRUM

Scrum được xây dựng dựa trên lý thuyết quản lý quy trình thực nghiệm, hay còn được gọi là **Thuyết thực nghiệm** (empiricism). Lý thuyết này khẳng định rằng tri thức xuất phát từ kinh nghiệm thực tế và việc đưa ra các quyết định cần dựa trên những gì đã biết. Do đó, Scrum sử dụng cách tiếp cận lặp và tăng trưởng để có được sự chính xác cao hơn trong các tiên lượng và kiểm soát được rủi ro. Ba trụ cột chính được thể hiện xuyên suốt trong Scrum để đạt được điều này là: Minh bạch, thanh tra và thích nghi.

### Minh bạch (transparency)

Muốn thành công với Scrum, thông tin liên quan tới quá trình phát triển phải minh bạch và thông suốt. Các thông tin đó có thể là: tầm nhìn (vision) về sản phẩm, yêu cầu khách hàng, tiến độ công việc, các khía cạnh và rào cản v.v.. Từ đó mọi người ở các vai trò khác nhau có đủ thông tin cần thiết để tiến hành các quyết định có giá trị nhằm nâng cao hiệu quả công việc. Các công cụ và cuộc họp trong Scrum luôn đảm bảo thông tin được minh bạch cho các bên.

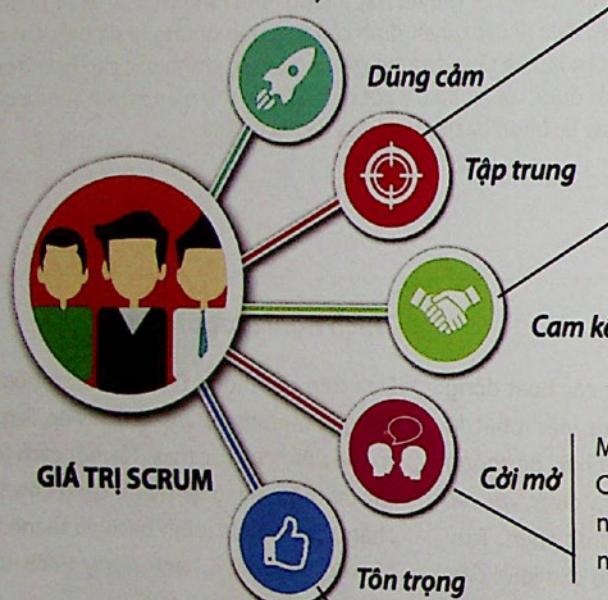
### Thanh tra (inspection)

Công tác thanh tra liên tục các hoạt động trong Scrum đảm bảo cho việc phát lộ các vấn đề cũng như giải pháp để thông tin đa dạng và hữu ích đến được với các bên tham gia quá trình phát triển. Truy xét kỹ càng và liên tục là cơ chế khởi đầu cho việc thích nghi và các cải tiến liên tục trong Scrum.

### Thích nghi (adaptation)

Dựa trên các thông tin minh bạch từ các quá trình thanh tra và làm việc, Scrum có thể phản hồi các thay đổi một cách tích cực, nhờ đó mang lại thành công cho sản phẩm. Các nỗ lực minh bạch và thanh tra đều để hướng tới hành động thích ứng nhanh chóng và hiệu quả.

## NĂM GIÁ TRỊ CỦA SCRUM



Để một người dám nói ra vấn đề của mình và chấp nhận rất nhiều loại rủi ro khi thay đổi, cam kết, họ cần là người dũng cảm. Về cơ bản, những giá trị khác không thể có nếu bạn không có sự dũng cảm.

Mọi người tập trung vào công việc trong Sprint & Mục tiêu Sprint của Nhóm. Khi Nhóm Phát triển đã cam kết với những việc trong Sprint, họ cần phải tập trung để hoàn thành những gì mà mình đã cam kết.

Mỗi thành viên cam kết với các thành viên khác về những điều mình làm và công việc đã được chọn ở buổi Lập kế hoạch Sprint.

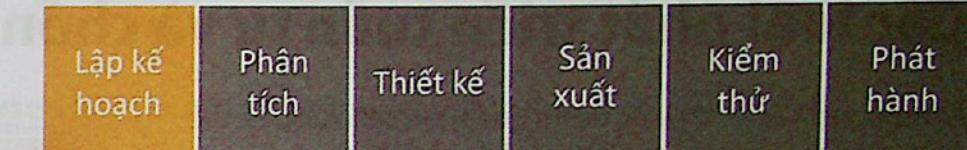
Ngoài ra, chúng ta liên tục cải tiến tức là thay đổi để trở thành một cá nhân tốt hơn, nhóm tốt hơn và tổ chức tốt hơn. Chúng ta luôn phải thay đổi để giữ vững lợi thế cạnh tranh và phục vụ khách hàng. Thực hiện thay đổi bao giờ cũng rất khó khăn, do đó chỉ với sự cam kết chúng ta mới có thể làm được.

Mọi thứ cần phải rõ ràng, minh bạch để mọi người có thể làm việc hiệu quả. Công việc phát triển sản phẩm ngày nay rất phức tạp, một người không thể nhìn và hiểu được hết tất cả mọi vấn đề. Do đó, nếu mọi người không cởi mở với nhau, thông tin bị che giấu rất nhiều và hiệu quả công việc khó có thể nâng cao.

Khi thiếu tôn trọng, mọi người khó thành thật trong chia sẻ. Ví dụ, khi một người không biết một điều gì đó và đi hỏi người khác. Người trả lời thay vì mong muốn giúp đỡ để người hỏi trở nên tốt hơn, độc lập lại phàn nàn, đánh giá người hỏi thì lần sau người hỏi sẽ khó mà cởi mở và nói sự thật được.

Không có tôn trọng, khó có sự cởi mở. Những công ty có văn hóa đổ lỗi khó có sự cởi mở.

## SPRINT - TRÁI TIM CỦA SCRUM



Cách phát triển truyền thống, các công việc được làm tuần tự, nên có rất nhiều rủi ro, lãng phí được phát hiện muộn.



Tất cả những công việc phải làm để hoàn thành một hoặc một nhóm tính năng & chức năng được đưa vào một Sprint.

Một Sprint không phải là một chu trình mini-waterfall truyền thống, mọi công việc được làm đồng thời với nhau.



Scrum hoạt động như thế nào

Thời gian

Cùng công việc nhưng được tổ chức khác và với ít yêu cầu hơn. Những lãng phí lớn sẽ bị phát lộ và có thể được xóa bỏ một cách hệ thống.

► Theo tài liệu thuyết trình "Scrum for Vietnam" của Ken Schwaber, 2012.

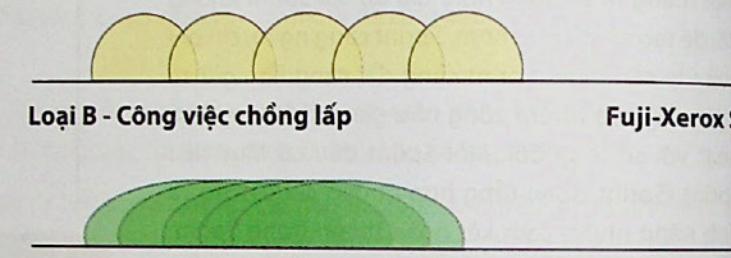
# Các loại phát triển tuần tự và chồng lấp

Khi sắp xếp các công việc trong quá trình sản xuất, có thể thực hiện theo một số hình thức khác nhau:

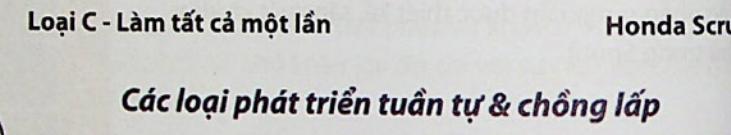
Loại A: Các giai đoạn tách biệt.



Loại B: Các giai đoạn có sự chồng lấp nhau, nhưng không hoàn toàn.



Loại C: Các giai đoạn hoàn toàn chồng lấp. Mọi việc đều được làm trong một khoảng thời gian ngắn.



## Các loại phát triển tuần tự & chồng lấp

Theo Nonaka & Takeuchi

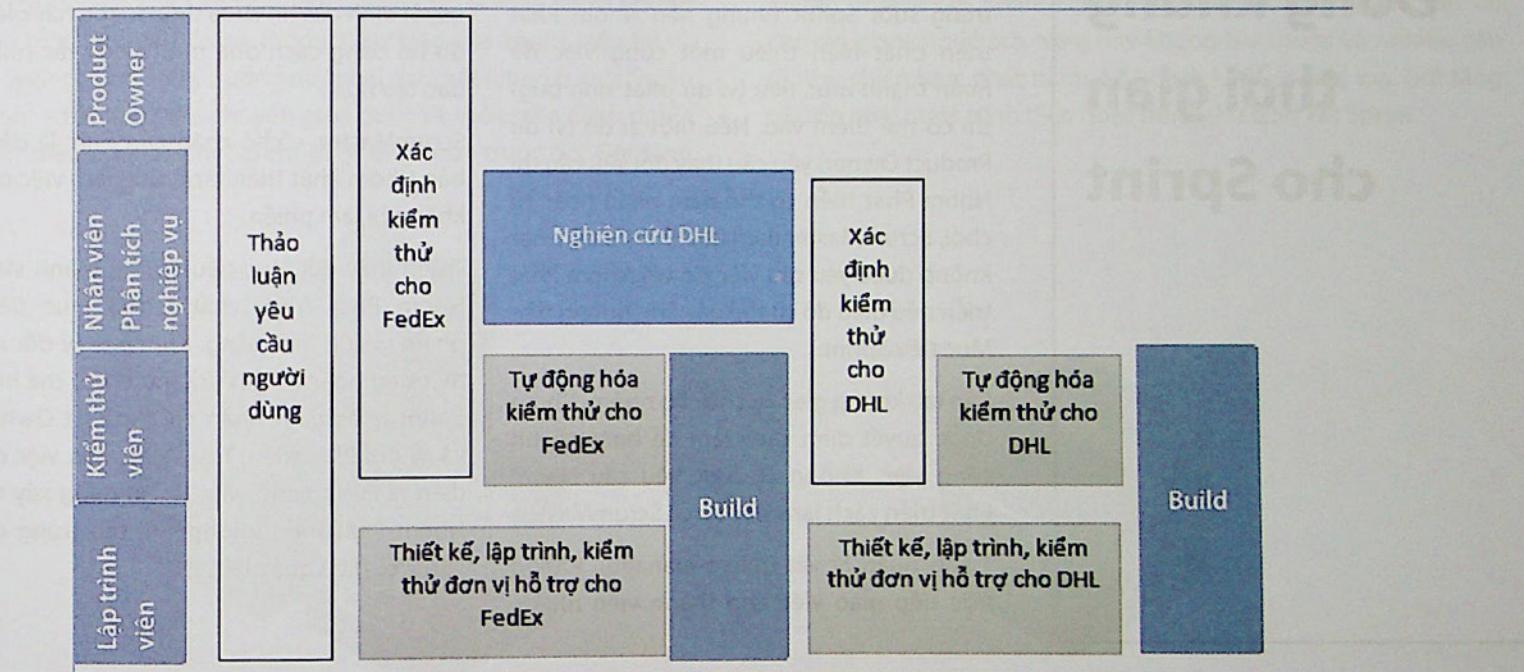
### Dùng và Nghĩ

Nhóm của bạn có thể làm việc theo kiểu chồng lấp loại C không? Thủ hình dung ai làm gì, như thế nào.

## Ví dụ về chồng lấp

Tất cả thành viên ở các vùng kỹ năng khác nhau đồng thời tham gia vào việc phát triển.

Ví dụ về phân công công việc trong một nhóm sản xuất phần mềm. Mọi công việc để hoàn thành sản phẩm từ phân tích yêu cầu, viết kiểm thử, thiết kế, lập trình, kiểm thử đơn vị, dựng sản phẩm được thực hiện đồng thời.



## Đóng khung thời gian cho Sprint

Không thay đổi yêu cầu sản phẩm trong suốt Sprint nhưng nếu Nhóm Phát triển phát hiện thiếu một công việc để hoàn thành mục tiêu (ví dụ phát sinh bug) thì có thể thêm vào. Nếu một ai đó (ví dụ Product Owner) yêu cầu thay đổi yêu cầu thì Nhóm Phát triển có thể đàm phán hoặc từ chối, ScrumMaster đảm bảo Product Owner không được yêu cầu việc đó với Nhóm Phát triển nếu điều đó có thể gây ảnh hưởng đến Mục tiêu Sprint.

Cần tạo không gian tự chủ cho nhóm. Nhóm được quyết định cách làm để hoàn thành công việc. Không ai được yêu cầu Nhóm Phát triển cách làm việc kể cả ScrumMaster.

Tránh quản lý vặt (micro-manage). Không trực tiếp giao việc cho thành viên nhóm.

Hãy để họ tự quyết định. Nếu khả năng quyết định của họ chưa thực sự tốt, hãy giúp đỡ họ bằng cách đưa ra các quy tắc hoặc đào tạo họ.

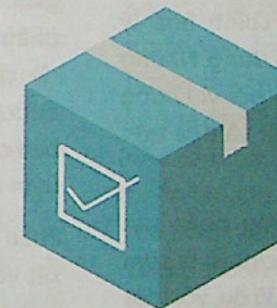
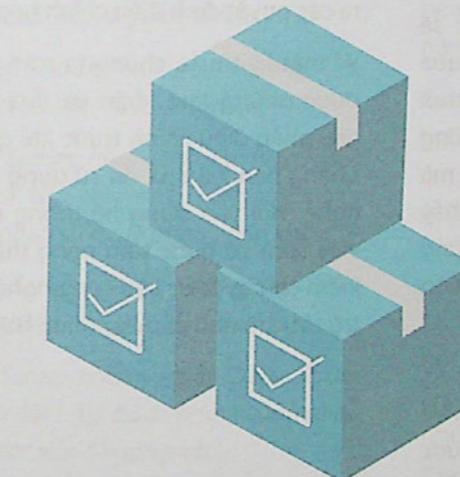
ScrumMaster – “chó chăn cừu”: tức là đảm bảo Nhóm Phát triển tập trung làm việc mà không bị làm phiền.

Tránh thay đổi Mục tiêu Sprint, thành viên Nhóm Phát triển, chất lượng mục tiêu, phạm vi của tính năng. Khi có thay đổi rất hệ trọng hoặc bất khả kháng thì có thể hủy Sprint hoặc đàm phán với Product Owner và Nhóm Phát triển. Tuy nhiên, nếu việc đó diễn ra liên tục thì vấn đề lớn đang xảy ra. Nhóm Phát triển không thể tập trung để làm việc hiệu quả nhất.

## Kết quả cuối mỗi Sprint: Gói tăng trưởng

Gói tăng trưởng (Increment) là cách gọi tắt của thuật ngữ “gói tăng trưởng có khả năng chuyển giao được”, là tổng tất cả những hạng mục đã hoàn thành Product Backlog của Sprint hiện tại và các Sprint trước. Tăng trưởng này phải đạt chất lượng ở mức “hoàn thành” – tức là có thể chuyển giao được và thỏa mãn Định nghĩa Hoàn thành mà nhóm Scrum đã thống nhất trước đó. Gói tăng

trưởng này phải có khả năng sử dụng được bởi khách hàng chứ không phải là sản phẩm alpha hay bản mẫu. Tuy nhiên, việc có chuyển giao cho khách hàng hay không tùy thuộc vào nhiều yếu tố như chiến lược phát hành, kế hoạch kinh doanh, v.v.. Gói tăng trưởng phải được trình diễn hoặc dùng thử ở Sơ kết Sprint.



## Những lợi ích từ gói tăng trưởng

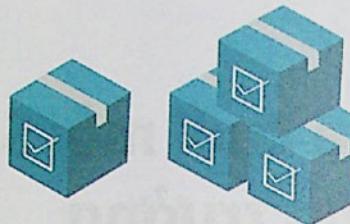
Việc có một phần tăng trưởng chuyển giao được sau một khoảng thời gian ngắn sẽ giúp nhanh chóng có được phản hồi từ người dùng hay khách hàng. Các phản hồi này sẽ được ghi nhận để đưa ra những điều chỉnh cho sản phẩm nếu cần thiết. Ví dụ, một ứng dụng dành cho các thiết bị di động, sau một khoảng thời gian ngắn với một lượng tính năng đủ dùng ban đầu thì chúng ta có thể phát hành tới một lượng người dùng nhất định. Sau đó, chúng ta sẽ lần lượt tích hợp thêm những tính năng khác hoặc có những điều chỉnh hợp lý dựa trên phản hồi của người dùng. Những phản hồi này thực sự giá trị bởi vì chúng xuất phát từ người dùng thật và dựa trên sản phẩm thật.

Việc chuyển giao được những phần sản phẩm nhanh chóng sẽ giúp chúng ta sớm thu được giá trị từ sản phẩm. Ví dụ, với một

ứng dụng dành cho thiết bị di động như đã nói ở trên, việc sớm mang đến tay người dùng sẽ giúp đẩy nhanh việc phát triển khách hàng lên song song với phát triển sản phẩm. Điều này có nghĩa là chúng ta cũng sớm thấy được giá trị của sản phẩm và giá trị này cũng đến tay người dùng sớm hơn.

Phần tăng trưởng chuyển giao được sẽ được gia tăng tính minh bạch trong quá trình sản xuất. Sau những khoảng thời gian ngắn, chúng ta đã có được những phản sản phẩm thực sự "hoàn thành" mà tất cả mọi người đều có thể nhìn thấy được, tương tác được, thay vì chỉ là những báo cáo, những thiết kế hay bản mẫu vốn tiềm ẩn nhiều rủi ro.

Phần tăng trưởng giúp giảm thiểu rủi ro, xét về cả mặt kinh doanh lẫn kỹ thuật. Về mặt kinh doanh, chúng ta sớm biết được



thực sự tính năng đó có mang lại giá trị hay không. Chẳng hạn, chúng ta muốn phát triển một phiên bản của hệ thống hiện tại dành riêng cho các thiết bị di động, nhưng không biết được phản ứng của thị trường. Do vậy, việc chúng ta nhanh chóng đưa ra được một phiên bản dùng được của sản phẩm sẽ giúp thăm dò thị trường và đưa ra các quyết định điều chỉnh hợp lý.

Về mặt kỹ thuật, chúng ta sớm phát hiện được những khó khăn và đưa ra những giải pháp điều chỉnh trước khi quá muộn. Chẳng hạn, bạn muốn sử dụng một công nghệ mới vào trong hệ thống của mình, việc sớm có được tính năng thật sẽ giúp kiểm chứng được liệu công nghệ đó có hỗ trợ tốt cho nhu cầu của mình hay không.

## ScrumBut: Cẩn thận với những điều chỉnh

ScrumBut là thuật ngữ để chỉ đến những tình huống áp dụng Scrum nhưng lại làm khác đi ở một số quy tắc hoặc hoạt động so với những khuyến nghị được đưa ra trong Hướng dẫn Scrum. ScrumBut là lý do chính dẫn đến việc các nhóm và tổ chức không đạt được lợi ích cao nhất khi sử dụng Scrum để giải quyết các vấn đề của mình, đặc biệt là đối với các đội nhóm và tổ chức mới bắt đầu áp dụng Scrum. Một số trường hợp ScrumBut thường xảy ra như là:

*"Tôi dùng Scrum nhưng không có ScrumMaster hoặc Product Owner."*

*"Tôi dùng Scrum nhưng Nhóm Phát triển không đảm bảo liên chức năng, vẫn phụ thuộc vào nhân sự ở các phòng ban khác."*

*"Tôi dùng Scrum nhưng không có Daily Scrum mà thay bằng Weekly Scrum."*

*"Tôi dùng Scrum nhưng phiên Daily Scrum thường kéo dài 1 giờ để nhóm cùng nhau giải quyết các vấn đề gấp phai."*

*"Tôi dùng Scrum nhưng cuối mỗi Sprint không nhất thiết phải bàn giao một Phần tăng trưởng của sản phẩm, thay vào đó có thể là một bản thiết kế hoàn chỉnh, một bản*

*mẫu hoặc phần quan trọng nhất của một chức năng nào đó."*

*"Tôi dùng Scrum nhưng độ dài của các Sprint thay đổi tùy theo khối lượng công việc và yêu cầu kinh doanh trong từng giai đoạn..." v.v..*

Những "điều chỉnh" như các trường hợp ở trên thoạt nhìn thì có vẻ không có vấn đề gì bởi vì nó giúp thích ứng với tình huống của các nhóm, nhưng thực chất thì chúng tiềm ẩn nguy cơ rất cao dẫn đến làm mất hẳn những giá trị mà Scrum mang lại.

Có nhiều lý do dẫn đến việc thực hiện ScrumBut, một trong số đó là do không am hiểu tường tận ý nghĩa của từng thành phần của Scrum, dẫn đến việc áp dụng một cách hình thức và không thực chất. Cần phải biết rằng, Scrum đã có một lịch sử tồn tại và tiến hóa dựa trên các lý thuyết khoa học và thực tiễn, tất cả các thành phần của Scrum đều được thiết kế để đảm bảo các nhóm và tổ chức đạt được giá trị cao nhất thông qua việc duy trì ba trụ cột là minh bạch, thanh tra và thích nghi. Do đó, việc tuân thủ các khuyến nghị của Scrum là cách làm an toàn nhất để được hưởng lợi

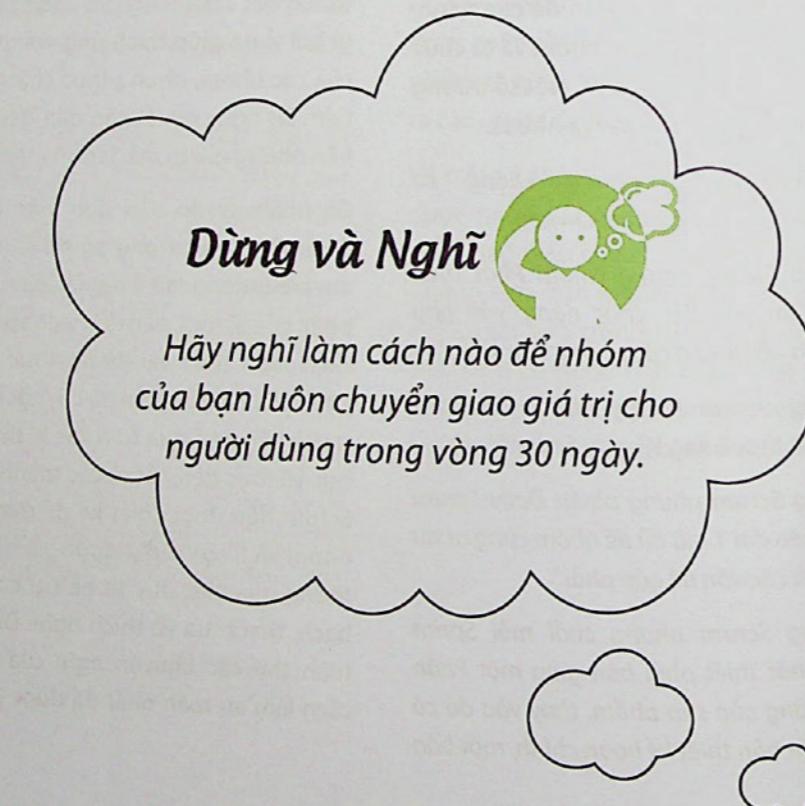
từ những giá trị mà Scrum mang lại. Những thay đổi đối với Scrum nếu không được suy xét cẩn thận và được thực hiện bởi những người am hiểu Scrum thì có thể dẫn đến việc phá hỏng hoàn toàn những gì mà các nhóm và tổ chức mong đợi.

Chẳng hạn, tính chất liên chức năng của Nhóm Phát triển là cách mà Scrum sử dụng để tăng tính chủ động, tăng khả năng tự tổ chức, tăng mức độ cam kết của nhóm, phát huy được năng lực của từng cá nhân nhằm đẩy nhanh năng suất và chất lượng sản phẩm. Trong trường hợp nhóm không đảm bảo đủ các kỹ năng cần thiết và phải phụ thuộc vào các phòng ban khác thì có thể nhóm sẽ không tự ra được các quyết định, không đảm bảo được tiến độ sản xuất cũng như cam kết của các thành viên. Tương tự, phần tăng trưởng ở cuối mỗi Sprint là cách mà Scrum sử dụng để gia tăng tính minh bạch, thanh tra và thích nghi một cách thực chất nhất. Trong trường hợp nhóm không bàn giao phần tăng trưởng mà chỉ bàn giao các bản thiết kế hoặc bản mẫu thì cả ba trụ cột của Scrum đều bị ảnh hưởng: Bản thiết kế tiềm ẩn rủi ro bởi vì sẽ có

những sai khác so với sản phẩm thật, những sai khác đó sẽ không được nhóm nhận ra và tồn tại nguy cơ gây hại về sau; Khi không có được sản phẩm thật thì Nhóm Phát triển, Product Owner và các bên liên quan không thể thực hiện tốt thao tác thanh tra để có các phản hồi xác đáng về sản phẩm; Từ đó, việc thích nghi trong giai đoạn tiếp theo sẽ không thể thực hiện được bởi vì nhóm không có đủ các thông tin cần thiết về sản phẩm hiện tại cũng như phản hồi của các

bên để điều chỉnh. Một thành phần khác đó là sự kiện Scrum Hàng ngày, tại sao Scrum lại quy định phiên trao đổi hàng ngày? Tại sao lại đóng khung trong 15 phút? Tại sao nhóm chỉ cập nhật và thảo luận nhanh chứ không tập trung giải quyết vấn đề? Tại sao không có sự tham gia của Product Owner? Trả lời được các câu hỏi đó là cách để nhóm hiểu ý nghĩa của Scrum Hàng ngày và thực hiện một cách hiệu quả sự kiện này.

Như vậy, có thể thấy rằng ScrumBut là một cái bẫy nguy hiểm, đặc biệt là đối với các nhóm và tổ chức chưa có am hiểu và thành thạo khi dùng Scrum. Cách an toàn nhất là “làm đúng sách” trước khi thực hiện các điều chỉnh. Và nếu muốn thực hiện các điều chỉnh thì hãy đảm bảo rằng nhóm đã hiểu được các tác động của việc điều chỉnh đó, cả những tác động tốt và những tác động xấu.



## 5 HIỂU LÂM VỀ SCRUM

### 1. Scrum chỉ áp dụng trong sản xuất các sản phẩm phần mềm.

Scrum được phổ biến trước tiên trong lĩnh vực phát triển phần mềm, nhưng ngày nay đã được ứng dụng rất thành công trong nhiều lĩnh vực khác như giáo dục, marketing, bán hàng, quản trị doanh nghiệp, các cơ quan chính phủ, v.v..

### 2. Việc tổ chức làm việc đồng thời (các giai đoạn chồng lấp) trong quá trình sản xuất là không thể đổi với một số loại sản phẩm.

Các tình huống thực tế đã chứng minh là có thể tái sắp xếp công việc và phân công nhiệm vụ trong rất nhiều tình huống thuộc các lĩnh vực khác nhau để các thành viên có thể tham gia sản xuất đồng thời. Do đó, cần loại bỏ quan niệm “không thể” này và bắt tay vào tìm cách để tổ chức lại công việc của nhóm.

### 3. Scrum là một phương pháp linh hoạt, do đó không kiểm soát được hoạt động của nhóm và của các thành viên dẫn đến việc chây ì không có cam kết hoàn thành công việc.

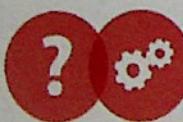
Để đạt được sự linh hoạt, Scrum yêu cầu tính kỷ luật rất cao ở tất cả các vai trò và hoạt động trong suốt quá trình sản xuất. Các tạo tác, sự kiện và quy tắc của Scrum đảm bảo tính kỷ luật được tuân thủ một cách triệt để. Nhóm và các thành viên hoạt động tự do trong khuôn khổ của các nguyên tắc và quy tắc đã được thống nhất.

### 4. Scrum không coi trọng việc lập kế hoạch bởi vì nó linh hoạt theo thời gian.

Lập kế hoạch là một việc được thực hiện rất cẩn thận và thường xuyên trong cả dài hạn và ngắn hạn. Product Owner luôn rà soát Product Backlog và cập nhật các điều chỉnh để sản phẩm có được giá trị cao nhất. Nhóm Scrum lập kế hoạch cho từng Sprint. Các thành viên Nhóm Phát triển tái lập kế hoạch hàng ngày để đảm bảo công việc phát triển bám sát với thực tế. Hoạt động lập kế hoạch quan trọng hơn so với một bản kế hoạch cố định.

### 5. Phần tăng trưởng ở cuối Sprint cần được chuyển giao ngay cho khách hàng.

Việc chuyển giao sản phẩm cho khách hàng phụ thuộc vào tình hình cụ thể, bao gồm những yếu tố như kế hoạch, vận hành, kinh doanh, nguồn lực, v.v.. Phần tăng trưởng ở cuối Sprint không nhất thiết phải được chuyển giao ngay, nhưng bắt buộc phải đạt được chất lượng của một sản phẩm đã sẵn sàng để sử dụng.



## CÂU HỎI ỨNG DỤNG

1. Hai trụ cột Thanh tra và Thích nghi của quá trình quản lý thực nghiệm bị ảnh hưởng như thế nào nếu trụ cột Minh bạch không đảm bảo?
2. Mô tả mối quan hệ giữa ba trụ cột của quá trình quản lý thực nghiệm.
3. Phân tích ưu và nhược điểm của việc Nhóm Phát triển tự xác định cách hoàn thành công việc trong phân đoạn thay vì một người thực hiện và truyền đạt lại.
4. Bạn có thể áp dụng Scrum để quản lý những công việc nào của mình?
5. Nhóm Scrum quản lý thay đổi như thế nào?
6. Mô tả lợi ích của bên liên quan (ví dụ: khách hàng, nhà phát triển, quản lý, v.v.) từ việc áp dụng Scrum?
7. Nếu áp dụng Scrum vào một dự án/công việc của bạn thì thành viên ở những chuyên môn khác nhau (kiểm thử, thiết kế, v.v.) sẽ tham gia theo trình tự như thế nào trong một Sprint?
8. Những lợi ích khi Nhóm Phát triển được làm việc trong một Sprint đóng khung?
9. Những khó khăn khi đóng khung một Sprint ở dự án/công việc của bạn? Cách vượt qua?

# 3

## NHÓM HIỆU NĂNG CAO

Trong chương này...

- Các giai đoạn phát triển của nhóm
- Nhóm liên chức năng và tự tổ chức
- Nhóm Scrum
- ScrumMaster
- Product Owner
- Nhóm Phát triển
- Đặc điểm của nhóm hiệu năng cao
- Team Canvas
- Các nghề nghiệp đặc thù Scrum



*Nếu tất cả mọi người cùng nhau tiến về phía trước, thì thành công sẽ tự nó đến.*

*Henry Ford*



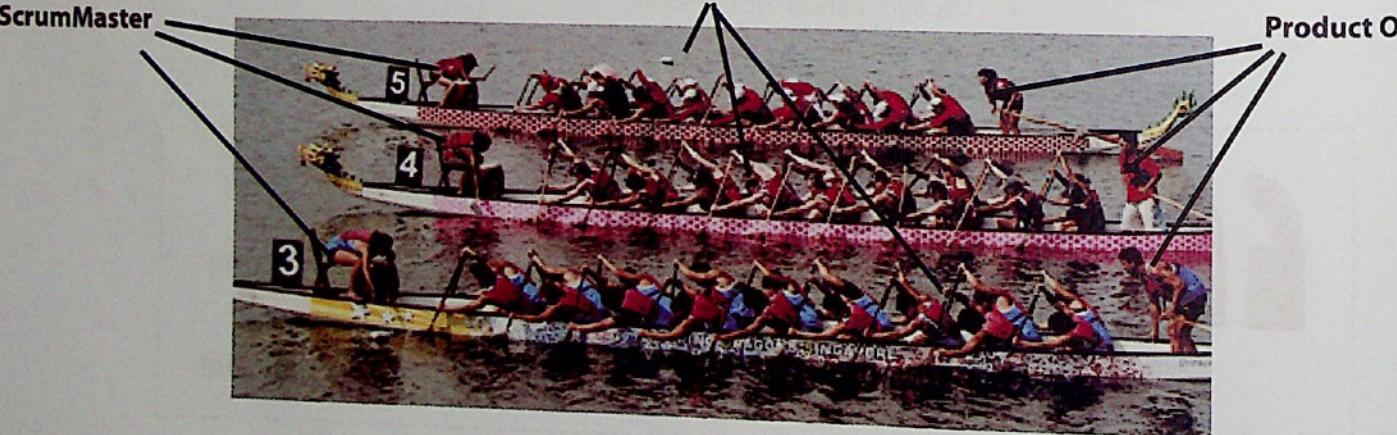
Tuyên ngôn Agile ghi rõ "Cá nhân và tương tác hơn là quy trình và công cụ". Bí quyết thành công của Scrum nằm ở các cá nhân và sự tương tác qua lại giữa các cá nhân trong nhóm Scrum.

Nhóm Scrum bao gồm các vai trò Product Owner, Nhóm Phát triển và ScrumMaster. Nhóm Scrum là nhóm tự tổ chức (self-organizing) và liên chức năng (cross-functional). Nhóm tự tổ chức tự mình chọn cách thức tốt nhất để hoàn thành công việc chứ không bị chỉ đạo bởi ai đó bên ngoài nhóm. Nhóm liên chức năng có đủ kỹ năng cần thiết để hoàn thành công việc mà không phụ thuộc vào bất kỳ người ngoài nào khác ngoài nhóm. Mô hình nhóm trong Scrum được thiết kế để tối ưu hóa sự linh hoạt, sáng tạo và năng suất. Nhóm Scrum chuyển giao sản phẩm theo phân đoạn lặp đi lặp lại và tăng dần, tối đa hóa cơ hội cho các phản hồi. Việc chuyển giao tăng dần (incremental) các gói sản phẩm đạt tiêu chuẩn "Hoàn thành" đảm bảo một phiên bản có thể sử dụng được của sản phẩm.

Nếu phải tìm một sự so sánh, chúng ta có thể so sánh cấu trúc của nhóm Scrum với nhóm đua thuyền rồng. Product Owner là người lái, định hướng con thuyền. Các tay chèo thuộc Nhóm Phát triển. ScrumMaster là người giúp giữ nhịp để nhóm đạt hiệu năng tốt nhất trong cả cuộc đua.

Nhóm Phát triển

Product Owner



## NHÓM TỰ TỔ CHỨC LIÊN CHỨC NĂNG

Nhóm Scrum có hai đặc điểm quan trọng là: tự tổ chức (self-organizing) và liên chức năng (cross-functional).

**Tự tổ chức** có nghĩa là nhóm cùng ra quyết định, cùng tổ chức công việc thông qua bàn bạc, phân công ngang hàng hướng đến mục tiêu chung. Trong nhóm tự tổ chức, không ai giao việc cho ai, mà cùng nhau chủ động lập kế hoạch, phân chia công việc hợp lý dựa trên các vai trò được định nghĩa rõ ràng từ trước. Các vai trò này được trao quyền để hoàn thành công việc hướng đến mục tiêu chung.

**Liên chức năng** là một cơ cấu nhóm không quá mỏm mẻ, và không phải lần đầu tiên xuất hiện trong Scrum. Một nhóm liên chức năng bao gồm nhiều cá nhân với các chuyên môn khác nhau đủ năng lực được kết hợp lại cùng làm việc hướng tới một mục tiêu chung.

Trong đội dự án, các cá nhân có thể đến từ nhiều phòng ban chức năng khác nhau, cũng có thể xuất phát từ bên ngoài (khách hàng, các cá nhân có liên quan, chuyên gia tư vấn, v.v.). Nhưng khi đã thành một đội (team), thì các cá nhân làm việc tập trung cho đội như là một đơn vị (unit) để hoàn tất mục tiêu chung. Bên trong nhóm liên chức năng không có các nhóm nhỏ khác. Ví dụ: một Nhóm Scrum "Alpha" về Content Marketing 9 người được thành lập với 1 Product Owner, 1 ScrumMaster, 2 người viết nội dung, 1 họa sỹ, 1 người chăm sóc cộng đồng, 2 người làm video, 1 chuyên gia nội dung, nhưng sẽ không phân chia chức năng thành các nhóm nhỏ khác như nhóm cộng đồng, nhóm video hay nhóm content... nữa, mà chỉ có một nhóm duy nhất "Alpha" với các cá nhân có chuyên môn khác nhau, hợp thành một đội thống nhất để làm việc hướng tới sản phẩm cần phát triển phục vụ các chiến lược marketing của công ty. Trong cách nói của các tác giả Scrum, thì dù bạn là tester (kiểm thử) hay copywriter (chuyên viết bài)... đều là

developer (nhà phát triển), họ là nhà phát triển nhưng có chuyên môn đặc thù là kiểm thử hay viết; developer trong trường hợp này không có ý nghĩa là coder/programmer (lập trình viên). Việc xóa nhòa các chức danh công việc (Job title) này có mục đích là để hướng mọi người vào một mục tiêu chung, không phân biệt "nhãn mác" (title): phát triển sản phẩm.

Khác với nhóm liên chức năng, nhóm chức năng (functional team) thường chỉ phụ trách một loại công việc đặc thù. Ví dụ, phòng thiết kế đồ họa thì không tham gia viết nội dung, phòng nội dung thì không có ai thiết kế. Công việc của nhóm chức năng thường có tính cộp lặp cao.

Có nhiều hiểu lầm phổ biến đối với người mới nghe tới Agile lần đầu, đơn cử: "Cá nhân trong nhóm liên chức năng biết làm tất cả mọi việc, trong nhóm Scrum không có ai là tester hết, không cần designer trong nhóm Scrum nữa, chỉ có developer thôi". *Thực chất là mỗi người mỗi việc*. Hay "Nhóm liên chức năng có nghĩa là ai cũng có thể làm được việc thay thế người khác, không cần phân vai rõ ràng, một người có vấn đề sẽ không ảnh hưởng đến cả nhóm". *Thực ra đây chỉ là trạng thái lý tưởng, không phải là cái bắt buộc ở một nhóm liên chức năng*.

Hay "*Cứ liên chức năng thì siêu năng suất*". Không hẳn, năng suất còn do cách thức phối hợp trong nhóm để đạt được mục tiêu chứ không đơn giản là chuyện cấu hình. Chúng ta hãy khảo sát một chút lý thuyết về nhóm để có thêm các hiểu biết về cách thức xây dựng một nhóm hiệu năng cao và vận dụng cho Nhóm Scrum.



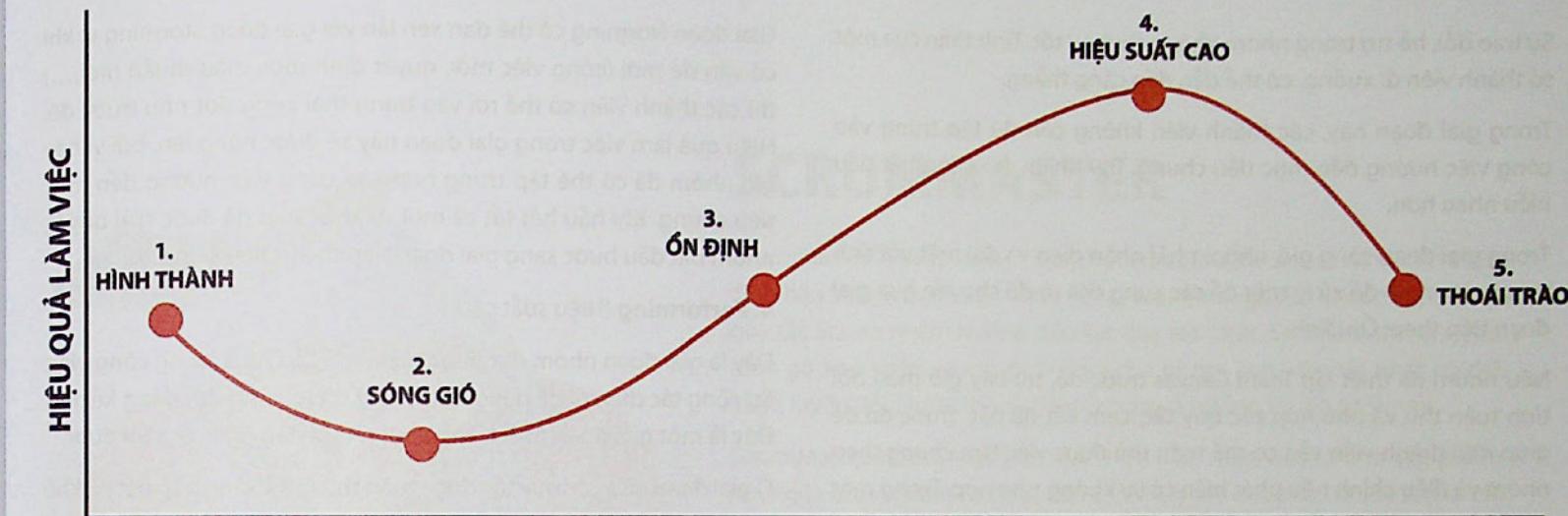
## Các giai đoạn phát triển nhóm

Để đạt hiệu năng cao cho đội nhóm nói chung và Nhóm Scrum nói riêng, chúng ta cần lưu ý các giai đoạn phát triển của nhóm để từ đó biết mình và nhóm cần làm gì để đạt hiệu năng tốt nhất.

Một nhóm cộng tác trải qua những giai đoạn khác nhau từ khi được bắt đầu thành lập cho đến khi hoạt động ổn định theo thời gian. Việc nhận diện các giai đoạn của nhóm là quan trọng để đưa ra các quyết định chính xác nhằm đảm bảo nhóm đạt được hiệu quả tốt nhất.

Bruce Tuckman (một nhà tâm lý học người Mỹ) đã đưa ra một mô hình để giải thích các giai đoạn này. Mô hình Tuckman chia chặng đường của một nhóm thành 5 giai đoạn.

Bruce Tuckman  
(ảnh: toolhero.com)



### CÁC GIAI ĐOẠN PHÁT TRIỂN CỦA NHÓM THEO MÔ HÌNH TUCKMAN

#### 1. Forming (Hình thành)

Đây là giai đoạn nhóm được thành lập, các thành viên vẫn còn đang lạ lẫm với nhau và bắt đầu tìm hiểu nhau để cộng tác vì công việc trước mắt. Ở giai đoạn này, các thành viên có thể chưa hiểu rõ mục đích chung của cả nhóm cũng như nhiệm vụ cụ thể của từng người trong nhóm. Nhóm có thể đưa ra những quyết định dựa trên sự đồng thuận, hiếm có các xung đột gay gắt do mọi người vẫn đang còn dè dặt với nhau. Tâm lý chung ở giai đoạn này đó là: Hưng phấn với công việc mới; Dè dặt trong việc tiếp cận và chia sẻ với các thành viên khác; Quan sát và thăm dò mọi người xung quanh; Tự định vị mình trong cấu trúc của nhóm.

Để vượt nhanh qua giai đoạn này, các thành viên sẽ phải tìm hiểu nhau nhiều hơn, một vài hoạt động dã ngoại là hữu ích và nhóm nên xây dựng Team Canvas là cơ chế để đi nhanh qua giai đoạn này và chuẩn bị nền tảng vượt qua giai đoạn tiếp theo: Sóng gió.

#### 2. Storming (Sóng gió)

Giai đoạn này xảy ra khi các thành viên bắt đầu bộc lộ mình và có thể phá vỡ những quy tắc của nhóm đã được thiết lập từ đầu. Đây là giai đoạn rất khó khăn đối với nhóm và dễ dẫn đến kết quả xấu.

