

# SCRUM CĂN BẢN

**Pete Deemer**

Viện đào tạo Scrum (ScrumTL.com)

**Gabrielle Benefield**

Viện đào tạo Scrum (ScrumTL.com)

**Craig Larman**

craiglarman.com

**Bas Vodde**

Odd-e.com

Dịch: Nguyễn Ngọc Anh, Phạm Anh Đới, Nguyễn Việt Khoa, Nguyễn Ngọc Tú

Hiệu đính: Dương Trọng Tấn

**phiên bản 1.2**

Nhấn gửi bạn đọc: Hiện đã có nhiều tài liệu mô tả ngắn gọn về Scrum trên mạng Internet, còn quyển sách này hướng tới việc cung cấp kiến thức chi tiết về hoạt động triển khai Scrum. Tuy nhiên, tài liệu này không phải là điểm cuối cùng trong việc học Scrum; các nhóm có ý định áp dụng Scrum nên trang bị cho mình các kiến thức trong *Agile Project Management with Scrum (Quản lý dự án với Scrum)* hoặc *Agile Software Development with Scrum (Phát triển phần mềm linh hoạt với Scrum)* của Ken Schwaber , và tham gia các khóa huấn luyện chất lượng cao về Scrum; các thông tin chi tiết có thể tham khảo tại [scrumalliance.org](http://scrumalliance.org). Chúng tôi xin gửi lời cảm ơn đến Ken Schwaber, Tiến sĩ Jeff Sutherland và Mike Cohn về những thông tin quý báu mà họ cung cấp.

**© 2010 Pete Deemer, Gabrielle Benefield, Craig Larman, Bas Vodde**

## Phương thức phát triển phần mềm truyền thống

Phương pháp phát triển phần mềm truyền thống, được sử dụng ở cả các công ty lớn lẫn nhỏ, thường là một chu trình tuần tự được gọi là “thác nước” (mô hình Waterfall). Có nhiều biến thể khác của mô hình thác nước (như V-Model chẳng hạn), nhưng nhìn chung đều bắt đầu với giai đoạn lập kế hoạch chi tiết, ở giai đoạn này sản phẩm đầu ra được xem xét cẩn thận, được thiết kế và viết tài liệu rất chi tiết. Những công việc cần thiết cho hoạt động thiết kế được xác định, và được tổ chức thông qua sử dụng các công cụ như biểu đồ Gantt và các ứng dụng như Microsoft Project. Nhóm phát triển ước lượng thời gian thực hiện dự án bằng cách cộng dồn thời gian ước tính của các cá nhân tham gia dự án ở mỗi bước phát triển. Một khi các bên liên quan xem xét kỹ lưỡng và phê chuẩn kế hoạch, nhóm phát triển mới bắt đầu tiến hành công việc. Các thành viên trong nhóm hoàn thành phần công việc thuộc về chuyên môn của mình và chuyển giao cho người khác trong dây chuyền sản xuất. Khi công việc hoàn thành, sản phẩm được đưa sang bộ phận kiểm thử (đôi khi gọi là bộ phận Quản lý Chất lượng - QA) để hoàn thành việc kiểm thử trước khi chuyển giao sản phẩm cho khách hàng. Trong suốt quá trình đó, độ lệch so với kế hoạch được kiểm soát nghiêm ngặt nhằm đảm bảo sản phẩm ra đời thực tế đúng với thiết kế.

Cách tiếp cận này có cả những ưu điểm và nhược điểm. Ưu điểm của nó là rất lô-gic – bạn phải suy nghĩ trước khi làm, viết ra tất cả mọi thứ, tuân thủ kế hoạch, và đảm bảo mọi thứ có thể đều được kiểm soát. Nhược điểm cơ bản của nó là không tính tới sự tham gia của con người.