Ở giai đoạn sóng gió, có thể xảy ra xung đột giữa các thành viên do những nguyên nhân khác nhau như: Phong cách làm việc, tính cách, cách cư xử, tranh cãi về các vấn đề hay giải pháp, văn hóa,... Các thành viên cũng có thể không hài lòng về công việc của nhau, dễ có các so sánh giữa mình với người khác,... Nhóm khó đi đến các quyết định dựa trên sự đồng thuận.

Các thành viên cũng có thể bắt đầu chất vấn về các quy tắc đã được thiết lập, muốn chỉnh sửa, thử nghiệm và có thể phá vỡ các quy tắc đó. Tệ hơn nữa, một số thành viên có thể tỏ ra không hợp tác, không cam kết trong công việc và không hài lòng với cách làm việc hiện tại.

Sự trao đổi, hỗ trợ trong nhóm không thực sự tốt. Tinh thần của một số thành viên đi xuống, có thể dẫn đến căng thẳng.

Trong giai đoạn này, các thành viên không còn đủ tập trung vào công việc hướng đến mục tiêu chung. Tuy nhiên, họ cũng bắt đầu hiểu nhau hơn.

Trong giai đoạn sóng gió, nhóm phải nhận diện và đổi mới với tình trạng của mình để xử lý triệt để các xung đột từ đó chuyển qua giai đoạn tiếp theo: Ốn định.

Nếu nhóm đã thiết lập Team Canvas trước đó, thì bây giờ theo dõi tính tuân thủ và phù hợp các quy tắc, cam kết đã đặt trước đó để giúp mọi thành viên vẫn có thể tuân thủ được việc làm chung theo nhóm và điều chỉnh nếu phát hiện có sự không phù hợp. Trong một nhóm Scrum, ScrumMaster cần nhận diện được khi nào nhóm đang trong giai đoạn sóng gió, theo sát các diễn biến để có điều chỉnh kịp thời, giúp nhóm vượt qua giai đoạn này nhanh nhất có thể.

### 3. Norming (Ốn định)

Giai đoạn Ốn định bắt đầu khi mọi người bắt đầu chấp nhận nhau, chấp nhận sự khác biệt, cố gắng giải quyết các mâu thuẫn, nhận biết thế mạnh của các thành viên khác và tôn trọng lẫn nhau.

Các thành viên bắt đầu trao đổi với nhau suôn sẻ hơn, tham khảo ý kiến lẫn nhau và yêu cầu sự trợ giúp khi cần thiết. Có thể bắt đầu có các ý kiến mang tính xây dựng. Mọi người bắt đầu nhìn vào mục tiêu chung và có cam kết mạnh mẽ hơn trong công việc. Có thể có các quy tắc mới được hình thành và tuân thủ để giảm thiểu mâu thuẫn, tạo không gian thuận lợi để các thành viên làm việc và cộng tác.

Giai đoạn Norming có thể đan xen lẫn với giai đoạn Storming vì khi có vấn đề mới (công việc mới, quyết định mới, mâu thuẫn mới,...) thì các thành viên có thể rơi vào trạng thái xung đột như trước đó. Hiệu quả làm việc trong giai đoạn này sẽ được nâng lên, bởi vì bây giờ nhóm đã có thể tập trung hơn vào công việc hướng đến mục tiêu chung. Khi hầu hết tất cả mọi sự khác biệt đã được giải quyết, nhóm bắt đầu bước sang giai đoạn tiếp theo: Hiệu suất cao.

### 4. Performing (Hiệu suất cao)

Đây là giai đoạn nhóm đạt được hiệu quả cao nhất trong công việc. Sự cộng tác diễn ra dễ dàng mà không có sự xung đột đáng kể nào. Đây là một giai đoạn mà không phải nhóm nào cũng đạt tới được.

Ở giai đoạn này, các quy tắc được tuân thủ mà không gặp bất cứ khăc nào. Các cơ chế hỗ trợ lẫn nhau trong nhóm phát huy hiệu quả tốt. Sự nhiệt tình và cam kết của các thành viên với mục tiêu chung là không còn nghi ngờ gì nữa.

Các thành viên cảm thấy rất thoải mái khi làm việc trong nhóm. Các thành viên mới gia nhập cũng sẽ nhanh chóng hòa nhập và làm việc hiệu quả. Nếu có thành viên rời nhóm thì hiệu quả làm việc của nhóm cũng không bị ảnh hưởng nghiêm trọng. Tinh thần chủ đạo được thể hiện ở giai đoạn này là tinh thần đồng đội.

Sau giai đoạn này, Mô hình Tuckman còn có thêm giai đoạn Adjour (Thoái trào). Khi đó nhóm hoàn thành nốt các công việc của mình và có thể không còn duy trì nhóm nữa.



## SCRUMMASTER

ScrumMaster là một vai trò then chốt trong Nhóm Scrum có trách nhiệm đảm bảo Scrum được vận hành đúng, tuân thủ nguyên lý, các kỹ thuật và quy tắc Scrum nhằm hướng đến kết quả tốt nhất. ScrumMaster là người tháo gỡ khó khăn và tạo điều kiện cho nhóm làm việc tốt nhất có thể, huấn luyện đội nhóm thành thục Scrum và hưởng lợi ích từ Scrum.

ScrumMaster không trực tiếp tham gia vào công việc làm ra sản phẩm, nhưng là chất kết dính để các bên phối hợp với nhau tạo ra sản phẩm tốt. ScrumMaster không phải là quản lý của Nhóm mà là một lãnh đạo theo phong cách phục vụ (Servant Leader). Với vai trò là lãnh đạo phục vụ, ScrumMaster cung cấp các dịch vụ cho Product Owner, Nhóm Phát triển và Tổ chức.

Trích Hướng dẫn Scrum



ScrumMaster sức mạnh của Scrum

## Với Nhóm Phát triển, ScrumMaster phục vụ theo nhiều cách như sau:

Đào tạo căn bản Scrum trong trường hợp các thành viên chưa biết về khung làm việc này.

Đảm bảo Nhóm Phát triển thực hiện tốt các sự kiện trong Scrum, bao gồm việc tuân thủ khung thời gian, đúng mục đích và nội dung của từng sự kiện.

Hỗ trợ Nhóm Phát triển tìm kiếm và sử dụng hiệu quả các công cụ hỗ trợ phát triển, chẳng hạn công cụ để quản lý Sprint Backlog, duy trì biểu đồ Sprint Burndown, các hệ thống quản lý mã nguồn, kiểm thử tự động, tích hợp liên tục cũng như các công cụ để giao tiếp trong Nhóm Phát triển như email hay hệ thống trao đổi và chia sẻ trực tuyến, v.v..

Bảo vệ Nhóm Phát triển trước những can thiệp bên ngoài trong quá trình triển khai một Sprint. Ví dụ, ScrumMaster cần giải thích rõ cho Product Owner những tác hại mà việc này gây ra cho Nhóm Phát triển và sản phẩm khi Product Owner muốn thay đổi các hạng mục Product Backlog đã lựa chọn của một Sprint đang diễn ra, đồng thời hướng dẫn Product Owner đưa các thay đổi này vào Sprint tiếp theo, v.v..

Đảm bảo thông tin được minh bạch và thông suốt trong Nhóm Phát triển, chẳng hạn như thông qua việc nhắc nhở nhóm luân phiên cập nhật Sprint Backlog và biểu đồ Sprint Burndown, đảm bảo Nhóm Phát triển thực hiện đúng sự kiện Scrum Hằng ngày, v.v..

Đảm bảo Nhóm Phát triển có đầy đủ các tài nguyên cần thiết phục vụ sản xuất. Ví dụ như, không gian làm việc, các công cụ hỗ trợ và những tài nguyên khác khi có nhu cầu, v.v..

Trong trường hợp Nhóm Phát triển gặp khó khăn với một vấn đề kỹ thuật nhất định, nghĩa là Nhóm Phát triển gặp trở ngại ScrumMaster có thể giải quyết bằng cách tìm kiếm sự trợ giúp từ các chuyên gia bên ngoài nhóm.

Trích Hướng dẫn Scrum

## ScrumMaster phục vụ tổ chức bằng nhiều cách, cụ thể như:

Giúp nâng cao năng lực Scrum trong tổ chức, chẳng hạn thông qua việc đào tạo Scrum cho các cấp quản lý và các phòng ban khác nhau, thành lập các nhóm học tập Scrum trong tổ chức.

Làm việc với các ScrumMaster khác để lập kế hoạch triển khai Scrum trong phạm vi tổ chức, gia tăng hiệu quả của việc áp dụng Scrum trong tổ chức của mình.

Giúp đỡ nhân viên và các bên hữu quan hiểu và sử dụng được Scrum cũng như quy trình phát triển sản phẩm thực nghiệm (empirical product development).

Tạo ra sự thay đổi làm tăng năng suất của Nhóm Scrum.

ScrumMaster phục vụ các đối tượng khác nhau với các công việc khác nhau, các công việc này thay đổi theo các giai đoạn của quá trình phát triển.

Đầu tiên, ScrumMaster làm việc với Tổ chức để chuẩn bị và có thể bắt đầu áp dụng Scrum. Đồng thời, ScrumMaster tập trung vào

việc xây dựng Nhóm Phát triển cùng với huấn luyện, giúp đỡ Product Owner để đưa nhóm đi vào hoạt động. Ở giai đoạn này, các công việc liên quan đến kỹ thuật phát triển chỉ dừng lại ở mức tối thiểu, đảm bảo vẫn hoàn thành được công việc.

Sau đó, ScrumMaster có thể tạm thời gác lại các công việc liên quan đến tổ chức để tập trung vào việc giúp Nhóm làm việc ổn định hơn, từng bước nâng cao các kỹ năng của từng cá nhân và của cả Nhóm. Dần dần ScrumMaster có thể giới thiệu và hỗ trợ nhóm áp dụng các kỹ thuật phát triển để nâng cao năng suất, chất lượng sản phẩm.

Khi Nhóm Scrum đã thực sự ổn định và vận hành tốt, ScrumMaster có thể toàn tâm toàn ý để tập trung vào việc tìm kiếm, giới thiệu và hỗ trợ nhóm áp dụng các kỹ thuật phát triển tốt hơn. Cùng với đó là hoạt động để mở rộng Scrum ra tổ chức, giao tiếp và trao đổi giữa các nhóm, hướng đến việc phát triển Scrum bền vững.

Trích Hướng dẫn Scrum

## ScrumMaster phục vụ Product Owner theo nhiều cách như sau:

Tìm kiếm các kỹ thuật để quản lý hiệu quả Product Backlog;

Giao tiếp tích cực với Nhóm Phát triển về tầm nhìn, mục đích và các hạng mục của Product Backlog;

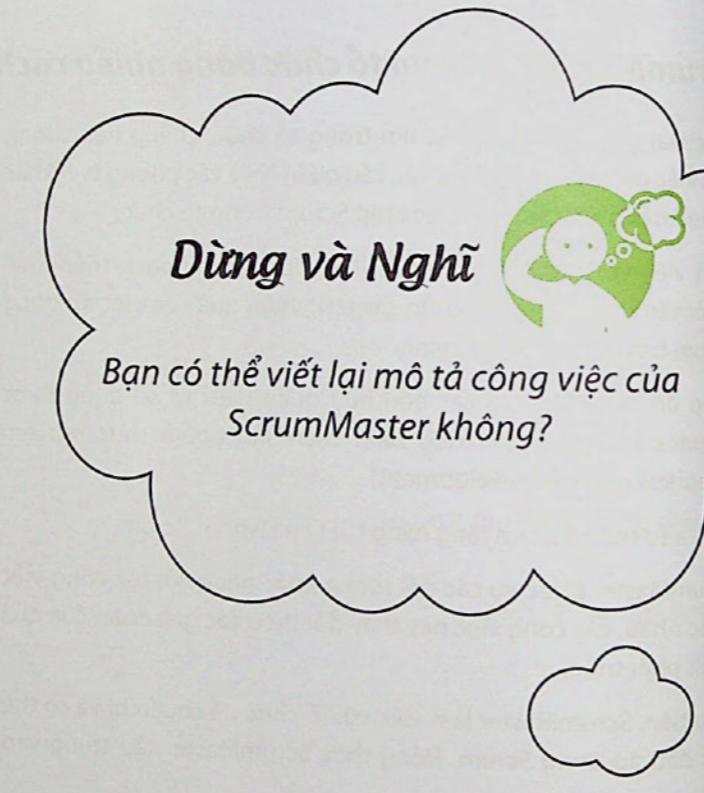
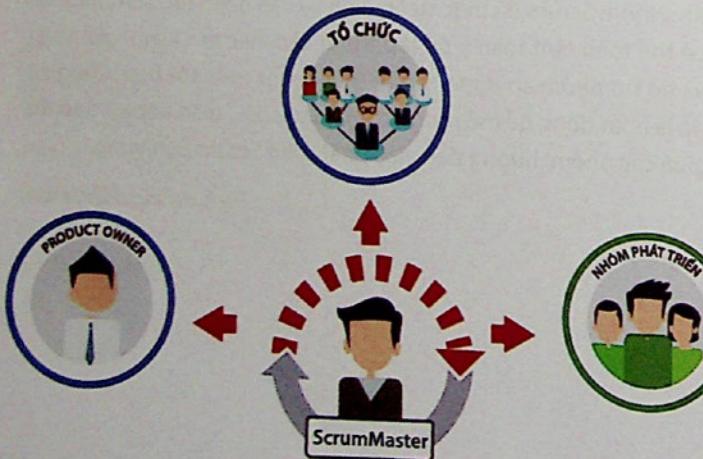
Dạy cho Nhóm Phát triển biết cách tạo ra các hạng mục Product Backlog thật rõ ràng và đơn giản;

Hiểu rõ việc lập kế hoạch dài hạn sản phẩm trong một môi trường thực nghiệm (empirical);

Hiểu rõ và thực hành sự linh hoạt (agility);

Thúc đẩy các sự kiện Scrum theo yêu cầu hoặc khi cần thiết.

Trích Hướng dẫn Scrum



## Công việc của ScrumMaster

### 1. Tổ chức các cuộc họp

Họp là cơ chế rất quan trọng để giao tiếp, báo cáo, giải quyết vấn đề, minh bạch thông tin,... trong nhóm. Scrum có rất nhiều cuộc họp để từ đó không cần hoặc cần ít hơn những cuộc họp và giao tiếp khác.

Công việc của ScrumMaster gắn liền với nhiều cuộc họp chính thức và không chính thức như họp Lập kế hoạch Sprint, họp Sơ kết Sprint, họp Cải tiến Sprint, họp viết yêu cầu, họp Scrum Hàng ngày và nhiều buổi họp đột xuất khác.

Khi một cuộc họp không được tổ chức tốt thường tốn rất nhiều thời gian của cả nhóm. Ví dụ, trong một buổi họp, cả nhóm 6 người phải đợi một thông tin chưa được chuẩn bị sẵn 5 phút thì tổng cả nhóm đã lãng phí 30 phút. Do đó, tổ chức các cuộc họp hiệu quả là việc đóng góp rất nhiều giá trị cho nhóm. ScrumMaster là người tổ chức sự chuẩn bị tốt cho các cuộc họp, điều phối và hỗ trợ để các cuộc họp thật hiệu quả.

### 2. Thanh tra, thu thập và bạch hóa thông tin

Để "quan sát" hiện trạng và cảm nhận thực tế làm việc của nhóm, ScrumMaster có thể bắt đầu với hàng loạt câu hỏi: Minh có nhìn thấy bảng công việc không? Khách hàng có biết việc gì đang diễn ra trên bảng không? Có gì được cập nhật từ hôm qua đến nay không? Có nhìn thấy biểu đồ Burndown không? Có điểm nào bất thường trên Biểu đồ Burndown không?

Để điều tra, "đào bới" thêm, ScrumMaster có thể dùng các câu hỏi Socratic: Tôi nhận thấy <tình huống>, chúng ta sẽ làm gì? Tôi quan sát thấy <tình huống>, nó có quan trọng không? Tôi thấy <cảm giác>, bạn có thấy điều đó? Chúng ta sẽ cố tìm lý do của <tình trạng>? Bạn nghĩ chúng ta cần làm gì? Ai có ý tưởng gì về <tình trạng>? Điều này có hiệu quả không? Bạn đã quyết định điều gì? Bạn nên làm gì [với tình huống\vấn đề này]?

Để điều tra nguyên nhân của vấn đề, ScrumMaster còn phải thành thạo và liên tục sử dụng 5WHYs (Hỏi 5 lần tại sao) để tìm ra cách thúc đẩy nhóm tiến lên.

Các thông tin quan sát được, điều tra được cần lưu trữ và tổ chức khoa học để tiện bê truy xuất (cá nhân và đội nhóm).

### 3. Loại bỏ trở ngại

Các trở ngại rất đa dạng, từ khó khăn trong giao tiếp, những hỏng hóc của trang thiết bị làm việc, hay khó thu xếp với khách hàng, hoặc quá thiếu thốn nguồn lực, v.v. Sử dụng 5WHYs để tìm những phương án cho các tình huống khó khăn. Ngoài ra, ScrumMaster có thể duy trì một danh sách trở ngại (Impediment Backlog) và cùng với các bên để tháo gỡ dần dần. Đây là một trong những công việc quan trọng và thường trực nhất của ScrumMaster.

#### 4. Tìm kiếm cải tiến

Product Owner của tôi làm việc thế nào?  
Nhóm Phát triển đang làm việc thế nào?  
Các kỹ thuật đang được dùng thế nào? Tổ chức đang làm việc ra sao? Ai cần được huấn luyện về cái gì? Quy trình có dư thừa cái gì không? Có thể mua sắm hay làm mới công cụ gì để tăng năng suất và gia tăng không khí vui vẻ không?

Đó là những câu hỏi mà ScrumMaster có thể tự mình đặt ra hàng ngày bên cạnh việc tổ chức sự kiện Cải tiến Sprint cho thật hiệu

qua để liên tục tìm kiếm các cải tiến, giúp nhóm làm việc hiệu quả hơn, vui vẻ hơn.

ScrumMaster có thể duy trì một kaizen log để ghi nhận theo dõi cải tiến qua thời gian, làm thành dữ liệu để tổng hợp và chia sẻ những cải tiến quan trọng cho các nhóm khác, hoặc chuyển giao cho ScrumMaster mới khi mình không còn làm việc trong Nhóm Scrum nữa.

#### 5. Huấn luyện (Coaching) Scrum cho nhóm

Thông qua seminar, workshop, coaching bằng hội thoại hoặc làm mẫu, v.v. để giúp thành viên thành thục Scrum và cách thức giải quyết vấn đề.

Hãy tham khảo thêm danh mục kiểm tra của ScrumMaster ở cuối chương này để dùng như là một tài liệu hướng dẫn khi một ScrumMaster bắt đầu đảm nhận vai trò này.



LƯU Ý!

#### ScrumMaster là:

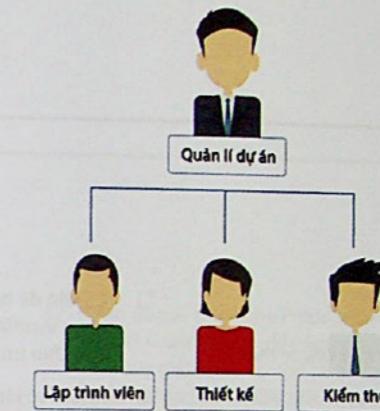
- Lãnh đạo Phục vụ
- Cố vấn
- Nhân tố thay đổi
- Người giải quyết vấn đề
- "Chó chăn cừu"
- Người làm việc toàn thời gian

#### ...không là:

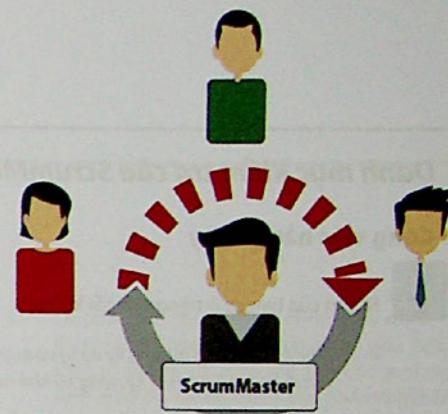
- Quản lý Dự án
- Nắm quyền đối với nhóm
- Kiến trúc sư trưởng
- Ai kiêm nhiệm cũng được

#### Ví dụ công việc một ngày của ScrumMaster:

1. Đến văn phòng trước Nhóm Phát triển 30 phút để chuẩn bị cà phê, không gian làm việc cho nhóm, loại bỏ những trở ngại để họ không bị phân tâm
2. Dẫn dắt nhóm họp Scrum Hàng ngày
3. Giải quyết khó khăn cho một vấn đề được nêu ra trong buổi Scrum Hàng ngày với quản lý
4. Yêu cầu Product Owner phân tách một vài hạng mục to trong Product Backlog thành những hạng mục nhỏ hơn và thêm tiêu chí chấp nhận
5. Dạy cho Nhóm Phát triển một kỹ thuật mới
6. Thực hiện một hành động cải tiến đã đưa ra ở buổi Cải tiến Sprint trước đó
7. Gặp Product Owner để làm rõ và xem xét lại một số hạng mục trên cùng của Product Backlog
8. Ăn trưa với một chuyên gia về nghiệp vụ sản phẩm
9. Cập nhật bảng công việc của nhóm
10. Uống cà phê cùng Product Owner để thảo luận về những nợ kỹ thuật mà do nhóm đã chấp nhận để hoàn thành một hạn chót trước đó
11. Ngăn một bên liên quan tới năn nỉ Nhóm Phát triển làm thêm một yêu cầu ngoài kế hoạch
12. Xây dựng đề cương để huấn luyện cho một thành viên mới
13. Huấn luyện một thành viên mới
14. Gặp gỡ với nhóm thực hành Scrum
15. Học một kỹ thuật mới
16. Ghi nhật ký nhóm



Quản lý dự án (PM) giao việc và kiểm soát



ScrumMaster tạo điều kiện

## Danh mục kiểm tra của ScrumMaster

### Công việc hằng ngày

- Nhóm của bạn có ở trạng thái tốt không?
  - *Mục tiêu rõ ràng (những kỳ vọng và quy định phải rõ ràng, và mục tiêu có thể đạt được, phù hợp với kỹ năng và khả năng của mỗi người).*
  - *Tập trung và có trọng điểm, tập trung cao độ vào một lĩnh vực nhất định cần được chú ý.*
  - *Không có cảm giác tự ti, để cao hành động và nhận thức.*
  - *Phản hồi trực tiếp và ngay lập tức (nhanh chóng nhìn thấy các thành công và thất bại của một chuỗi hoạt động, nhờ thế có thể điều chỉnh hành vi nếu cần thiết).*
  - *Cân bằng giữa cấp độ khả năng và thử thách (hoạt động đưa ra không quá khó cũng không quá dễ).*
  - *Mỗi người đều có khả năng tự kiểm soát trong mỗi tình huống hay hoạt động.*
  - *Mỗi hoạt động đều hiển nhiên đem lại kết quả, vì thế không cần quá nhiều cố gắng trong hành động.*
- Các thành viên có thích nhau không, có thư giãn cùng nhau không và có vui mừng trước thành công của những thành viên khác không?
- Các thành viên nhóm có cùng nhau giữ cho mỗi người đều phải đạt được những tiêu chuẩn cao, và thúc đẩy mỗi người đều phát triển?
- Có vấn đề hoặc cơ hội nào mà nhóm đang không thảo luận cùng nhau vì những vấn đề hoặc cơ hội đó có thể gây ra tình trạng quá khó chịu trong nhóm không?
- Các thành viên có thích nhau không, có thư giãn cùng nhau không và có vui mừng trước thành công của những thành viên khác không?
- Các thành viên nhóm có cùng nhau giữ cho mỗi người đều phải đạt được những tiêu chuẩn cao, và thúc đẩy mỗi người đều phát triển?

- Có vấn đề hoặc cơ hội nào mà nhóm đang không thảo luận cùng nhau vì những vấn đề hoặc cơ hội đó có thể gây ra tình trạng quá khó chịu trong nhóm không?
- Nhóm có tập trung liên tục vào các mục tiêu Sprint không? Có thể bạn nên thực hiện một bước kiểm tra giữa Sprint để có thể tái rà soát các tiêu chuẩn chấp thuận của các hạng mục trong Product Backlog đã cam kết hoàn thành trong Sprint hiện tại.
- Bảng phân công nhiệm vụ của Sprint có phản ánh đúng những công việc mà nhóm đang thực sự làm không? Cần nhận thức rõ “vấn đề tiềm ẩn” của những công việc không được nêu ra và những nhiệm vụ có thể tồn tại một ngày mới có thể hoàn thành. Những công việc không liên quan đến những cam kết trong Sprint sẽ là những trở ngại cho việc hoàn thành các cam kết đó.
- Bảng phân công nhiệm vụ của nhóm bạn có cập nhật không?
- Các thành viên nhóm có biết về các công cụ tự quản lý của nhóm không, các công cụ đó có tiện dụng không?
- Những công cụ quản lý có được những người ngoài nhóm tôn trọng đúng mức không? Sự giám sát quá mức đối với các hoạt động thường nhật của những người bên ngoài nhóm có thể hủy hoại sự minh bạch và cơ chế tự quản lý trong nhóm.
- Các thành viên nhóm có tự nguyện nhận nhiệm vụ không?
- Sự cần thiết của việc hoàn trả nợ kỹ thuật dần dần sẽ giúp cho việc lập trình trở nên dễ chịu hơn, đã được ghi nhận rõ ràng trong khái niệm hoàn thành chưa?
- Các thành viên nhóm có đang cùng nhau chịu trách nhiệm về mọi khía cạnh của sản phẩm chung (kiểm thử, tài liệu cho người dùng, v.v.) mà không quan tâm đến chức danh của mỗi người không?

### Công việc theo Sprint

- Product Backlog có được sắp xếp theo hiểu biết mới nhất của nhóm không?
- Các yêu cầu và mong muốn từ tất cả các bên liên quan có được ghi nhận trong Product Backlog không? Ghi nhớ: backlog là quan trọng.
- Khối lượng của Product Backlog có thể quản lý được không? Để duy trì số lượng hạng mục trong giới hạn có thể quản lý được, hãy giữ những hạng mục chi tiết ở trên cùng, còn các hạng mục trừ tượng nói chung ở dưới. Sẽ phản tác dụng nếu chúng ta phân tích quá sâu vào những hạng mục ở phía dưới của Product Backlog. Những yêu cầu đặt ra sẽ thay đổi trong những cuộc trao đổi liên tục giữa sản phẩm đang phát triển và các bên liên quan hoặc khách hàng.
- Có yêu cầu nào (đặc biệt là những yêu cầu ở phía trên cùng của Product Backlog) có thể được thể hiện tốt hơn bằng những user story độc lập, có thể thương lượng được, có thể đánh giá được, có thể ước lượng được, nhỏ và có thể kiểm thử được?
- Bạn đã giải thích cho Product Owner của bạn về nợ kỹ thuật và cách để tránh nó chưa? Một trong những giải pháp có thể là thêm việc viết kiểm thử tự động và tái cấu trúc vào định nghĩa “hoàn thành” cho mỗi hạng mục.
- Backlog có phải là một biểu đồ thông tin, mà các bên liên quan có thể lập tức nhận thấy được không?
- Nếu bạn đang sử dụng một công cụ tự động trong quản lý backlog, mọi người có biết cách để sử dụng nó dễ dàng không? Các công cụ quản lý tự động cũng dẫn tới nguy cơ trở thành sự tắc nghẽn thông tin nếu thiếu đi sự truyền tải thông tin chủ động từ ScrumMaster.
- Bạn có thể giúp truyền tải thông tin bằng các bản in cho mỗi người không?
- Bạn có thể giúp truyền tải thông tin bằng cách tạo ra các biểu đồ to và rõ ràng không?
- Bạn đã giúp Product Owner sắp xếp các hạng mục backlog vào các thời điểm phát hành thích hợp hoặc trong những nhóm ưu tiên chưa?
- Có phải mỗi người trong nhóm đều nhận thức được liệu kế hoạch phát hành còn phù hợp với thực tế không? Bạn có thể thử cho mọi người xem Biểu đồ tương quan sản phẩm/phát hành với thời gian thực hiện sau khi các hạng mục đã được xác nhận “hoàn thành” trong mỗi cuộc họp Sơ kết Sprint. Biểu đồ thể hiện tỷ lệ các hạng mục Product Backlog đã được hoàn thành và những hạng mục mới được thêm vào để cho phép phát hiện sớm những biến động trong quy mô hoặc kế hoạch thực hiện.
- Product Owner của bạn đã điều chỉnh kế hoạch phát hành sau buổi họp Sơ kết Sprint gần nhất chưa? Chỉ có số ít Product Owner đã bàn giao sản phẩm được kiểm thử đầy đủ đúng hạn sắp xếp lại kế hoạch phát hành mỗi Sprint. Điều này có thể sẽ khiến cho một vài sản phẩm cần được phát hành sau vì có những việc quan trọng hơn được phát hiện ra.
- Bạn đã từng thử nhiều cách thức và địa điểm cho các buổi họp Cải tiến Sprint chưa?



Danh mục kiểm tra của ScrumMaster



# PRODUCT OWNER

Product Owner là một trong ba vai trò Scrum. Vai trò này chịu trách nhiệm định hướng sản phẩm trong suốt quá trình sản xuất. Nhiệm vụ của Product Owner là tối ưu hóa giá trị của sản phẩm thông qua việc tận dụng tốt nhất khả năng sản xuất của Nhóm Phát triển.

Để đảm đương tốt vai trò này, Product Owner cần là người có hiểu biết về tầm nhìn của sản phẩm và những yêu cầu để hiện thực hóa tầm nhìn đó. Có thể Product Owner chưa cần biết ngay từ đầu thật cụ thể là sẽ làm những gì, nhưng có hiểu biết sâu sắc tại sao lại xây dựng sản phẩm này.

Product Owner là một người duy nhất, không phải là một nhóm người. Điều này không có nghĩa là một mình Product Owner phải làm mọi việc, mà có thể trao đổi, nhận sự hỗ trợ từ những người khác. Tuy nhiên, Product Owner vẫn là người đại diện duy nhất và chịu trách nhiệm về sản phẩm đang xây dựng. Cụ thể, Product Owner là người duy nhất chịu trách nhiệm quản lý Product Backlog - nơi lưu trữ các hạng mục cần phát triển của sản phẩm.

Cách thức quản lý Product Backlog được bàn kỹ trong Chương 5 và một phần trong Chương 9 (về sử dụng User Story và ước tính linh hoạt).

Ngoài việc quản lý Product Backlog, Product Owner còn phải tích cực tham gia vào các cuộc họp của nhóm để cung cấp các thông tin cần thiết, chủ động quản lý một Kế hoạch Phát hành cho sản phẩm.

## Sáu nhiệm vụ chính của một Product Owner

1. Công việc đầu tiên của Product Owner là tìm hiểu và phân tích kỹ về sản phẩm dự định phát triển, lên danh sách các tính năng mong muốn bằng việc liệt kê chúng trong một Product Backlog. Đây là danh sách các hạng mục mà Nhóm Phát triển dựa vào để làm việc và chuyển thành các tính năng của sản phẩm thật. Danh sách này không phải là cố định mà được điều chỉnh trong suốt quá trình phát triển của sản phẩm sao cho phù hợp nhất.
2. Các hạng mục trong Product Backlog, sẽ được Product Owner đánh giá giá trị và sắp xếp. Hạng mục nào có giá trị cao hơn sẽ được đưa lên trên để đưa vào sản xuất sớm nhất. Khái niệm "giá trị" ở đây phụ thuộc vào rất nhiều yếu tố, ví dụ như lợi nhuận thu về, mong muốn của khách hàng, các mục tiêu chiến lược, sự phụ thuộc vào các hạng mục khác, rủi ro và nhiều yếu tố khác nữa. Nếu không có được một giá trị cụ thể bằng con số, nhưng Product Owner có thể so sánh dựa theo giá trị tương đối để thực hiện việc sắp xếp. Điều này cũng tương tự như việc, chúng ta không biết chính xác kích thước của quả bưởi, nhưng biết chắc chắn là nó to hơn quả cam.
3. Nhiệm vụ thứ ba của Product Owner là tối ưu hóa lợi nhuận trên vốn đầu tư (ROI). Trong khi, khả năng sản xuất của Nhóm Phát triển thường có một giới hạn nhất định, Product Owner là người luôn phải tìm cách để sử dụng tốt nhất khả năng này. Vì vậy, ngoài việc sắp xếp các hạng mục trong Product Backlog thì Product Owner cũng cần phải biết nói "không" để loại bỏ những hạng mục không cần thiết.
4. Product Backlog là một tạo tác (artifact) quan trọng. Product Owner phải giữ cho nó luôn được minh bạch và rõ ràng đối với tất cả mọi người. Nói cách khác Product Owner phải đảm bảo tính minh bạch của Product Backlog.
5. Một nhiệm vụ quan trọng nữa của Product Owner là phải luôn luôn sẵn sàng giải thích cho Nhóm Phát triển hiểu rõ các hạng mục của Product Backlog. Điều này đảm bảo Nhóm Phát triển hiểu đúng và đầy đủ về từng hạng mục mà họ triển khai.
6. Nhiệm vụ của Product Owner còn là theo dõi tiến độ phát triển sản phẩm, đảm bảo minh bạch và có thể giải trình với các bên liên quan khi có yêu cầu. Product Owner có thể sử dụng các công cụ như Biểu đồ Burndown, hay các công cụ khác để thực hiện nhiệm vụ này.

Ngoài các nhiệm vụ trên, Product Owner là người có quyền và chịu trách nhiệm khi quyết định hủy Sprint (dừng Sprint bất thường).

## Ví dụ công việc một ngày của Product Owner:

1. Tham gia nghe buổi Scrum Hàng ngày
2. Tham gia giải quyết một khó khăn của Nhóm Phát triển liên quan tới sản phẩm
3. Giải đáp thắc mắc cho Nhóm Phát triển về một User Story đang phát triển
4. Trao đổi qua Skype với một bên liên quan mà không tham dự được buổi Sơ kết Sprint
5. Ăn trưa với một người sử dụng dịch vụ để nhận biết thêm yêu cầu
6. Gặp gỡ một bên liên quan để thảo luận về một số điểm cần có trong bản phát hành tới
7. Sắp xếp lại Product Backlog
8. Làm việc với ScrumMaster để cải tiến buổi Sơ kết Sprint tiếp theo
9. Cập nhật tiến độ dự án
10. Gặp giám đốc để trao đổi về bản phát hành tiếp theo
11. Làm việc với Nhóm phát triển và ScrumMaster về những thuật ngữ xem cách thức và thời điểm xử lý
12. Tác User Story lớn
13. Thu thập thông tin về sản phẩm với đội vận hành dịch vụ sản phẩm

### Danh mục kiểm tra của Product Owner



### Danh mục kiểm tra của Product Owner

#### Kiểm tra hàng ngày

- Đã làm rõ yêu cầu sản phẩm cho Nhóm Phát triển (nếu có)
- Đã làm mịn các hạng mục của Product Backlog, đặc biệt các hạng mục ở trên

#### Kiểm tra theo Sprint

- Đã làm việc với các bên liên quan để cập nhật những kỳ vọng mới về sản phẩm
- Đã sắp xếp các hạng mục trong Product Backlog đảm bảo tối ưu hóa ROI
- Đã làm mịn các hạng mục của Product Backlog
- Đã xác định tiêu chí chấp nhận cho các hạng mục sẽ phát triển trong Sprint tới
- Đã tham gia Lập kế hoạch Sprint để trình bày về tổng quan sản phẩm, từng hạng mục sẽ làm trong Sprint tới và giải đáp mọi thắc mắc về yêu cầu trong Lập kế hoạch Sprint
- Đã tham gia và đưa ra phản hồi về sản phẩm trong Sơ kết Sprint



Product Owner

## Dùng và Nghĩ

Bạn có thể viết lại mô tả công việc của Product Owner không?



# NHÓM PHÁT TRIỂN

Nhóm Phát triển là đội ngũ trực tiếp làm ra sản phẩm, nhóm này bao gồm các chuyên gia có nhiệm vụ chuyển giao phần tăng trưởng có thể chuyển giao được ở cuối mỗi Sprint. Nhóm Phát triển trong Scrum cũng là một nhóm tự tổ chức và liên chức năng. Nhóm được trao quyền để tự định hướng và đưa ra các quyết định liên quan đến công việc sản xuất. Tự tổ chức cũng có nghĩa là nhóm có toàn quyền lựa chọn công cụ, kỹ thuật và cách thức để hoàn thành công việc. Trong quá trình sản xuất, họ tự tiến hành ước lượng, phân bổ, theo dõi, điều tiết công việc theo hình thức tập thể.

Khi bắt đầu mỗi Sprint, Nhóm Phát triển tự lựa chọn các hạng mục từ danh sách đã được sắp xếp của Product Backlog sao cho phù hợp với khả năng sản xuất của mình. Đây là một khác biệt so với một số mô hình phát triển truyền thống - các mô hình hoạt động theo hình thức ra lệnh và kiểm soát. Trong mô hình sản xuất truyền thống các công việc thường được bộ phận quản lý phân chia và gán trách nhiệm cho từng người. Với cách thức làm việc như của Nhóm Phát triển trong Scrum sẽ giúp làm tăng ý thức sở hữu và tính cam kết của các thành viên.



Đối với mỗi hạng mục công việc, Nhóm Phát triển là những người ước lượng kích thước tốt nhất bởi vì họ là lực lượng trực tiếp sản xuất, là những người am hiểu về công việc và các yếu tố liên quan đến nó. Điều này không chỉ cần thiết đối với nhóm mà còn hữu ích đối với Product Owner khi quản lý Product Backlog.

Nhóm Phát triển thường xuyên theo dõi tiến độ công việc để biết được khả năng, tốc độ sản xuất, tình hình đạt được Mục tiêu Sprint và các mức quan trọng của sản phẩm. Nhờ theo dõi liên tục và sự minh bạch về thông tin nên Nhóm Scrum nói chung, Nhóm Phát triển nói riêng có những điều chỉnh hợp lý trong tổ chức sản xuất.



Một đặc điểm quan trọng của Nhóm Phát triển đó là tính liên chức năng. Liên chức năng đối với Nhóm Phát triển có nghĩa là nhóm được trang bị đầy đủ các kỹ năng cần thiết để hoàn thành công việc và chuyển giao phần tăng trưởng ở cuối mỗi Sprint mà không cần sự hỗ trợ từ bên ngoài. Đặc điểm này là quan trọng vì nó giúp Nhóm Phát triển có thể hoạt động độc lập, trở thành một đơn vị sản xuất hoàn chỉnh, tổ chức vận hành và chuyển giao sản phẩm có chất lượng. Đặc điểm này giúp Nhóm Phát triển đưa ra các quyết định nhanh chóng, kịp thời mà không phụ thuộc và chờ đợi từ các cá nhân, đơn vị khác bên ngoài.

Việc tổ chức nhóm liên chức năng là rất khác biệt so với cách tổ chức theo phòng/ban chuyên môn truyền thống. Ở những hệ thống có các phòng ban chuyên môn, công việc sản xuất đòi hỏi sự cộng tác, phối hợp giữa những bộ phận khá tách biệt. Điều này làm giảm đáng kể sự linh hoạt, gây lãng phí và làm chậm tốc độ sản xuất. Đối với một nhóm liên chức năng, sự cộng tác và đồng bộ diễn ra nhanh chóng và tối ưu giúp cho năng suất và hiệu quả sản xuất được nâng lên.



Nhóm Phát triển không có sự phân chia chức danh hay vai trò, tất cả các thành viên đều có vai trò giống nhau. Mỗi thành viên thường có thể mạnh ở một vài kỹ năng nhất định và những kỹ năng đó góp lại trở thành kỹ năng chung của nhóm chứ không còn là của riêng cá nhân nào. Như vậy, chúng ta sẽ không thấy những chức danh như "nhân viên phát triển nội dung (copywriter)", "họa sỹ (art designer)", "nhà phân tích nghiệp vụ (BA)", "nhà thiết kế (designer)", "lập trình viên (coder)", "kiểm thử viên (tester)" trong Nhóm Phát triển. Thay vào đó, tất cả các thành viên đều được gọi chung là "nhà phát triển (developer)" và kỹ năng tổng thể của nhóm là phân tích nghiệp vụ, phát triển nội dung, vẽ, thiết kế, lập trình, kiểm thử. Việc xóa bỏ sự phân chia vai trò giúp làm tăng tính sở hữu tập thể, trách nhiệm tập thể và bình đẳng trong công việc. Đồng thời cũng giúp thúc đẩy việc học tập trong Nhóm Phát triển. Các thành viên không chỉ tham gia giải quyết những hạng mục công việc mà mình có thể mạnh nhất, mà còn giúp sức trong những công việc khác. Điều này đòi hỏi sự trao đổi, học tập và rèn luyện những kỹ năng khác diễn ra giữa các thành viên. Theo thời gian mỗi thành viên sẽ học được những kỹ năng mới ngoài những kỹ năng vốn có của mình. Ví dụ, một thành viên với kỹ năng chính là thiết kế thì có thể có thêm kỹ năng mới là kiểm thử, một thành viên với kỹ năng chính là phân tích nghiệp vụ thì có thể có thêm kỹ năng mới là lập trình.

## Nhóm Phát triển gồm bao nhiêu thành viên là đủ?



### Có cần sự ổn định không?

Nhóm Phát triển vận hành tốt nhất với số lượng thành viên vừa phải, quy định của Scrum là từ 3 đến 9 người. Nhóm quá ít thành viên sẽ dẫn đến khả năng sản xuất nhỏ và có thể thiếu các kỹ năng cần thiết (không liên chức năng), khó để tạo được các phần tăng trưởng hoàn chỉnh đáng kể ở cuối mỗi Sprint. Nhóm có quá nhiều thành viên sẽ làm tăng tính phức tạp trong quản lý, gây khó khăn trong tương tác, giao tiếp, đảm bảo minh bạch và làm giảm tính linh hoạt của nhóm. Nếu dự án lớn cần nhiều người, chúng ta cần chia nhỏ nhóm và sử dụng kỹ thuật Scrum ở quy mô lớn ở các chương sau.

Sử dụng mô hình Tuckman, chúng ta có thể thấy Nhóm Phát triển đòi hỏi tính ổn định cao để đạt được năng suất và chất lượng cao nhất trong sản xuất. Điều này sẽ thuận lợi cho các thành viên trong cộng tác, giao tiếp, tận dụng tối ưu sự đóng góp của mỗi người. Sự ổn định cũng giúp nhóm điều chỉnh các thói quen hoạt động của mình một cách phù hợp và hiệu quả nhất. Giữ cho nhóm ổn định cũng giúp khả năng tiên lượng khi lập kế hoạch tốt hơn do các yếu tố như hiệu suất và tốc độ sản xuất của nhóm ổn định.

Tuy vậy, trong thực tế việc thay đổi thành viên đôi khi là khó tránh khỏi do nhiều nguyên nhân khác nhau. Khi đó, nhóm biết rằng việc đó sẽ ảnh hưởng tới năng suất trong ngắn hạn.