Ví dụ, cách tiếp cận này yêu cầu tất cả những ý tưởng tốt cần được đưa ra ngay từ đầu, để đưa vào kế hoạch. Nhưng như chúng ta biết, các ý tưởng tốt xuất hiện trong suốt quá trình phát triển – lúc bắt đầu, khi giữa chừng, và thậm chí đôi khi xuất hiện trước ngày giới thiệu sản phẩm, và một quy trình không cho phép những thay đổi sẽ kiềm chế sự đổi mới. Với mô hình thác nước, ý kiến hay nhưng đưa ra muộn không phải là một món quà mà là một nguy cơ.

Mô hình thác nước còn đề cao việc viết ra mọi thứ như là phương thức chính để trao đổi những thông tin quan trọng. Một giả định xem ra hợp lý đó là khi tôi viết ra giấy mọi thứ trong suy nghĩ của tôi, sẽ đáng tin cậy hơn khi truyền đạt cho người khác trong nhóm, hơn nữa, nếu nó ở trên giấy, sẽ là bằng chứng rằng tôi đã hoàn thành công việc của mình. Trên thực tế cho thấy, phần lớn các tài liệu mô tả yêu cầu của khách hàng chi tiết hơn 50 trang sẽ không được đọc. Và khi họ phải đọc, thường xuất hiện sự hiểu lầm. Một tài liệu được viết ra là một bức tranh không đầy đủ về các ý tưởng của tôi; khi bạn đọc nó, bạn hiểu theo cách khác xa với điều tôi nghĩ hoặc tôi muốn nói. Sẽ không có gì ngạc nhiên khi nảy sinh những hiểu lầm nghiêm trọng.

Một điều nữa xảy ra là việc phát sinh các ý tưởng khi lần đầu bạn sử dụng sản phẩm. Bạn đột nhiên nghĩ ra 20 cách để làm cho nó tốt hơn. Không may, những ý tưởng có giá trị này hay đến vào cuối quá trình phát triển, khi mọi thay đổi đều rất khó khăn và rắc rối – nói cách khác sẽ phải trả giá đắt khi muốn làm lại cho đúng, ít nhất khi sử dụng một phương pháp truyền thống.

Con người không có khả năng đoán trước được tương lai. Ví như, không thể đoán trước được một thông báo không như mong đợi từ đối thủ cạnh tranh. Các vấn đề kỹ thuật có thể xuất hiện bất ngờ làm thay đổi định hướng công việc. Hơn nữa, con người đặc biệt kém khi lên kế hoạch

cho những việc không chắc chắn trong tương lai xa – sẽ là ảo tưởng để đoán rằng bạn sẽ sử dụng một tuần như thế nào trong tám tháng tới kể từ hôm nay. Đã có nhiều biểu đồ Gantt được xây dựng cẩn thận mà vẫn thất bại.

Thêm vào đó, chu trình phát triển tuần tự kích thích mối quan hệ đối kháng khi chuyển giao công việc giữa người này và người kia. “Anh ta yêu cầu tôi xây dựng điều mà không có trong mô tả”. “Cô ấy đang thay đổi suy nghĩ”. “Tôi không chịu trách nhiệm cho cái mà tôi không kiểm soát”. Và điều này cho phép ta thấy bộ mặt thật của phát triển tuần tự – nó chẳng mấy thú vị. Mô hình thác nước là nguyên nhân gây ra những bức xúc lớn cho những người xây dựng sản phẩm. Thành quả tạo ra mà không thể hiện tính sáng tạo, kỹ năng, và đam mê của chính người người tạo nên nó. Con người không phải máy móc, và nếu mọi người phải làm việc như một cỗ máy thì thường đem lại kết quả không vui vẻ gì.

Một quy trình cứng nhắc, khó thay đổi sẽ cho ra những sản phẩm tầm thường. Khách hàng có thể có được điều họ yêu cầu lúc đầu (ít nhất hai bước chuyển đổi được gỡ bỏ), nhưng đó có phải là điều khách hàng thật sự muốn khi nhìn thấy sản phẩm hay không? Bằng cách thu thập tất cả các yêu cầu khách hàng và ép cứng chúng, sản phẩm chỉ “tốt” như ý tưởng ban đầu, thay vì trở thành tốt nhất khi mọi người đã tìm hiểu và khám phá ra những điều mới.

Nhiều người sử dụng chu trình tuần tự phải trải nghiệm những thiếu sót này không biết bao nhiêu lần. Nhưng có vẻ như rất hợp lý khi phản ứng thông thường hướng vào bên trong: “Chỉ cần chúng ta làm tốt hơn, nó đã hoạt động rồi” – nếu chúng ta lên kế hoạch, làm tài liệu tốt hơn, chống lại sự thay đổi nhiều hơn, mọi thứ có lẽ đã diễn ra suôn sẻ. Nhưng không may nhiều nhóm phát triển lại gặp điều ngược lại: càng cố gắng bao nhiêu, kết quả càng tồi tệ bấy nhiêu! Nhiều nhà quản lý đã đầu tư cả danh tiếng của họ – và nhiều tài nguyên khác – vào mô hình thác nước; việc họ chuyển sang một mô hình khác biệt về cơ bản là sự thừa nhận rõ ràng họ đang sai lầm. *Scrum* chính là điều khác biệt cơ bản như vậy...

## Phát triển linh hoạt và Scrum

Các phương pháp phát triển linh hoạt ra đời dựa trên niềm tin rằng bằng việc tiếp cận thực tại cuộc sống và phát triển sản phẩm để học hỏi, cải tiến, và thay đổi sẽ mang lại các kết quả tốt hơn. Các nguyên tắc phát triển linh hoạt nhấn mạnh đến việc con người có thể nhanh chóng tạo ra phần mềm chạy được, hơn là dành nhiều thời gian của giai đoạn đầu cho việc viết các đặc tả kỹ thuật. Phương pháp phát triển linh hoạt tập trung vào nhóm liên chức năng (cross-functional) được trao quyền để đưa ra các quyết định, thay vì một hệ thống phân cấp và phân chia chức năng. Và phương pháp này còn tập trung vào các phân đoạn lặp nhanh, với việc tiếp nhận thông tin liên tục từ phía khách hàng trong quá trình phát triển. Khi mọi người bắt đầu tìm hiểu về phát triển linh hoạt hoặc Scrum, thường có một chút mơ hồ trong nhận thức – điều này có vẻ rất giống những thuở đầu khi chúng ta nói “hãy làm đi.”

Hiện nay phương pháp phát triển linh hoạt phổ biến nhất là Scrum. Phương pháp này chịu ảnh hưởng lớn từ bài viết trên “*Harvard Business Review*” năm 1986, dựa trên các thông tin thực tiễn của các nhóm phát triển sản phẩm thành công; bài viết này giới thiệu về thuật ngữ “Bóng