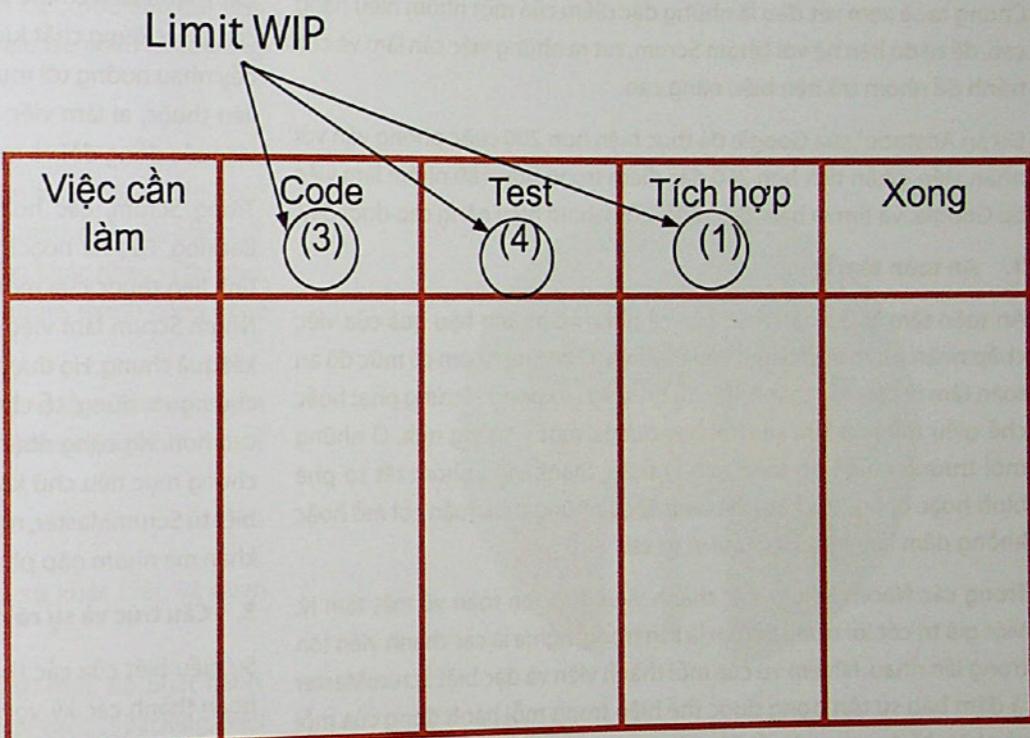
## Nhóm Phát triển đóng góp những gì cho sản phẩm?

Ngoài việc trực tiếp làm ra sản phẩm, sự am hiểu của Nhóm Phát triển về sản phẩm là rất quan trọng để đi đến thành công. Đặc biệt là đối với các sản phẩm mới. Ở đó, sản phẩm là cả một quá trình tìm tòi, thử nghiệm, điều chỉnh mà ngay cả Product Owner và các bên liên quan đều không biết rõ ràng từ trước.

Nhóm Phát triển còn giúp sức cho Product Owner trong việc duy trì Product Backlog. Có những công việc mà Nhóm Phát triển là những người làm tốt nhất, chẳng hạn như việc ước tính kích thước của các hạng mục, các yêu cầu liên quan đến công nghệ, sự phụ thuộc giữa các hạng mục, v.v.. Đây là những yếu tố có ảnh hưởng lớn đến nội dung của Product Backlog.



Nhóm Phát triển hoạt động hướng vào mục tiêu chung, thay vì hướng vào công việc của từng cá nhân. Và tuân thủ nguyên tắc giới hạn lượng việc đang làm (Limit WIP - Limit Work-In-Progress).



# Đặc điểm của nhóm hiệu năng cao

Liệu Nhóm Scrum có hiệu năng cao?

Chúng ta sẽ xem xét đâu là những đặc điểm của một nhóm hiệu năng cao, để từ đó liên hệ với Nhóm Scrum, rút ra những việc cần làm và cần tránh để nhóm trở nên hiệu năng cao.

Dự án Aristotle<sup>1</sup> của Google đã thực hiện hơn 200 cuộc phỏng vấn với nhân viên, phân tích hơn 250 đặc điểm trong hơn 180 nhóm làm việc tại Google, và tìm ra năm đặc điểm của nhóm hiệu năng cao dưới đây.

## 1. An toàn tâm lý

An toàn tâm lý là nhận thức của cá nhân về những hậu quả của việc chấp nhận rủi ro mang tính liên cá nhân. Ở những nhóm có mức độ an toàn tâm lý cao, các thành viên tự tin rằng sẽ không ai trừng phạt hoặc chế giễu mình vì làm sai, hỏi hay đưa ra một ý tưởng mới. Ở những môi trường có độ an toàn tâm lý thấp, thành viên nhóm rất sợ phê bình hoặc bị người khác phê bình, ít có những thảo luận cởi mở hoặc không dám làm một việc mới vì sợ sai.

Trong các Nhóm Scrum, các thành viên được an toàn về mặt tâm lý. Một giá trị cốt lõi trong Scrum là tôn trọng, nghĩa là các thành viên tôn trọng lẫn nhau. Nhiệm vụ của mỗi thành viên và đặc biệt ScrumMaster là đảm bảo sự tôn trọng được thể hiện trong mỗi hành động của mỗi cá nhân. Ví dụ, trong buổi Cải tiến Sprint, một thành viên hỏi về thiện chí của một thành viên khác, ScrumMaster ngay lập tức yêu cầu dừng việc đó lại và hướng dẫn thảo luận trên vấn đề, phương pháp làm việc chứ không phải chỉ trích cá nhân. Bên cạnh đó, một giá trị cốt lõi khác là Cởi mở quy định rằng mọi thứ cần phải được minh bạch, rõ ràng để mọi người trong nhóm có thể làm việc hiệu quả với nhau.

## 2. Tính liên thuộc

Các thành viên có thể tin tưởng về khả năng thành viên hoàn thành công việc đúng chất lượng và thời gian. Các công việc có giá trị thúc đẩy nhau hướng tới mục tiêu chung. Trong các nhóm không có tính liên thuộc, ai làm việc của người ấy mà không quan tâm tới công việc của đồng đội và mục tiêu chung.

Trong Scrum, các hoạt động lập kế hoạch từ làm mìn Product Backlog, Lập kế hoạch Sprint, Scrum Hàng ngày đã làm rõ những tính liên thuộc của mỗi công việc và cá nhân. Các thành viên trong Nhóm Scrum làm việc dưới sự định hướng rằng họ cần cam kết kết quả chung. Họ được trao quyền, thấy ý nghĩa công việc của mình cho người dùng, tổ chức và các thành viên khác nên tính cam kết cao hơn. Họ cũng nhận được sự giúp đỡ của Nhóm Scrum vì họ chung mục tiêu chứ không phải mục tiêu của từng cá nhân và đặc biệt từ ScrumMaster, người có nhiệm vụ chính là tháo gỡ những khăk mà nhóm gặp phải.

## 3. Cấu trúc và sự rõ ràng

Sự hiểu biết của các thành viên về kỳ vọng công việc, quy trình hoàn thành các kỳ vọng và hệ quả từ hiệu năng của một người quan trọng với nhóm. Mục tiêu cho cá nhân hoặc nhóm cần phải rõ ràng, thử thách và có khả năng hoàn thành.

Như chúng ta thấy, trong Nhóm Scrum chúng ta có cấu trúc nhớ với ba vai trò Product Owner, ScrumMaster, Nhóm Phát triển và rõ ràng trong cách làm việc: Quản lý Product Backlog, Lập kế hoạch Sprint, Scrum Hàng ngày, Sơ kết Sprint, Cải tiến Sprint, thế nào xong với Định nghĩa Hoàn thành, Tiêu chí chấp nhận cho mỗi hạng mục của Product Backlog, Quy ước làm việc của Nhóm. Các Nhóm

Scrum thiết lập một hệ thống Mục tiêu Sprint có ý nghĩa, thảo luận về những mục tiêu ngắn hạn này cùng với các mục tiêu dài hơn được mô tả trong các kế hoạch phát hành (release plan) hoặc trong các mục tiêu OKR (Objective-Key Results).

## 4. Ý nghĩa công việc

Các thành viên thấy mục đích của chính bản thân công việc hoặc ở kết quả là rất quan trọng đối với mức độ hiệu quả của nhóm. Ý nghĩa này có thể khác biệt giữa các thành viên khác nhau ví dụ như: An toàn tài chính, hỗ trợ gia đình, giúp đỡ nhóm thành công hoặc để thể hiện mình.

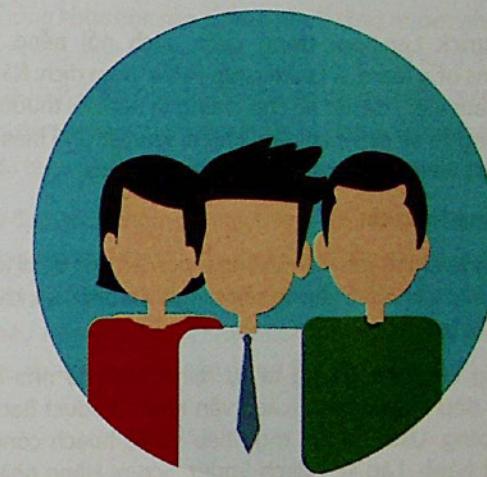
Yêu cầu sản phẩm thường được viết dưới dạng User Story là ngôn ngữ của người dùng. Do đó, các nhà phát triển cũng nhìn thấy ý nghĩa của một công việc mình làm.

## 5. Ảnh hưởng cá nhân

Cá nhân thấy rằng công việc của họ tạo ra sự khác biệt, và quan trọng đối với công việc chung.

Trong Scrum, cả nhóm cùng lập kế hoạch từ mức độ phát hành, Sprint, hàng ngày và có rất nhiều công cụ như Product Backlog, Biểu đồ Burndown, để mỗi thành viên đều nhìn thấy bức tranh toàn cảnh và giúp họ biết được ảnh hưởng công việc của họ với kết quả cuối của nhóm, sản phẩm và tổ chức.

Như vậy chúng ta thấy trong Scrum, nhóm được tổ chức tốt hoàn toàn hội tụ đủ các đặc trưng của một nhóm hiệu năng cao. Tuy nhiên, trong quá trình hoạt động, nhóm vẫn phải liên tục cải tiến để duy trì những yếu tố quan trọng đảm bảo hiệu năng cao này.



<sup>1</sup> <https://rework.withgoogle.com/print/guides/5721312655835136/>

# Năm rối loạn của nhóm

Liệu nhóm Scrum có tuyệt đối tránh được những rối loạn của một nhóm kém hiệu quả? Câu trả lời là không. Chúng ta có thể tưởng tượng Scrum như một chiếc gương, giúp ta nhận biết các vấn đề mình gặp phải, nhưng việc xử lý các vấn đề là phải tự chúng ta làm lấy. Tuy nhiên, Scrum không dừng lại ở việc phát lộ vấn đề (mặc dù đó cũng đã là một việc rất quan trọng), Scrum còn cung cấp cho chúng ta một số công cụ để chúng ta giải quyết hệ thống những vấn đề đó.

Tác giả Patrick Lencioni trong cuốn sách nổi tiếng "The Five Dysfunctions of a Team: A Leadership Fable" (tạm dịch: Năm rối loạn của nhóm làm việc) đã liệt kê cho chúng ta năm lỗi thường gặp của nhóm. Chúng ta sẽ điểm lại năm khiếm khuyết phổ biến này cùng với cách thức xử lý của Scrum để vượt qua chúng.

## 1. Thiếu vắng lòng tin

Điều này xảy ra khi thành viên không muốn trở nên dễ bị tổn thương trước thành viên khác và không sẵn sàng thừa nhận lỗi, khuyết điểm hoặc yêu cầu trợ giúp.

Scrum cung cấp cho chúng ta sự minh bạch từ nhu cầu người dùng, mục tiêu kinh doanh (cách vận hành Product Backlog, User Story Mapping, User Story), mục tiêu và kế hoạch công việc (Kế hoạch phát hành, Lập kế hoạch Sprint, Scrum Hàng ngày, Biểu đồ Burndown, Sprint Backlog), phương pháp làm việc (Cải tiến Sprint) nên khả năng tin tưởng giữa các thành viên trong nhóm trở nên dễ dàng hơn. Thông tin càng minh bạch, sự tin tưởng càng có cơ hội được nâng cao.

## 2. Sự xung đột

Nhóm này mong muốn giữ một sự an hòa giả tạo, ngăn cản sự xung đột. Nhóm thiếu vắng lòng tin, không có khả năng có những cuộc thảo luận sâu sắc để giải quyết những xung đột, vấn đề then chốt. Từ đó nhóm thay vì thảo luận thẳng thắn để giải quyết vấn đề lại thành ra những bình luận, bàn tán, chê bai.

Các cơ chế thanh tra và minh bạch như chúng ta đã phân tích thì các xung đột được phát lộ, nên dù từng thành viên có sợ hoặc muốn giấu đi thì vẫn phải đổi mặt, do đó chúng ta có thể giúp nhóm đổi mặt với nỗi sợ này. Với vai trò thành viên Nhóm Scrum, chúng ta cần làm phát lộ những xung đột bằng việc bạch hóa thông tin. Ngoài ra, ScrumMaster cũng phải giúp nhóm thiết lập được các cơ chế giải quyết xung đột hữu hiệu, để khi có những xung đột, chúng được xử lý một cách có hệ thống, tháo gỡ khó khăn và thúc đẩy nhóm tiến lên.

## 3. Thiếu cam kết

Sự thiếu rõ ràng ngăn cản thành viên đưa ra những quyết định mà họ sẽ tuân theo.

Trong Scrum, nhóm và cá nhân được trao quyền, được tôn trọng cũng như nhận thức rõ hằng ngày những ảnh hưởng công việc của họ với nhóm, sản phẩm, tổ chức. Cùng với đó là việc cam kết không còn quá khó khăn, những cam kết của họ có cơ sở là những việc họ đã làm trước đó (sử dụng vận tốc Sprint để cam kết cho Sprint, sử dụng ước lượng các công việc trước đó để đưa ra ước lượng, cam kết chỉ cho mỗi ngày) và đặc biệt họ có cả nhóm và ScrumMaster chuyên trách để hỗ trợ, giải quyết khó khăn. Chúng ta cần giúp đỡ từng cá nhân để việc cam kết trở nên dễ dàng hơn thông qua việc cung cấp dữ liệu cũng như phản hồi về cách họ đã làm. Cam kết cũng là một trong những giá trị cốt lõi của Scrum, bên cạnh Tập trung, Dũng cảm, Cởi mở và Tôn trọng.

## 4. Tránh nhận trách nhiệm giải trình

Nhu cầu né tránh sự khó chịu giữa các thành viên ngăn cản các thành viên giữ

trách nhiệm giải trình với nhau. Biểu hiện của việc né tránh trách nhiệm giải trình là các thành viên không cam kết với một kế hoạch rõ ràng và yêu cầu sự cam kết của các thành viên khác.

Scrum chia vai trò rõ ràng và cung cấp luồng làm việc rõ ràng để nhóm cũng như mỗi thành viên cam kết, giải trình (Sơ kết Sprint giải trình về sản phẩm, Cải tiến Sprint giải trình quy trình, Scrum Hàng ngày giải trình tiến độ công việc cá nhân và Nhóm Phát triển). Với vai trò Product Owner và ScrumMaster hãy đảm bảo Nhóm Scrum thực hiện nghiêm túc các sự kiện để gia tăng tránh nhiệm giải trình của mỗi cá nhân cũng như Nhóm Scrum, Nhóm Phát triển.

## 5. Không chú ý vào kết quả của nhóm

Các thành viên tập trung vào mục tiêu cá nhân và hình ảnh cá nhân làm cho nhóm mất sự tập trung vào thành công chung của nhóm. Biểu hiện của hiện tượng này là các thành viên đặt ưu tiên về nghề nghiệp, công việc cá nhân, sự ghi nhận cá nhân lên trên mục tiêu của nhóm.

Mỗi việc làm trong Nhóm Scrum đều đảm bảo phục vụ mục đích chung của nhóm và được thanh tra bởi tất cả các thành viên. Trong Nhóm Scrum chỉ còn nhà phát triển - không còn các vai trò khác, Product Backlog thể hiện đúng nhu cầu người dùng, có mục tiêu Sprint, mỗi công việc phục vụ yêu cầu sản phẩm, minh bạch tiến độ chung và luôn nhìn toàn cảnh của sản phẩm để gia tăng khả năng chú ý vào kết quả nhóm của mỗi thành viên.



# Văn hóa nhóm nhỏ

Một nguyên tắc quan trọng của Agile là thu nhỏ phạm vi công việc (descaling works) và trao quyền cho các nhóm nhỏ. Như chúng ta có thấy rằng ở Microsoft họ làm việc ở những nhóm 12 người. Jeff Bezos, CEO của Amazon tự hào về văn hóa nhóm nhỏ của công ty mình: "Không bao giờ có cuộc họp mà hai chiếc pizza không đủ để cả nhóm cùng ăn".

Có bằng chứng khoa học nào chứng minh rằng nhóm nhỏ là tốt cho hiệu năng? Theo một nghiên cứu về quan hệ của kích thước nhóm và hiệu quả thực hiện những nhiệm vụ phức hợp<sup>1</sup>, hiệu năng của nhóm tăng dần theo kích thước nhóm và đạt giá trị tối ưu ở 8 thành viên, sau đó hiệu năng giảm dần khi lượng thành viên tăng.

Scrum quy định một nhóm nên từ 3-9 người. Nếu dự án lớn cần phải huy động nguồn lực lớn hơn, thì dự án sẽ được chia thành phần nhỏ hơn được triển khai bởi một nhóm Scrum nhỏ và làm việc đồng bộ hóa với nhau trên cùng một Product Backlog chung. Chương 8 sẽ bàn kỹ hơn về cách thức để vận dụng Scrum ở quy mô lớn hơn.

<sup>1</sup> <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0153048>



## Dùng và Nghĩ

Dưới đây là một bài viết của Seth Godin, một chuyên gia hàng đầu về marketing và quản trị. Bạn hãy đọc chậm và liên hệ với nhóm của mình xem có điều gì hữu ích, điều gì thú vị.

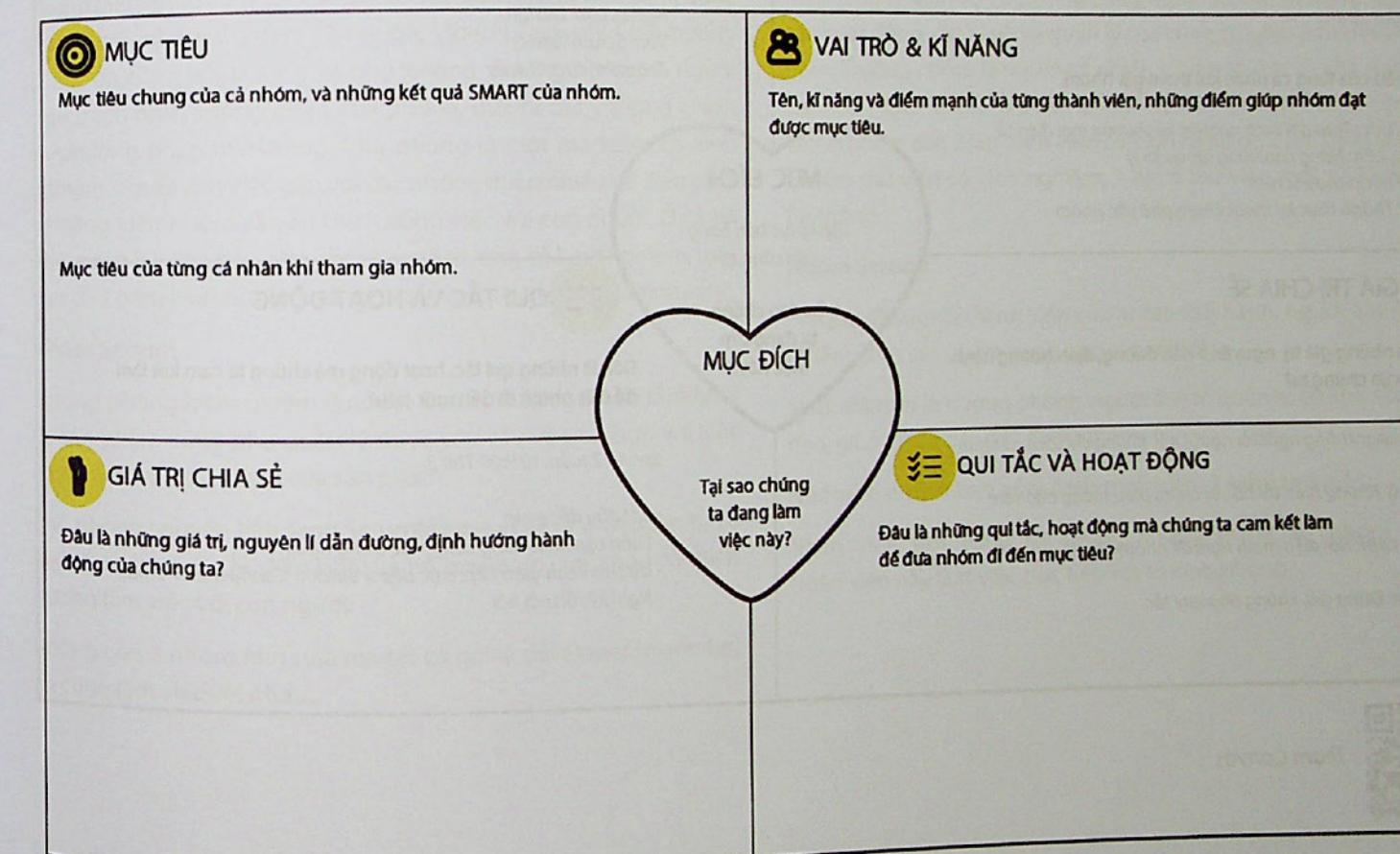
### TUYÊN NGÔN CHO NHỮNG NHÓM NHỎ ĐANG LÀM NHỮNG VIỆC QUAN TRỌNG

- Chúng ta luôn bị hạn-chót bám đuổi, bởi vì thời gian là tài sản quý giá nhất của chúng ta.
- Nếu bạn hứa, hãy đặt hạn-chót. Không có hạn-chót, không có lời hứa.
- Nếu bạn đặt một hạn-chót, hãy tuân thủ.
- Nếu bạn không thể兑现约会, hãy thông báo sớm và thường xuyên. Kế hoạch B được chuẩn bị tốt vẫn hơn là chỉ hy vọng.
- Đẹp mờ hồn độn của riêng bạn.
- Đẹp mờ hồn độn của người khác nữa.
- Giao tiếp, giao tiếp, giao tiếp!
- Hãy đặt câu hỏi đối với các chiến lược và giả định.
- Đừng đặt câu hỏi về thiện chí, nỗ lực hay ý định.
- Sẽ không chuyên nghiệp nếu nói câu này: "Tôi sẽ biết điều đó khi tôi nhìn thấy nó". Chúng ta mô tả và thảo luận, cả những thứ trừ tượng.
- Những dự án lớn hầu như không quan trọng bằng các cam kết đáng sợ.
- Nếu việc bạn đang làm bây giờ không quan trọng mấy đối với nhiệm vụ chung, hãy giúp đỡ người khác hoàn thành công việc.
- Hãy cứ phạm lỗi, đối mặt với chúng, giải quyết chúng, chia sẻ bài học với người khác.
- Phần mềm giá rẻ, đáng tin cậy có thể nhảm chán, nhưng thường là tốt hơn. Bởi vì chúng rẻ và đáng tin cậy.
- Hệ thống cấp bậc của ngày hôm qua là gần như không quan trọng như cơ cấu dự án hiện nay.
- Cắt đi những thứ phải cắt đi, hãy bỏ kế hoạch triển khai đầy đủ, chờ đợi đến khi bạn có thể tìm ra cách hoàn thành công việc.
- Chúng ta hầu như đang làm những việc chưa từng làm, do đó sẽ không có gì đáng ngạc nhiên nếu bạn đang ngạc nhiên.
- Quan tâm nhiều hơn.
- Nếu một người ngoài cuộc có thể làm cho nó nhanh hơn và hơn chúng ta, đừng ngần ngại.
- Luôn luôn tìm kiếm các nguồn lực bên ngoài. Một hộp danh thiếp (rolodex) tốt hơn luôn tốt hơn, ngay cả khi chúng ta không còn dùng hộp danh thiếp nữa.
- Nói chuyện với tất cả mọi người như thể họ là ông chủ của bối cảnh kinh doanh của bạn, người sáng lập hay nhân viên của bối cảnh kinh doanh của bạn. Như nhau cả thôi.
- Bạn có thể làm được những điều trên vì đó là chuyện hết sức cá nhân.

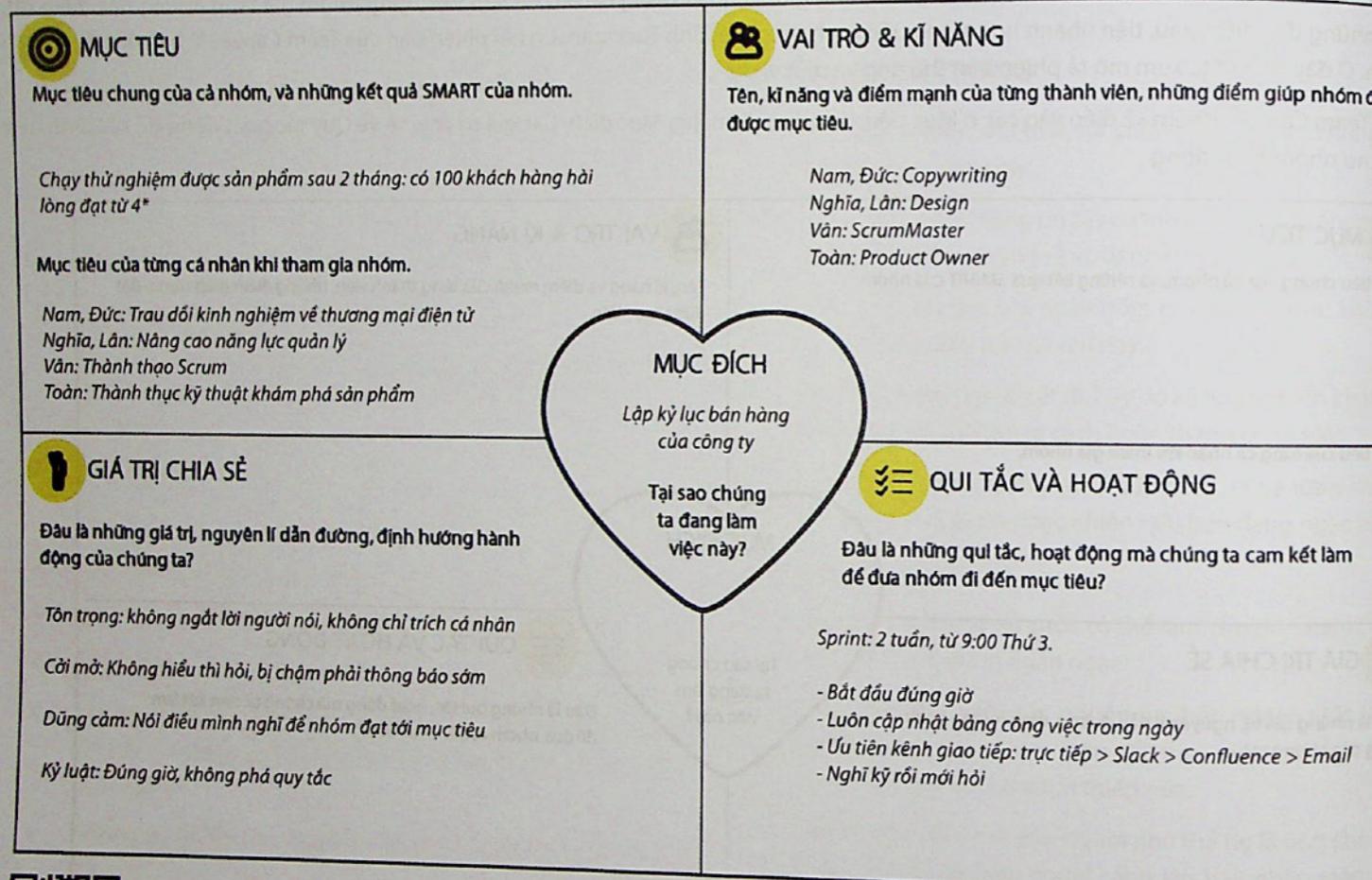
## TEAM CANVAS

Team Canvas<sup>1</sup> là công cụ tốt để nhóm tập trung vào những điều quan trọng cần có để làm việc, giúp nhóm gia tăng những đặc điểm tốt, giảm những đặc điểm xấu, tiến nhanh hơn trong tiến trình theo mô hình Tuckman. Có hai phiên bản của Team Canvas là bản đầy đủ và bản cơ bản. Ở đây chúng ta xem mô tả phiên bản thu gọn và một ví dụ.

Trong Team Canvas, nhóm sẽ điền vào các ô Mục tiêu, Vai trò và Kỹ năng, Mục đích, Các giá trị chia sẻ và Quy tắc hoạt động để xác định nền tảng cho nhóm hoạt động.



Trong ví dụ dưới đây, một nhóm marketing thiết lập một dự án xác lập kí lục kinh doanh mới cho công ty. Nhóm đã dùng Team Canvas để thảo luận về các điều thiết yếu nhất, để cùng nhau cộng tác đạt được mục tiêu.



Team Canvas

## Ví dụ xây dựng một nhóm Marketing

Phòng Marketing của một công ty có 10 người thành viên đang được chia thành nhóm Content (3 người), Video (3 người), Community (2 người) cùng với trưởng và phó phòng. Trưởng phòng là người chịu trách nhiệm tổng quan của phòng, đưa ra các ý tưởng chính và phương pháp marketing. Phó phòng là một marketer có kinh nghiệm, người làm việc gần với các nhóm, thiên nhiều về điều phối và động viên nhóm và yêu thích công việc với con người. Ở cả ba nhóm còn lại, đều bao gồm một bạn tổ trưởng có kinh nghiệm, một bạn có 2 năm kinh nghiệm và một bạn mới (trừ nhóm Community).

### Nhóm Scrum:

Trưởng phòng kiêm nhiệm Product Owner vì là người có tầm nhìn về sản phẩm cũng như quản lý được yêu cầu và phù hợp với việc chịu trách nhiệm chung của sản phẩm.

Phó phòng chuyển hẳn sang ScrumMaster vì đã là vị trí quản lý, có kinh nghiệm quản lý, điều phối và hỗ trợ nhóm cũng như có thiên hướng làm việc với con người.

Không còn 3 nhóm nhỏ nữa mà tất cả gọi là developer (marketer) và cùng làm việc với nhau.

## Ví dụ xây dựng một nhóm Phần mềm cho nội bộ

Một công ty dịch vụ có phòng công nghệ thông tin để tự động hóa các hoạt động vận hành và quản lý của công ty, gồm 8 thành viên bao gồm: Trưởng nhóm là người có nhiều kinh nghiệm và khả năng lãnh đạo tốt. Người đặt ra yêu cầu là một thành viên của nhóm vận hành. Trong nhóm còn 2 lập trình viên có kinh nghiệm, 2 lập trình viên mới, 1 kiểm thử viên có kinh nghiệm, 1 kiểm thử viên mới và 1 vận hành hệ thống.

### Nhóm Scrum:

Product Owner là thành viên của nhóm vận hành, người này hiểu rất rõ tầm nhìn sản phẩm cũng như các yêu cầu.

ScrumMaster là trưởng phòng, người ở vị trí quản lý, có khả năng lãnh đạo, giúp đỡ và dẫn dắt được nhóm.

Nhóm Phát triển bao gồm 4 lập trình viên, 2 kiểm thử viên.

Nhân viên vận hành hệ thống không tham gia nhóm Scrum. Nhân viên này làm việc trực tiếp với trưởng phòng.

## Ví dụ xây dựng một nhóm gia công Phần mềm

Trong một nhóm của một công ty gia công phần mềm có:

Quản trị dự án (PM) – là một nhân viên làm việc có tính cách khá cởi mở, giao tiếp tốt, có nghĩa vụ quản lý tiến độ chung của dự án.

Phiên dịch (Comtor) – bạn nữ có 1 năm kinh nghiệm với phần mềm, đảm nhiệm dịch các tài liệu từ tiếng Nhật sang tiếng Việt và ngược lại.

Kiểm thử viên (Senior tester) – giàu kinh nghiệm với hệ thống sản phẩm đang làm, hiểu rõ nhu cầu khách hàng.

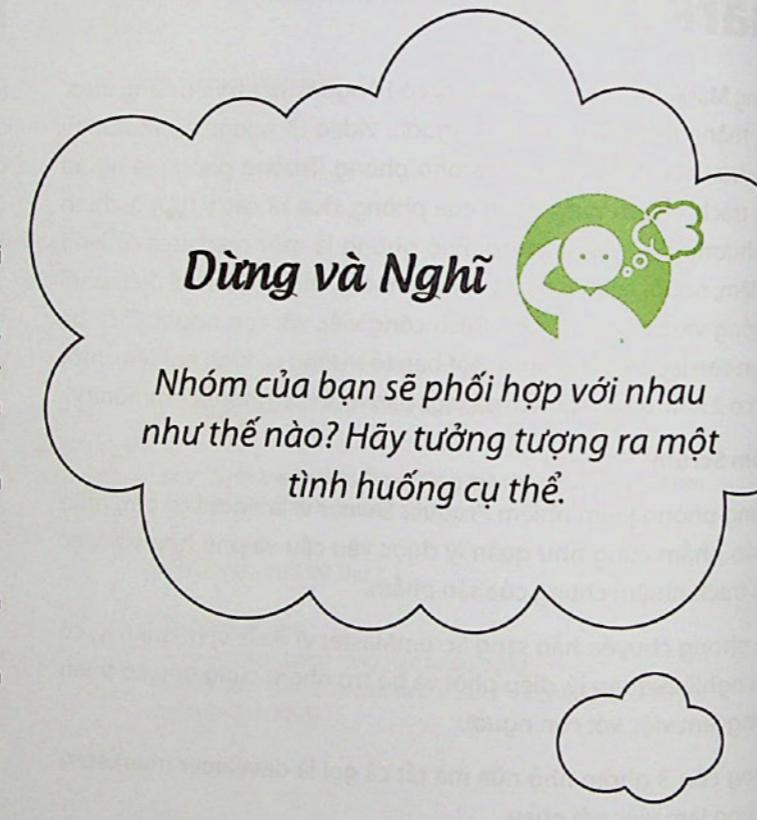
Nhóm có thêm 2 bạn kiểm thử viên mới và 4 bạn lập trình viên có tính cách khá vui nhưng ít nói.

### Nhóm Scrum:

PM chuyển hẳn sang vai trò ScrumMaster và trưởng đơn vị làm nhiệm vụ phân bổ nguồn lực và quản lý chung.

Senior Tester làm Product Owner vì là người nắm rõ nghiệp vụ sản phẩm và có khả năng làm rõ yêu cầu sản phẩm.

Các thành viên còn lại là các nhà phát triển.



## NHỮNG HIỂU LẦM THƯỜNG GẶP

### 1. Product Owner là Product Manager.

Tùy vào mô tả công việc của Product Manager ở từng doanh nghiệp, nhìn chung Product Manager có vai trò:

- Định vị sản phẩm so với các chiến lược của tổ chức, mô hình kinh doanh, đối thủ, v.v..
- Quản lý các mục tiêu và kế hoạch phát hành lớn và thậm chí cho từng khu vực địa lý, từng đối tượng khách hàng
- Một Product Manager thường quản lý nhiều sản phẩm

Như vậy Product Manager không phải một thành viên của Nhóm Scrum, không làm việc thường xuyên với Nhóm Phát triển.

### 2. Product Owner là Business Analyst (BA).

Tùy vào mô hình công ty mà hai vai trò có mục đích, công việc và kỹ năng khác nhau. Ở đây chúng ta so sánh dựa theo tiêu chuẩn Scrum Guide và PMI.

Product Owner và BA là hai vai trò khác nhau, có mục đích công việc cũng như bộ kỹ năng khá khác biệt mặc dù có một số điểm chung. Trong Nhóm Scrum không có vai trò BA. Chúng ta có thể so sánh hai vai trò này bằng bảng sau.

	Product Owner	Business Analyst
Mục đích công việc	Tối đa hóa giá trị của sản phẩm do Nhóm Phát triển làm ra	Giải pháp phù hợp với nhu cầu của kinh doanh
Công việc chính	<ul style="list-style-type: none"> <li>Tìm hiểu và phân tích kỹ về sản phẩm dự định phát triển</li> <li>Đánh giá giá trị và sắp xếp các yêu cầu sản phẩm</li> <li>Tối ưu hóa lợi nhuận trên vốn đầu tư (ROI)</li> <li>Minh bạch hóa Product Backlog cho Nhóm Phát triển và các bên liên quan khác</li> <li>Giải thích cho Nhóm Phát triển hiểu rõ các hạng mục của Product Backlog</li> <li>Theo dõi tiến độ phát triển sản phẩm, đảm bảo minh bạch và có thể giải trình với các bên liên quan khi có yêu cầu</li> </ul>	<ul style="list-style-type: none"> <li>Hiểu mục tiêu và vấn đề của doanh nghiệp</li> <li>Phân tích nhu cầu và giải pháp</li> <li>Tạo sự thay đổi</li> <li>Hỗ trợ sự cộng tác của các bên liên quan của sản phẩm</li> </ul>
Năng lực kỹ thuật mô tả yêu cầu	<ul style="list-style-type: none"> <li>Có cần, nhưng có thể không cần nhiều</li> </ul>	<ul style="list-style-type: none"> <li>Là năng lực chính</li> </ul>
Năng lực nghiệp vụ sản phẩm	<ul style="list-style-type: none"> <li>Là năng lực cốt lõi</li> </ul>	<ul style="list-style-type: none"> <li>Có cần, nhưng chỉ cần cơ bản</li> </ul>

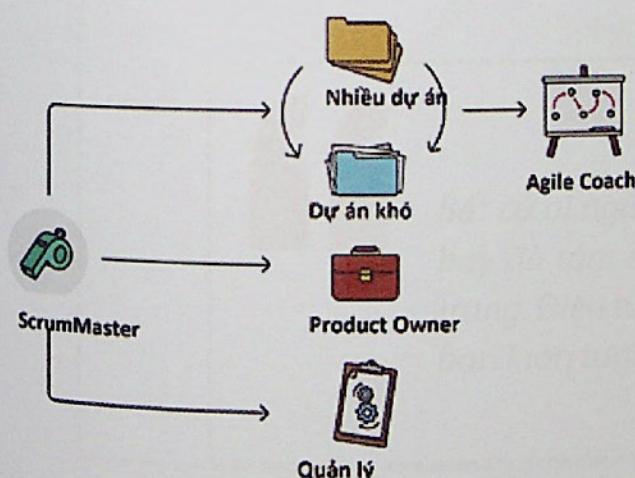
### 3. ScrumMaster là Project Manager

Project Manager là quản lý và chịu trách nhiệm toàn bộ về dự án và nhóm.

ScrumMaster không chịu trách nhiệm về dự án và không phải quản lý của nhóm. ScrumMaster giúp nhóm thành công nhờ vào việc đứng ngoài công việc cụ thể của nhóm để trợ giúp, huấn luyện và tháo gỡ khó khăn để Nhóm Phát triển và Product Owner làm việc hiệu quả nhất.

### 4. ScrumMaster chỉ là một công việc không phải một nghề, không có lộ trình công danh

ScrumMaster là một nghề toàn thời gian, một vị trí quản lý trong tổ chức, có lộ trình công danh. Thông thường một ScrumMaster có một số hướng phát triển khi không còn là ScrumMaster nữa hoặc vẫn là ScrumMaster và đôi khi đảm nhiệm những vị trí Agile Coach, CSM, CSPO, CSD, A-CSM, A-CSPO, A-CSD, CSP-SM, CSP-PO, CSP-D, CST, CEC, CTC.



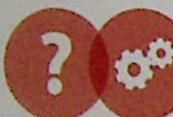
quản lý, chuyển qua Product Owner hoặc đơn giản là ScrumMaster của những dự án, nhóm khó, đôi khi là ScrumMaster chuyên đi xây dựng nhóm cho công ty.

ScrumAlliance, hiệp hội Scrum lớn nhất thế giới, cung cấp hệ thống gồm nhiều chứng chỉ cho mức độ và lộ trình nghề nghiệp khác nhau. Theo đó ScrumMaster có thể phát triển từ công việc của ScrumMaster, đến Scrum Professional, rồi hướng đến Scrum Coach hoặc Scrum Trainer.



### 5. ScrumMaster có thể kiêm Product Owner

Một người có thể kiêm nhiệm ScrumMaster và nhà phát triển, nhưng không thể kiêm nhiệm Product Owner. Hai vai trò này có mục tiêu công việc khác nhau (Phục vụ team và phục vụ sản phẩm), nên khi một người đóng hai vai trò ScrumMaster và Product Owner cùng lúc, thường họ sẽ trở về vai trò PM truyền thống, phá vỡ quy tắc của Scrum.



## CÂU HỎI ỨNG DỤNG

1. ScrumMaster có phải là người quản lý dự án hay không? Tại sao nói ScrumMaster là người lãnh đạo kiểu mới?
2. Cần có những tố chất gì để trở thành một ScrumMaster tốt?
3. Để làm tốt vai trò Product Owner bạn cần những kỹ năng nào?
4. Công việc chủ đạo của Product Owner với Product Backlog là gì?
5. ScrumMaster làm gì để giúp Nhóm Phát triển đạt được năng suất cao?
6. Tính liên chức năng và tự tổ chức giúp ích gì cho Nhóm Phát triển?
7. Nhóm Scrum sử dụng những công cụ chủ đạo nào để tăng cường sự minh bạch?
8. Nhóm của bạn có đủ tính chất để trở thành một Nhóm Scrum không? Cần làm gì để có thể bù đắp những tính chất còn thiếu (nếu có)?
9. Nếu áp dụng Scrum cho một nhóm của bạn, ai sẽ đóng vai trò ScrumMaster?
10. Quy tắc làm việc nhóm của bạn có những mục nào là quan trọng nhất?

# 4

## CÁC SỰ KIỆN SCRUM

Trong chương này...

- Sprint
- Lập kế hoạch Sprint
- Scrum Hàng ngày
- Sơ kết Sprint
- Cải tiến Sprint

“

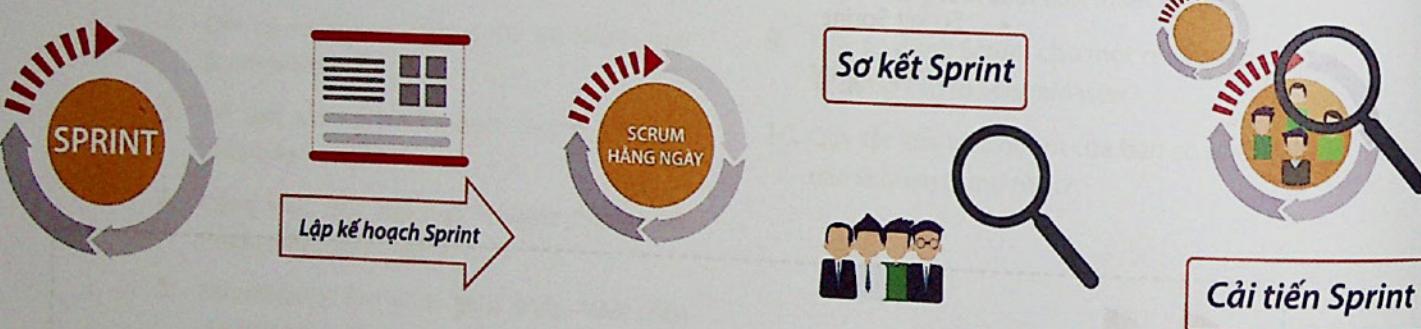
*Bất cứ ai ngừng học hỏi đều trở nên già cỗi, cho dù ở tuổi hai mươi hay đã tám mươi tuổi. Bất cứ ai không ngừng học hỏi vẫn mãi trẻ trung. Điều tuyệt vời nhất trong cuộc sống chính là giữ cho trí tuệ của bạn luôn tươi trẻ.*

*Henry Ford*

# SPRINT

Các sự kiện trong Scrum hình thành một thứ tự lặp đi lặp lại các công việc có chủ đích nhằm xây dựng một thói quen làm việc hiệu quả. Không gọi là các cuộc họp (dù chúng là các cuộc họp), các Sự kiện Scrum là cơ hội để cộng tác nhóm. Tất cả các Sự kiện Scrum

đều được đóng khung thời gian (time-boxed), nghĩa là mỗi sự kiện có giới hạn thời gian tối đa. Điều này đảm bảo thời lượng vừa đủ để tránh lãng phí thời gian không cần thiết cho sự kiện.



Sprint là phân đoạn mà mọi việc làm của Nhóm Scrum được diễn ra. Những công việc lập kế hoạch, thực thi, cộng tác, tháo gỡ khó khăn, tổng kết, cải tiến đều được diễn ra trong một khung thời gian có độ dài xác định để hoàn thành một mục tiêu cụ thể là chuyển giao giá trị ở cuối chu kỳ. Cơ chế thu nhỏ phạm vi công việc trong vòng một Sprint giúp Nhóm Scrum có khả năng liên tục chuyển giao giá trị, nhận về các phản hồi từ phía khách hàng (hoặc người dùng cuối) để điều chỉnh hướng đi cho thời gian tiếp theo. Về mặt nội bộ, việc lặp đi lặp lại một kỉ luật gồm nhiều việc quan trọng dưới dạng một tập hợp các sự kiện bắt buộc có cấu trúc rõ ràng giúp nhóm xây dựng, duy trì và cải tiến khả năng cộng tác nội bộ và với khách hàng, để ngày càng gia tăng sự hiệu quả. Độ dài của Sprint có thể kéo dài từ 1-4 tuần, tùy thuộc vào bối cảnh của dự án, đặc trưng của dự án, yêu cầu về thông tin phản hồi. Một dự án có nhiều biến động thì có xu hướng có Sprint ngắn hơn. Một dự án RnD thuộc dạng "khó

đoán" và không bị sức ép về tiến độ phải phát hành sản phẩm thì có thể lựa chọn độ dài lớn hơn. Nhưng cần lưu ý, một Sprint càng ngắn, thì áp lực thường lớn hơn, và công bô ra để "quản lý" (through qua các tương tác trong các sự kiện) thường chiếm tỷ lệ lớn hơn. Như chúng ta được biết trong Chương 1, độ dài phổ biến ở Microsoft là 3 tuần, với nhu cầu phát hành sản phẩm là từ vài tuần đến vài tháng. Nhiều nhóm chọn một chu kỳ 2 tuần làm việc để thiết lập độ dài Sprint và cảm thấy phù hợp với chu kỳ đó. Nhóm của bạn hoàn toàn có thể thí điểm xem dự án của mình phù hợp với kích thước Sprint như thế nào, rồi điều chỉnh. Nhưng sự điều chỉnh chỉ nên diễn ra rất ít lần, khi thật cần thiết.

Bao gồm cả bản thân Sprint vốn chứa tất cả các sự kiện khác, mỗi sự kiện trong Scrum là một cơ hội chính thức để thực hiện cơ chế thanh tra và thích nghi. Mỗi sự kiện đều có quy cách rõ ràng. Nếu không thực hiện được hoặc thực hiện không đúng các sự kiện này có thể dẫn đến giảm sự minh bạch và đánh mất cơ hội để thanh tra và thích nghi.

## Các sự kiện trong Scrum bao gồm:

1. Sprint
2. Lập kế hoạch Sprint
3. Scrum Hàng ngày
4. Sơ kết Sprint
5. Cải tiến Sprint

## Bảng tóm tắt các Sự kiện Scrum

Sự kiện	Mục đích	Thành phần (* là không bắt buộc)	Khung thời gian	Diễn biến
Sprint	Hiện thực hóa các hạng mục ưu tiên trong Product Backlog	Nhóm Scrum, người có liên quan*	1-4 tuần	Toàn bộ nhóm cộng tác với nhau, tự tổ chức để đặt ra mục tiêu Sprint, lập kế hoạch, thực thi, thích ứng, rà soát và rút ra bài học để cải tiến.
Lập kế hoạch Sprint, phần 1	Xác định việc cần làm (WHAT), lựa chọn hạng mục và làm rõ	Nhóm Scrum, người có liên quan*	4 giờ cho Sprint 4 tuần	Product Owner trình bày tầm nhìn về sản phẩm, mục tiêu Sprint và các hạng mục. Nhóm phát triển hỏi kĩ về các hạng mục cũng như mục tiêu.
Lập kế hoạch Sprint, phần 2	Xác định HOW, phân rã thành nhiệm vụ cụ thể và đưa lên Sprint Backlog	Nhóm Scrum, người có liên quan*	4 giờ cho Sprint 4 tuần	Nhóm Phát triển thảo luận, phân tích và dùng các kỹ thuật phù hợp để bênh công việc, làm ra Sprint Backlog cho Sprint. Có thể hỏi thêm Product Owner nếu phát sinh câu hỏi.
Scrum Hàng ngày	Thanh tra-Thích nghi liên tục	ScrumMaster, Nhóm Phát triển	15 phút	Mỗi thành viên cập nhật cho nhóm tiến độ công việc của mình, vấn đề gặp phải và dự kiến trong ngày tiếp theo.
Sơ kết Sprint	Rà soát kết quả làm được, nhận về phản hồi liên quan đến sản phẩm	Nhóm Scrum, người có liên quan*	4 giờ cho Sprint 4 tuần	Nhóm trình diễn sản phẩm, Product Owner phản hồi.
Cải tiến Sprint	Cải tiến liên tục cách làm việc để hiệu quả và vui vẻ hơn.	ScrumMaster, Nhóm Phát triển	3 giờ cho Sprint 4 tuần	Nhóm dùng các kỹ thuật cải tiến để thảo luận về kinh nghiệm vừa trải qua, những bài học rút ra và lập kế hoạch cải tiến cách làm cho Sprint tới.

### Chúng ta cần lưu ý trong suốt Sprint:

- Không cho phép có bất kỳ sự thay đổi nào ảnh hưởng đến Mục tiêu Sprint;
- Thành phần Nhóm Phát triển được giữ nguyên;
- Mục tiêu chất lượng không được cắt giảm;
- Phạm vi có thể được làm rõ và tái thương lượng giữa Product Owner và Nhóm Phát triển.

Mỗi Sprint có thể được coi như một tiểu dự án với độ dài tối đa là 1 tháng. Mỗi Sprint có một mục tiêu rõ ràng xác định việc phải xây dựng trong Sprint, một bản thiết kế và bản kế hoạch linh hoạt sẽ hướng dẫn quá trình xây dựng đó.

### Hủy một Sprint:

Sprint có thể bị hủy trước khi kết thúc khung thời gian. Chỉ có Product Owner mới đủ thẩm quyền dừng Sprint, mặc dù Product Owner có thể chịu ảnh hưởng bởi những bên hữu quan khác, bởi Nhóm Phát triển hoặc bởi ScrumMaster.

Một Sprint có thể bị hủy nếu như Mục tiêu Sprint không còn phù hợp nữa. Điều này xảy ra khi công ty chuyển hướng kinh doanh hoặc khi tình thế công nghệ có sự thay đổi. Nhìn chung, Sprint có thể bị hủy nếu nó không mang lại điều gì có ích. Thế nhưng, do thời gian mỗi Sprint tương đối ngắn, nên việc hủy một Sprint không mấy khi xảy ra.

Khi Sprint bị hủy, các phần sản phẩm đã hoàn chỉnh được xem xét lại. Nếu phần nào đó của công việc có thể chuyển giao được, thì Product Owner có thể chấp nhận chúng. Các hạng mục Product Backlog chưa hoàn tất sẽ được ước lượng lại và trả về Product Backlog để phát triển tiếp. Các phần việc đã thực hiện trên đó sẽ nhanh chóng hết tác dụng và phải thường xuyên được ước lượng lại.

Việc hủy Sprint sẽ gây lãng phí tài nguyên, do mọi người phải mất thời gian, công sức để lên kế hoạch cho một Sprint mới. Việc hủy Sprint thường gây tổn hại nhất định cho Nhóm Phát triển, và rất ít khi xảy ra.



Product Owner



# LẬP KẾ HOẠCH SPRINT

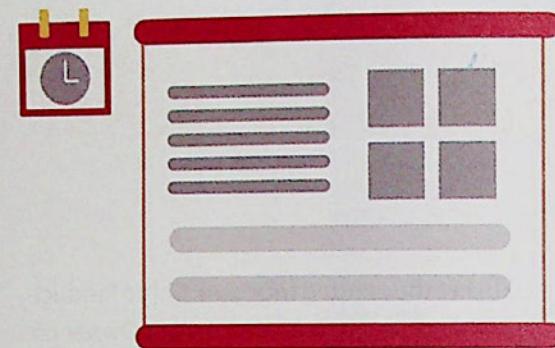
Một Sprint bắt đầu bằng buổi Lập kế hoạch Sprint để xác định Mục tiêu Sprint và lên kế hoạch các công việc cần thực hiện. Sự kiện này được chia làm hai phần: Phần thứ nhất để lựa chọn các công việc cần làm trong Sprint và Phần thứ hai để quyết định cách thức hoàn thành các công việc đã lựa chọn trước đó.

**Toàn bộ buổi Lập kế hoạch Sprint sẽ lần lượt trả lời các câu hỏi:**

- 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 hai Product Owner có thể vắng mặt nhưng phải luôn sẵn sàng để hỗ trợ Nhóm Phát triển làm rõ các hạng mục khi cần thiết, chẳng hạn thông qua điện thoại hay các hệ thống trao đổi trực tuyến.

Buổi Lập kế hoạch Sprint kéo dài trong 8 giờ đối với Sprint 1 tháng, đối với các Sprint ngắn hơn thì khung thời gian cho sự kiện có thể ngắn hơn. Ví dụ, đối với Sprint 2 tuần khung thời gian cho buổi Lập kế hoạch là khoảng 4 giờ. Thời gian dành cho hai phần của sự kiện này là bằng nhau, mỗi phần chiếm một nửa khung thời gian.



## Phần 1: Làm gì trong Sprint này?

Kết thúc Phần 1 của buổi Lập kế hoạch Sprint; Nhóm Scrum sẽ đưa ra câu trả lời cho câu hỏi: Chúng ta sẽ làm gì trong Sprint này? Cụ thể kết quả của phần này là Mục tiêu Sprint và danh sách các hạng mục Product Backlog được lựa chọn để phát triển trong Sprint.

Phần 1 được bắt đầu bằng việc Product Owner trình bày mục tiêu mong muốn đạt được trong Sprint này. Sau đó, Product Owner làm rõ thêm các hạng mục ở phần trên của Product Backlog (những hạng mục có độ ưu tiên cao nhất) để cả nhóm hiểu kỹ về hơn các hạng mục này. Product Owner cần làm rõ số lượng hạng mục Product Backlog nhiều hơn số lượng mà Nhóm Phát triển có thể hoàn thành trong cho một Sprint (con số này dựa vào các Sprint trước đó hoặc dựa trên kinh nghiệm). Ví dụ, nếu ước lượng thấy nhóm có khả năng làm khoảng 5 hạng mục thì ít nhất cả nhóm phải hiểu rõ khoảng 8 hạng mục hoặc hơn thế. Việc đó giúp nhóm có đầy đủ thông tin để đưa ra các quyết định lựa chọn một cách chính xác cho Sprint này.

### Năng lực của nhóm

Bạn cần phải tính được số lượng người làm việc và làm bao nhiêu giờ trong Sprint này. Vì có thể có ai đó sẽ nghỉ hoặc phải giảm thời lượng làm việc để phục vụ cho nhiệm vụ khác. Ngoài ra, bạn cũng cần phải tính thời gian “không năng suất” của mỗi ngày. Bởi thực chất một người chỉ có thể tập trung làm việc trong một thời gian nhất định trong ngày. Nếu thời gian làm việc mỗi ngày là 8 tiếng, thì thời gian “năng suất” – tức thời gian mà mỗi thành viên có

Thông thường, những hạng mục này đã được phân tích kỹ và làm rõ từ một vài Sprint trước thông qua việc làm mịn Product Backlog, do đó công việc trong phần này không tốn quá nhiều thời gian. Sau đó, cả nhóm sẽ hợp tác để tìm hiểu công việc phải làm trong Sprint.

Căn cứ vào Mục tiêu Sprint và năng lực hiện có, Nhóm Phát triển lựa chọn những hạng mục mà họ tin là có thể hoàn thành trong Sprint này, bắt đầu từ những hạng mục đứng đầu trong Product Backlog - nói cách khác, họ bắt đầu với những hạng mục có độ ưu tiên cao nhất do Product Owner 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 Product Owner. Điều này sẽ giúp Nhóm Phát triển đưa ra các 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.

thể dành để thực sự làm việc trên các hạng mục Product Backlog – có thể chỉ là 5 hoặc 6 giờ.

Để tránh được những ước lượng quá lạc quan (cũng có nghĩa là khả năng lớn sẽ phải làm thêm ngoài giờ hoặc không thể hoàn thành được mục tiêu đã định), bạn sẽ phải hết sức thực tế trong việc tính toán năng lực của cả nhóm trong Sprint.

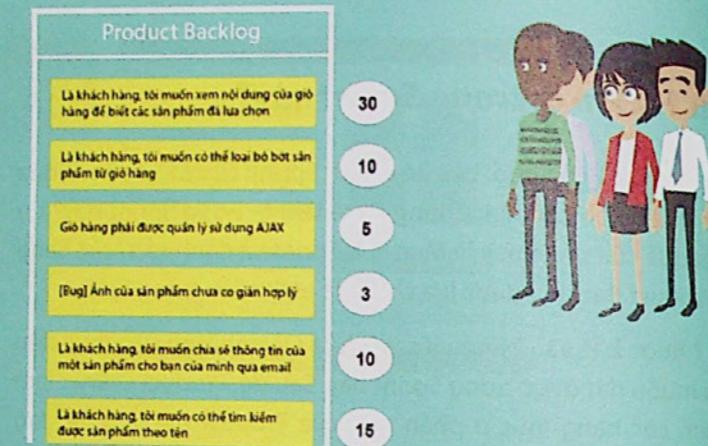
Để quyết định những hạng mục nào của Product Backlog sẽ triển khai trong Sprint này, Nhóm Phát triển sử dụng tốc độ sản xuất trung bình của mình trong quá khứ đồng thời tính khả năng của nhóm trong Sprint hiện tại.

Ví dụ, vận tốc trung bình của Nhóm Phát triển 7 người là 80 point trong một Sprint (nhóm này ước tính các hạng mục Product Backlog theo point - điểm). Trong Sprint hiện tại, có một thành viên xin nghỉ phép 3 ngày. Vậy, khả năng của nhóm trong Sprint này chỉ đạt được 74 point. Do đó, nhóm chỉ lựa chọn số lượng hạng mục đủ để làm trong khả năng 74 point của mình.

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 Product Owner. Điều này thường xảy ra khi có sự phụ thuộc giữa các hạng mục hoặc nhóm cảm thấy hạng mục có độ ưu tiên thấp đó rất phù hợp để phát triển cùng với các hạng mục khác đã lựa chọn.

Kết thúc Phần 1, Nhóm Phát triển và Product Owner thống nhất lại Mục tiêu Sprint và danh sách các hạng mục sẽ chuyển giao trong Sprint này. Mục tiêu Sprint là một đoạn mô tả ngắn về kết quả kỳ vọng đạt được sau khi Sprint kết thúc. Mục tiêu 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 giỏ hàng, hiển thị trang thanh toán."



#### Ví dụ: Lịch trình họp Lập kế hoạch Sprint cho một Sprint

- 8:00 – 8:30. Phiên họp bắt đầu. Product Owner trình bày mục tiêu của Sprint và tóm tắt các hạng mục Product Backlog dự kiến.
- 8:30 – 9:00. Nhóm Phát triển ước tính thời gian và tìm hiểu rõ các việc cần phải làm. Nhóm Phát triển và Product Owner cùng trao đổi về những điều nghi vấn về hạng mục Product Backlog (nếu có).
- 9:00 – 9:30. Nhóm chọn hạng mục để đưa vào Sprint. Thực hiện tính toán tốc độ để đảm bảo tính khả thi của Mục tiêu Sprint.
- 9:30 - 11:00. Chọn thời gian và địa điểm để họp Scrum Hàng ngày (nếu có thay đổi so với Sprint trước). Chia nhỏ hơn các hạng mục thành các đầu việc phải làm. Trong lúc thảo luận về chia nhỏ cách làm, nhóm có thể sẽ phải vẽ ra những phác thảo về thiết kế hệ thống, giao diện và những quy ước hợp tác nhất định trong viết mã lệnh.
- 11:00 - 11:15. Cả nhóm chốt lại kế hoạch của Sprint. Cập nhật các công việc lên Sprint Backlog và chuẩn bị vào Sprint làm việc. Phiên họp kết thúc.

#### Tốc độ (Velocity)

Tốc độ được tính bằng số lượng đơn vị được hoàn thành trong mỗi Sprint. Nếu bạn dùng đơn vị là điểm (point), thì tốc độ chính là số điểm mà nhóm hoàn thành được trong một Sprint. Qua thời gian, tốc độ của nhóm có thể sẽ tương đối ổn định. Đó là tiền đề quan trọng để nhóm có thể phỏng đoán khối lượng công việc của nhóm trong mỗi Sprint.

#### Dừng và Nghỉ



Vì sao việc đặt mục tiêu Sprint lại quan trọng? Nói rộng ra, tại sao việc đặt mục tiêu lại quan trọng?

**Để đặt mục tiêu và lập kế hoạch hiệu quả cho Sprint, Nhóm Scrum có thể sử dụng SMART Rubric như là công cụ hướng dẫn**

	4	3	2	1
<b>Cụ thể</b> <b>Specific</b>	Kết quả mong muốn rất rõ ràng	Kết quả mong muốn chỉ rõ ràng một phần	Kết quả mong muốn không rõ ràng	Không có kết quả mong muốn
<b>Đo được</b> <b>Measurable</b>	Sẽ đo được bởi vì những chỉ số được cung cấp sẽ chỉ rõ tiến độ	Chỉ đo được một phần bởi vì những chỉ số cung cấp không chỉ rõ toàn bộ tiến độ	Có cung cấp chỉ số, nhưng chúng không chỉ ra được tiến độ	Không đo được. Không chỉ ra cách để đo
<b>Khả thi</b> <b>Attainable</b>	Mình chắc chắn đạt được mục tiêu này với những khả năng hiện tại	Mình có thể đạt được mục tiêu này với những khả năng hiện tại	Mình hơi lo ngại với khả năng hiện tại. Có thể mình sẽ không đạt được	Mình hoàn toàn không thể làm được điều này
<b>Thực tế</b> <b>Realistic</b>	Mục tiêu này liên quan chặt chẽ đến định hướng của mình, nó phù hợp với khả năng và mong muốn của mình.	Mục tiêu này hơi liên quan đến định hướng của mình, nó gần phù hợp với khả năng và mong muốn của mình.	Mục tiêu này ít liên quan đến định hướng của mình. Nó ít phù hợp với khả năng và mong muốn của mình.	Mục tiêu này chẳng liên quan gì đến định hướng của mình. Nó cũng không phù hợp với khả năng và mong muốn của mình.
<b>Ràng buộc thời gian</b> <b>Time-bound</b>	Có một thời điểm hoàn thành rõ ràng nằm trong khoảng thời gian kiểm soát được	Không có một thời điểm hoàn thành rõ ràng nhưng vẫn nằm trong khoảng thời gian kiểm soát được	Không có một thời điểm hoàn thành rõ ràng và có một khoảng thời gian không hoàn toàn cụ thể	Không có bất cứ thông tin nào về thời gian

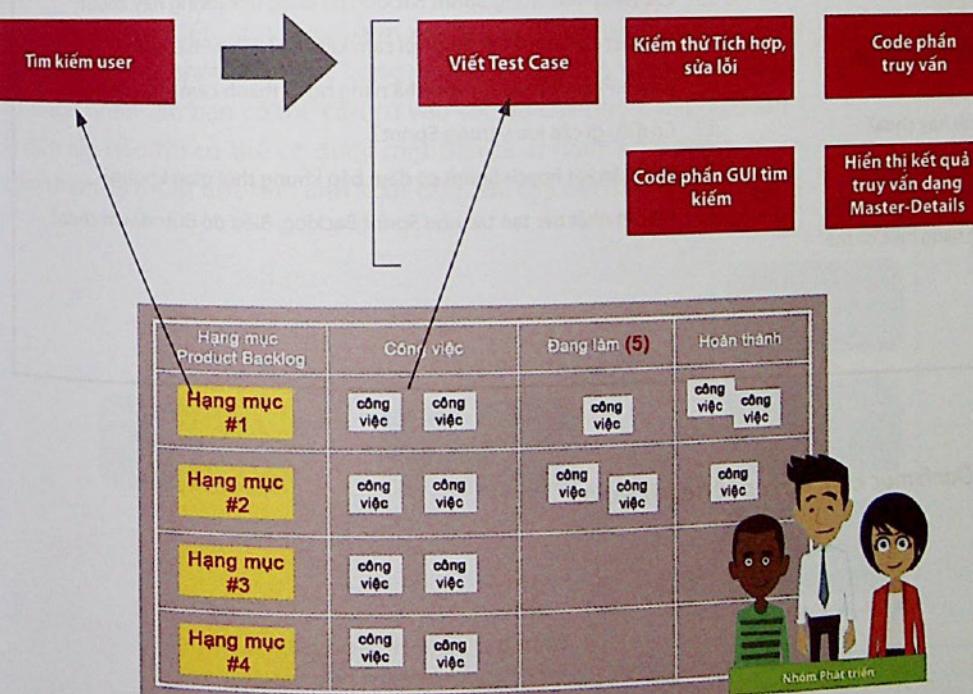


SMART Rubric

**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 bảng công việc được Nhóm Phát triển sử dụng trong suốt Sprint, bao gồm các hạng mục Product Backlog đã lựa chọn và danh sách công việc tương ứng với từng hạng mục Product Backlog 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 thường đủ nhỏ để hoàn thành trong vòng một ngày hoặc ít hơn.



Một số loại công việc thường thấy như: thiết kế, viết kiểm thử, viết mã, viết tài liệu, nghiên cứu kỹ thuật hoặc công nghệ mới, v.v.. Tất cả các công việc này đều phải được Nhóm Phát triển ước lượng nỗ lực để hoàn thành từng công việc. Thường các nỗ lực này được lượng hóa theo đơn vị giờ hoặc point và chúng được Nhóm Phát triển điều chỉnh thường xuyên trong quá trình triển khai. Một trong các kỹ thuật thường dùng để ước lượng là Planning Poker (xem Chương 9).

Tổng nỗ lực cho tất cả các hạng mục trên Sprint Backlog được nhóm sử dụng để theo dõi tiến độ Sprint. Giá trị này được cập nhật ngay vào Sprint Backlog và đồng thời cũng được sử dụng để tạo Biểu đồ Sprint Burndown nhằm theo dõi tiến độ Sprint. Sau mỗi ngày làm việc, các thành viên sẽ cập nhật đồng thời Sprint Backlog và Biểu đồ Burndown với các giá trị mới.

Nếu Nhóm Phát triển thấy rằng lượng công việc vừa lựa chọn là quá nhiều hay quá ít so với khả năng của nhóm họ có thể trao đổi và thương lượng với Product Owner để loại bỏ bớt hoặc chọn thêm các hạng mục khác. Trong Phần 2, Nhóm Phát triển có thể cần thêm sự hỗ trợ về mặt kỹ thuật từ bên ngoài nhóm để việc phân tích, ước lượng công việc tốt hơn.

### Danh mục kiểm tra Lập kế hoạch Sprint

- Buổi lập kế hoạch có diễn ra đúng giờ không?
- Có thành viên nào thiếu không? Lý do vì sao?
- Product Backlog có sẵn sàng trước buổi lập kế hoạch?
- Product Owner có sơ kết lại tình hình sản phẩm hiện tại đầu buổi lập kế hoạch?
- Product Owner có đưa ra mong muốn mục tiêu Sprint đầu buổi lập kế hoạch?
- Khả năng (capacity) của Nhóm Phát triển đã được tính hay chưa?
- Có những vấn đề, sự kiện nào đặc biệt ảnh hưởng tới khả năng của nhóm không (ví dụ nghỉ hè, thành viên nào có việc riêng)?
- Nhóm Phát triển và Product Owner có rà soát những hạng mục có thể sẽ phát triển trong Sprint?
- Những hạng mục có thể sẽ phát triển trong Sprint đã có tiêu chí chấp nhận?
- Nhóm Phát triển đã phân rã các hạng mục sẽ phát triển thành các công việc đưa vào Sprint Backlog?
- Các hạng mục trong Sprint Backlog đã được ước lượng hay chưa?
- Product Owner có hài lòng với cam kết của Nhóm Phát triển?
- Nhóm Phát triển có tin vào khả năng hoàn thành cam kết không?
- Có điều gì cần lưu ý trong Sprint?
- Buổi Lập kế hoạch Sprint có đảm bảo khung thời gian không?
- Đã cập nhật các tạo tác như Sprint Backlog, Biểu đồ Burndown chưa?

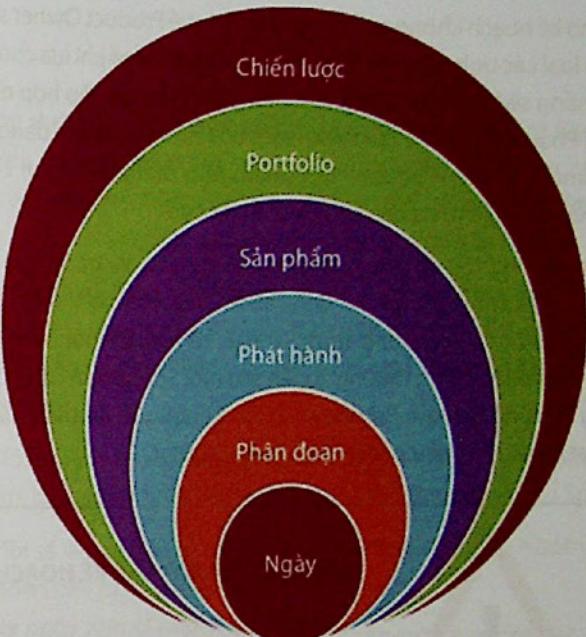


Danh mục kiểm tra Lập kế hoạch Sprint

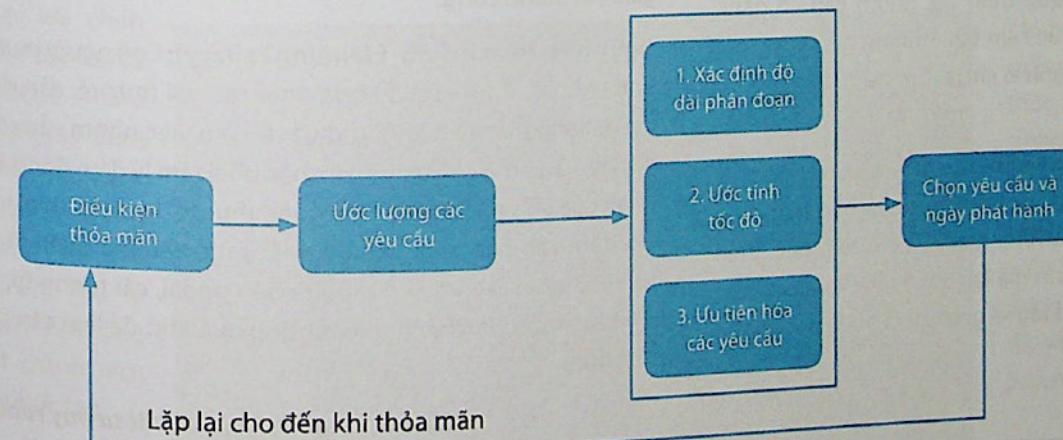
### Lập kế hoạch dài hơi hơn

Có thể bạn sẽ hỏi: Trong Scrum cơ chế lập kế hoạch theo Sprint quá ngắn, có đủ tốt cho việc hoạch định không? Câu trả lời rất rõ ràng: Không đủ. Việc lập kế hoạch có thể diễn ra ở nhiều cấp độ khác nhau. Lập kế hoạch Sprint dành cho Nhóm Phát triển, đây là một nét mới. Ngoài ra, các công tác lập kế hoạch dài hơi hơn, như Lập kế hoạch Phát hành, Lập kế hoạch Danh mục Sản phẩm (Portfolio) hay Chiến lược sản phẩm vẫn cần được thực thi hiệu quả ở các cấp cao hơn trong tổ chức.

Công việc lập kế hoạch cho sản phẩm thường được giao cho Product Owner. Kế hoạch này sẽ phải dựa nhiều vào việc ước tính kích thước các hạng mục Product Backlog, tính toán tốc độ hoàn thành công việc của Nhóm Phát triển, cũng như xem xét các ưu tiên kinh doanh đối với sản phẩm đang phát triển. Khi công việc ước tính cho từng hạng mục đã được hoàn tất, bạn có thể căn cứ vào tốc độ của nhóm mà dự đoán là đến khi nào thì có thể có được một bản phát hành có ý nghĩa. Hãy xem thêm kỹ thuật ước tính linh hoạt được đề cập đến trong Chương 9.



Các mức độ lập kế hoạch



Để cho kế hoạch không quá cứng nhắc, có thể Product Owner sẽ cần phân loại các tính năng **Phải có, Nên có và Có thể có** khi lựa chọn các tính năng sẽ hiện diện trong bản phát hành sắp tới. Tập hợp những phần Phải có để có thể tạo thành một bộ tính năng thuộc dạng MVP (Minimum Viable Product – sản phẩm hữu dụng tối thiểu) có thể chuyển giao giá trị cơ bản tới người dùng cuối.

Nên nhớ rằng tốc độ ước tính là con số dự báo, và trong quá trình hướng đến mục tiêu phát hành, nhóm của bạn có thể gặp phải những vấn đề không tính trước. Vì thế, kế hoạch phát hành cũng phải được hiệu chỉnh thường xuyên sau mỗi Sprint. Bạn cũng có thể dùng Biểu đồ Release Burndown (xem Chương 5) để theo dõi tiến độ đạt được mục tiêu phát hành qua các Sprint.



### NHỮNG CẠM BẪY LẬP KẾ HOẠCH

Lập kế hoạch luôn là một công việc khó khăn, đòi hỏi từng cá nhân và nhóm phải rèn luyện để ngày càng hiệu quả hơn. Trong quá trình đó, nhóm cần cố gắng tránh các cạm bẫy thường gặp được các nhà nghiên cứu về tâm lý học hành vi chỉ ra. Có thể kể đến một số như dưới đây.

**Hiệu ứng Tự tin thái quá (Overconfidence effect):** Con người có xu hướng đánh giá cao năng lực và kiến thức của bản thân hơn mức mình đang có (đặc biệt là nam giới). Việc ước lượng theo nhóm cùng với việc liên tục đánh giá lại, ước lượng lại trong các hoạt động Làm mìn Product Backlog sẽ giúp giảm thiểu các sai sót.

Giống như một công việc lập kế hoạch bất kỳ, việc hoạch định luôn bao gồm hai phần: Mục tiêu và Việc cần làm. Một số nhóm Scrum có thể dùng phương pháp SMART để xác định mục tiêu phát hành, một số nhóm khác dùng phương pháp OKR để thiết lập mục tiêu. Không giống như SMART, các mục tiêu OKR (viết tắt của Objectives – Mục tiêu, Key Results – các kết quả then chốt) hướng đến xác lập một mục tiêu có tính thử thách cao hơn, vốn rất phù hợp trong các nhóm khởi nghiệp, hoặc đang làm việc trong các dự án đòi hỏi tính sáng tạo cao độ. Hãy nghiên cứu thêm công cụ này để gia tăng độ hiệu quả trong lập kế hoạch dành cho Nhóm Scrum của bạn.

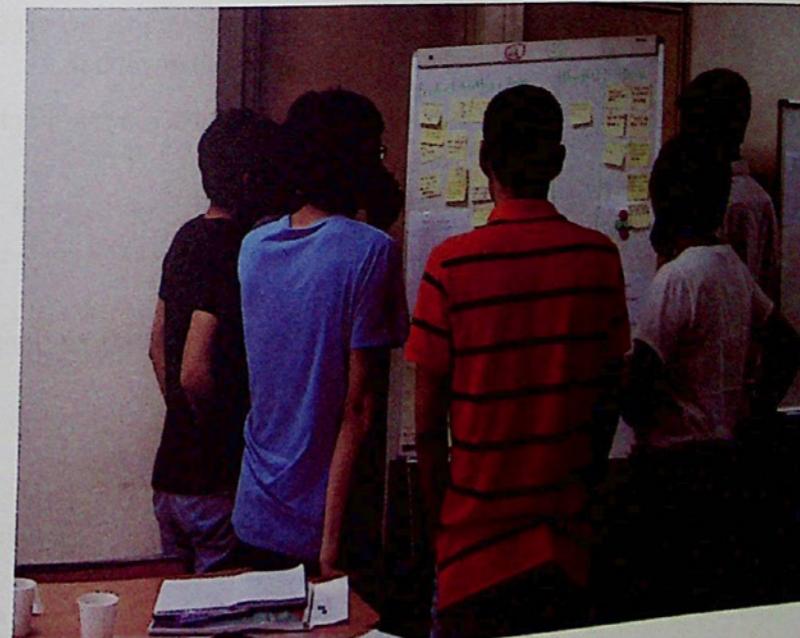
Theo R. Dobelli, *Nghệ thuật tư duy rành mạch*.

## SCRUM HẰNG NGÀY

Scrum Hằng ngày là một nghi thức quan trọng diễn ra đều đặn hằng ngày. Đây là một buổi trao đổi ngắn không kéo dài quá 15 phút với mục đích giúp Nhóm Phát triển đồng bộ công việc và lập kế hoạch cho ngày làm việc tiếp theo. Đây là một sự kiện dành cho Nhóm Phát triển, tất cả các thành viên của Nhóm Phát triển đều phải tham dự. ScrumMaster thường tham dự nhưng chỉ để đảm bảo Nhóm Phát triển thực hiện đúng công việc này chứ không trực tiếp tham gia điều hành hay phát biểu đóng góp vào nội dung của sự kiện. Product Owner và một số người khác có thể tham dự nhưng chỉ được lắng nghe và không đóng bất cứ vai trò nào trong sự kiện này.

Buổi Scrum Hằng ngày nên được diễn ra tại một địa điểm và khung thời gian cố định để giảm thiểu sự phức tạp. Thời điểm tổ chức được Nhóm Phát triển lựa chọn sao cho thuận lợi nhất đối với tất cả các thành viên. Thường thì các nhóm lựa chọn thời điểm Scrum Hằng ngày là đầu buổi sáng để khởi động cho một ngày làm việc mới.

Việc lựa chọn thời điểm tùy thuộc vào nhóm, điều quan trọng là đảm bảo sự kiện này luôn luôn diễn ra đúng thời điểm đã lựa chọn nhằm tạo ra thói quen và không biến nó thành một sự kiện phức tạp.



Thông thường, Nhóm Phát triển họp Scrum trước Sprint Backlog và Biểu đồ Burndown của mình để có thể cập nhật luôn những thay đổi vào Sprint Backlog cũng như có thể nắm được thông tin tổng thể về những việc cần phải làm còn lại trong Sprint.

Nội dung buổi Scrum Hằng ngày chỉ gói trong khoảng thời gian 15 phút, các thành viên Nhóm Phát triển thường sử dụng ba câu hỏi cẩn bản để giúp thanh tra và thích nghi trong sự kiện Scrum Hằng ngày:

- Tôi đã làm được những gì từ buổi Scrum Hằng ngày trước cho tới bây giờ để giúp Nhóm Phát triển đạt được Mục tiêu Sprint?
- Tôi sẽ làm những gì từ bây giờ tới hôm sau để giúp Nhóm Phát triển tiến tới Mục tiêu Sprint?
- Tôi đang gặp những khó khăn gì cần trả cả nhóm đạt được Mục tiêu Sprint này?

Mặc dù ba câu hỏi này là rất hữu ích, nhưng không bắt buộc. Nhóm Phát triển hoàn toàn có thể chọn phương thức khác để đạt được mục đích quan trọng nhất của mình là tái lập được kế hoạch hành động cho 24 giờ tiếp theo. Nhiều Nhóm Phát triển mắc lỗi khi thực hiện Scrum Hằng ngày và không thể đạt được mục tiêu của buổi họp này do sa đà vào thảo luận sâu trong suốt buổi Scrum Hằng ngày, đặc biệt khi phát

hiện ra vấn đề nào đó. Hãy đảm bảo mỗi thành viên chỉ trả lời ba câu hỏi trên, việc trao đổi giải quyết các vấn đề có thể thực hiện ngay sau khi Scrum Hằng ngày kết thúc. Việc này giúp đảm bảo thời lượng của buổi trao đổi không kéo dài quá 15 phút đồng thời giữ tập trung vào việc đồng bộ và phối hợp công việc giữa các thành viên thay vì giải quyết vấn đề.

### Nhiệm vụ của ScrumMaster trong Scrum Hằng ngày



Làm sao để cải thiện Scrum Hằng ngày?

Nhiệm vụ của ScrumMaster trong buổi Scrum Hằng ngày là đảm bảo điều kiện thuận lợi cho Nhóm Phát triển tiến hành sự kiện, ví dụ như địa điểm, đảm bảo nhóm giữ đúng khung thời gian tối đa 15 phút, ghi nhận các trở ngại được nhóm phát lộ, đảm bảo các thành viên tập trung vào đồng bộ và phối hợp công việc hiện tại thay vì sa đà vào các vấn đề khác.

Ngoài ra, ScrumMaster quan sát trong suốt cuộc họp để cảm nhận không khí và những vấn đề tiềm ẩn. Nếu các vấn đề được phát biểu, ScrumMaster ghi lại và có thể quản lý trong một Danh sách Trở ngại (Impediment Backlog) và giải quyết dần.

Để tháo gỡ khó khăn, đôi khi ScrumMaster phải vận dụng kỹ thuật hỏi 5 lần tại sao (5WHYS).

Các nhóm nên đứng trong họp Scrum Hằng ngày để tăng sự tập trung. Đó là lý do sự kiện này còn được gọi là Standup Meeting. Khi họp, các nhóm có thể dùng một vật trao tay, như một quả bóng hoặc một cái gậy để truyền tay nhau, như thể trong một cuộc chạy tiếp sức. Hãy sáng tạo hết sức để cuộc họp Scrum Hằng ngày của bạn thật vui vẻ và hiệu quả.

## 5WHYS (Hỏi năm lần “Tại sao”)



Đây là kỹ thuật có nguồn gốc từ phương thức sản xuất Toyota đã được dùng phổ biến trong hầu hết các lĩnh vực. Để điều tra nguyên nhân và các lựa chọn để giải quyết vấn đề, người điều tra sẽ hỏi 5 lần liên tiếp tại sao.

Dưới đây là ví dụ về một cuộc điều tra nguyên nhân của việc trả lại hàng:

Vấn đề: “Khách hàng phàn nàn về chất lượng lô hàng vừa giao”

Tại sao 1: Tại sao chất lượng lô hàng kém?

Trả lời: Do nguyên vật liệu đầu vào kém chất lượng.

Tại sao 2: Tại sao chất lượng nguyên vật liệu kém?

Trả lời: Do trong quá trình sản xuất bị thiếu nguyên vật liệu nên phải mua bổ sung từ nguồn khác.

Tại sao 3: Tại sao lại thiếu nguyên vật liệu?

Trả lời: Do ước tính không chính xác nên mua thiếu.

Tại sao 4: Tại sao lại ước tính nguyên vật liệu chưa chính xác?

Trả lời: Do không biết yêu cầu nguyên vật liệu của bản thiết kế mới.

Tại sao 5: Tại sao lại không biết yêu cầu nguyên vật liệu của thiết kế mới?

Trả lời: Tại vì chưa làm thử trước đó.

Thay vì quay ra đổ lỗi cho khách hàng ngay từ đầu, hoặc đổ lỗi cho đồng nghiệp, hoặc đổ lỗi cho bên giao hàng, chúng ta có thể nhìn ra vấn đề gốc rễ nằm ở quy trình sản xuất để khắc phục và tránh lặp lại.

Một ngộ nhận thường thấy là dùng 5WHYS luôn cho ra nguyên nhân gốc rễ (root cause), kỳ thực 5WHYS chỉ giúp ta tìm ra được những khả năng. Và từ đó ta có thể ra quyết định chính xác hơn.

### Những lỗi nên tránh khi Scrum Hằng ngày

1. Scrum Hằng ngày trở thành một buổi báo cáo công việc cho ScrumMaster
2. Scrum Hằng ngày biến thành nơi dành cho ScrumMaster cập nhật tiến độ công việc
3. Scrum Hằng ngày biến thành một phiên thảo luận để giải quyết vấn đề
4. Scrum Hằng ngày diễn ra quá xa nơi làm việc
5. Nhóm Phát triển không thấy giá trị của buổi Scrum Hằng ngày
6. Các thành viên thường báo cáo là không gặp vấn đề gì
7. Scrum Hằng ngày không diễn ra như là một thói quen, hôm làm hôm bỏ
8. ScrumMaster giao việc cho thành viên trong Scrum Hằng ngày
9. Mọi người nói xong là xong việc, không nghe người khác nói tiếp

# SƠ KẾT SPRINT



Buổi Sơ kết Sprint sẽ được tiến hành khi thời gian triển khai Sprint đã hết để kiểm tra phần tăng trưởng đạt được trong Sprint vừa qua. Đây là một hoạt động thanh tra và thích nghi đối với sản phẩm đang được xây dựng. Kết thúc buổi Sơ kết Sprint lộ trình sản phẩm và Product Backlog có thể được điều chỉnh để phù hợp hơn với tình hình phát triển mới.

Tham dự buổi Sơ kết Sprint có Nhóm Phát triển, Product Owner, ScrumMaster. Ngoài ra, Product Owner có thể mời thêm những người khác nếu phù hợp, chẳng hạn như khách hàng, người dùng, các bên liên quan, các lãnh đạo, và bất cứ ai quan tâm. Tất cả những người tham dự đều hoàn toàn tự do trong việc đưa ra các câu hỏi và đóng góp ý kiến của mình.

Khung thời gian của buổi Sơ kết Sprint là một giờ tương ứng với một tuần làm việc của Sprint. Có nghĩa là, đối với Sprint 4 tuần thì thời gian tối đa là 4 giờ, với Sprint 2 tuần là 2 giờ. Bạn cũng có thể điều chỉnh thời lượng họp theo nhu cầu thực tiễn của nhóm.

Một nội dung quan trọng của buổi Sơ kết Sprint là Nhóm Phát triển và Product Owner trao đổi với nhau để tìm hiểu về tình hình, ghi nhận các khuyến nghị của nhau. Đây là cơ hội để Product Owner tìm hiểu và nắm tình hình của sản phẩm và của Nhóm Phát triển. Còn đối với Nhóm Phát triển đây là cơ hội để tìm hiểu và nắm tình hình của Product Owner và của thị trường.

Nội dung của Sơ kết Sprint như sau:

1. Product Owner trình bày Mục tiêu Sprint
2. Nhóm Phát triển trình bày kết quả đã hoàn thành
3. Trực tiếp sử dụng sản phẩm



## Tiến trình một cuộc họp Sơ kết Sprint



### Một số lưu ý:

1. Không trình bày những tính năng chưa "hoàn thành"
2. Các phản hồi được đưa ra – Product Backlog có thể được đánh giá lại độ ưu tiên
3. Product Owner nên sử dụng kỹ thuật kiểm thử chấp nhận để đánh giá các tính năng.
4. Đây không là buổi demo sản phẩm. Trong thực tế, việc demo sản phẩm chỉ là một nội dung của buổi Sơ kết

Sprint. Nếu chỉ tập trung vào demo sản phẩm thì sẽ bỏ qua một nội dung quan trọng khác liên quan đến việc thảo luận và cộng tác giữa các thành viên tham gia. Từ đó gây hiểu nhầm và thực hiện sai mục đích thực sự của buổi Sơ kết Sprint đó là thanh tra và thích nghi sản phẩm đang được xây dựng.