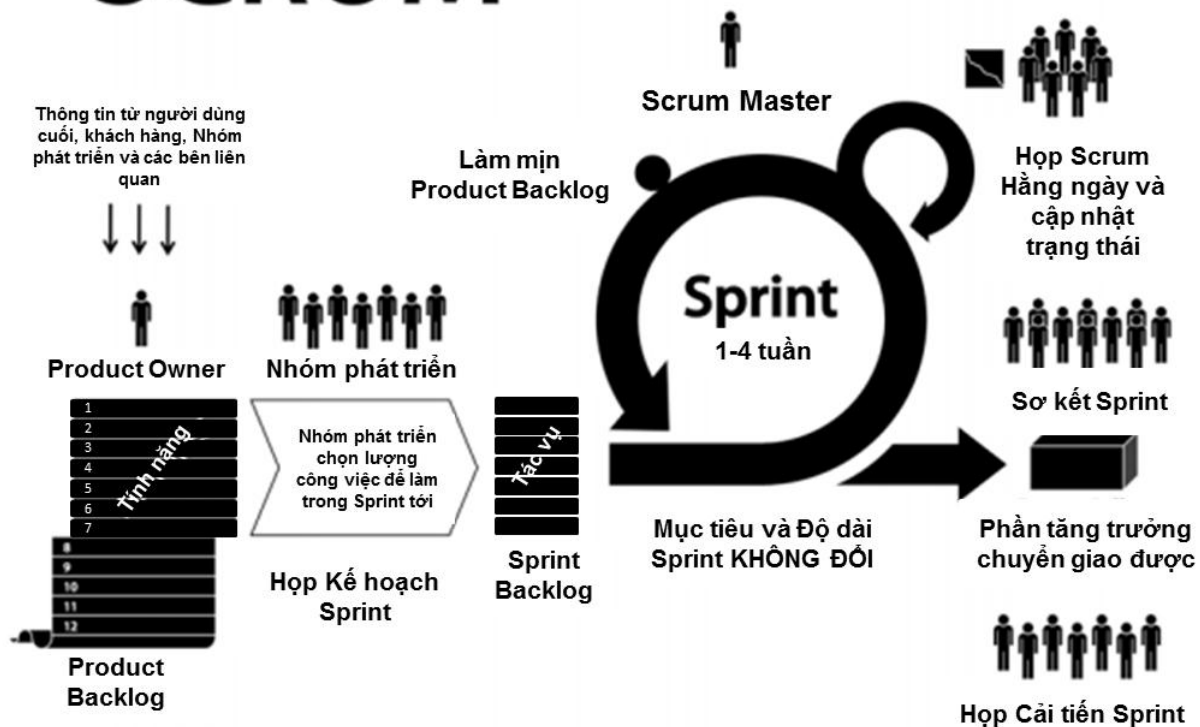
bầu dục”, sau đó thuật ngữ này được chuyển thành “Scrum” trong bài “*Wicked Problems, Righteous Solutions*” (được viết năm 1991 bởi DeGrace và Stahl) nói về sự phát triển thành công đối với trò chơi Bóng bầu dục (Rugby), trong trò chơi này một nhóm sẽ tự tổ chức (self-organizing) để cùng nhau phát triển sản phẩm. Phương pháp này được Ken Schwaber và Ts. Jeff Sutherland chuẩn hóa vào năm 1993. Ngày nay, Scrum được sử dụng ở nhiều công ty lớn và nhỏ, trong đó có Yahoo!, Google, Lockheed Martin, Motorola, SAP, Cisco, GE, CapitalOne và Cục dự trữ liên bang Hoa Kỳ (US Federal Reserve). Báo cáo từ nhiều nhóm phát triển sử dụng Scrum cho thấy có những cải tiến đáng kể về năng suất và tinh thần, và một số trường hợp chuyển đổi thành công sang sử dụng Scrum. Đối với các nhà phát triển sản phẩm – nhiều người trong số họ chán ngấy với tình trạng thay đổi xoành xoạch các phương pháp quản lý – điều này rất quan trọng. Scrum thật đơn giản và rất mạnh mẽ.

## Tóm tắt Scrum

Scrum là một khung làm việc với đặc tính lặp (iterative) và tiệm tiến (incremental) dành cho các dự án và phát triển sản phẩm hay ứng dụng. Scrum tổ chức việc phát triển thành các chu trình làm việc gọi là **Sprint**. Mỗi Sprint không quá một tháng, và không có ngắt quãng giữa các Sprint. Các Sprint được đóng khung thời gian (*time boxed*) – kết thúc vào một ngày xác định kể cả khi công việc đã hoàn thành hoặc chưa, và *không bao giờ kéo dài thêm*. Bắt đầu mỗi Sprint, nhóm liên chức năng (cross-functional) chọn các hạng mục (các yêu cầu khách hàng) từ danh sách ưu tiên hóa. Nhóm Phát triển cam kết hoàn thành các hạng mục khi kết thúc Sprint. Trong Sprint, các hạng mục đã được chọn không thay đổi. Mỗi ngày Nhóm Phát triển tập hợp trong thời gian ngắn để thanh tra quá trình thực hiện, và điều chỉnh những bước cần thiết tiếp theo để hoàn thành công việc còn lại. Kết thúc Sprint, nhóm sơ kết lại Sprint với các bên liên quan và trình bày các việc đã làm. Mọi người nhận phản hồi để đưa vào Sprint tiếp theo. Scrum quan tâm tới phần thật sự đã “hoàn thành” của sản phẩm ở cuối mỗi Sprint; đối với phần mềm, có nghĩa là mã nguồn hoàn chỉnh, đã được kiểm thử đầy đủ, có thể chuyển giao được. Các vai trò chính, đồ nghề và sự kiện được tóm tắt ở **Hình 1**.