Danh mục kiểm tra Sơ kết Sprint

- Buổi Sơ kết Sprint có thiếu thành viên nào trong Nhóm Scrum?
- Buổi Sơ kết Sprint có diễn ra đúng thời gian?
- Nhóm Scrum có trình bày Mục tiêu Sprint?
- Nhóm Phát triển chỉ trình bày những hạng mục đã hoàn thành?
- Product Owner và các bên liên quan có kiểm tra sản phẩm?
- Product Owner có đưa ra quyết định chấp nhận hoặc không chấp nhận phản hồi trưởng?
- Product Owner và các bên liên quan có đưa ra phản hồi về sản phẩm?
- Buổi Sơ kết Sprint có giữ đúng khung thời gian?



## CẢI TIẾN SPRINT

Cải tiến Sprint là sự kiện diễn ra ở cuối Sprint, ngay sau buổi Sơ kết Sprint và trước phiên Lập kế hoạch Sprint tiếp theo. Mục đích của sự kiện này là để cải thiện cách làm việc cho hiệu quả hơn và thú vị hơn sau mỗi Sprint.

Tài liệu hướng dẫn Scrum ghi: *"Mục đích của hoạt động Cải tiến Sprint (Sprint Retrospective) là để: 1. Thanh tra Sprint vừa rồi đã diễn ra như thế nào trên các phương diện con người, quan hệ, quy trình và công cụ; 2. Nhận diện trình tự và sắp xếp các thứ đã làm tốt, và những thứ cần phải cải tiến; 3. Lập một kế hoạch cải tiến cách làm việc của Nhóm Scrum".*

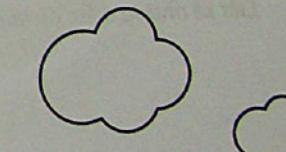
Nhóm Phát triển và ScrumMaster bắt buộc phải tham gia buổi Cải tiến Sprint. Product Owner có thể tham gia hoặc không. ScrumMaster cũng có thể mời những người khác cùng tham gia sự kiện này nhằm đóng góp ý kiến cho nhóm. ScrumMaster cố gắng hướng dẫn nhóm thực hiện đúng từng bước của nhóm trong hoạt động họp cải tiến, có được phương án cụ thể, và đưa ra một kế hoạch cải tiến thật SMART. Cần hết sức cụ thể, không tham lam khi cải tiến. Một ví dụ về cải tiến là: Thay đổi vị trí ngồi của nhân viên để có ánh sáng tối ưu và gia tăng giao tiếp mặt đối mặt (cải tiến về tương tác). Một cải tiến khác có thể là "Đổi trình tự việc A lên trước việc B" để hợp lý hơn (cải tiến về quy trình). Hoặc một ví dụ khác là "làm biểu mẫu trực tuyến cho các việc để ghi nhận các lỗi của sản phẩm nhằm giảm thiểu việc tiếp nhận trực tiếp mất thời gian" v.v..

Thời gian tối đa dành cho buổi Cải tiến Sprint là 3 giờ đối với Sprint 1 tháng. Đối với các Sprint ngắn hơn thì thời gian thường ngắn hơn, khoảng 45 phút tương ứng với một tuần làm việc của Sprint. Chẳng hạn, với Sprint 2 tuần thì khung thời gian là khoảng 1 giờ 30 phút.

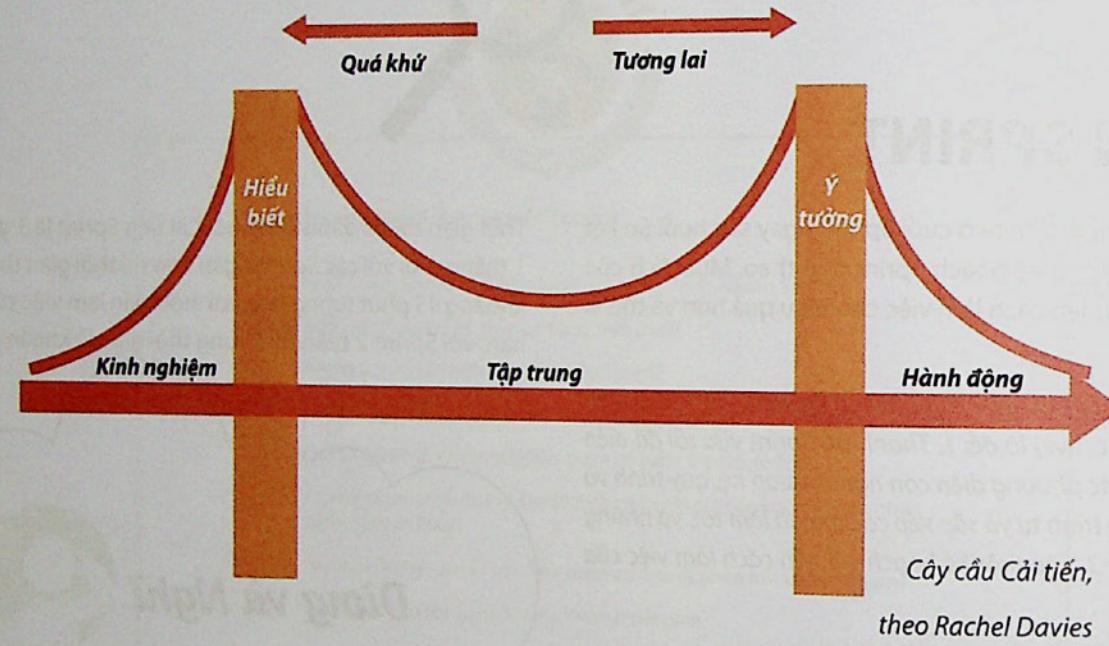
### Dừng và Nghỉ



Giả sử sau mỗi Sprint, nhóm của bạn cải thiện năng suất 3%. Hãy tính xem, cả năm nhóm bạn có thể tăng năng suất được bao nhiêu?



Dừng và nhìn lại, tìm kiếm các cải tiến và xây dựng tổ chức học tập.



Cần có một người giữ vai trò hỗ trợ cho buổi Cải tiến Sprint, người này có thể là ScrumMaster hoặc một người khác bên ngoài nhóm. Hoặc có một cách làm khá phổ biến đó là trao đổi chéo giữa các ScrumMaster, có nghĩa là ScrumMaster của nhóm này hỗ trợ Cải tiến Sprint cho nhóm khác.

Các hoạt động chính của buổi Cải tiến Sprint là:

- *Liệt kê những điểm đã làm tốt*
- *Liệt kê những điểm đã làm chưa tốt*

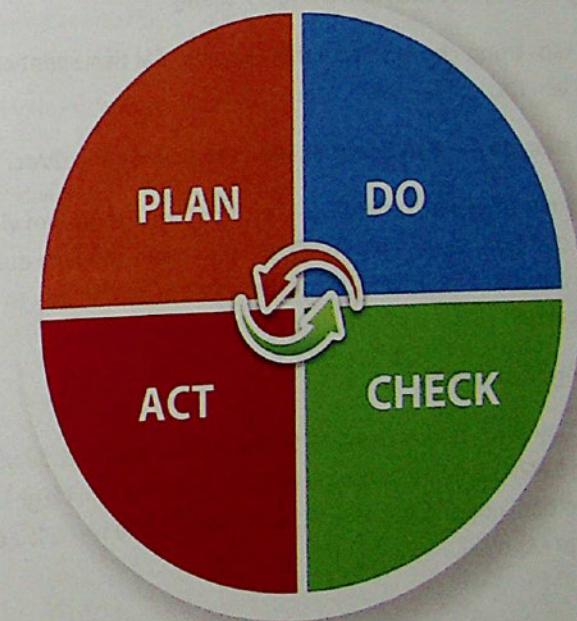
- *Đưa ra một vài hành động cải tiến*
- *Kế hoạch cải tiến cho Sprint sau*

Cách đơn giản nhất là cả nhóm cùng họp và bắt đầu với câu hỏi **“Chúng ta đã làm tốt những gì?”**, rồi trả lời câu hỏi tiếp theo **“Có điều gì cần loại bỏ?”** và **“Có điều gì có thể làm tốt hơn? Bằng cách nào?”**. Rồi sau đó cùng thống nhất một kế hoạch hành động cho Sprint sau.

Trong thực tiễn, các buổi Cải tiến Sprint nên được đặt vào một chu trình khép kín **Plan-Do-Check-Act** (Lập kế hoạch – Thực thi – Kiểm tra – Thích ứng) vốn là quy trình cải tiến liên tục (Kaizen) đã rất nổi tiếng. Theo đó, sau khi có được kế hoạch cải tiến, thì Nhóm Scrum cần đưa vào thực thi trong Sprint tiếp theo, sau đó theo dõi xem cải tiến đó có hiệu quả không, kiểm tra trong Sprint tới và thích ứng. Cho nên, một trong các trọng tâm của buổi họp cải tiến là phải kiểm tra lại xem kế hoạch cải tiến lần trước có hiệu quả không, và phải chỉnh sửa thế nào. Để thực hiện được việc này, Nhóm Scrum có thể duy trì cho mình một Bản ghi Cải tiến (Kaizen Log) để theo dõi mức độ hiệu quả của từng cải tiến qua thời gian. Tài liệu này ScrumMaster nên duy trì, cập nhật thường xuyên và có thể đóng vai trò như là đầu vào cho các cải tiến mang tính hệ thống hơn và nhân rộng ra các nhóm khác.

Trong một tổ chức có hệ quản trị tri thức hiệu quả, các cải tiến này có thể được phân loại và chia sẻ rộng rãi (dưới dạng wiki, blog hoặc tin tức nội bộ) để lan truyền sáng kiến ở quy mô lớn hơn.

Nhiều nhóm thực hiện việc cải tiến liên tục rất tùy hứng, không kiểm soát, không theo dõi, thì kết quả cải tiến thường không rõ ràng và ít giá trị. Hoạt động cải tiến liên tục cần phải trở thành thói quen của từng cá nhân và nhóm, và dần dà thành văn hóa của tổ chức thì sẽ đạt hiệu quả cao nhất.





## Những điều cần tránh khi họp Cải tiến Sprint

### 1. Họp Cải tiến Sprint ở quán bia cho thoải mái.

Thực ra nhóm cần ở một không gian riêng tư, tập trung và làm việc có đầu tư chất xám mới ra được kế hoạch cải tiến có chất lượng.

### 2. Thay đổi triệt để cách làm trong... 2 tháng.

Hãy tập trung cho Sprint tới thôi. Hãy nhớ Cải tiến Sprint cần phải liên tục, có hiệu quả tức thì.

### 3. Quá bận, bỏ Cải tiến Sprint đi, tuần sau ta làm cũng được.

Lưu ý rằng Cải tiến Sprint là để cho nhóm bớt bận đi, làm việc hiệu quả hơn, và vui hơn. Nhiều nhóm rơi vào vòng luẩn quẩn "quá bận để cải tiến", và không thể nào thoát ra. Cần duy trì kỷ luật cải tiến.

### 4. Không rõ ai làm gì.

Như vậy là kế hoạch chưa đủ SMART. Nhóm cần triệt để vận dụng 5W2H để lập kế hoạch cải tiến. Xác định rõ ai làm cái gì, khi nào, ở đâu, như thế nào, do đặc ra sao và mất bao nhiêu lâu để nhìn ra kết quả.

### 5. Người ngoài (bao gồm lãnh đạo) tham gia vào họp cải tiến.

Cần lưu ý, trong hoạt động cải tiến có đề cập cởi mở đến những vấn đề tế nhị về con người, các mối quan hệ và quy trình. ScrumMaster cần phải đảm bảo không khí an toàn cho buổi họp cải tiến, bao gồm việc không cho những người ngoài tham dự cuộc họp có phần "kín đáo" của Nhóm Scrum.

### 6. Không xem xét hiệu quả của các cải tiến lần trước.

Như vậy là vi phạm chu trình cải tiến liên tục PDCA. Cách làm này dễ dẫn đến nhóm sẽ từ bỏ cải tiến vì không đo đếm được hiệu quả thực sự.

### 7. Buổi họp Cải tiến Sprint thành phiên đồ lõi hoặc đấu tố.

Khi thảo luận, cần tập trung vào vấn đề, kể cả những vấn đề nhân, nhưng ScrumMaster cần phải giữ không khí của buổi làm việc không quá căng thẳng để biến thành buổi phàn nàn dài dằng dặc hoặc buổi cãi vã không hồi kết. ScrumMaster cần thiết lập các khung thời gian hợp lý cho mỗi phần của một buổi họp cải tiến (nên sử dụng các hình thức chặt chẽ như GladSadMad hoặc StopStartContinue, Lean Coffee để đạt được hiệu quả cao hơn).

## Danh mục kiểm tra Cải tiến Sprint

- Nhóm Phát triển có mời thêm thành viên khác tham gia?
- Có thành viên nào của Nhóm Phát triển thiếu không? Lý do vì sao?
- Địa điểm, các văn phòng phẩm đã được chuẩn bị?
- Buổi Cải tiến có diễn ra đúng giờ?
- Mọi thành viên đã rõ mục đích, quy tắc và cách thức thực hiện buổi Cải tiến?
- Có rà soát các hành động cải tiến ở những Sprint trước?
- Đã chuẩn bị các dữ liệu cần thiết cho việc cải tiến?
- Có thành viên nào không tham gia tích cực vào buổi làm việc? Tại sao?
- Có hành động cụ thể được tìm ra?
- Buổi Cải tiến Sprint có giữ đúng khung thời gian?

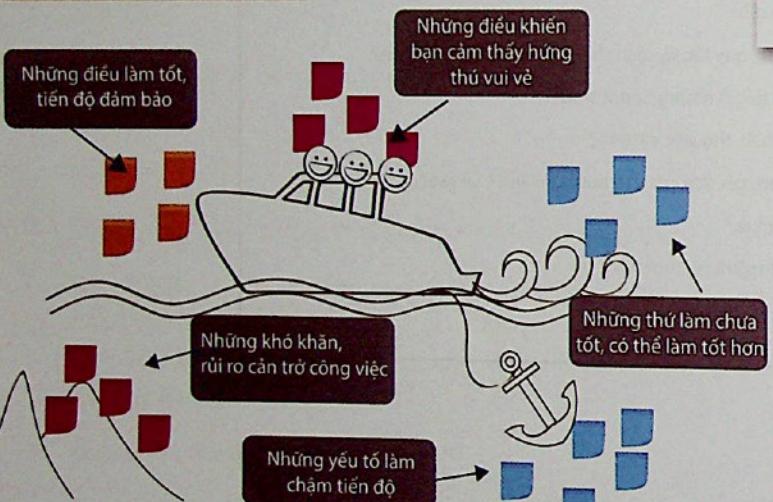


Danh mục kiểm tra Cải tiến Sprint

# Các kỹ thuật Cải tiến Sprint

Có rất nhiều kỹ thuật cụ thể để tiến hành buổi Cải tiến Sprint. Ví dụ như Glad-Sad-Mad, SpeedBoat, Start-Stop-Continue, 5whys, Lean Coffee, v.v.. Trong đó, kỹ thuật đơn giản và được sử dụng phổ biến là Glad-Sad-Mad. Nhóm của bạn hãy mạnh dạn sáng tạo các hình thức cải tiến khác để hoạt động này hiệu quả hơn, vui hơn.

## SpeedBoat



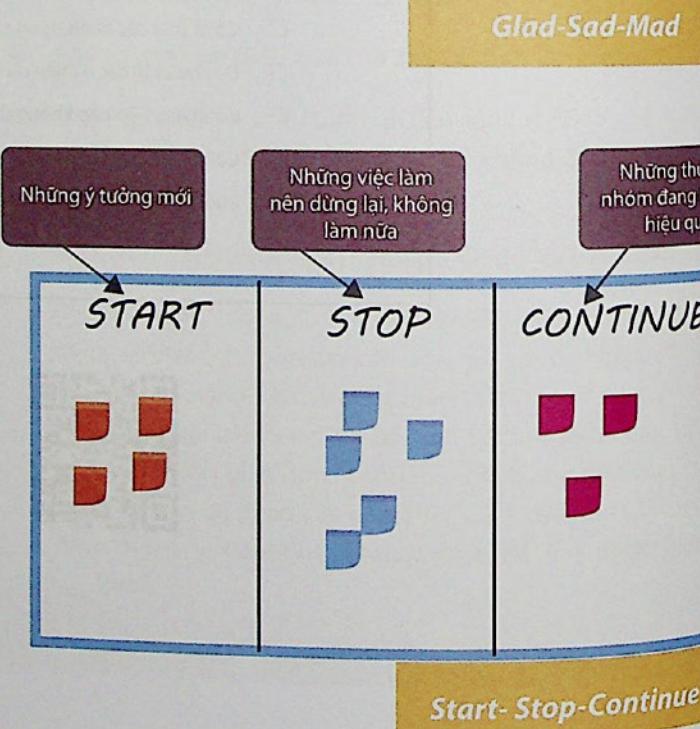
Các kỹ thuật Cải tiến Sprint



## Glad-Sad-Mad

# Glad-Sad-Mad

Glad-Sad-Mad là một kỹ thuật để thực hiện buổi cải tiến dựa trên việc phân loại các ý kiến của thành viên thành 3 nhóm: Glad (vui mừng), Sad (buồn) và Mad (tức giận). Những hạng mục nào mà thành viên đưa ra cảm thấy hài lòng thì được phân loại vào mục Glad. Những hạng mục nào mà họ chưa cảm thấy hài lòng và có thể cải tiến được thì đưa vào mục Sad. Những hạng mục nào mà gây cản trở nghiêm trọng và mình mong muốn loại bỏ nó ngay thì đưa vào mục Mad. Đây là một kỹ thuật được sử dụng phổ biến trong các buổi Cải tiến Sprint.



## Start- Stop-Continue

## Cách tiến hành cụ thể:

### 1. Chuẩn bị

Để bắt đầu thì cần chuẩn bị một tấm bảng (tờ giấy - cỡ A0) đủ lớn, chia thành 3 cột: Glad, Sad, Mad và đủ rộng để dán ý kiến của tất cả các thành viên.

### 2. Thu thập dữ liệu

Người hỗ trợ yêu cầu các thành viên ghi tất cả những quan sát của mình vào từng mảnh giấy dán. Mỗi ý kiến trên một mảnh riêng biệt, sau đó dán lên các cột tương ứng trên bảng. Mỗi ý kiến nên

được viết ngắn gọn, súc tích đủ để mọi người hiểu đúng về vấn đề. Hoạt động này cần sử dụng khung thời gian, chừng hạn 10 hoặc 15 phút.

### 3. Tổng hợp thông tin

Người hỗ trợ gom các ý kiến trùng nhau lại thành một (loại bỏ), sau đó các thành viên trong nhóm bỏ phiếu để

lựa chọn ra những hạng mục mà mình muốn thảo luận. Cách làm phổ biến thường là sử dụng "bỏ phiếu chấm", tức là mỗi một thành viên được lựa chọn tối đa 3 hạng mục bằng cách ghi một dấu chấm vào mỗi một tờ giấy dán mà mình lựa chọn. Những hạng mục nào có nhiều dấu chấm thì sẽ được thảo luận trước. Người hỗ trợ sẽ sắp xếp lại trật tự các tờ giấy dán dựa trên kết quả bỏ phiếu của cả nhóm.

### 4. 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 là đã hết thời gian đóng khung. Việc thảo luận nên tập trung vào việc đưa ra những hành động cải tiến có thể thực hiện được trong thời gian tới. Kể cả trong những hạng mục ở cột "Glad" thì cũng vẫn có thể cải tiến để nó tốt hơn.

# SpeedBoat

SpeedBoat là kỹ thuật để tiến hành buổi cải tiến dựa trên hình ảnh ẩn dụ của một con tàu cao tốc. Mục đích của kỹ thuật này là xác định độ hài lòng của khách hàng đối với dịch vụ mà mình cung cấp. Kỹ thuật này cũng được sử dụng phổ biến trong các buổi Cải tiến Sprint.

## Cách tiến hành cụ thể:

### 1. Chuẩn bị

Người hỗ trợ vẽ lên bảng (hoặc tờ giấy đủ rộng, cỡ A0) hình ảnh một chiếc tàu cao tốc đang lướt sóng, bổ sung thêm các hình ảnh hỗ trợ và cản trở chuyển động của con tàu như mỏ neo, sóng lớn, v.v..

### 2. Thu thập dữ liệu

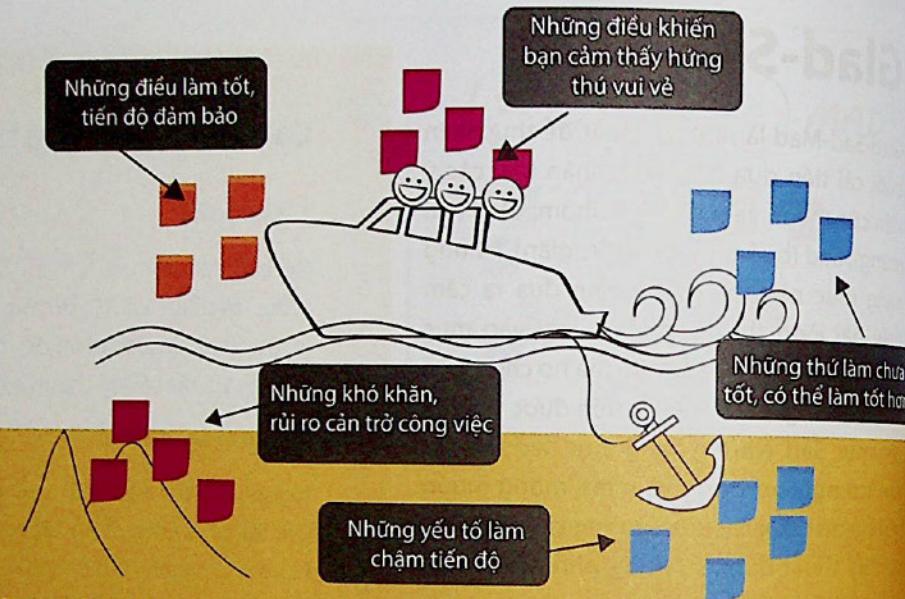
Yêu cầu tất cả mọi người cùng viết ra những ý kiến của mình về cả những yếu tố trở ngại và tích cực trong công việc (chẳng hạn trong Sprint vừa qua). Mỗi một ý kiến được viết lên một tờ giấy dán và lần lượt được dán vào các vùng thích hợp trên hình vẽ.

### 3. Tổng hợp thông tin

Người hỗ trợ yêu cầu các thành viên nhóm các ý kiến có liên quan lại với nhau. Sau đó, các thành viên sẽ bỏ phiếu để lựa chọn những hạng mục mà mình muốn thực hiện cải tiến. Có thể sử dụng hình thức "bỏ phiếu chấm", tức là mỗi thành viên được lựa chọn 3 hạng mục và mình muốn thảo luận bằng cách ghi một dấu chấm lên đấy. Những hạng mục nào có nhiều dấu chấm thì sẽ được lựa chọn để cải tiến.

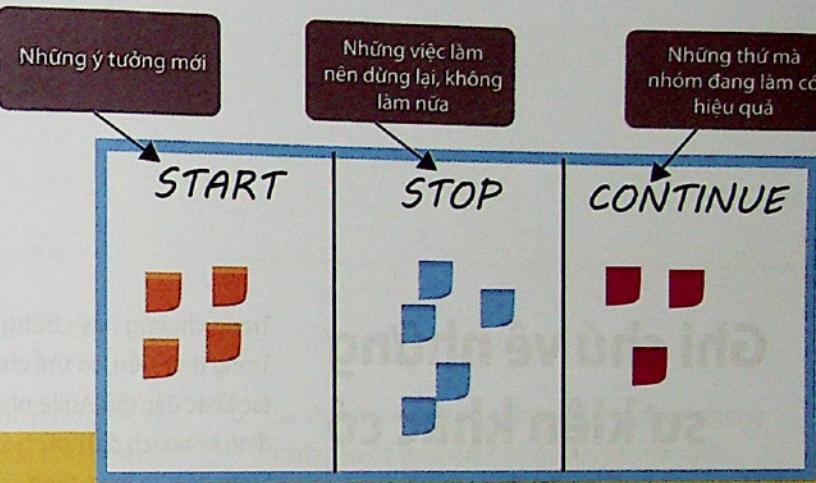
### 4. Thảo luận và đưa ra hành động cải tiến

Khi cả nhóm đã lựa chọn được các hạng mục cần cải tiến thảo luận để tìm ra nguyên nhân gốc rễ của vấn đề. Sau đó để xuất các thay đổi cần thiết thực hiện trong thời gian tới.



# Start-Stop- Continue

Một kỹ thuật khác cũng thường được sử dụng khi thực hành họp Cải tiến Sprint là Start-Stop-Continue.



## Cách tiến hành cụ thể:

### 1. Chuẩn bị

Để bắt đầu thì cần chuẩn bị một tấm bảng (tờ giấy - cỡ A0) đủ lớn, chia thành 3 cột: Start, Stop, Continue và đủ rộng để dán ý kiến của tất cả các thành viên.

### 2. Thu thập dữ liệu

Người hỗ trợ yêu cầu các thành viên ghi tất cả những quan sát của mình vào từng mảnh giấy dán. Mỗi ý kiến trên một mảnh riêng biệt, sau đó dán lên các cột tương ứng trên bảng. Mỗi ý kiến nên được viết ngắn gọn, súc tích đủ để mọi người hiểu đúng về vấn đề. Hoạt động này cần sử 15 phút.

### 3. Tổng hợp thông tin

Mỗi thành viên nên thuyết trình nhanh về từng ý kiến của mình trong thời gian ngắn. Sau đó cả nhóm bỏ phiếu hoặc thuyết phục nhau để lọc ra những ý tưởng tốt nhất, khả thi nhất, có giá trị cao nhất. Và chọn ra để đưa thành việc làm chung của nhóm.

### 4. Đưa ra hành động cải tiến

Nhóm thảo luận thật kỹ và đưa ra một kế hoạch cải tiến cho Sprint sau. Kế hoạch cải tiến cần phải khả thi, rõ ràng, cụ thể và đo lường được. Trong kế hoạch đó, cần làm rõ ai phải làm gì, ở đâu, khi nào, như thế nào và đo đếm ra sao.

## Ghi chú về những sự kiện khác có liên quan

Trong chương này chúng ta đã khảo sát những sự kiện chính yếu của Scrum. Trong thực tiễn, có thể chúng ta cần tổ chức thêm những phiên làm việc cộng tác khác đặc thù Agile như Lập kế hoạch Phát hành (Release Planning) để xác định kế hoạch dài hơi cho sự ra đời và phát triển sản phẩm; Workshop viết User Story để chuẩn bị Product Backlog và có thể kiêm thêm ước lượng các hạng mục; Họp Làm mịn Product Backlog để cập nhật các hạng mục và độ ưu tiên trong Product Backlog; Workshop Hình dung Kiến trúc tổng thể (Architecture Envisioning Workshop) để thống nhất chung cho toàn đội phát triển về kiến trúc tổng thể được sử dụng trong sản phẩm.

Ngoài ra cũng có thể có hàng loạt những sự kiện khác tùy nhu cầu như: Họp khởi công dự án, Huấn luyện trước dự án, Họp nghiệm thu và tổng kết dự án, Họp Kích não tạo ý tưởng (Brainstorming Session) v.v..

Nhóm của bạn nên tìm hiểu thêm và có một kế hoạch tổng thể cho những sự kiện như thế này để thúc đẩy thành công của dự án.



## CÂU HỎI ỨNG DỤNG

1. Khung thời gian cho buổi Lập kế hoạch Sprint với một nhóm triển khai Sprint 3 tuần là bao nhiêu?
2. Kết quả cần đạt được sau khi Lập kế hoạch Sprint kết thúc là gì? Nếu bạn là ScrumMaster, khi khung thời gian của sự kiện này đã qua mà nhóm chưa đạt được kết quả theo yêu cầu bạn sẽ làm gì? Tại sao?
3. Nhóm Phát triển và Product Owner cộng tác như thế nào trong Lập kế hoạch Sprint?
4. Bạn thử lý giải việc đóng khung thời gian cho Scrum Hằng ngày chỉ là 15 phút?
5. Nếu trong buổi Scrum Hằng ngày, các thành viên Nhóm Phát triển dành thời gian để giải quyết các vấn đề bạn sẽ làm gì?
6. Là thành viên Nhóm Phát triển bạn có những cách nào để theo dõi tiến độ của nhóm mình?
7. ScrumMaster cần có hành động gì khi thấy đường tiến độ của Nhóm Phát triển trên biểu đồ Sprint Burndown đi trên đường cơ sở?
8. Nhóm Phát triển làm những gì trong buổi Sơ kết Sprint?
9. Thành viên Nhóm Phát triển làm gì trong buổi Cải tiến Sprint?
10. Theo bạn, dấu hiệu của một Sprint thành công sẽ như thế nào?

# 5

## CÁC TẠO TÁC VÀ CÔNG CỤ

Trong chương này...

- Quản lý sản phẩm bằng Product Backlog
- Duy trì một Product Backlog tốt theo tiêu chuẩn DEEP
- Sử dụng Sprint Backlog để quản lý công việc
- Một số kiểu Sprint Backlog thông dụng
- Chuyển giao Phần tăng trưởng cuối mỗi Sprint
- Xây dựng Định nghĩa Hoàn thành để nâng cao chất lượng
- Sử dụng các biểu đồ Burndown để theo dõi tiến độ
- Minh bạch thông tin là điều bắt buộc

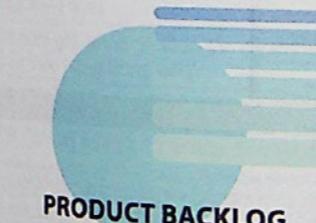
“

Hành trình vạn dặm bắt đầu với một bước chân nhỏ.

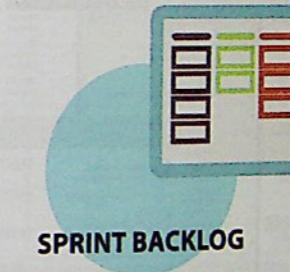
Lão Tử

### Tạo tác Scrum

Các tạo tác Scrum gồm một công cụ quản lý yêu cầu sản phẩm gọi là Product Backlog, một công cụ kế hoạch thích ứng trực quan cho nhóm cộng tác là Sprint Backlog và bản thân gói tăng trưởng chuyển giao được (Increment) ở cuối Sprint. Tạo tác (artifact) ở đây vừa là công cụ vừa là đồ được Nhóm Scrum tạo ra và duy trì liên tục trong quá trình làm việc. Các tạo tác cung cấp sự minh bạch, hỗ trợ quá trình thanh tra và thích nghi. Việc sử dụng và duy trì tốt các tạo tác này giúp cho tất cả các bên liên quan có được thông tin đầy đủ, hiểu đúng và có cùng một cách hiểu về tình trạng phát triển sản phẩm. Ngoài các tạo tác tiêu chuẩn của Scrum, các nhóm có thể dùng những công cụ khác để theo dõi tiến độ và thúc đẩy cộng tác.



PRODUCT BACKLOG



SPRINT BACKLOG



PHẦN TĂNG TRƯỞNG

- Danh sách được sắp xếp mô tả các yêu cầu của sản phẩm
- Có thể chỉnh sửa được
- Các hạng mục Product Backlog được lựa chọn cho Sprint và bản kế hoạch công việc chi tiết để hoàn thành các hạng mục đó
- Được cập nhật hằng ngày
- Tổng tất cả các hạng mục Product Backlog đã được hoàn thành trong một Sprint và giá trị của những phần tăng trưởng của những Sprint trước
- Được chuyển giao ở cuối mỗi Sprint

# PRODUCT BACKLOG

Product Backlog là danh sách các tính năng mong muốn của sản phẩm. Danh sách này được sắp xếp dựa trên đánh giá của Product Owner với từng hạng mục. Các hạng mục ở phía trên của danh sách được thảo luận kỹ với nhiều chi tiết cụ thể, chúng sẽ được Nhóm Phát triển lựa chọn để đưa vào sản xuất sớm, các hạng mục nằm ở phía cuối của danh sách thường được thảo luận ở mức vừa phải với một số đặc điểm chính và được lựa chọn phát triển muộn hơn.



## Ví dụ 1: Product Backlog dùng Spreadsheet

Các ước tính theo tiêu chí  
khác nhau (tùy thuộc nhóm)

Sprint

Hạng mục	Chi tiết (Wiki URL)	Độ ưu tiên	Giá trị Ước tính	Nỗ lực Ước tính	Tổng nỗ lực còn lại đến Sprint					
					1	2	3	4	5	6
Là khách hàng tôi muốn đưa một cuốn sách vào giỏ hàng (xem phác họa UI trong wiki)	...	1	7	5						
Là khách hàng, tôi muốn xóa một cuốn sách khỏi giỏ hàng	...	2	6	2						
Cải thiện hiệu năng xử lý giao dịch (xem yêu cầu các số đo hiệu năng trong wiki)	...	3	6	13						
Tìm kiếm giải pháp để tăng tốc độ kiểm tra thẻ ghi nợ (xem yêu cầu số đo hiệu năng trong wiki)	...	4	6	20						
Cập nhật tất cả các máy chủ lên Apache 2.2.3	...	5	5	13						
Phân tích và sửa lỗi xử lý hóa đơn (bugzilla ID 14823)	...	6	2	3						
Là một khách hàng, tôi muốn tạo một danh sách yêu thích	...	7	7	40						
Là một khách hàng, tôi muốn thêm các hạng mục trong danh sách	...	8	4	20						

Cột Hạng mục chứa các mô tả  
yêu cầu sơ bộ

Cột chi tiết được dùng để liên kết  
tới mô tả chi tiết của yêu cầu (bằng  
Wiki hoặc văn bản)

Phân cập nhật tiến độ hiện  
thực hóa các hạng mục qua  
các Sprint

## Ví dụ 2: Product Backlog dùng bảng vật lý

Story		Ràng buộc	Mô hình
Hạng mục đã sẵn sàng	1. Story 2. Story 3. Story 4. Story		
	5. Story 6. Story 7. Story 8. Story		
Theme A	Epic Epic Epic	Ràng buộc hiệu năng	
Theme B	Epic Epic	Ràng buộc Ôn định	
Theme C	Epic Epic Epic	Ràng buộc tương thích	

Cột Story chứa các mô tả ngắn gọn về yêu cầu sản phẩm. Các mục trong phần Hạng mục đã sẵn sàng đủ chi tiết và có thể đưa vào sản xuất. Các mục chưa chi tiết được xếp bên dưới theo phân loại nhóm (Theme) khác nhau.

Cột Ràng buộc lưu giữ các yêu cầu phi chức năng của hệ thống.

Cột Mô hình chứa các phác thảo thiết kế kỹ thuật của hệ thống.

## Ví dụ 3: Product Backlog sử dụng công cụ quản lý dự án

Scrum Học viện Agile Team Visible

Product Backlog

Là quản lý, tôi muốn có thể viết một bài blog mới

Là khách hàng, tôi muốn nhận được email thông báo về đơn hàng mới

Là quản lý, tôi muốn xem được thống kê đơn bán hàng theo ngày và tháng

Cập nhật Staging Server lên phiên bản 16.04

Tìm kiếm và lựa chọn thư viện Javascript để lưu trữ dữ liệu ở phía người dùng

Add a card...

Cần làm

Thông báo về đơn hàng

Là quản lý, tôi muốn xem tất cả các câu hỏi của khách hàng và gửi câu trả lời

[Bug] Ảnh của sản phẩm chưa có giãn hợp lý

Là khách hàng, tôi muốn kiểm tra được tình trạng đơn hàng bằng mã đơn hàng

Là khách hàng, tôi muốn xem được danh sách các sản phẩm phân loại theo nhóm

Là khách hàng, tôi muốn xem danh sách sản phẩm theo trật tự giá tăng dần hoặc giảm dần

Là quản lý, tôi muốn có thể lựa chọn một sản phẩm nổi bật để đưa lên đầu tiên

Là khách hàng, tôi muốn có thể tìm

Sprint Backlog

Là khách hàng, tôi muốn xem nội dung của giỏ hàng để biết các sản phẩm đã lựa chọn

Giỏ hàng phải được quản lý sử dụng AJAX

Là khách hàng, tôi muốn có thể bỏ sản phẩm từ giỏ hàng

Add a card...

Hoàn thành

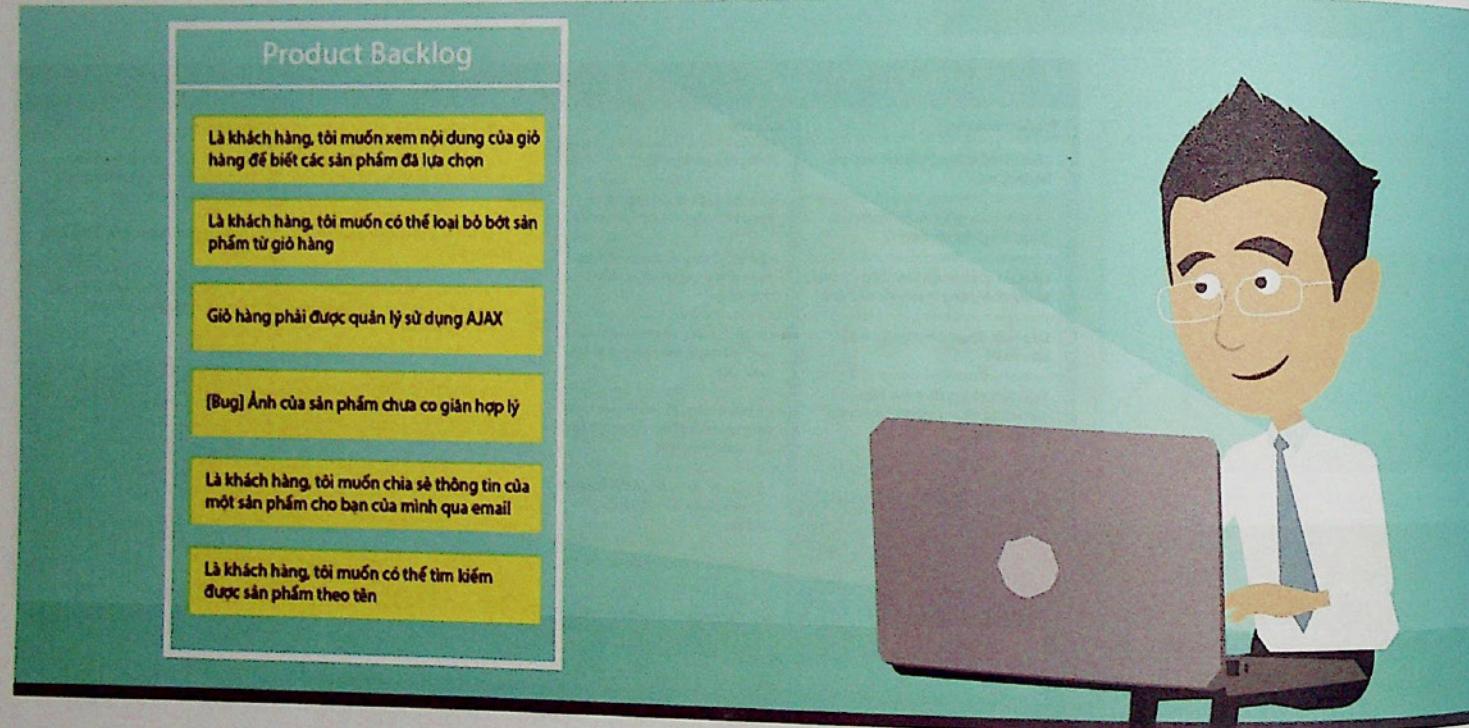
Là khách hàng, tôi muốn xem nội dung của giỏ hàng để biết các sản phẩm đã lựa chọn

Giỏ hàng phải được quản lý sử dụng AJAX

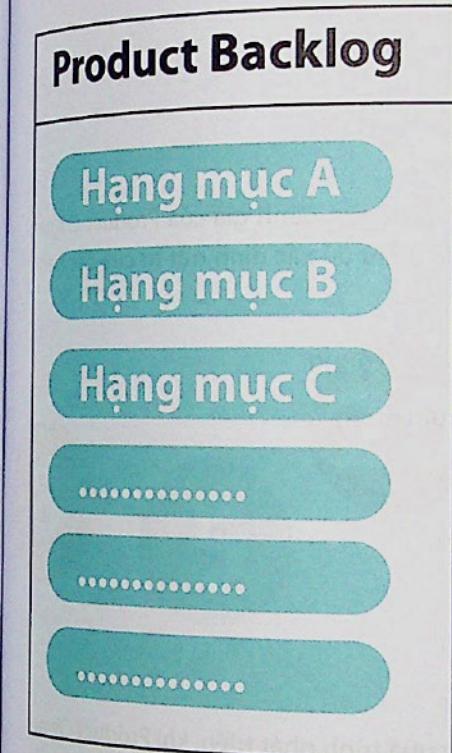
Là khách hàng, tôi muốn có thể bỏ sản phẩm từ giỏ hàng

Add a card...

Product Backlog này gồm các cột và chứa các trạng thái hiện thực hóa của từng hạng mục. Mỗi hạng mục được viết trên một thẻ. Thông tin của mỗi thẻ có thể sẽ rất chi tiết khi được mở ra.



Product Owner là người chịu trách nhiệm quản lý và bảo trì Product Backlog, bao gồm xác định nội dung, đánh giá và sắp xếp các hạng mục, làm mịn các hạng mục, làm rõ và giữ cho nội dung của Product Backlog luôn minh bạch với tất cả các bên.

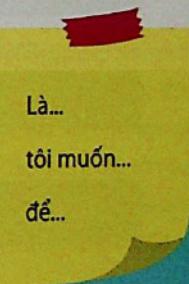


## Hạng mục Product Backlog

Product Backlog chứa các hạng mục Product Backlog. Các hạng mục này thường mô tả tính năng của sản phẩm, ngoài ra còn có thể có các hạng mục khác như lỗi, công việc liên quan đến kỹ thuật, công việc nghiên cứu.

Scrum không quy định hình thức cụ thể để thể hiện hạng mục Product Backlog, nhưng trong thực tế thì kỹ thuật được sử dụng phổ biến là User Story (xem chi tiết trong Chương 9).

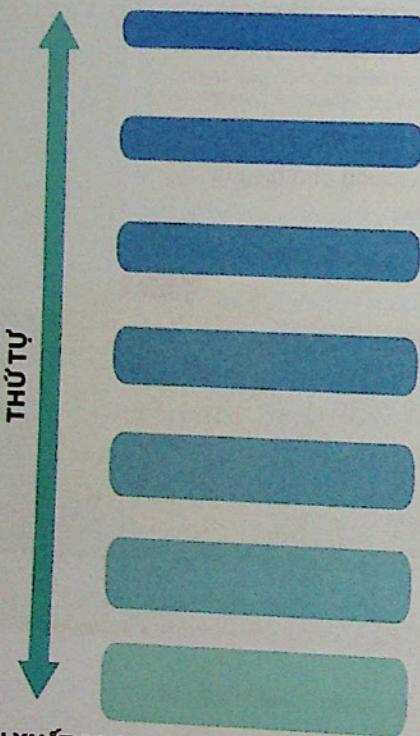
Scrum không quy định hình thức cụ thể để thể hiện hạng mục Product Backlog, nhưng trong thực tế thì kỹ thuật được sử dụng phổ biến là User Story.



Product Backlog template

# Đánh giá độ ưu tiên

SẢN XUẤT TRƯỚC



Các hạng mục trong Product Backlog được sắp xếp dựa trên đánh giá của Product Owner. Product Owner cần phải xem xét một số yếu tố khác nhau để xác định trật tự của các hạng mục, chẳng hạn như:

- Giá trị đối với phần lớn khách hàng hay người dùng
- Giá trị đối với những khách hàng hay người dùng quan trọng
- Chi phí
- Thời gian để triển khai
- Mối quan hệ với các hạng mục khác
- Rủi ro và cơ hội

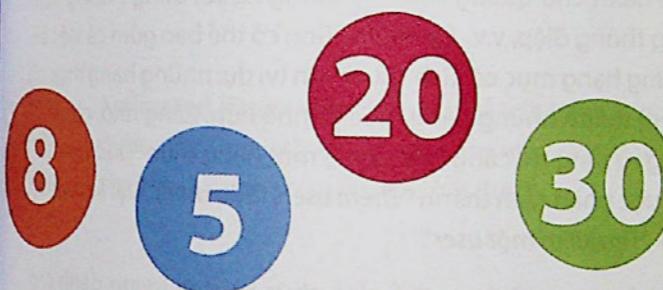
Trật tự của các hạng mục có thể được thay đổi trong quá trình phát triển, khi Product Owner có thêm những thông tin mới, hiểu biết nhiều hơn về sản phẩm và có thể có những điều chỉnh về lộ trình của sản phẩm.

# Ước tính các hạng mục Product Backlog

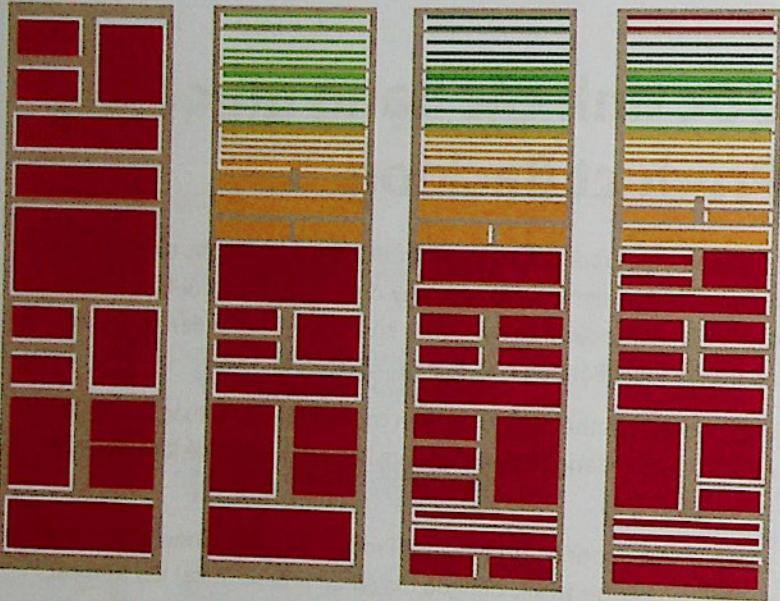
Mỗi hạng mục Product Backlog được ước tính kích thước, tức là lượng nỗ lực cần để triển khai nó. Việc này được thực hiện bởi Nhóm Phát triển. Kích thước của một hạng mục có thể ảnh hưởng đến thứ tự của nó trên Product Backlog.

Việc ước tính có thể được thực hiện theo nhiều cách khác nhau. Các nhóm Scrum thường sử dụng trò chơi Planning Poker để làm việc này (xem Chương 9).

Kết thúc việc ước tính các hạng mục Product Backlog, Nhóm Phát triển có thể gán các giá trị xác định độ lớn (ví dụ như số điểm) lên trên các hạng mục Product Backlog. Bạn cũng có thể gán kích thước giống như cỡ áo (ví dụ: XS, S, M, L, XL, XXL, XXXL) để đánh dấu độ lớn tương ứng của các hạng mục. Việc này sẽ giúp bạn quản lý Product Backlog thuận lợi hơn. Đặc biệt, nó sẽ giúp bạn làm mịn Product Backlog sau này.



Hướng dẫn Ước tính Linh hoạt với  
Planning Poker



## Làm mịn Product Backlog

Các hạng mục Product Backlog có kích thước khác nhau, mức độ chi tiết khác nhau và chúng thường cần phải được làm mịn trước khi sẵn sàng để đưa vào sản xuất. Những hạng mục ở phần trên thường “mịn” hơn so với những hạng mục ở dưới.

Việc làm mịn có thể đơn giản là chi tiết hóa một hạng mục nào đó còn chung chung. Ví dụ như hạng mục “Chạy quảng cáo Facebook” được bổ sung thêm các thông tin về thời gian bắt đầu và kết thúc, kinh phí dành cho quảng cáo, đối tượng người dùng hướng đến, nội dung thông điệp, v.v.. Làm mịn cũng có thể bao gồm cả việc bê nhô những hạng mục có kích thước lớn (ví dụ: những hạng mục cỡ XL trở lên) thành những hạng mục cỡ nhỏ hơn. Càng nhỏ, càng chi tiết, càng rõ ràng thì càng mịn. Ví dụ một hạng mục “Quản lý user” có thể được phân tách thành “Thêm user”, “Sửa user”, “Liệt kê toàn bộ user” và “Tim kiếm một user”.

Scrum không quy định cụ thể cách thức và thời lượng dành cho hoạt động làm mịn Product Backlog. Tuy nhiên, khuyến cáo rằng các nhóm không nên dành quá 10% thời gian của mỗi Sprint cho hoạt động này. Hoạt động làm mịn Product Backlog đảm bảo cho các phiên Lập kế hoạch Sprint tiếp theo được thuận lợi, cũng như luôn cập nhật trạng thái hiểu biết tốt nhất về yêu cầu sản phẩm. Phiên họp làm mịn Product Backlog này diễn ra trong Sprint, có thể là theo nhu cầu, hoặc cố định, tùy sắp đặt của nhóm.

## Tiêu chuẩn DEEP dành cho Product Backlog

DEEP là một tiêu chuẩn được sử dụng khá phổ biến hướng đến việc xây dựng một Product Backlog tốt.

DEEP quy định 4 tính chất mà một Product Backlog nên có, bao gồm:

**D** **Detailed appropriately (Đủ chi tiết):** Những hạng mục ở phía trên của Product Backlog thì có nhiều chi tiết cụ thể hơn, rõ ràng hơn so với những hạng mục ở phía dưới.

**E** **Estimated (Được ước tính):** Tất cả các hạng mục Product Backlog phải được ước tính. Độ chính xác của việc ước tính các hạng mục ở trên thường là cao hơn so với phần ở dưới.

**E** **Emergent (Tiến hóa):** Product Backlog không phải là một danh sách cố định các hạng mục cần phát triển mà nó luôn được tiến hóa trong suốt quá trình phát triển dựa theo những hiểu biết học được.

**P** **Prioritized (Được sắp xếp):** Các hạng mục trong Product Backlog được sắp xếp theo độ ưu tiên. Những hạng mục nào cần đưa vào sản xuất sớm sẽ nằm phía trên của Product Backlog.

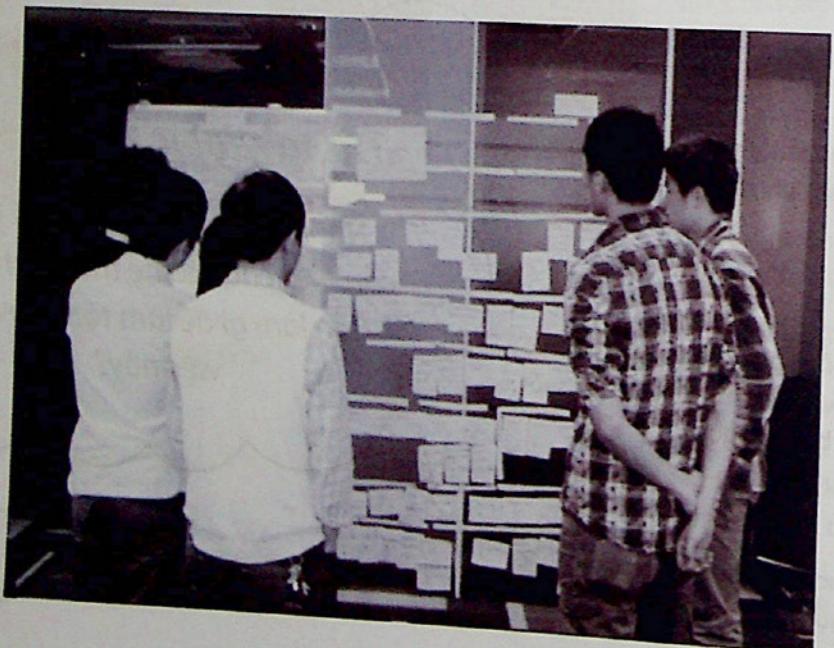
### Dùng và Nghĩ

Nhóm của bạn có quản lý yêu cầu một cách có hệ thống không?  
Cần làm gì để làm tốt hơn nữa việc này?



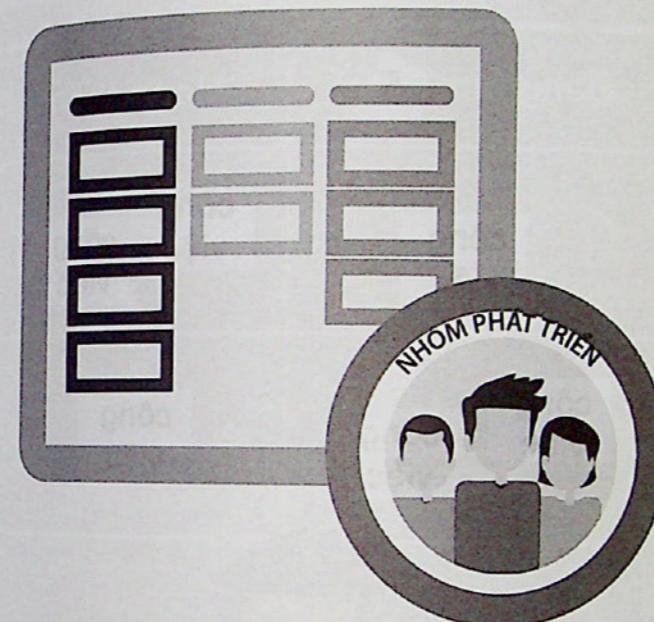
## SPRINT BACKLOG

Sprint Backlog là bảng công việc được Nhóm Phát triển sử dụng để quản lý quá trình sản xuất trong một Sprint. Nội dung cơ bản của một Sprint Backlog bao gồm danh sách các hạng mục Product Backlog đã lựa chọn cho Sprint hiện tại và danh sách các công việc cần thực hiện trong Sprint để hoàn thành các hạng mục Product Backlog đó.



Một nhóm Scrum đang đứng họp Scrum Hằng ngày trước Sprint Backlog được dán lên tường.

Thông thường, sau buổi họp này, các thành viên sẽ cập nhật Sprint Backlog và biểu đồ Sprint Burndown.



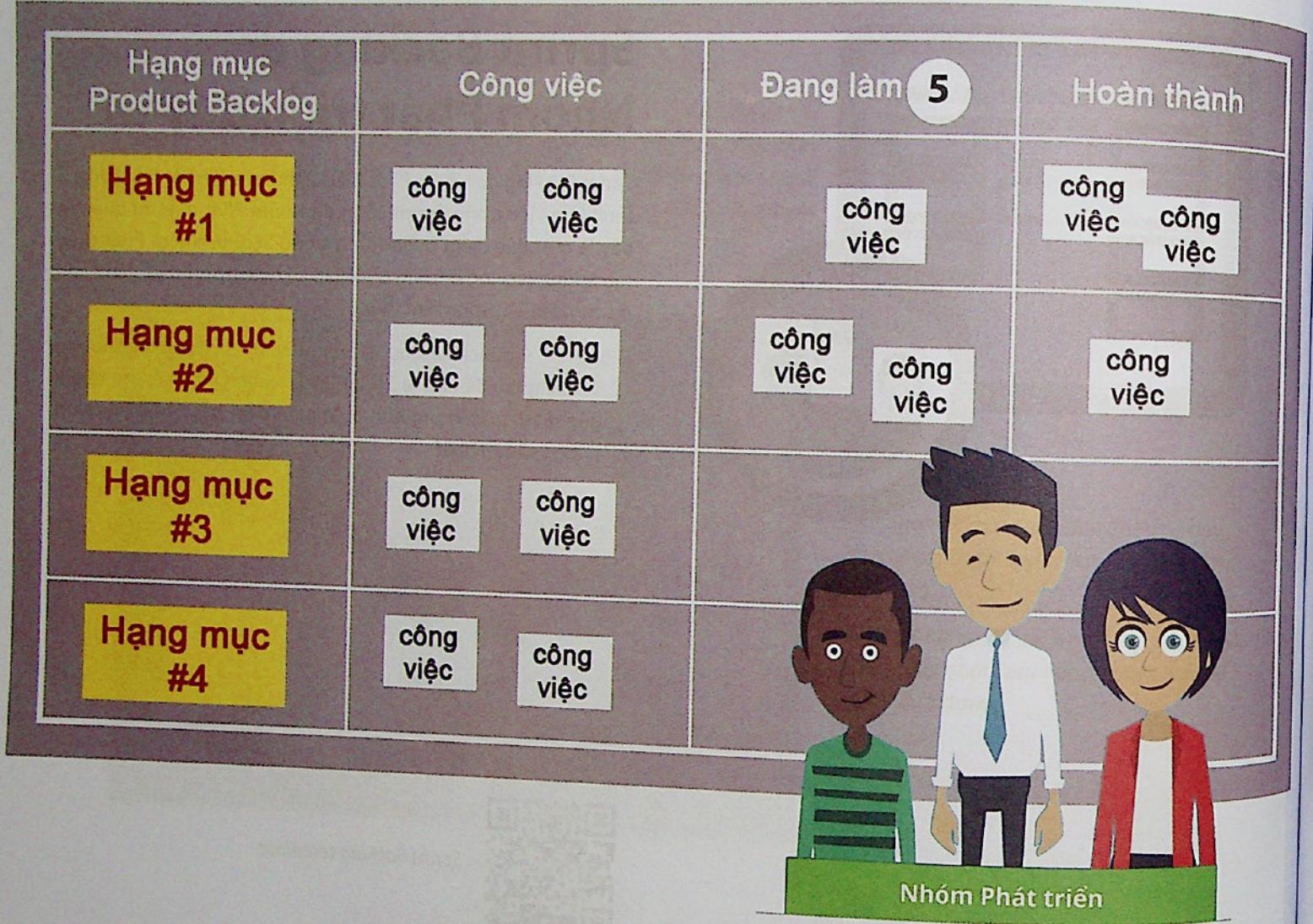
## Sprint Backlog do Nhóm Phát triển duy trì

Sprint Backlog do Nhóm Phát triển sở hữu và vận hành bởi vì quá trình làm việc trong Sprint cũng do Nhóm Phát triển tự quản lý, đây là một đặc điểm quan trọng trong Scrum. Nhóm Phát triển tạo Sprint Backlog ở trong buổi Lập kế hoạch Sprint và vận hành nó cho đến khi Sprint kết thúc.

Giống như Product Backlog, các nhóm có thể sử dụng các phương tiện khác nhau để làm ra Sprint Backlog tiện lợi nhất cho mình. Nhưng điều quan trọng hơn cả là luôn đảm bảo Sprint Backlog hiện diện và cập nhật.



Sprint Backlog template



## Sprint Backlog dạng Spreadsheet

Có nhiều cách để thể hiện Sprint Backlog, tùy theo lựa chọn và tính phù hợp đối với nhóm.

Ví dụ, cách trình bày dưới dạng một bảng tính truyền thống như sau:

Danh sách các hạng mục Product Backlog được liệt kê ở cột đầu tiên. Cột thứ 2 chứa danh sách các công việc sẽ thực hiện tương ứng với từng hạng mục. Cột thứ 3 ghi tên của người sẽ triển khai công việc đó. Cột thứ 4 ghi khối lượng công việc ước tính ban đầu tương ứng với từng công việc.

Hạng mục trong Product Backlog	Công việc trong Sprint	Người thực hiện	Ước lượng khối lượng công việc còn lại tối ngày ..					
			Ước tính khối lượng công việc ban đầu	1	2	3	4	5
Là Người mua, Tôi muốn đưa một cuốn sách vào giỏ hàng (xem giao diện phác thảo trên wiki)	Thay đổi cơ sở dữ liệu		5					
	Tạo trang web (UI)		8					
	tạo trang web (Lập trình JavaScript)		13					
	viết acceptance test tự động			13				
	cập nhật trang giúp đỡ người mua.			3				
	...							
Tăng tốc độ xử lý giao dịch	trộn mã DCP và hoàn thành kiểm thử mức tăng.				5			
	hoàn thành máy đặt hàng cho pRank				8			
	chuyển DCP và người đọc sang sử dụng pRank http API				13			

Hạng mục Product Backlog được đưa vào Sprint

Các công việc cần làm để hiện thực hóa yêu cầu

Các ước lượng về nỗ lực còn lại để làm việc cụ thể, cập nhật qua từng ngày

Các cột tiếp theo tương ứng với từng ngày làm việc trong Sprint.

Sau mỗi ngày làm việc, nhóm ước tính lại khối lượng công việc còn lại cho từng công việc và cập nhật vào bảng. Với cấu trúc Sprint Backlog như thế này, nhóm có thể dùng công cụ bảng tính truyền thống để quản lý.

## Sprint Backlog dạng Kanban

Một dạng cấu trúc khác được ưa chuộng và áp dụng rộng rãi hiện nay là bảng Kanban bao gồm các cột.

Với dạng này, Sprint Backlog được trình bày như sau:

Hạng mục Product Backlog	Công việc	Đang làm 5	Hoàn thành
Hạng mục #1	công việc	công việc	công việc
Hạng mục #2	công việc	công việc	công việc
Hạng mục #3	công việc	công việc	
Hạng mục #4	công việc	công việc	

Cột đầu tiên chứa các hạng mục Product Backlog.

Cột thứ 2 chứa các công việc tương ứng với từng hạng mục Product Backlog.

Cột thứ 3 chứa các công việc đang được triển khai. Lưu ý là ở đây chúng ta có một giới hạn các công việc đang làm, có nghĩa là tại một thời điểm thì nhóm không được phép thực hiện đồng thời số công việc vượt quá giới hạn đã quy định.

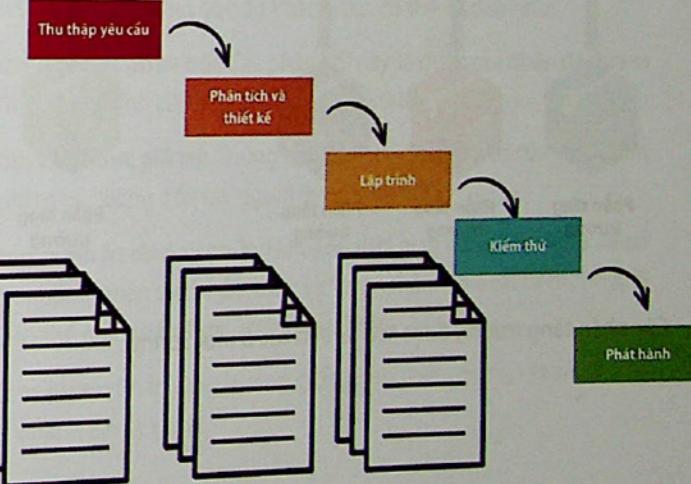
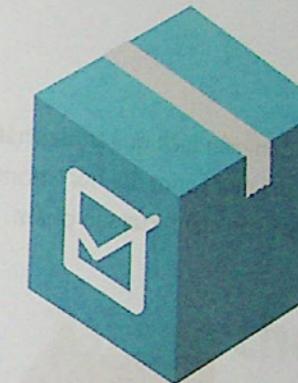
Cột cuối cùng chứa các công việc đã được hoàn thành.

Đây là một bảng Kanban cơ bản, trong thực tế thì cột "Đang làm" thường được chia thành các cột nhỏ hơn để thể hiện cụ thể trạng thái của công việc. Ví dụ, *Đang viết mã*, *Đang rà soát*, *Đang kiểm thử*.

## PHẦN TĂNG TRƯỞNG CHUYỂN GIAO ĐƯỢC

Phần tăng trưởng (tên gọi ngắn của Phần tăng trưởng Sản phẩm Có khả năng Chuyển giao được) là phần sản phẩm Nhóm Phát triển tạo ra cuối mỗi Sprint.

Đây là một khái niệm quan trọng trong Scrum, tạo ra sự khác biệt lớn về mặt sản phẩm so với các phương pháp truyền thống.

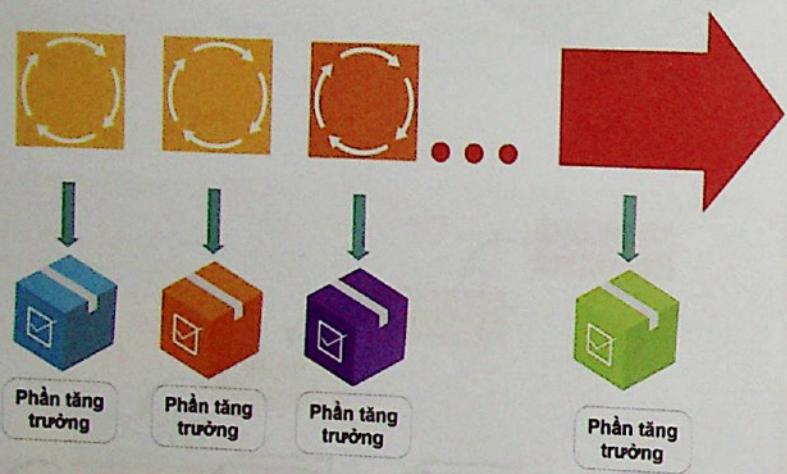


### Khác biệt với tư duy phát triển truyền thống

Ở các phương pháp phát triển truyền thống, quá trình phát triển sản phẩm được tiến hành lần lượt theo từng bước, chẳng hạn đi từ thu thập yêu cầu, phân tích và thiết kế, xây dựng, kiểm thử rồi đi đến phát hành. Do đó, ở phần lớn thời gian đầu, kết quả thu được là rất nhiều tài liệu, bao gồm cả tài liệu mô tả yêu cầu lẫn tài liệu thiết kế. Trong khi đó thiếu vắng một sản phẩm thật có thể sử dụng được bởi người dùng.

Trong phương pháp phát triển truyền thống, sản phẩm chỉ được hình thành ở giai đoạn cuối của chu trình phát triển

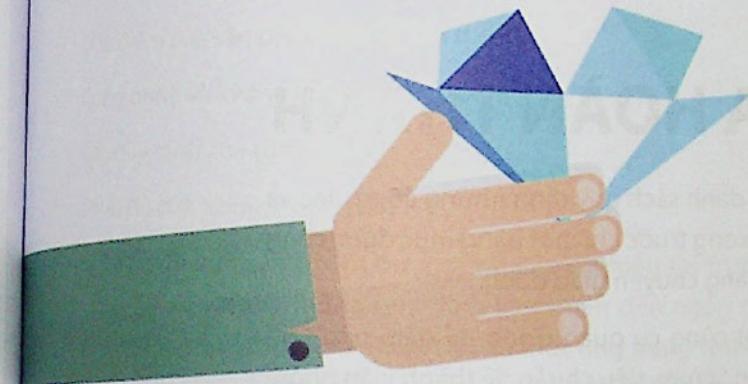
Scrum có một hướng tiếp cận khác, không chỉ đơn giản tách quá trình phát triển thành các Sprint nhỏ liên tiếp nhau, mà cuối mỗi Sprint đòi hỏi Nhóm Phát triển phải chuyển giao một phần tính năng hoàn chỉnh của sản phẩm.



Các phản tăng trưởng được tạo ra liên tục ở mỗi Sprint

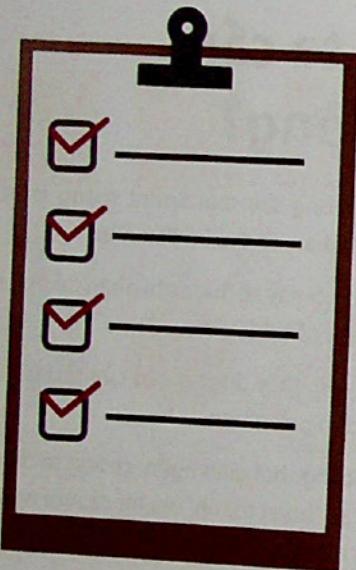
## Các đặc điểm của Phản tăng trưởng

- Sử dụng được:** Tính năng đạt được sau mỗi Sprint là thật và sử dụng được chứ không chỉ là một bản thiết kế hay một bản mẫu.
- Có khả năng chuyển giao được:** Phản tăng trưởng ở cuối mỗi Sprint cần phải có khả năng chuyển giao được. Điều này không có nghĩa là nó phải lập tức được chuyển giao ngay, mà có nghĩa là nó đạt được trạng thái sẵn sàng chuyển giao.
- Hoàn thành:** Phản tăng trưởng phải tuân thủ theo Định nghĩa Hoàn thành đã được thống nhất trước đó.



## Tại sao chúng ta cần Phản tăng trưởng?

- Việc tạo ra được các phản tăng trưởng sau mỗi Sprint mang lại rất nhiều lợi ích từ những góc độ khác nhau, có thể kể đến như:
- Sớm có được phản hồi:** Các phản hồi này sẽ được ghi nhận để đưa ra những điều chỉnh cho sản phẩm nếu cần thiết.
  - Sớm thu được giá trị:** Chúng ta sớm thấy được giá trị của sản phẩm và giá trị này cũng đến tay người dùng sớm hơn.
  - Tăng minh bạch:** Sau những khoảng thời gian ngắn, chúng ta đã có được những phản sản phẩm thực sự "hoàn thành" mà tất cả mọi người đều có thể nhìn thấy được, tương tác được.
  - Giảm rủi ro:** Phản tăng trưởng giúp giảm thiểu rủi ro, xét về cả mặt kinh doanh lẫn kỹ thuật.



## ĐỊNH NGHĨA HOÀN THÀNH

Định nghĩa Hoàn thành là một danh sách quy định những công việc và tiêu chuẩn cần được thực hiện xong trước khi một hạng mục được công nhận là đạt trạng thái có khả năng chuyển giao được.

Định nghĩa Hoàn thành là một công cụ quan trọng để đảm bảo tính minh bạch của các tạo tác trong Scrum, tiêu chuẩn để thành viên nhóm tự kiểm tra và đạt được chất lượng tự thân.

Ngoài ra, Định nghĩa Hoàn thành giúp mọi người hiểu chung về yêu cầu công việc, ngăn ngừa những mâu thuẫn do hiểu nhầm hoặc không hiểu rõ. Gia tăng tính minh bạch về yêu cầu giúp nhóm tự tổ chức hiệu quả hơn.

## Nội dung của Định nghĩa Hoàn thành

Nội dung của Định nghĩa Hoàn thành dành cho từng nhóm và từng sản phẩm là khác nhau, điều này phụ thuộc vào một số yếu tố như:

- Đặc điểm của sản phẩm đang xây dựng
- Công nghệ và kỹ thuật đang được sử dụng
- Các quy định của tổ chức
- Trình độ của Nhóm Phát triển

Chẳng hạn, nếu một sản phẩm được cung cấp ở nhiều thị trường khác nhau thì có thể có những quy định liên quan đến ngôn ngữ hiển thị, tiền tệ, định dạng thời gian. Nếu là một ứng dụng web thì có thể có những quy định liên quan đến việc hỗ trợ tốt trên các loại trình duyệt khác nhau. Nếu tổ chức có những quy ước liên quan đến việc phát triển thì có thể có những quy định liên quan đến việc tuân thủ các quy ước này. Nếu trình độ của Nhóm Phát triển cao thì có thể có những quy định rất chặt chẽ về chất lượng sản phẩm. Chẳng hạn, trong sản xuất phần mềm, có thể có các quy định như phương thức không vượt quá 8 dòng mã, độ phủ của kiểm thử là 100%, thực hiện tái cấu trúc mã nguồn, v.v.. Trong sản xuất video marketing, có thể có các quy định như độ dài không quá 2 phút, thời gian xuất hiện của thông điệp chính phải tối thiểu 30% thời lượng, âm thanh không có tiếng vang (echo), v.v..

### Ví dụ về một Định nghĩa Hoàn thành trong Phát triển phần mềm

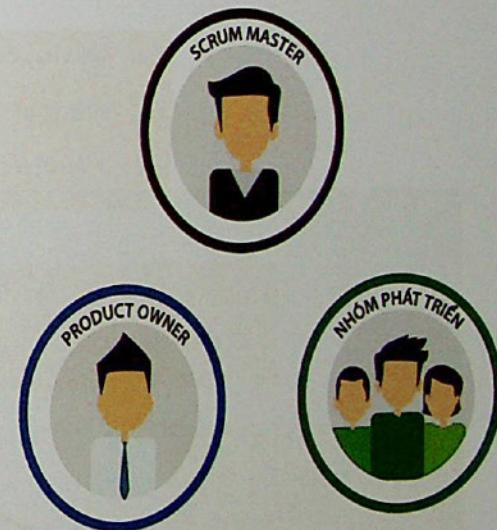
- |  |
|--|
| <input checked="" type="checkbox"/> Thiết kế đã được kiểm tra chéo |
| <input checked="" type="checkbox"/> Mã nguồn đã được kiểm tra chéo |
| <input checked="" type="checkbox"/> Mã nguồn đã được tái cấu trúc  |
| <input type="checkbox"/> Mã nguồn phù hợp với quy ước của công ty  |
| <input type="checkbox"/> Mã nguồn đã được đẩy lên hệ thống lưu trữ |
| <input type="checkbox"/> Đã có kiểm thử đơn vị                     |
| <input type="checkbox"/> Đã có kiểm thử tích hợp                   |
| <input type="checkbox"/> Đã có kiểm thử chấp nhận                  |
| <input type="checkbox"/> Không có lỗi được phát hiện               |

## Ví dụ về một Định nghĩa Hoàn thành trong phát triển học liệu eLearning

- Tên video ngắn gọn, phù hợp nội dung của video
- Với mỗi nội dung trong video đều có ví dụ hoặc demo cụ thể
- Nội dung của video đáp ứng ít nhất 1 chuẩn đầu ra và không quá 3 chuẩn đầu ra của bài học
- Có file slide gốc dạng Powerpoint, được review chuyên môn
- Có file thu âm gốc, khớp với transcript
- Không có lỗi chuyên môn
- Mỗi slide là một ý nhỏ
- Sử dụng template slide nền trắng, chữ đen, xanh navy, cam
- Sử dụng font chữ Myriad Pro, size không nhỏ hơn 24
- Không nhiều hơn 6-8 từ mỗi dòng
- Tuân thủ nguyên tắc 6x6, không quá 6 gạch đầu dòng trong 1 slide và không quá 6 từ mỗi dòng
- Hình ảnh và âm thanh có nội dung khớp nhau
- Mở đầu và kết thúc mỗi video đều có logo
- Không có lỗi chính tả
- Không nói vấp, nói ngọng
- Mỗi video không dài quá 10 phút

## Nhóm Scrum xây dựng Định nghĩa Hoàn thành

Việc xây dựng Định nghĩa Hoàn thành cần có sự tham gia của toàn bộ Nhóm Scrum, tức là bao gồm các thành viên Nhóm Phát triển, Product Owner và ScrumMaster. Thông thường, Product Owner là người có vai trò xác nhận cuối cùng đối với Định nghĩa Hoàn thành. Định nghĩa Hoàn thành cũng không phải là một tài liệu cố định mà sẽ được cải tiến theo quá trình trưởng thành của nhóm để phản ánh sự trưởng thành này cũng như nâng cao chất lượng của sản phẩm.

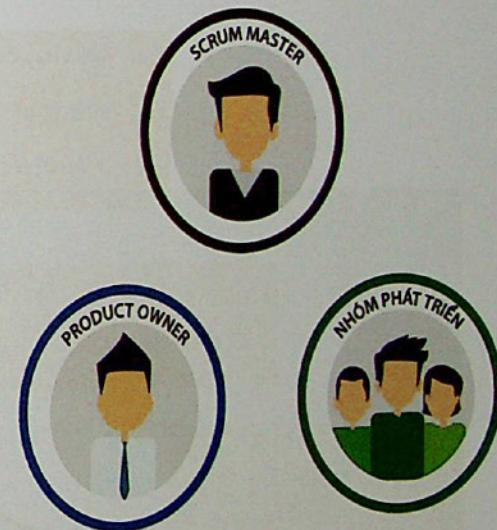


## Ví dụ về một Định nghĩa Hoàn thành trong phát triển học liệu eLearning

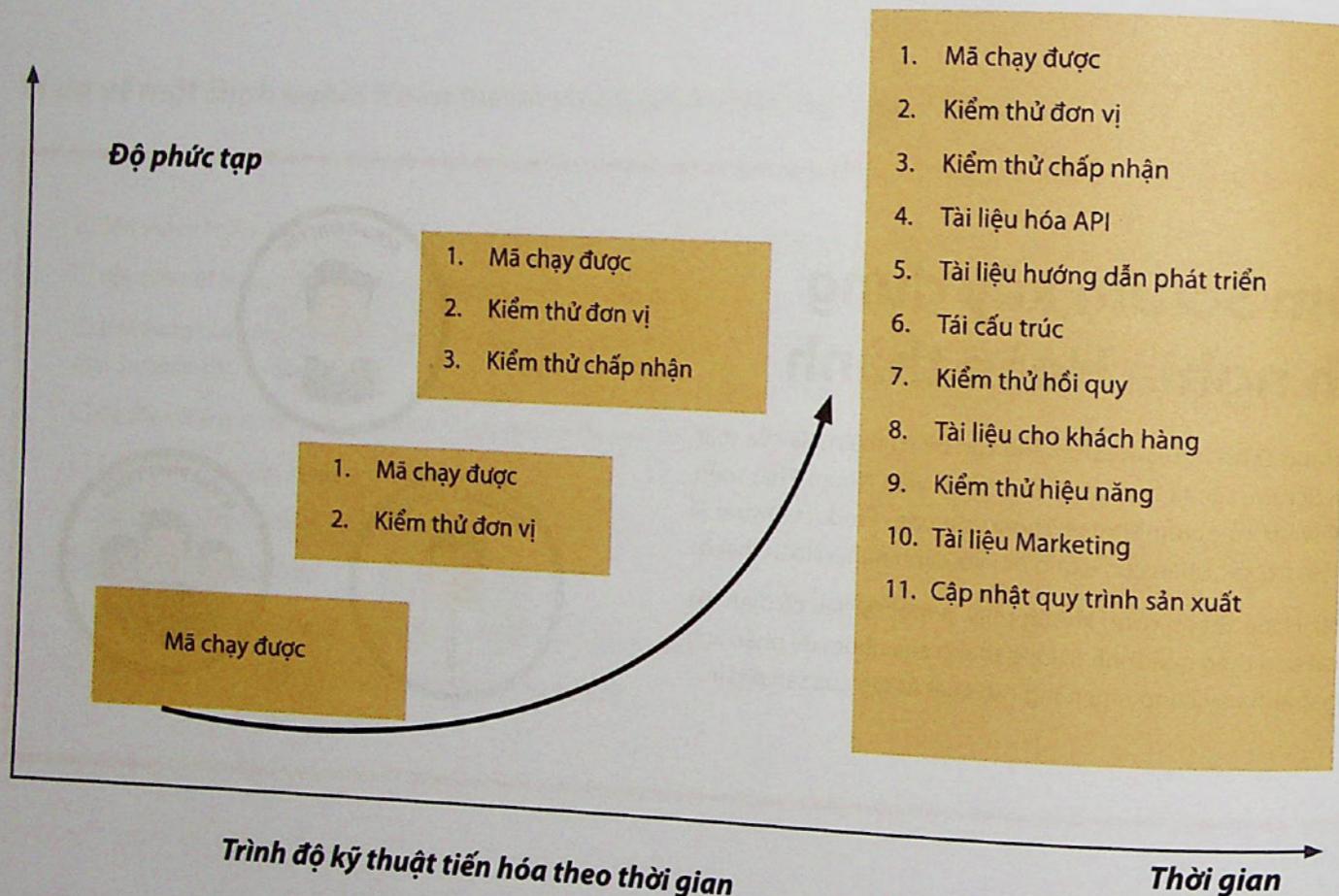
- Tên video ngắn gọn, phù hợp nội dung của video
- Với mỗi nội dung trong video đều có ví dụ hoặc demo cụ thể
- Nội dung của video đáp ứng ít nhất 1 chuẩn đầu ra và không quá 3 chuẩn đầu ra của bài học
- Có file slide gốc dạng Powerpoint, được review chuyên môn
- Có file thu âm gốc, khớp với transcript
- Không có lỗi chuyên môn
- Mỗi slide là một ý nhỏ
- Sử dụng template slide nền trắng, chữ đen, xanh navy, cam
- Sử dụng font chữ Myriad Pro, size không nhỏ hơn 24
- Không nhiều hơn 6-8 từ mỗi dòng
- Tuân thủ nguyên tắc 6x6, không quá 6 gạch đầu dòng trong 1 slide và không quá 6 từ mỗi dòng
- Hình ảnh và âm thanh có nội dung khớp nhau
- Mở đầu và kết thúc mỗi video đều có logo
- Không có lỗi chính tả
- Không nói vấp, nói ngọng
- Mỗi video không dài quá 10 phút

## Nhóm Scrum xây dựng Định nghĩa Hoàn thành

Việc xây dựng Định nghĩa Hoàn thành cần có sự tham gia của toàn bộ Nhóm Scrum, tức là bao gồm các thành viên Nhóm Phát triển, Product Owner và ScrumMaster. Thông thường, Product Owner là người có vai trò xác nhận cuối cùng đối với Định nghĩa Hoàn thành. Định nghĩa Hoàn thành cũng không phải là một tài liệu cố định mà sẽ được cải tiến theo quá trình trưởng thành của nhóm để phản ánh sự trưởng thành này cũng như nâng cao chất lượng của sản phẩm.



## Quá trình tiến hóa của “Hoàn thành”



**Ví dụ về tiến hóa của một Định nghĩa Hoàn thành trong Phát triển phần mềm**

## Xây dựng Định nghĩa Hoàn thành cho từng cấp độ khác nhau

Các ví dụ về Định nghĩa Hoàn thành được đề cập ở trên là áp dụng cho các User story để được công nhận là đạt trạng thái sẵn sàng chuyển giao. Tuy nhiên, trên thực tế thì Định nghĩa Hoàn thành có thể được áp dụng cho những cấp độ khác nữa, chẳng hạn như đối với các User story để quy định trạng thái sẵn sàng để đưa vào phát triển, đối với các công việc trong Sprint và đối với các Phiên bản phát hành.

### Định nghĩa Hoàn thành dành cho User Story

Đối với các User story, chúng ta có thể xây dựng Định nghĩa Hoàn thành để quy định trạng thái của một User story được công nhận là sẵn sàng để đưa vào phát triển trong một Sprint. Loại Định nghĩa Hoàn thành này dành cho User story còn được gọi là Định nghĩa Sẵn sàng.

- User story đã được ước tính
- Có danh sách các tiêu chí chấp nhận
- User story là duy nhất
- Có định kèm tài liệu phù hợp (chẳng hạn như bản mẫu, dữ liệu,...)
- Giá trị ước tính ban đầu không nhiều hơn 5 điểm

## Định nghĩa Hoàn thành dành cho công việc trong Sprint

Đối với các công việc trong Sprint, Nhóm Phát triển có thể đưa ra Định nghĩa Hoàn thành để quy định trạng thái cuối của từng công việc. Một phần quan trọng của Định nghĩa Hoàn thành này thường đề cập đến các yêu cầu về mặt kỹ thuật cũng như quy trình để cộng tác trong nhóm.

- Mã nguồn đã được kiểm tra chéo
- Mã nguồn có đầy đủ chú thích
- Mã nguồn đã được đưa lên hệ thống lưu trữ
- Đã cập nhật thời gian của công việc trên Sprint Backlog về 0

## Định nghĩa Hoàn thành dành cho Phiên bản Phát hành

Đối với các Phiên bản Phát hành, Nhóm Scrum và các bên liên quan có thể xây dựng Định nghĩa Hoàn thành nhằm đưa ra các quy định mà một bản phát hành cần phải thỏa mãn. Nội dung của Định nghĩa Hoàn thành này thường tập trung vào các điều kiện để cài đặt sản phẩm, sử dụng sản phẩm, tính ổn định của sản phẩm và việc hỗ trợ người dùng.

- Tất cả mã nguồn đều đã được triển khai trên máy chủ thực tế
- Sản phẩm được dùng thử bởi 10 người ngoài Nhóm Phát triển trong vòng 5 ngày
- Không có lỗi được phát hiện
- Nhóm chăm sóc khách hàng đã được huấn luyện về các tính năng mới

## Định nghĩa Hoàn thành khác với Tiêu chí chấp nhận

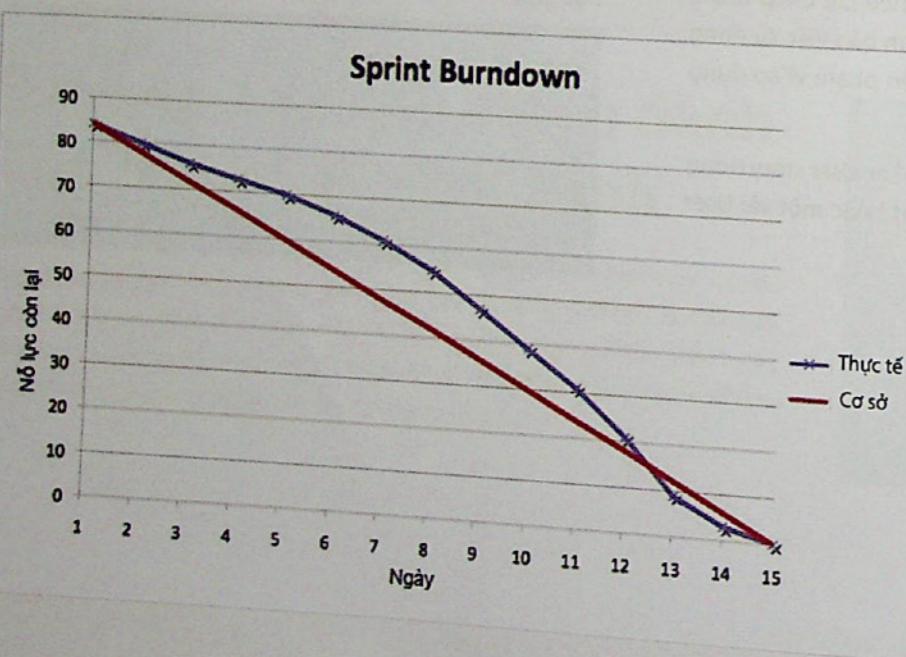
Đối với User story, có hai khái niệm thường gây ra nhầm lẫn cho những người mới bắt đầu đó là Định nghĩa Hoàn thành và Tiêu chí Chấp nhận. Việc phân biệt rõ hai khái niệm này là cần thiết để đảm bảo việc sử dụng đúng và hiệu quả. Chúng ta có thể phân biệt dựa trên phạm vi áp dụng của chúng.

Lưu ý, Định nghĩa Hoàn thành được áp dụng cho tất cả các User story trong khi đó Tiêu chí Chấp nhận chỉ được áp dụng cho một hoặc một vài User story nhất định.