Điểm mấu chốt nhất của Scrum chính là cơ chế “thanh tra và thích nghi”. Vì trong quá trình phát triển không bao giờ tránh được việc phải học hỏi, đổi mới, và các yếu tố bất ngờ, Scrum nhấn mạnh việc đi từng bước nhỏ, thanh tra cả kết quả sản phẩm lẫn hiệu quả công việc thực tại, để từ đó thích ứng với các mục tiêu của sản phẩm cũng như quy trình thực tế. Và cứ thế lặp đi lặp lại như vậy.

# SCRUM



Hình1. Scrum

## Các vai trò trong Scrum

Trong Scrum có ba vai trò: Product Owner (Chủ Sản phẩm), Nhóm Phát triển, và Scrum Master. Tất cả được gọi là Nhóm Scrum. Product Owner chịu trách nhiệm tối đa hóa lợi nhuận trên vốn đầu tư (Return On Investment - ROI) bằng cách chỉ ra các chức năng của sản phẩm, chuyển chúng thành danh sách được ưu tiên hóa, quyết định chức năng nào sẽ ở trên đầu danh sách cho Sprint tiếp theo, và liên tục điều chỉnh lại sự ưu tiên và cải tiến danh sách đó. Product Owner có trách nhiệm về lợi nhuận và tổn thất cho sản phẩm, giả sử rằng đó là sản phẩm thương mại. Trong trường hợp đó là ứng dụng nội bộ, Product Owner không chịu trách nhiệm cho ROI theo hướng của một sản phẩm thương mại (mà sinh ra lợi nhuận), nhưng vẫn phải có trách nhiệm tối đa hóa ROI – theo hướng là chọn – trong mỗi Sprint – các hạng mục có giá trị thương mại cao nhất, chi phí thấp nhất. Trong thực tế, “giá trị” là một từ không rõ ràng và sự sắp xếp thứ tự ưu tiên có thể bị ảnh hưởng bởi sự mong muốn của các khách hàng chủ chốt, có tính đến các mục tiêu chiến lược, rủi ro, việc cải tiến và các yếu tố khác nữa. Trong một số trường hợp, Product Owner và khách hàng là một; điều này thường gặp với các ứng dụng nội bộ. Trường hợp khác, khách hàng có thể là hàng triệu người với những nhu cầu khác nhau, khi đó vai trò của Product Owner giống như vị trí Product Manager (Giám đốc Sản phẩm) hoặc Product Marketing Manager (Giám đốc Tiếp thị Sản phẩm) trong nhiều tổ chức. Tuy nhiên, Product Owner khác so với Product Manager truyền thống bởi vì họ chủ động và thường xuyên

tương tác với Nhóm Phát triển, chủ quan đưa ra sự ưu tiên và kiểm tra kết quả mỗi vòng phát triển hai hoặc bốn tuần, thay vì ủy quyền quyết định cho người quản lý dự án. Một điều quan trọng cần chú ý là trong Scrum chỉ có duy nhất một người ở vị trí – và có quyền quyết định – Product Owner, và người đó có trách nhiệm với giá trị của sản phẩm.

Nhóm Phát triển xây dựng sản phẩm mà Product Owner chỉ định: ví dụ một ứng dụng hay một website. Nhóm Phát triển trong Scrum là “liên chức năng” (cross-functional) – bao gồm tất cả các chuyên môn cần thiết để đưa ra sản phẩm có thể chuyển giao được sau mỗi Sprint – và còn là nhóm tự quản (self-organizing), với mức độ độc lập và trách nhiệm rất cao. Nhóm quyết định cam kết cái gì, và làm thế nào để hoàn thành tốt nhất cam kết đó; trong Scrum Nhóm được gọi là các chú “lợn” và những người còn lại trong tổ chức gọi là “gà” (xuất phát từ câu chuyện vui về một con lợn và một con gà quyết định mở cửa hàng với tên gọi “Thịt lợn và Trứng”, con lợn có những suy nghĩ khác bởi vì “nó thì bị thịt còn con gà chỉ tham gia”).