Một ví dụ về Tiêu chí Chấp nhận cho tính năng thanh toán trực tuyến

- Hoạt động tốt với các loại thẻ Visa, Master
- Thời gian phản hồi không vượt quá 20 giây
- Hỗ trợ thanh toán bằng USD, EURO, VND

# BIỂU ĐỒ SPRINT BURNDOWN

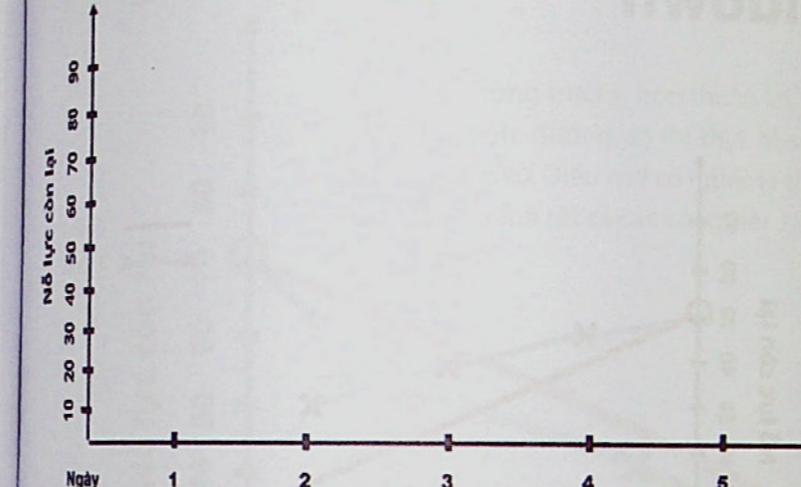


Scrum không quy định cách thức để Nhóm Phát triển theo dõi tiến độ sản xuất trong Sprint. Rất nhiều nhóm sử dụng biểu đồ Sprint Burndown như một công cụ để trực quan hóa tiến độ hướng tới Mục tiêu Sprint. Nhóm Phát triển tập trung sự quan tâm của mình vào lượng nỗ lực còn lại cần thực hiện để đạt được Mục tiêu Sprint chứ không phải là mình đã làm được những gì.

Nhóm Phát triển tạo ra Sprint Burndown sau buổi Lập kế hoạch Sprint và cập nhật nó hằng ngày để thấy được tiến độ thực của mình. Nếu có những biểu hiện không tốt về mặt tiến độ công việc thì nhóm có thể đưa ra các quyết định điều chỉnh nhanh chóng một cách chủ động để hướng đến việc hoàn thành Mục tiêu Sprint đúng như cam kết của mình.

Nhóm Phát triển dựa vào Sprint Backlog để tính tổng lượng nỗ lực còn lại cần thực hiện để đạt được mục tiêu Sprint. Mỗi công việc trên Sprint Backlog đều đã được ước tính trong buổi Lập kế hoạch Sprint, tổng các giá trị ước tính của những công việc chưa hoàn thành chính là lượng nỗ lực còn lại.

Hằng ngày, khi đã có một số công việc được hoàn thành, nhóm cập nhật lại lượng nỗ lực cho các công việc chưa hoàn thành.



## Cách vẽ Sprint Burndown

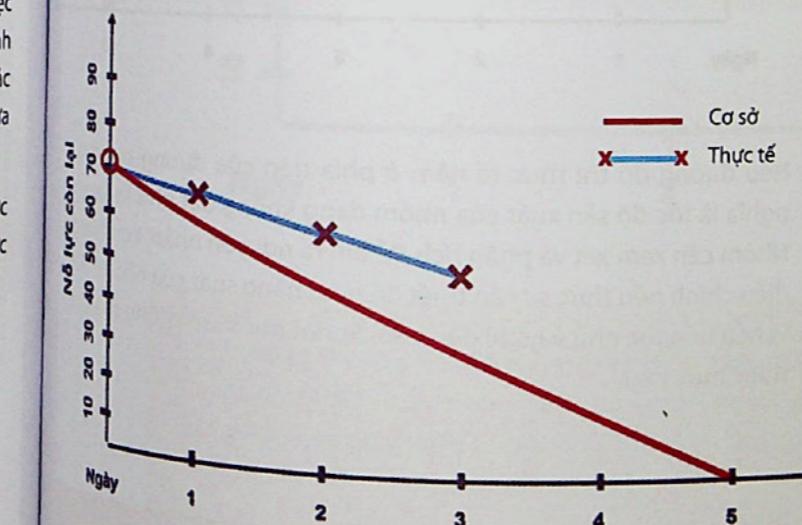
Biểu đồ Sprint Burndown có 2 trục. Trục hoành thể hiện thời gian, được chia thành các ngày tương ứng với thời gian của một Sprint. Ví dụ, đối với Sprint 1 tuần có 5 ngày làm việc thì chúng ta sẽ chia thành các ngày: 1, 2, 3, 4, 5.

Trục tung thể hiện lượng nỗ lực còn lại cần thiết để hoàn thành tất cả các công việc. Giá trị này được lấy từ Sprint Backlog.

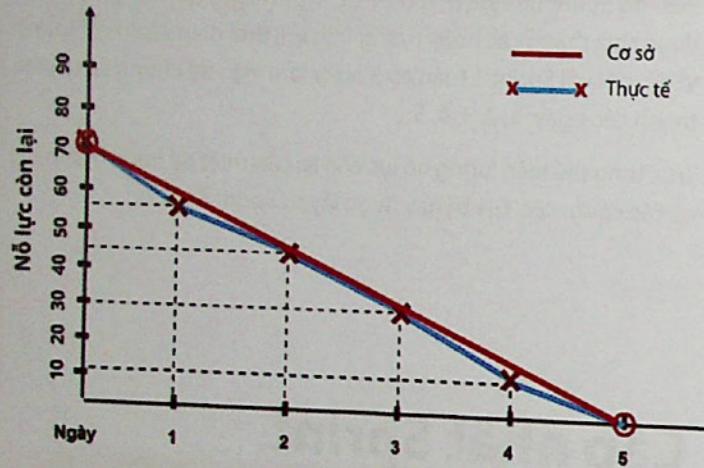
## Cập nhật Sprint Burndown

Hằng ngày, Nhóm Phát triển tính lượng nỗ lực còn lại và cập nhật vào Sprint Burndown.

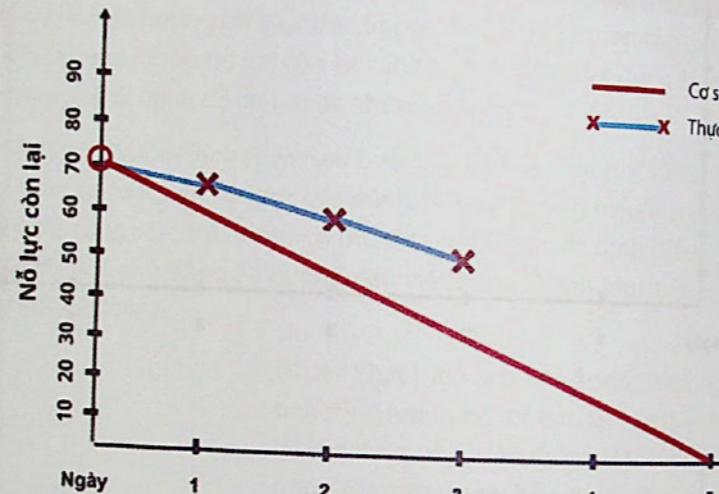
Trong ví dụ này, tổng lượng nỗ lực còn lại theo từng ngày lần lượt là 71, 58, 45:



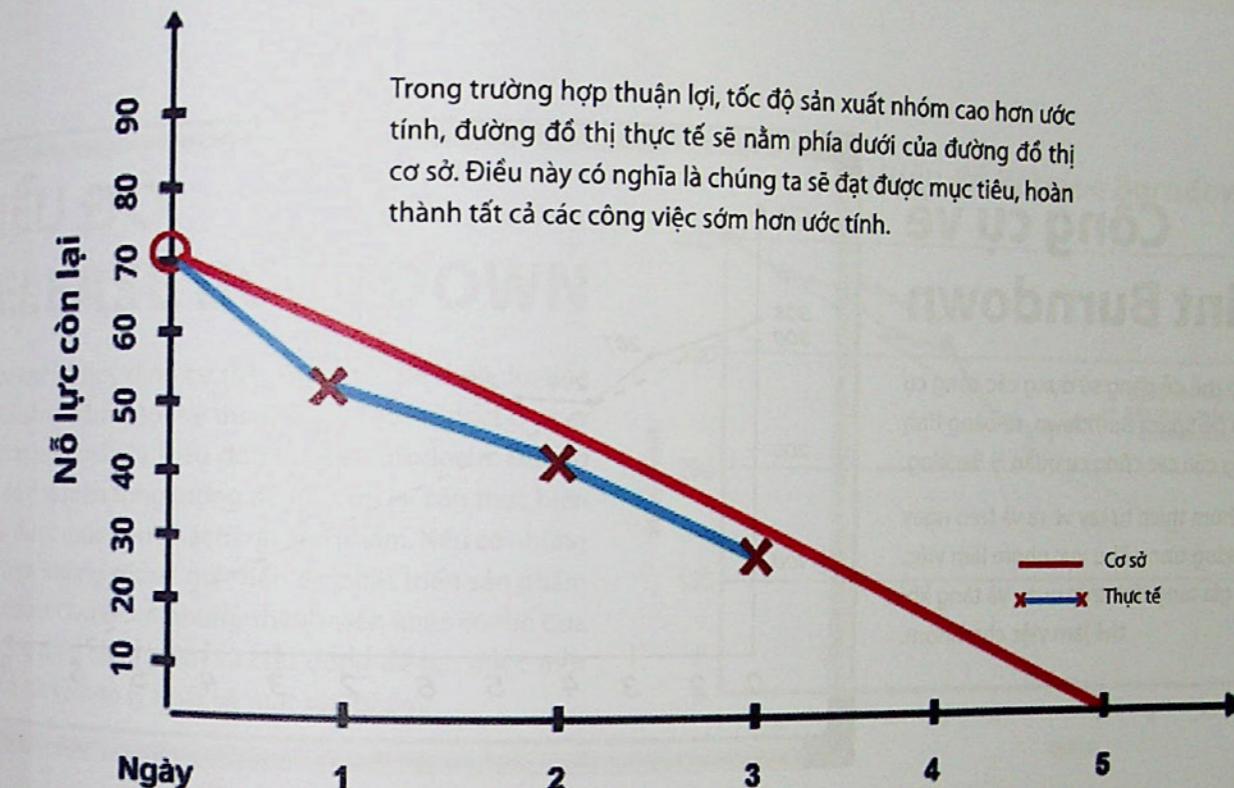
# Đường cơ sở của Sprint Burndown



Trong biểu đồ Sprint Burndown, chúng ta có một đường cơ sở được nối từ điểm khởi đầu cho đến điểm mong muốn đạt được khi kết thúc Sprint. Tức là, trong trường hợp lý tưởng nhất, tốc độ sản xuất của nhóm luôn ổn định và lượng nỗ lực công việc còn lại sau từng ngày sẽ giảm dần đều trong suốt Sprint, đạt được giá trị mong muốn vào đúng ngày cuối cùng.



Nếu đường đồ thị thực tế nằm ở phía trên của đường cơ sở có nghĩa là tốc độ sản xuất của nhóm đang không đạt như kỳ vọng. Nhóm cần xem xét và phân tích để tìm ra nguyên nhân, có những điều chỉnh nếu thực sự cần thiết để tăng năng suất của nhóm. Bởi vì nếu tiếp tục như vậy thì đến cuối Sprint nhóm sẽ không thể đạt được mục tiêu.

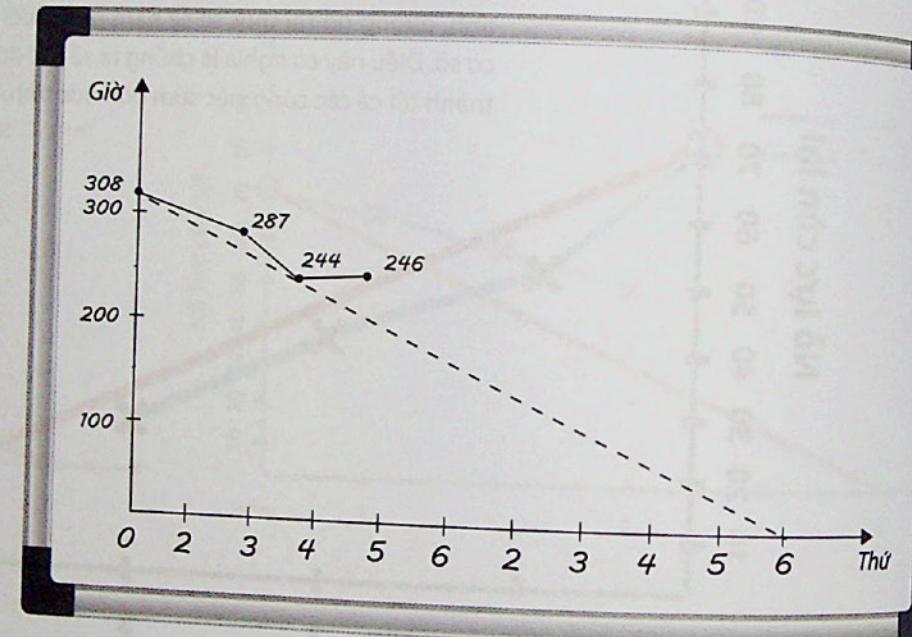


Trong trường hợp thuận lợi, tốc độ sản xuất nhóm cao hơn ước tính, đường đồ thị thực tế sẽ nằm phía dưới của đường đồ thị cơ sở. Điều này có nghĩa là chúng ta sẽ đạt được mục tiêu, hoàn thành tất cả các công việc sớm hơn ước tính.

Thường thì việc này không gây ra tác hại gì với Sprint hiện tại. Nhưng nếu sự chênh lệch là khá lớn và xảy ra nhiều thì nhóm cần chú ý hơn đến việc lập kế hoạch. Bởi vì những yếu tố có thể gây ra hiện tượng này đó là việc ước tính nỗ lực dành cho các công việc hoặc ước tính khả năng của nhóm đang quá sai lệch so với thực tế.

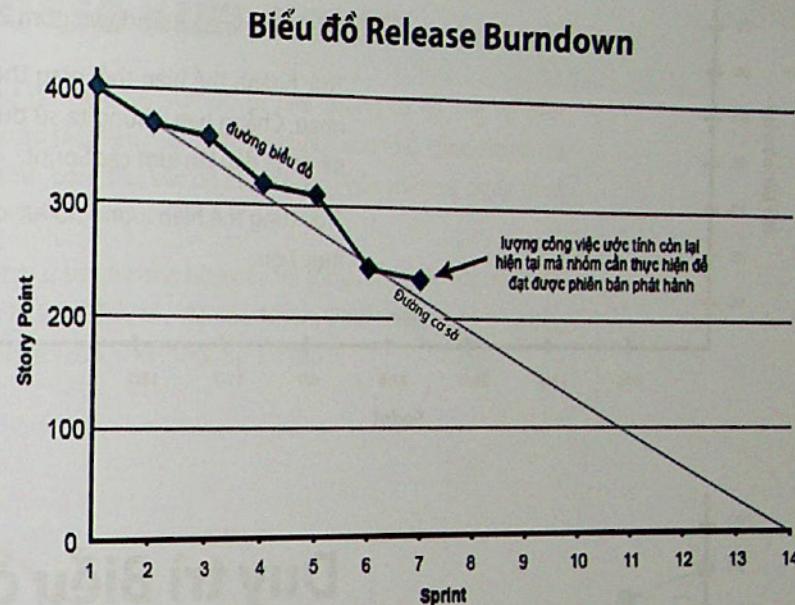
## Công cụ vẽ Sprint Burndown

Trong thực tế, có thể dễ dàng sử dụng các công cụ điện tử để vẽ biểu đồ Sprint Burndown, từ bảng tính cho đến tính năng của các công cụ quản lý Backlog. Tuy nhiên, nhiều nhóm thích tự tay vẽ ra và treo ngay cạnh Sprint Backlog trong khu vực nhóm làm việc. Cách làm này giúp gia tăng tính trực quan và tăng khí thế làm việc cho nhóm.



## BIỂU ĐỒ RELEASE BURNDOWN

Scrum không quy định cụ thể cách thức để theo dõi tiến độ của sản phẩm. Một kỹ thuật được các Product Owner sử dụng phổ biến là biểu đồ Release Burndown. Đường biểu đồ thể hiện tổng lượng nỗ lực còn lại cần thực hiện để đạt được mục tiêu Phát hành Sản phẩm. Nếu có những biểu hiện không tốt về mặt tiến độ phát triển sản phẩm thì Product Owner và những thành viên khác có thể đưa ra những điều chỉnh sớm và chủ động để đạt được một kết quả tốt và hợp lý nhất về mặt sản phẩm.



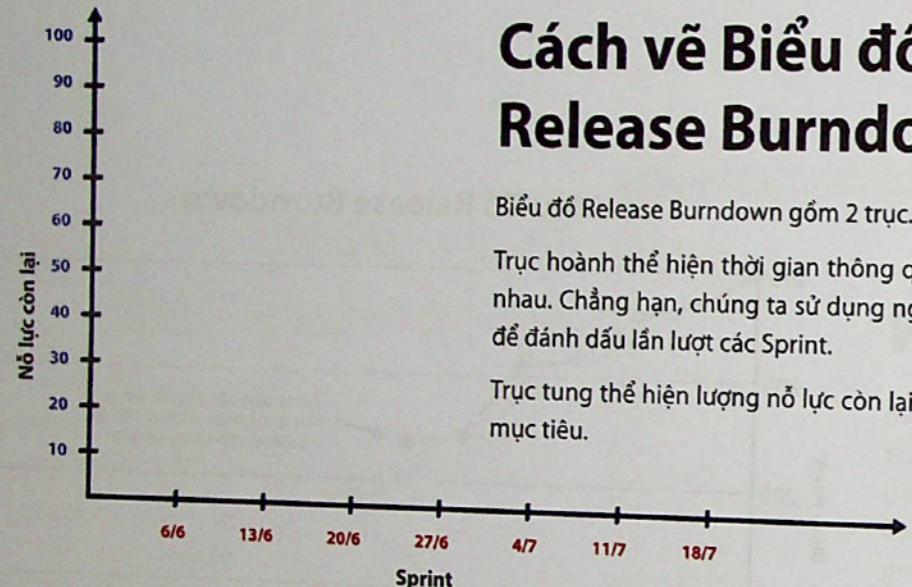
Burndown chart sample

# Cách vẽ Biểu đồ Release Burndown

Biểu đồ Release Burndown gồm 2 trục.

Trục hoành thể hiện thời gian thông qua các Sprint liên tiếp nhau. Chẳng hạn, chúng ta sử dụng ngày bắt đầu của Sprint để đánh dấu lần lượt các Sprint.

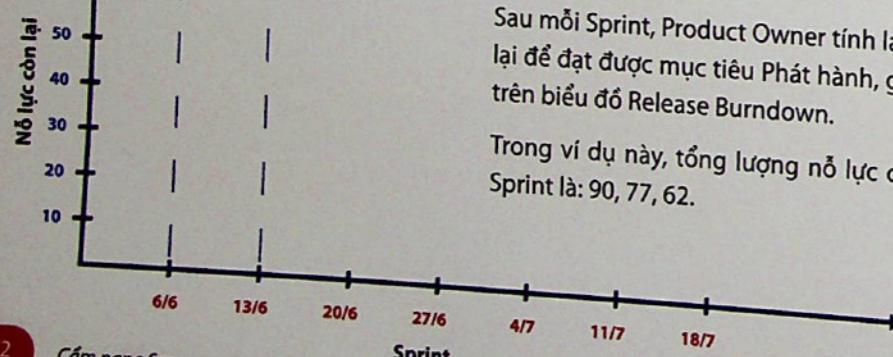
Trục tung thể hiện lượng nỗ lực còn lại cho đến khi đạt được mục tiêu.



## Duy trì Biểu đồ Release Burndown

Sau mỗi Sprint, Product Owner tính lại tổng lượng nỗ lực còn lại để đạt được mục tiêu Phát hành, giá trị này được thể hiện trên biểu đồ Release Burndown.

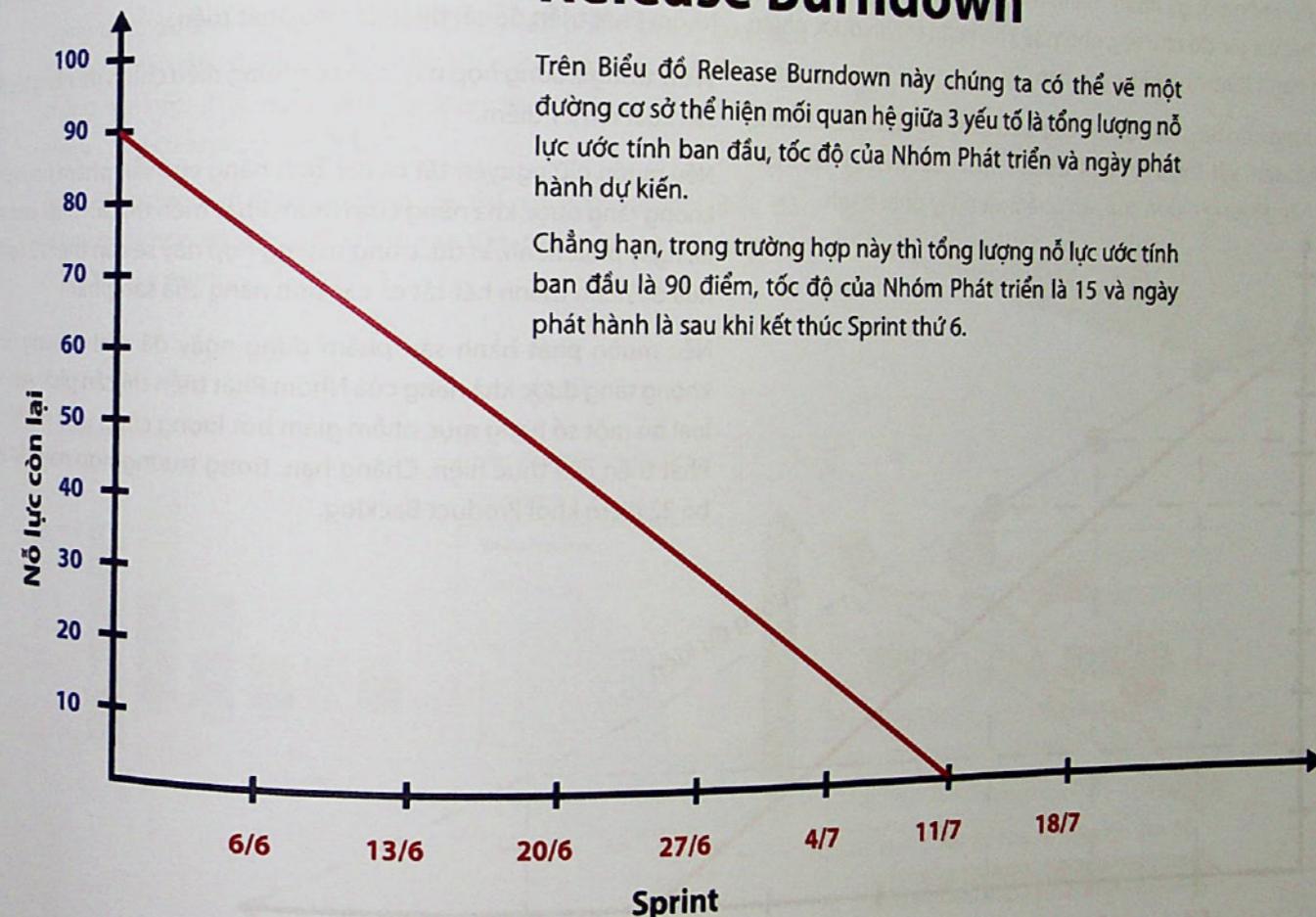
Trong ví dụ này, tổng lượng nỗ lực còn lại lần lượt sau mỗi Sprint là: 90, 77, 62.



## Đường cơ sở của Biểu đồ Release Burndown

Trên Biểu đồ Release Burndown này chúng ta có thể vẽ một đường cơ sở thể hiện mối quan hệ giữa 3 yếu tố là tổng lượng nỗ lực ước tính ban đầu, tốc độ của Nhóm Phát triển và ngày phát hành dự kiến.

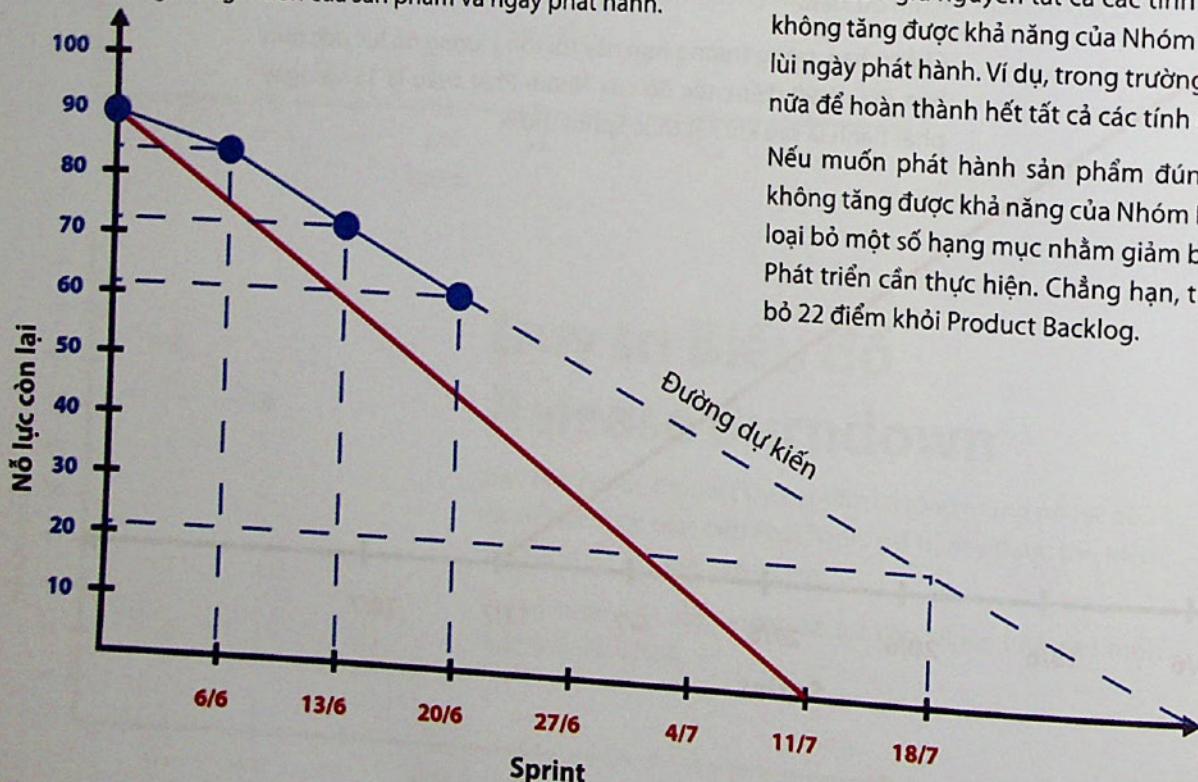
Chẳng hạn, trong trường hợp này thì tổng lượng nỗ lực ước tính ban đầu là 90 điểm, tốc độ của Nhóm Phát triển là 15 và ngày phát hành là sau khi kết thúc Sprint thứ 6.



## Chậm tiến độ so với dự kiến

Nếu Nhóm Phát triển không hoàn thành tốt các mục tiêu Sprint, dẫn đến tiến độ không được hoàn thành theo như dự kiến. Có nghĩa là, nếu tiếp tục với tốc độ như vậy, nhóm sẽ khó hoàn thành được phiên bản phát hành theo như kế hoạch ban đầu.

Trong trường hợp này, việc đưa ra các điều chỉnh là cần thiết. Chúng ta cần phải xem xét 3 yếu tố là tốc độ sản xuất của Nhóm Phát triển, các tính năng mong muốn của sản phẩm và ngày phát hành.



Nếu muốn giữ nguyên các tính năng mong muốn của sản phẩm và không thay đổi ngày phát hành thì cần có các điều chỉnh đối với Nhóm Phát triển để cải thiện tốc độ phát triển.

Ví dụ, trong trường hợp này, cần có những điều chỉnh để nâng tốc độ sản xuất lên 21 điểm.

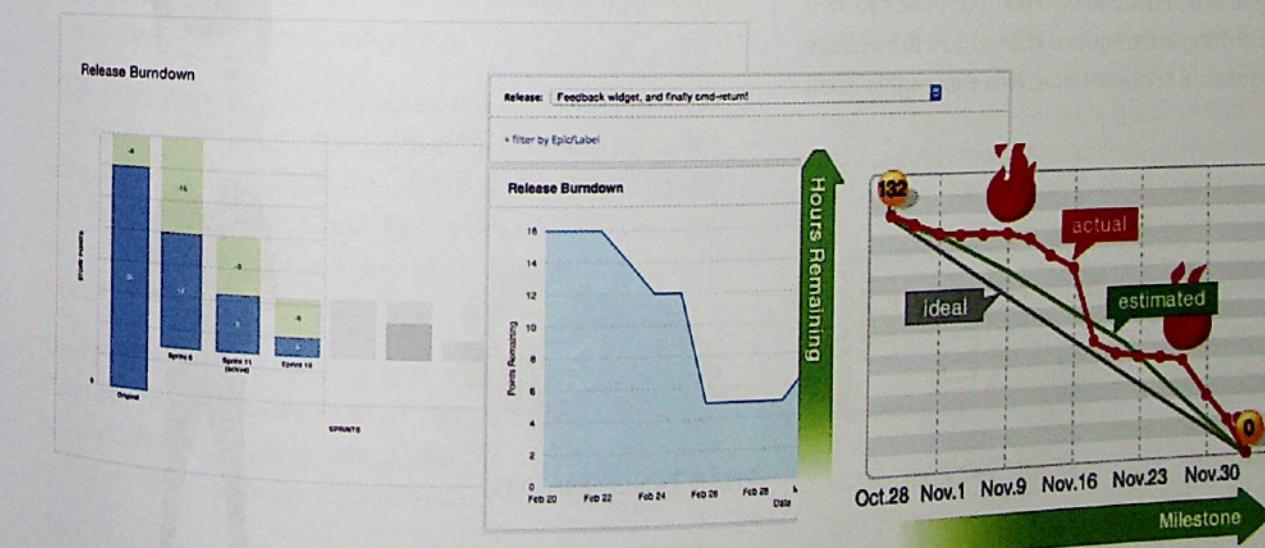
Nếu muốn giữ nguyên tất cả các tính năng của sản phẩm nhưng lại không tăng được khả năng của Nhóm Phát triển thì cần phải xem xét lùi ngày phát hành. Ví dụ, trong trường hợp này sẽ cần thêm 2 Sprint nữa để hoàn thành hết tất cả các tính năng của sản phẩm.

Nếu muốn phát hành sản phẩm đúng ngày đã định nhưng cũng không tăng được khả năng của Nhóm Phát triển thì cần phải xem xét loại bỏ một số hạng mục nhằm giảm bớt lượng công việc mà Nhóm Phát triển cần thực hiện. Chẳng hạn, trong trường hợp này cần loại bỏ 22 điểm khỏi Product Backlog.

## Công cụ vẽ Biểu đồ Release Burndown

Trong thực tế, tính năng theo dõi tiến độ sản phẩm thông qua Biểu đồ Burndown đều được hỗ trợ ở hầu hết các công cụ quản lý Backlog, chẳng hạn như JIRA, Assembla, Backlog, Pivotal Tracker hay Excel và nhiều công cụ khác.

Tuy nhiên, Product Owner cũng có thể tự thiết kế một Biểu đồ Release Burndown vật lý để nâng cao tính trực quan và dễ dàng tùy chỉnh theo ý muốn.



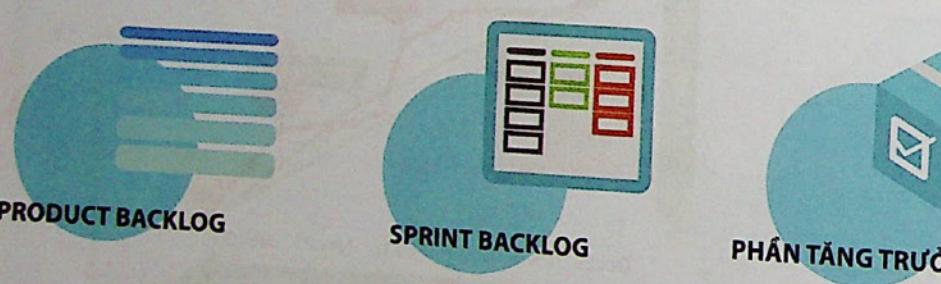
# Tính minh bạch của các tạo tác

Đảm bảo cho các tạo tác của nhóm luôn được cập nhật và phản ánh đúng tình trạng hiện tại là điều kiện tối cần thiết để có được thành công của từng Sprint và thành công của sản phẩm. Các tạo tác là môi trường, là công cụ, là kết quả của giao tiếp giữa các bên tham gia trong quá trình sản xuất. Nếu thông tin không được giữ minh bạch trên các tạo tác thì sẽ gây cản trở cho quá trình cộng tác và ra quyết định, dẫn đến các kết quả tai hại.

Nếu Product Backlog không được cập nhật, không được làm mịn thường xuyên thì ngay lập tức sẽ ảnh hưởng đến việc lập kế hoạch sản xuất của từng Sprint. Nếu Sprint Backlog không được cập nhật thì rủi ro hằng ngày sẽ bị tích tụ do nhóm không có được các điều chỉnh cần thiết, nhóm đứng trước nguy cơ không hoàn thành được toàn bộ công việc trước khi Sprint kết thúc. Nếu Định nghĩa Hoàn

thành không rõ ràng và phù hợp với yêu cầu chất lượng và năng lực của nhóm thì chất lượng của sản phẩm sẽ bị ảnh hưởng, giao tiếp giữa các bên cũng sẽ khó khăn.

Đối với tất cả các tạo tác và công cụ khác của nhóm, việc đảm bảo tính minh bạch là yêu cầu bắt buộc. Do đó, một trong số các nhiệm vụ chủ chốt của ScrumMaster là thường xuyên thanh tra các tạo tác và thúc giục các bên đảm bảo cho các tạo tác luôn ở tình trạng tốt nhất.



## 5 hiểu lầm về các tạo tác của Scrum

### 1. Sprint Backlog là do ScrumMaster quản lý.

Sprint Backlog là bản kế hoạch công việc của Nhóm Phát triển, do đó nó được Nhóm Phát triển quản lý. ScrumMaster có nhiệm vụ hỗ trợ Nhóm Phát triển đảm bảo sử dụng đúng Sprint Backlog và cập nhật thường xuyên để thông tin được minh bạch.

### 2. Việc ước tính kích thước của các hạng mục Product Backlog được thực hiện bởi Product Owner bởi vì đây là người chịu trách nhiệm quản lý Product Backlog.

Nhóm Phát triển mới là những người ước tính kích thước của các hạng mục Product Backlog, bởi vì họ là những người sản xuất trực tiếp và có hiểu biết tốt nhất về khối lượng công việc dành cho các hạng mục. Nhóm Phát triển ước tính các hạng mục dựa trên dữ liệu về các hạng mục mà mình đã phát triển trước đó.

### 3. Tiến độ phát triển trong từng Sprint được theo dõi và cập nhật bởi Product Owner, bởi vì đây là người chịu trách nhiệm về thành công của sản phẩm. Kể cả biểu đồ Sprint Burndown cũng được duy trì bởi Product Owner.

Nhóm Phát triển là những người chịu trách nhiệm về tiến độ phát triển trong từng Sprint. Kể cả biểu đồ Sprint Burndown cũng do Nhóm Phát triển duy trì. Mặc dù Product Owner chịu

trách nhiệm về thành công của sản phẩm, nhưng Nhóm Phát triển được trao quyền tự quản lý trong từng Sprint để có thể tổ chức sản xuất tốt nhất.

### 4. Để ước tính kích thước của các hạng mục Product Backlog thì đơn vị thời gian tuyệt đối (chẳng hạn như giờ hoặc ngày làm việc) là chính xác hơn các đơn vị đo tương đối (chẳng hạn như điểm).

Việc ước tính kích thước của các hạng mục sử dụng các đơn vị thời gian tuyệt đối là khó chính xác, mất thời gian và khó đạt được đồng thuận. Mục đích của việc ước tính là để lập kế hoạch cho tốt, chứ không phải là để có được độ chính xác cao nhất về mặt thời gian. Các phương pháp ước tính tương đối hoạt động dựa trên dữ liệu thực tế mà nhóm vừa trải qua, do đó hỗ trợ rất tốt cho việc lập kế hoạch hiện tại.

### 5. Định nghĩa Hoàn thành với càng nhiều tiêu chuẩn khắt khe thì càng tốt.

Điều này chỉ đúng nếu năng lực của Nhóm Phát triển đáp ứng được các tiêu chuẩn được đưa ra. Trong quá trình xây dựng và cập nhật Định nghĩa Hoàn thành, nhóm cần cân bằng giữa tiêu chuẩn chất lượng và năng lực của nhóm để đạt được tiêu chuẩn đó. Các tiêu chuẩn này sẽ được nâng cao dần lên cùng sự trưởng thành của nhóm.



## CÂU HỎI ỨNG DỤNG

1. Khi Nhóm Phát triển nhận thấy cần bổ sung thêm một hạng mục vào Product Backlog họ phải làm gì? Tại sao?
2. Sprint chạy được 1 ngày thì Product Owner nhận thấy cần phải hoàn thành sớm một hạng mục có giá trị nào đó và chỉ ta gán luôn vào Sprint Backlog, với cương vị ScrumMaster bạn sẽ làm gì?
3. Tại sao Nhóm Phát triển ước tính nỗ lực cần thiết để hoàn thành các hạng mục trong Sprint Backlog?
4. Những giá trị của biểu đồ Sprint Burndown là gì?
5. Đâu là mối liên hệ giữa Product Backlog và Sprint Backlog?
6. Kết thúc một Sprint thì tạo tác nào nói lên sự thành công của Sprint? Tại sao?
7. Các hạng mục trong Product Backlog được sắp xếp ưu tiên như thế nào? Ai có quyền cao nhất trong việc đó?
8. Bạn giải thích với một đồng nghiệp như thế nào khi anh ta nói: "Product Owner là người quản lý Product Backlog nên sẽ là người ước tính các hạng mục trong đó?"
9. Bạn tư vấn công cụ nào làm Product Backlog và Sprint Backlog cho một Nhóm Scrum trong một Startup cùng làm việc ở một văn phòng? Tại sao?
10. Đâu là những nguyên nhân có thể khiến một Nhóm Phát triển gặp khó khăn trong việc xác định hạng mục nào đó của Product Backlog đã hoàn thành hay chưa?

# 6

## SCRUM TRONG TỔ CHỨC

Trong chương này...

- Các vấn đề khi đưa Scrum vào tổ chức
- Các tình huống áp dụng Scrum
- Quản trị dự án linh hoạt với Scrum
- Các khía cạnh văn hóa cần quan tâm
- Lộ trình áp dụng Scrum
- Nhận diện và vượt qua những trở ngại
- Dẫn dắt sự thay đổi
- Phát triển năng lực và gây dựng tổ chức học tập
- Chuyển đổi Scrum tại các tổ chức khác nhau

“

Scrum dễ hiểu, nhưng khó tinh thông.

Ken Schwaber

# CÁC VẤN ĐỀ TỔ CHỨC CẦN LƯU TÂM

Khi quyết định đưa Scrum vào tổ chức của mình, bạn sẽ phải lường trước và đương đầu một số vấn đề tiềm năng để tổ chức của bạn đón nhận Scrum thành công và gặt hái những lợi ích do Scrum mang lại.

Những vấn đề nổi cộm có thể dễ dàng nhìn ra như: Thay đổi các vai trò công việc của cá nhân và phòng ban hiện có, thay đổi cách thức đánh giá kết quả và vẽ lại lộ trình nghề nghiệp cho hàng loạt vị trí cũ và mới.

Scrum có thể gây xung đột vai trò trong tổ chức: Bạn có thể sẽ phải định nghĩa lại công việc của quản trị dự án do cách thức quản trị dự án đã thay đổi, sắp xếp lại chức năng và nhiệm vụ của những phòng ban chức năng để tổ chức những nhóm Scrum liên chức năng, những phòng ban truyền thống như phòng đảm bảo chất lượng, các phòng kiểm thử, phòng thiết kế,... có thể sẽ biến mất khỏi cơ cấu tổ chức, hoặc sẽ phải định vị lại chức năng và nhiệm vụ.

Song song với việc thay đổi các vị trí, phòng ban, tổ chức của bạn cũng sẽ phải thay đổi lại cách thức đánh giá hiệu suất, định nghĩa lại sự thành công để vừa đảm bảo cá nhân vẫn nỗ lực, lại vừa khuyến khích tinh thần hợp tác; vừa đánh giá đúng giá trị của cá nhân, lại vừa đánh giá thành

tích dựa trên sự chung sức trong tập thể. Đây là vấn đề kỹ thuật, nhưng cũng là vấn đề mang tính giá trị quan cần sự thay đổi sâu sắc trong tư duy của lãnh đạo.

Khi đưa Scrum vào, hàng loạt vai trò mới phát sinh như: ScrumMaster, Product Owner, Agile Coach, Scrum Coach, và những công việc của từng thành viên trong đội phát triển sẽ thay đổi về chất. Điều này đòi hỏi bộ phận quản lý nhân sự và lãnh đạo phải quy hoạch lại lộ trình nghề nghiệp, hệ thống mô tả nghiệp vụ và những chính sách đi kèm để phát triển đội ngũ vững mạnh đáp ứng nhiệm vụ theo cách làm mới mà Scrum mang lại. Không chú trọng tới yếu tố này thì những cách áp dụng Scrum chỉ mang tính bề mặt, thiếu bền vững và khó lòng gặt hái được thành công lâu dài.

Cuối cùng, việc du nhập một cách làm mới, một công cụ mới hay một công nghệ mới về bản chất là một sự thay đổi, lãnh đạo công ty sẽ phải dẫn dắt sự thay đổi để đảm bảo sự chuyển đổi cách thức làm việc được thành công, vượt qua những trở ngại và phản đối để có những thay đổi căn bản, cắm rễ lâu bền vào văn hóa của công ty.

# CÁC TÌNH HUỐNG VÀ MỨC ĐỘ SỬ DỤNG SCRUM



Ở phạm vi hẹp nhất, Scrum được dùng với mục đích **thí điểm**. Khi doanh nghiệp chưa sẵn sàng để dùng Scrum cho các dự án quan trọng, thì mô hình này tỏ ra phù hợp. Theo đó Nhóm Scrum được lập ra và thực hiện một nhiệm vụ (thật hoặc giả lập) ngắn hạn chừng vài Sprint để kiểm tra xem Scrum có phù hợp với doanh nghiệp mình không, đội ngũ có dễ dàng làm chủ Scrum không, và có những vấn đề phát sinh nào. Nhiều doanh nghiệp chọn bước thí điểm là bước khởi đầu trước khi áp dụng đồng loạt Scrum cho toàn bộ tổ chức.

Mức độ phổ biến nhất là các doanh nghiệp sử dụng Scrum như một khuôn khổ mới cho **quản trị dự án**. Theo đó, tùy theo tính trọng yếu mà một số dự án sẽ được tổ chức dưới dạng Scrum. Đây là hình thức đầu tư ngắn hạn, thu lợi nhanh, phù hợp với các công việc thời vụ hoặc doanh nghiệp chưa có chiến lược sử dụng Scrum lâu dài.

Mức độ cao hơn của việc vận dụng Scrum là dùng nó làm quy trình tiêu chuẩn để **phát triển sản phẩm**. Trong trường hợp này toàn bộ đội ngũ phát triển được tổ chức dưới dạng các nhóm Scrum, nếu sản phẩm lớn hơn sẽ áp dụng những cách vận dụng Scrum ở quy mô lớn (tham khảo Chương 8) để tổ chức phát triển sản phẩm. Đây là cách làm đòi hỏi chiến lược lâu dài, đầu tư bài bản cho con người, lộ trình nghề nghiệp, hạ tầng công nghệ và cơ sở vật chất đi kèm.

Trong trường hợp bên trên, Scrum vẫn chỉ đóng vai trò thu hẹp trong đội ngũ phát triển sản phẩm. Trường hợp sử dụng Scrum với phạm vi lớn hơn là khi **toàn bộ tổ chức** tạo lập được văn hóa Scrum để cùng hoạt động. Tổ chức sẽ lấy các giá trị Scrum làm giá trị căn bản của văn hóa công ty, các lãnh đạo sẽ vận dụng tư duy Agile và các quy trình Scrum cho các lĩnh vực vận hành, phát triển sản phẩm và dịch vụ. Thông thường, các công ty nhỏ hơn (như các startup và người đến vài chục người) cho đến những công ty cỡ vừa sẽ dễ dàng vận dụng Scrum theo cách này. Các doanh nghiệp lớn hơn sẽ đòi hỏi một chiến lược chuyển đổi bài bản cùng với các kế hoạch quản trị thay đổi đường dài.

## Lộ trình đơn giản để đưa Scrum vào tổ chức



## Câu chuyện từ thực địa:

EcoLife<sup>1</sup> là một startup công nghệ nhỏ đã được 3 năm, chia thành ba phòng: Vận hành, Công nghệ, Marketing & Bán hàng. Trước chuyển đổi, giám đốc là người đưa ra ý tưởng kinh doanh, sau đó ra đề bài cho từng bộ phận. Phòng Marketing & Bán hàng cũng như phòng Vận hành đưa yêu cầu cho trưởng phòng Công nghệ. Các yêu cầu chia thành hai loại: yêu cầu dài hạn mang tính chiến lược và yêu cầu nhỏ cần gấp. Phòng Công nghệ có quy trình Waterfall để phát triển và sử dụng nhiều công cụ khác nhau, tuy nhiên những yêu cầu thay đổi liên tục do tình hình kinh doanh cũng như yêu cầu không rõ ràng – chỉ mang tính sơ lược. Sau đó phòng Công nghệ yêu cầu các thành viên khác đưa ra yêu cầu chi tiết nhưng do không nắm rõ phương pháp và không biết về hệ thống bên dưới nên các yêu cầu vẫn khó thực thi.

Vấn đề với cách làm cũ là hầu hết các yêu cầu dài hạn mang tính chiến lược bị trượt hạn chót và chậm. Các yêu cầu nhỏ không thống kê được. Giám đốc cũng như hai phòng ban còn lại đều rất không hài lòng về phòng Công nghệ. Phòng Công nghệ phải làm việc quá thời gian làm cho các thành viên đều mệt mỏi. Chất lượng sản phẩm càng ngày càng không kiểm soát được.

## Nhân rộng Scrum tại doanh nghiệp gia công phần mềm

Mục tiêu giai đoạn chuyển đổi Agile đầu tiên là tăng chất lượng sản phẩm và giảm thời gian hoàn thành các yêu cầu, ưu tiên các yêu cầu chiến lược. Trong quá trình tư vấn, nhà tư vấn đã lựa chọn trưởng phòng kinh doanh làm Product Owner vì người này hiểu rõ sản phẩm (dù không định hướng) và có khả năng dành 40% thời gian để thống nhất yêu cầu dài hạn và yêu cầu ngắn hạn với các bên và làm rõ yêu cầu. Trưởng phòng Công nghệ làm ScrumMaster. Các thành viên còn lại hình thành Nhóm Phát triển. Dùng Scrum với Sprint 1 tuần vì sản phẩm không yêu cầu quá khắt khe và nhóm có thể hoàn thành được cũng như nhu cầu thay đổi nhanh. Sau khi tư vấn, Nhóm Scrum cùng với giám đốc được đào tạo cơ bản về Scrum trong một khóa đào tạo cấp tập 2 tuần dựa trên khung kiến thức của Scrum Alliance. Cuối cùng, nhóm được huấn luyện trong vòng hai tháng để đảm bảo không làm sai, có thói quen tốt và năng lực để cải tiến mà không phá vỡ những nguyên lý quan trọng của Scrum.

Sau giai đoạn chuyển đổi đầu tiên, nhóm Công nghệ đã làm chủ được tình hình.

Những ước lượng của họ đã đáng tin hơn. Lượng lỗi giảm xuống hơn 3/4, hầu hết các yêu cầu chiến lược được thực hiện như kỳ vọng và một số lần bị chậm tiến độ phát hành cũng đã được thông báo sớm với giám đốc và các bên liên quan để điều chỉnh kịp thời các kế hoạch hoạt động khác.

Chứng kiến sự thành công của phòng Công nghệ, trưởng phòng Marketing cũng mong muốn đưa Scrum vào phòng mình để nâng cao tính linh hoạt. Các chiến dịch marketing đang không đạt được kết quả như kỳ vọng. Trung bình khoảng 2 tháng nhóm mới nhận biết được phản hồi từ thị trường để thay đổi chiến thuật. Các thành viên bị động, không có được nhiều sự sáng tạo đáng kể, lỗi xảy ra nhiều. Mục tiêu chuyển đổi Agile ở phòng Marketing là rút ngắn vòng phản hồi từ 2 tháng xuống còn 2 tuần và giảm 2/3 lỗi gấp phai ở các nội dung marketing. Trong quá trình tư vấn, trưởng phòng Marketing được lựa chọn vào vai trò Product Owner, một thành viên có nhiều kinh nghiệm được chọn là ScrumMaster, các thành viên còn lại tham gia Nhóm Phát triển. Vì kinh nghiệm chuyển

đổi Agile đã có khi thực hiện tại nhóm Công nghệ, Nhóm Marketing chỉ mất 6 tuần để đạt được mục tiêu đề ra.

Từ những thay đổi tích cực trong kết quả công việc và tinh thần làm việc của nhân viên, ban giám đốc quyết định chính thức hóa Agile ở toàn bộ công ty sau 5 tháng triển khai. Phòng Vận hành được tư vấn sử dụng Scrumban thay vì Scrum.

Lượng công việc tăng nhiều hơn, áp lực tăng hơn, tuy nhiên các mục tiêu chính vẫn đạt được sự thích ứng với các thay đổi. Xung đột giữa các bộ phận có thể giải quyết được và mang tính chất giải quyết vấn đề.

## Nhận diện và vượt qua các trở lực từ yếu tố văn hóa

Một trong những lo ngại rất phổ biến ở những nơi chuẩn bị du nhập Scrum là lo ngại những đặc thù văn hóa bản địa sẽ không phù hợp với văn hóa làm việc của Scrum vốn đánh giá cao giá trị tự chủ, làm việc nhóm, hướng đến khách hàng và sẵn sàng thay đổi để thích nghi.

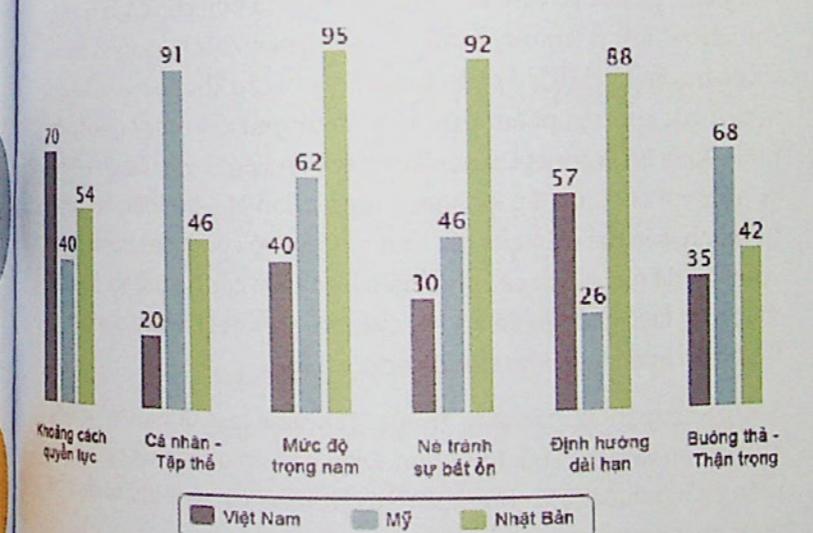
Chúng ta có thể vận dụng mô hình do Tiến sĩ Hofstede đề xuất để hiểu thêm về đặc thù văn hóa quốc gia và có cách để vượt qua những trở ngại thuộc phạm trù văn hóa. Tất nhiên mô hình này thật khó mà đầy đủ, nhưng trong chừng mực nào đó, nó hữu ích để việc du nhập Scrum vào một tổ chức mới thuận lợi hơn.

Bạn có thể truy xuất thông tin về mức độ của các yếu tố văn hóa theo khung Hofstede trên trang web: <https://geert-hofstede.com/countries.html>

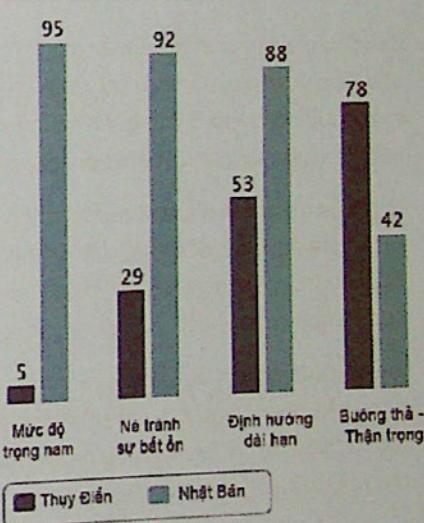


## “Người Nhật không phù hợp với Scrum”

Việt Nam  
sо sánh với Mỹ và Nhật Bản



Thụy Điển  
sо sánh với Nhật Bản



Chúng tôi nghe câu “Người Nhật không phù hợp với Scrum” từ nhiều đồng nghiệp người Nhật Bản. Thật kỳ lạ, những tư duy nguyên thủy của Agile, Scrum chịu sự ảnh hưởng rất lớn của tư duy tinh gọn (Lean Thinking), nhưng Agile lại không lan tỏa nhanh ở đó. Bạn hãy nhìn vào những biểu đồ về các chỉ số trong mô hình Hofstede so sánh bốn nước: Nhật Bản, Việt Nam, Mỹ và Thụy Điển – nơi Agile rất phổ biến và dễ được chấp nhận, và tự mình rút ra nhận xét.

## Câu chuyện từ thực địa: Scrum tại doanh nghiệp lớn

VieTop<sup>1</sup> là công ty cung cấp dịch vụ phát triển phần mềm cho các công ty thành viên khác của cùng công ty mẹ là một tập đoàn viễn thông lớn. Tất cả các dịch vụ phần mềm của VieTop đều chia sẻ trên cùng một nền tảng và cơ sở dữ liệu khách hàng và nhân viên chung.

Ban đầu công ty phát triển phần mềm rất chậm (cycle time rất lớn), hầu hết các yêu cầu trung bình (mất khoảng 30 ngày làm việc với một người) đều mất 3 tháng. Với những yêu cầu phức tạp hơn phải mất 6 tháng thậm chí cả năm trời mới xong.

Mục tiêu giai đoạn đầu của chương trình chuyển đổi Agile là tạo ra các nhóm Agile để giảm thời gian phát triển xuống còn 1 tháng với yêu cầu trung bình và tối đa 4 tháng với yêu cầu lớn. Các yêu cầu nhỏ và gấp có thể hoàn thành trong vòng 1 tuần.

VieTop được tư vấn để tổ chức các nhóm theo sản phẩm hoặc một dòng sản phẩm. Sau đó một số nhóm được chọn để tư vấn, đào tạo và huấn luyện. Sau 6 tháng những nhóm đạt được mục tiêu sẽ chuyển giao dần sang các nhóm còn lại.

Sau hai năm, VieTop nhận thấy một số vấn đề khác còn tồn đọng là sản phẩm ngày càng khó dùng, thời gian phát triển vẫn khá chậm so với nhu cầu mới, các thành viên làm việc không năng suất vì luôn phải chạy theo các hạn chót bất ngờ.

Mục tiêu của chương trình tiếp theo là trong vòng 8 tháng chuyển đổi là giảm thời gian phát triển xuống 70% và duy trì chất lượng sản phẩm.

Công ty mẹ được tư vấn để trao quyền nhiều hơn cho công ty con, cho phép VieTop có quyền tư vấn ngược lại khách hàng và thậm chí có quyền từ chối khi phân tích thấy yêu cầu không mang lại giá trị tích cực cho sản phẩm thay vì thực thi yêu cầu "được giao" như trước. Toàn bộ Product Owner làm việc hằng ngày với nhau và định kỳ ngồi với nhau để lập kế hoạch tổng quan cấp độ toàn bộ công ty. Thành viên đại diện của các nhóm đồng bộ công việc hằng ngày với nhau để giải quyết các phần tích hợp giữa các hệ thống. Sau đó thực hiện việc đào tạo về Scrum cho các Product Owner cũng như thực hiện huấn luyện họ trong vòng 3 tháng.

Sau chương trình, các phụ thuộc, các yêu cầu được giải quyết nhanh hơn và cũng chủ động hơn. Điều đó làm giảm thời gian phát triển xuống dưới 70% như mục tiêu đặt ra và các sản phẩm duy trì được chất lượng.



## Dùng và Nghĩ

Trong một hội thảo Agile năm 2009, chuyên gia Jean Tabaka có liệt kê 11 nguyên nhân hàng đầu dẫn đến thất bại trong việc áp dụng Agile như dưới đây. Bạn có kế hoạch nào để tránh những điều này?

### Top 11 nguyên nhân thất bại trong việc áp dụng Scrum

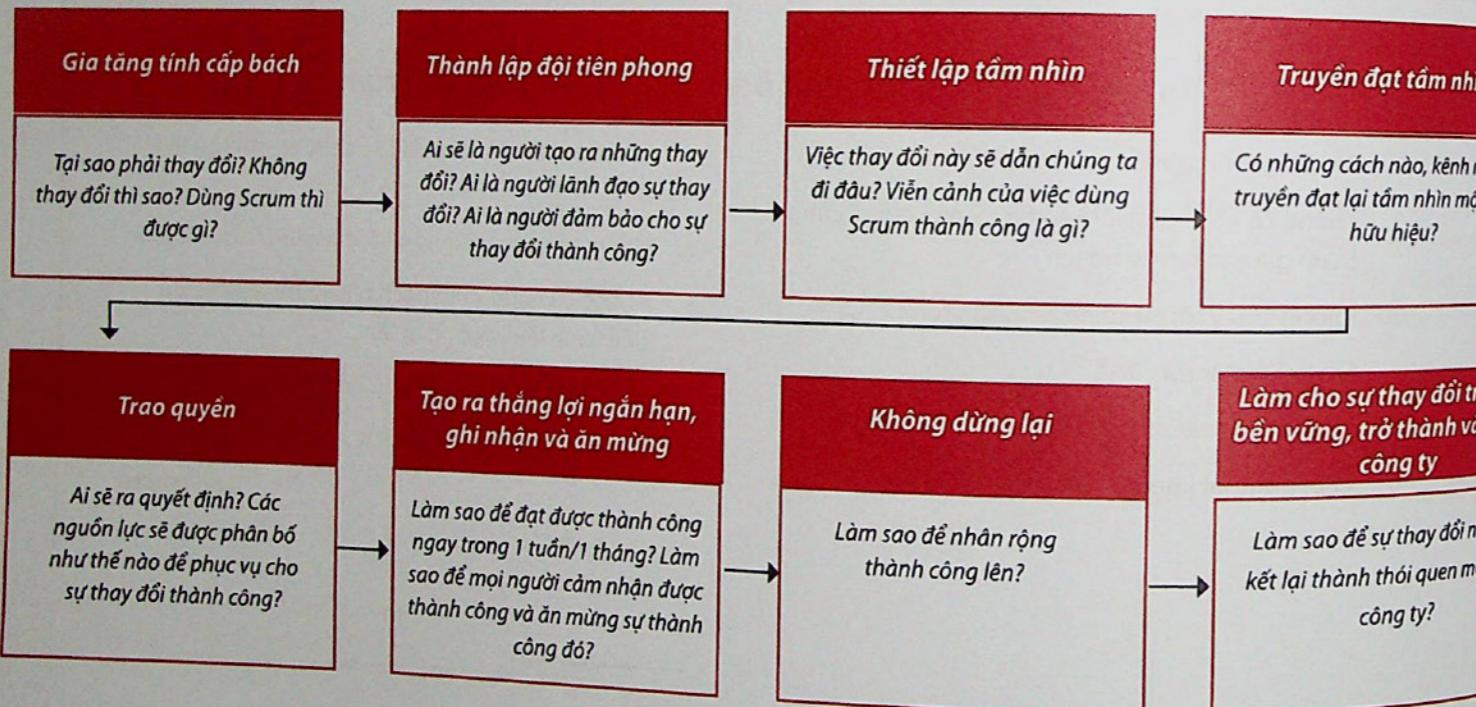
1. Sử dụng không hiệu quả hoạt động cải tiến
2. Không có khả năng lôi kéo tất cả mọi người cùng tham gia vào lập kế hoạch
3. Không chú ý tới cơ sở hạ tầng cần thiết
4. ScrumMaster tồi
5. Product Owner không giữ được sự nhất quán
6. Khôi phục lại khuôn mẫu trước đây
7. Chỉ quan tâm tới "cam kết sổ sách" từ phía quản lý điều hành
8. Nhóm thiếu thẩm quyền và khả năng ra quyết định
9. Không có người chịu trách nhiệm truyền đạt khi tiến hành làm việc phân tán
10. Văn hóa của tổ chức không hỗ trợ việc học tập
11. Từ chối chấp nhận một cách gay gắt

Jean Tabaka<sup>1</sup>

<sup>1</sup> Theo: <https://www.stickyminds.com/article/11-ways-agile-adoptions-fail>

# QUẢN LÝ SỰ THAY ĐỔI

Thay đổi cách nghĩ cách làm là sự thay đổi căn cơ, mất công mất sức mất thời giờ và đặc biệt đòi hỏi sự lưu tâm của các cấp lãnh đạo. Phạm vi áp dụng càng lớn thì việc quản lý thay đổi càng quan trọng. Chúng ta có thể vận dụng mô hình quản trị sự thay đổi của John Kotter<sup>1</sup> để quản lý sự thay đổi khi tiếp nhận Scrum vào tổ chức. Bạn cần có một kế hoạch thay đổi chu toàn, triển khai và giám sát, đánh giá liên tục và thích nghi để hướng tới mục tiêu định trước. Một kế hoạch thay đổi có thể được tổ chức dưới dạng dự án gồm một Nhóm Scrum, với "Phản tăng trưởng" chính là những thay đổi bạn muốn có ở trong tổ chức.



Cần lưu ý rằng, trong khi trình tự các bước để tạo lập sự thay đổi do Kotter giới thiệu tương đối tuyến tính, chúng ta nên tiếp cận việc những bước chuyển đổi này theo cách Agile hơn. Dưới đây là một số gợi ý:

## Nhìn thấy-Cảm nhận-Thay đổi

Hãy trực quan hóa vấn đề, nhu cầu thay đổi, cách thức thay đổi và những lợi ích mà sự thay đổi mang lại. Hãy để mọi người cảm nhận thấy sự thay đổi là cần thiết, cảm nhận luồng gió thay đổi phả vào mỗi góc của tổ chức. Bằng các nỗ lực tác động vào cảm xúc và nhận thức của từng cá nhân, sự thay đổi sẽ diễn ra thuận lợi và mạnh mẽ.

## Giảm phạm vi thay đổi

Các kế hoạch lớn và ào ạt trong quản trị thay đổi thường vấp phải sự kháng cự lớn, quá phức tạp và ít khi thành công. Hãy đặt các mục tiêu nhỏ, khả thi để đạt được những chiến thắng nhỏ (quick-wins). Việc này không chỉ gia tăng sự tự tin trong chuyển đổi, mà còn thu lại nhiều kinh nghiệm tốt để tiếp tục thúc đẩy các thay đổi tiếp theo. Nhiều thay đổi cùng một lúc thường đồng nghĩa với tỷ lệ thất bại cao.

## Thích ứng hơn là theo một kế hoạch định sẵn

Các thay đổi thường phức tạp, do vậy hãy sẵn sàng cho các thay đổi trong thực thi. Đó cũng là cách tiếp cận theo phong cách Agile, việc dẫn dắt thay đổi cũng nên rất thực nghiệm (empirical) – thu ngắn phạm vi, vòng phản hồi ngắn và điều chỉnh liên tục trên con đường đạt đến mục tiêu cuối cùng.

## Từ dưới lên và từ trên xuống cùng lúc

Không nên chỉ trông chờ vào các chương trình thay đổi từ cấp cao nhất. Bạn cũng không nên chỉ trông chờ vào sự tự giác từ dưới lên. Một quá trình thay đổi cần có sự tương tác. Lãnh đạo cần thường xuyên truyền đạt những tầm nhìn của sự thay đổi, làm rõ từng chi tiết, liên kết những sự thay đổi với các mục tiêu của tổ chức, trong khi có các hỗ trợ về quy trình, công nghệ, nguồn lực và con người để việc thay đổi được thuận lợi. Trong khi đó, các nhà quản lý hoặc ScrumMaster cần biết cách tạo lập những thay đổi từ cấp trực tiếp làm việc (các nhóm Scrum), kể các câu chuyện thành công/thất bại, rút ra các bài học và thúc đẩy sự thay đổi lên cấp độ mới.

## Tập trung vào tháo gỡ khó khăn

Khó khăn có thể đến từ tri thức nhưng cũng có thể đến từ quy trình và đặc biệt là từ con người. Hãy sẵn sàng tư thế để đón nhận những sự phản đối, trực trặc từ sớm. Phân tích thấu đáo nguyên nhân gốc rễ, giải quyết triệt để càng sớm càng tốt để thúc đẩy sự thay đổi tiến lên phía trước. Vai trò của các lãnh đạo thay đổi hoặc ScrumMaster có tay nghề luôn rất quan trọng trong việc thiết lập các nền tảng tốt cho sự thay đổi, cũng như có các cơ chế hữu hiệu để tháo gỡ khó khăn cho tổ chức.

<sup>1</sup> Xem thêm *Dẫn dắt sự thay đổi*, J. Kotter, NXB Khoa học xã hội, 2018

# THAY ĐỔI TƯ DUY

Theo nhà tâm lý học nổi tiếng Howard Gardner, sự thay đổi trong tư duy con người diễn ra rất chậm chạp. Chúng ta cần hiểu rõ từng đối tượng trước khi có được các chương trình thay đổi hiệu quả.

Cũng theo Gardner<sup>1</sup>, có một số yếu tố đòn bẩy để tác động vào sự thay đổi của tư duy con người (7 Rs), bao gồm:

**Lý luận (Reason):** Dùng lý lẽ để thuyết phục bao gồm việc trình bày tất cả các thông tin về ý tưởng, cả mặt tốt lẫn mặt xấu.

**Nghiên cứu (Research):** Thu thập các bằng chứng khoa học, các con số và thông tin liên quan đến ý tưởng, từ đó thuyết phục thay đổi.

**Cộng hưởng (Resonance):** Tạo mối liên kết với người được thuyết phục. Sự cộng hưởng đến từ mối quan hệ cá nhân, sự đáng tin cậy, cảm giác tôn trọng, thông qua giao tiếp, tương tác, ẩn dụ... để tạo sự thay đổi trong nhận thức. Gardner cho rằng đây là công cụ mạnh nhất để thay đổi tư duy.

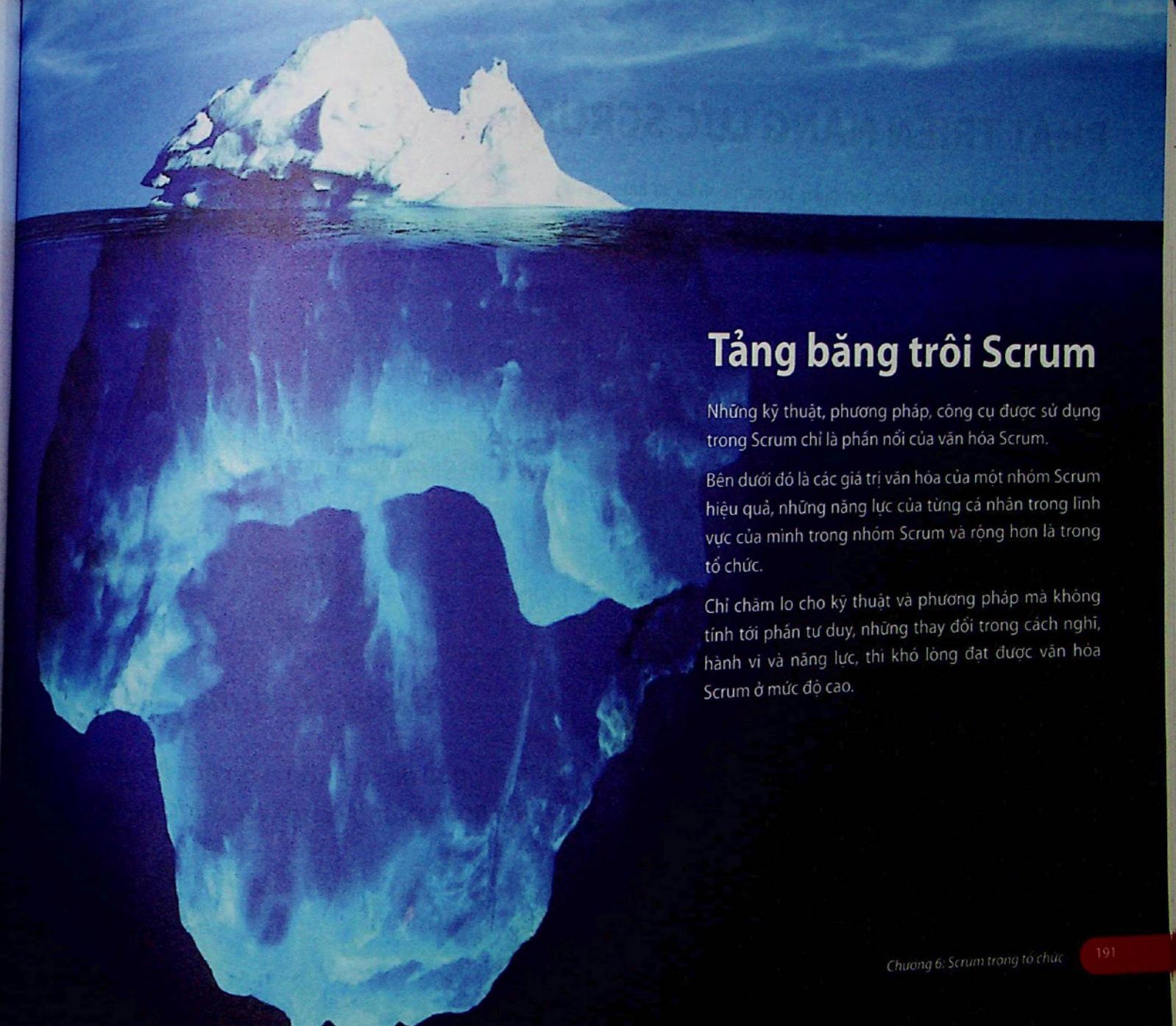
**Diễn đạt theo nhiều cách (Representational Redescriptions):** Thông tin có thể được chế biến lại, diễn giải lại dưới các dạng thức khác nhau như văn bản, đồ họa, âm thanh, hình ảnh, biểu đồ và các chuyện kể để việc tiếp thu ý tưởng dễ dàng, thuận tiện và gần gũi hơn với người tiếp nhận.

**Tài nguyên và Phần thưởng (Resources and Rewards):** Thưởng cho người tiếp nhận ý tưởng mới để thay đổi hành vi. Phần thưởng ở đây có thể là bằng vật chất, hoặc tinh thần. Ngoài ra, giá trị có được sau khi tiếp nhận ý tưởng mới cũng có tác động thay đổi tư duy về ý tưởng đó.

**Các sự kiện thực tế (Real world events):** Liên hệ các sự kiện trong thực tế đối với ý tưởng mới từ đó thay đổi nhận thức về vấn đề đang tiếp cận. Các sự kiện này có thể gần gũi ngay sát nơi làm việc, hoặc cũng có thể là các sự kiện đang diễn ra ở đâu đó trên thế giới như: các phát kiến khoa học nổi bật, làn sóng công nghệ mới, chiến tranh thương mại, khủng hoảng tài chính toàn cầu, hay xung đột vũ trang, v.v. Khi con người cảm thấy đe dọa, thì nhận thức có thể cũng sẽ thay đổi.

**Sự chống cự (Resistance):** Nhìn ra những yếu tố cản trở và phản đối chính yếu (hiển hiện và ngầm ẩn) đối với các ý tưởng và tháo gỡ từng thứ một. Có nhiều cản trở đến từ những lối tư duy theo quán tính, trình độ hiểu biết, hoặc do các quy tắc ràng buộc quá khắt khe cản trở khả năng tiếp thu ý tưởng mới.

ScrumMaster hoặc nhà lãnh đạo thay đổi có thể vận dụng các đòn bẩy này phù hợp với từng đối tượng và thời điểm để tạo những thay đổi trong tư duy của từng cá nhân trong tổ chức. Song song với việc tạo lập các thay đổi về quy trình, việc thay đổi tư duy thường đóng vai trò to lớn mang tính quyết định đối với sự chuyển biến trong tổ chức.



## Tảng băng trôi Scrum

Những kỹ thuật, phương pháp, công cụ được sử dụng trong Scrum chỉ là phần nổi của văn hóa Scrum.

Bên dưới đó là các giá trị văn hóa của một nhóm Scrum hiệu quả, những năng lực của từng cá nhân trong lĩnh vực của mình trong nhóm Scrum và rộng hơn là trong tổ chức.

Chi chăm lo cho kỹ thuật và phương pháp mà không tính tới phần tư duy, những thay đổi trong cách nghĩ, hành vi và năng lực, thì khó lòng đạt được văn hóa Scrum ở mức độ cao.

<sup>1</sup> Thay đổi tư duy, H. Gardner, NXB Khoa học xã hội, 2014

# PHÁT TRIỂN NĂNG LỰC SCRUM

Không thể thành công với việc vận dụng Scrum mà thiếu sự lưu tâm đúng mức tới việc học tập ở các cấp. Kể cả ở phạm vi hẹp nhất, với mục đích là học hỏi, việc thí điểm Scrum cũng đòi hỏi những nỗ lực học tập nghiêm túc. Việc học hỏi Scrum và những tri thức liên quan có thể được tổ chức dưới dạng các dự án học tập ngắn hoặc dài hạn, tùy quy mô và phạm vi. Trong đó ghi rõ mục tiêu (học gì) và cách thức đạt được mục tiêu đó (học thế nào). Bạn có nhiều lựa chọn để học Scrum, từ đọc sách, xem các bài giảng trên YouTube, tới việc tham gia các khóa học tập trung hoặc các khóa học online về Scrum và Agile trên các cổng MOOC<sup>1</sup> lớn như Coursera hoặc edX. Ngoài ra, bạn có thể cân nhắc mời các nhà đào tạo, huấn luyện viên Scrum chuyên nghiệp bên ngoài về kèm cặp đội ngũ để gia tốc quá trình học hỏi và vận dụng Scrum tại công ty. Nếu công ty đã có sẵn những chuyên gia nội bộ, việc tổ chức học tập dưới dạng các seminar, workshop kết hợp với kèm cặp trong công việc (on-job-training) là một lựa chọn rất tốt để thúc đẩy học hỏi. Ở quy mô lớn hơn, bạn có thể có những chương trình học tập diện rộng, với nhiều hoạt động đa dạng hơn.

Cần lưu ý việc thành thực một kỹ năng nào đó luôn cần thời gian, tránh đốt cháy giai đoạn. Bạn cần phải trải qua các bước học hỏi, bắt chước, trước khi có thể tự mình xoay sở và giải quyết các vấn đề (theo từng bước: làm thô, làm tinh hơn, rồi tiến đến thành thạo). Do vậy, việc tránh ScrumBut không đơn thuần là vấn đề làm Scrum đúng cách, mà còn là vấn đề lĩnh hội và rèn luyện kỹ năng căn bản trước khi điều chỉnh. Không chỉ các nhà huấn luyện Scrum nói điều này, các huấn luyện viên thể thao hay các nhà giáo chuyên nghiệp cũng thấm nhuần các quy tắc học tập tương tự.



## Dùng và Nghĩ

*Làm thế nào để cả nhóm vẫn học hỏi trong khi phải hoàn thành công việc được ủy nhiệm?*

## GÂY DỰNG VĂN HÓA HỌC TẬP

Tùy từng mức độ và điều kiện mà một tổ chức có thể có các chiến lược và chiến thuật khác nhau đối với việc học tập Scrum nói riêng và học tập nói chung. Nhưng cần lưu ý, học tập và quản trị tri thức (cả tiềm ẩn trong các cá nhân dưới dạng kỹ năng, know-how, hay tường minh dưới dạng văn bản, quy trình...) ngày càng trở nên quan trọng và trở thành động lực mới của tổ chức. Do vậy việc quan tâm tới xây dựng một tổ chức học tập (learning organization) không còn là chủ đề lạ lẫm với nhiều công ty hiện đại. Các công ty có thể gây dựng tổ chức học tập theo từng cấp độ từ hẹp tới lớn. Tiếp cận từ dưới lên, công ty có thể khuyến khích và thưởng những nỗ lực học tập chủ động từ phía cá nhân thông qua các chương trình học tập trung hoặc học trực tuyến với chi phí hợp lý. Mức độ tiếp theo, công ty chủ động đưa việc học tập vào trong quy trình làm việc. Điều này dễ dàng thực hiện khi vận dụng Scrum

do các hoạt động cải tiến retrospective hay kaizen về bản chất là việc học tập nhóm được nhúng trong công việc. Tổ chức các không gian chia sẻ kiến thức, thúc đẩy nghiên cứu và nâng cao trình độ với các cộng đồng thực hành (community of practices) hay các nhóm cùng sở thích (interest groups). Bước tiếp theo là lập các đội ngũ chuyên biệt trong việc tổng hợp các tri thức từ bên trong hay bên ngoài tổ chức để thúc đẩy đổi mới một cách có hệ thống và chủ đích. Có thể cân nhắc tới việc thiết lập một hệ thống quản trị tri thức ở mức công ty và duy trì lâu dài. Cuối cùng, lãnh đạo có thể có một chiến lược nhất quán cho việc lấy tri thức làm trọng tâm trong học thuyết quản trị của công ty, và hướng sự quản trị ở các cấp sang hình thức quản trị dựa vào tri thức (knowledge-based management) nhằm biến tri thức và sự học hỏi trở thành động lực phát triển quan trọng của tổ chức.

<sup>1</sup> MOOC: viết tắt của Massive Open Online Courses (các khóa học trực tuyến đại chúng mở).



## Dùng và Nghĩ

Trên tạp chí *Harvard Business Review* năm 2015, bài báo “*Tại sao một tổ chức không học hỏi*”<sup>1</sup> của hai tác giả Gino và Staats có đưa ra một số câu trả lời quan trọng. Đây là một phần của danh sách:

1. Sợ thất bại
2. Ngủ quên trên chiến thắng trong quá khứ
3. Không tin là mình có thể “khá được” vì đã thử ràng buộc
4. Đổ tại số phận
5. Quá bận để học
6. Quy trình quá cứng nhắc, không có chỗ cho học hỏi
7. Ỷ lại chuyên gia, không cần khuyến khích học hỏi ở các cấp
8. Khoán trăng việc học hỏi của tổ chức cho bộ phận đào tạo của công ty

Bạn thấy gì từ danh sách này?

<sup>1</sup> Xem <https://hbr.org/2015/11/why-organizations-dont-learn>

## ĐO LƯỜNG VÀ ĐÁNH GIÁ HIỆU SUẤT

Khi chuyển đổi Scrum ở quy mô lớn, tổ chức cần xây dựng những bộ đo hữu ích để phục vụ việc ra quyết định ở các cấp, giảm thiểu các nỗ lực thừa thãi trong quá trình nỗ lực đạt được mục tiêu của tổ chức. Tuy từng tổ chức mà bộ công cụ đo đếm là khác nhau. Chúng có thể bao gồm: các bộ đo về *năng suất* (ví dụ: lượng công việc hoàn thành qua mỗi Sprint), bộ đo về *giá trị* (ví dụ: giá trị kinh tế trên mỗi nhóm), bộ đo về *chất lượng* (ví dụ: tỷ lệ lỗi hoặc lượng thời gian ngắt dịch vụ), hoặc bộ đo về *phát triển bền vững* (ví dụ: chỉ số hạnh phúc). Các chỉ số này nên được thiết lập, theo dõi và minh bạch hóa đối với tổ chức để đảm bảo lợi ích tối đa.

Bên cạnh đó, những thay đổi trong cách đánh giá hiệu suất nhân viên cũng cần được tính đến. Thay vì đánh giá mỗi năm một lần, tần suất đánh giá trong các nhóm Scrum có thể nhiều hơn, có tính phản hồi cao hơn để hướng tới sự cải thiện thay vì chỉ dành cho mục đích ghi công như cách đánh giá truyền thống.

Pia-Maria Maria, tác giả cuốn sách “*Agile People*”, gợi ý rằng một cuộc hội thoại đánh giá nhân viên kiểu mới trong nhóm Agile có thể bao gồm các đặc trưng rất khác:

- Cuộc hội thoại cởi mở, không đánh giá (non-judgment)
- Để người khác cung cấp thêm thông tin về hiệu suất của nhân viên đó, bên cạnh quản lý trực tiếp
- Không trực tiếp liên hệ giữa hiệu suất với thưởng phạt trong suốt quá trình hội thoại
- Một năm hội thoại nhiều lần

<sup>1</sup> Xem *Agile People: A Radical Approach for HR & Managers*, Pia-Maria Thoren, NXB Lioncrest, 2018.

- Tập trung vào cải thiện hiệu suất trong tương lai, không tập trung vào đánh giá thành tích trong quá khứ
- Để nhân viên cùng lựa chọn cách thức hội thoại
- Để nhân viên tự đánh giá về đóng góp của bản thân vào thành công chung của cả nhóm qua mỗi Sprint
- Không gọi nó là buổi đánh giá hiệu suất (performance review) mà có thể gọi là buổi hội thoại cải tiến hoặc trao đổi phản hồi

Bên cạnh đó, hệ thống tưởng thường cũng cần thay đổi để phù hợp với văn hóa nuôi dưỡng của Agile. Thay vì chỉ ghi công những kết quả, cần phải đánh giá cao những nỗ lực và sự trưởng thành (ở cấp độ cá nhân cũng như nhóm).

Đặc biệt, việc đánh giá hiệu quả của nhân viên cần dựa vào việc hoàn thành các mục tiêu nhóm và mục tiêu cá nhân. Tốt nhất là các mục tiêu được thiết lập theo công cụ OKR, và chính bản thân nhân viên được biết, liên tục theo dõi và tự đánh giá các kết quả đạt được. Bản thân chu kỳ đặt mục tiêu ngắn theo OKR (thường là theo quý hoặc ngắn hơn) cũng giúp nhân viên có được phản hồi nhiều hơn về hiệu suất của mình, của nhóm mình để từ đó có các phương án thích ứng tốt hơn cho tương lai.

Văn hóa Agile đề cao học thuyết tạo động lực dựa trên sự tự chủ (*Autonomy*), phát triển năng lực (*Mastery*) và mục đích của công việc (*purpose*). Do đó, các phần thưởng, nếu có, thì cần tập trung thường cho nhóm thay vì thưởng cá nhân, dân chủ hóa việc ghi nhận công lao của từng thành viên (ví dụ: đánh giá ngang hàng) và không nên chú trọng vào các giá trị tài chính.



## 5 hiểu lầm thường gặp

### 1. Scrum chỉ phù hợp với công ty có nhân sự ở trình độ cao.

Thực ra, Scrum được làm ra để tất cả mọi người có thể phát huy được thế mạnh của mình. Ken Schwaber nói: "Scrum works for idiots" (Scrum hữu hiệu ngay cả với kẻ dốt nhất).

### 2. Chỉ cần họp hàng ngày cộng với dùng JIRA là đủ Agile rồi.

Đây là cách tiếp cận rất phổ biến của nhiều người bắt đầu "áp dụng Agile". Cách tiếp cận này nghe rất hấp dẫn vì tính đơn giản của nó: Chỉ cần mua một công cụ đắt tiền nhất thế giới và chọn một kỹ thuật Agile vào, thế là đủ để Agile rồi.

Thực ra cách làm này đi ngược lại với giá trị căn bản của Agile: "Cá nhân và tương tác hơn là quy trình và công cụ". Các công cuộc chuyển đổi Agile nên dành nhiều nỗ lực vào con người, đội ngũ và khách hàng. Tập trung vào năng lực của đội ngũ, xây dựng các nhóm làm việc hiệu năng cao và gây dựng văn hóa thay vì chỉ "đầu tư cho công nghệ". Cố nhiên, việc đưa vào các công cụ tối tân cũng có giá trị lớn của riêng nó.

### 3. Doanh nghiệp startup không cần quy trình, vì phải đi với tốc độ nhanh nên không cần quan tâm tới Scrum vậy.

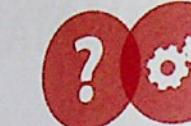
Thực ra chính việc xây dựng một Nhóm Scrum hiệu năng cao mới giúp doanh nghiệp khởi nghiệp có khả năng di chuyển với tốc độ nhanh vì có khả năng chuyển giao giá trị sớm và thường xuyên.

### 4. Chuyển đổi Agile tức là lãnh đạo trao hết quyền cho cấp dưới, để mình có nhiều thời gian đi chơi golf.

Thực ra, mấu chốt của trao truyền không phải là để không phải làm gì nữa, mà là ai giỏi với việc gì nhất thì nên được giao việc đó vì điều đó có thể mang lại hiệu quả cao nhất. Trao quyền thường phải đi kèm với giám sát (từ phía cấp trên) và trách nhiệm giải trình (từ phía cấp dưới) thì mới mang lại hiệu quả cao, nếu bạn không muốn phá vỡ hệ thống quản lý.

### 5. Chỉ cần thiết lập quy trình Scrum, sau đó giao hết cho QA kiểm soát là được.

Scrum không phải là quy trình, mà nó là một trò chơi (cho nên tài liệu Hướng dẫn Scrum được coi như cuốn sách mỏng định nghĩa luật chơi) mà có một số thứ được đặt theo trình tự nhất định để đạt hiệu quả cao. Scrum thật sự nằm ở năng lực của cá nhân, của nhóm và của tổ chức trong việc giải quyết vấn đề và đạt được các mục tiêu thử thách. Việc thiết lập quy trình và yêu cầu "tuân thủ" là một cách làm có thể hữu ích trong hệ thống khác, nhưng với Scrum có thể là lợi bất cập hại vì nó gây mất tập trung vào cái cần phải tập trung.



## CÂU HỎI ỨNG DỤNG

1. Tại sao tổ chức của bạn cần phải thay đổi? Nếu không thay đổi thì sao?
2. Tổ chức của bạn được gì khi áp dụng Scrum?
3. Liệt kê những thuận lợi khi tổ chức của bạn áp dụng Scrum về mặt văn hóa, cơ cấu tổ chức, con người, sự ủng hộ của lãnh đạo, v.v.?
4. Liệt kê những khó khăn khi bạn đưa Scrum vào tổ chức của mình về mặt văn hóa, cơ cấu tổ chức, con người, sự ủng hộ của lãnh đạo, v.v.?
5. Những ai nên tham gia nhóm tiên phong tạo sự thay đổi trong công ty bạn nếu áp dụng Scrum?
6. Bạn sẽ bắt đầu từ đâu để đưa Scrum vào tổ chức của mình?
7. Các quản lý sẽ có thay đổi công việc như thế nào nếu như tổ chức của bạn áp dụng Scrum?
8. Cần áp dụng những phương thức đào tạo, huấn luyện gì cho thành viên của nhóm/công ty của bạn khi áp dụng Scrum?
9. Định nghĩa Hoàn thành nên như thế nào để đảm bảo tiêu chuẩn chất lượng của tổ chức của bạn?
10. Có nên thay đổi hệ thống công cụ trong tổ chức để đảm bảo thông tin minh bạch nếu tổ chức của bạn áp dụng Scrum? Việc thay đổi nên như thế nào?