Nhóm Phát triển trong Scrum gồm bảy (cộng hoặc trừ hai) người, và để sản xuất phần mềm Nhóm nên gồm những thành viên có kỹ năng về phân tích, phát triển, kiểm thử, thiết kế giao diện, thiết kế cơ sở dữ liệu, kiến trúc, viết tài liệu, và những thứ khác. Nhóm phát triển sản phẩm và cung cấp ý tưởng cho Product Owner cách làm ra sản phẩm tốt. Trong Scrum, Nhóm Phát triển đạt năng suất và hiệu quả nhất nếu tất cả các thành viên dành 100 phần trăm khả năng cho một sản phẩm trong Sprint; tránh tham gia đồng thời nhiều sản phẩm hoặc dự án. Các nhóm ổn định thường đi liền với năng suất cao, do đó tránh thay đổi thành viên của Nhóm phát triển. Các đội phát triển ứng dụng có nhiều người được sắp xếp thành nhiều Nhóm Scrum, mỗi nhóm tập trung vào một tính năng riêng của sản phẩm, với sự phối hợp chặt chẽ khả năng của mình. Do nhóm thường làm tất cả các công việc (lên kế hoạch, phân tích, lập trình, kiểm thử) cho một tính năng hoàn chỉnh hướng tới khách hàng, nên các Nhóm phát triển còn được gọi là *Nhóm tính năng (feature teams)*.

Scrum Master giúp Nhóm Phát triển học và áp dụng Scrum để đạt được giá trị thương mại. Scrum Master làm tất cả những gì có thể trong quyền hạn để giúp Nhóm và Product Owner thành công. Scrum Master không phải là người quản lý của Nhóm Phát triển hay Quản lý dự án; thay vào đó Scrum Master phục vụ Nhóm Phát triển, bảo vệ các thành viên khỏi sự can thiệp từ bên ngoài, đào tạo và hướng dẫn Product Owner cũng như Nhóm Phát triển vận dụng Scrum một cách khéo léo. Scrum Master đảm bảo rằng mọi người (gồm cả Product Owner, và những người quản lý khác) hiểu và làm theo Scrum, để họ lãnh đạo tổ chức vượt qua những khó khăn thường gặp trong quá trình thay đổi và thành công với phương pháp phát triển linh hoạt. Vì Scrum giúp phát lộ nhiều trở ngại và nguy cơ cho Nhóm Phát triển và Product Owner, nên rất quan trọng khi có một Scrum Master luôn hết mình giúp đỡ giải quyết các trở ngại đó, nếu không Nhóm Phát triển hoặc Product Owner sẽ khó thành công được. Scrum Master nên là người làm việc toàn thời gian, mặc dù một nhóm nhỏ có thể cử một thành viên đóng vai trò này (khi đó sẽ làm ít công việc thường lệ hơn). Scrum Master tốt có thể xuất phát từ mọi vị trí hoặc chuyên môn: Kỹ sư, Thiết kế, Kiểm thử, Quản lý sản phẩm, Quản lý dự án, hoặc Quản lý Chất lượng.

Scrum Master và Product Owner không thể là cùng một người; tại những thời điểm, Scrum Master có thể phải gây áp lực với Product Owner (ví như khi Product Owner muốn yêu cầu thêm một chức năng mới ở giữa Sprint chẳng hạn). Không giống như người quản lý dự án,

Scrum Master không nói cho mọi người phải làm cái gì hoặc phân việc cho ai đó – họ chỉ làm cho quy trình dễ dàng, thuận tiện, giúp đỡ Nhóm Phát triển tự tổ chức và quản lý. Nếu Scrum Master đã nắm giữ một vị trí quản lý trước đó, họ sẽ phải thay đổi tư duy và cách thức tương tác với Nhóm phát triển để có thể thành công với Scrum.

Lưu ý là không có vị trí quản lý dự án trong Scrum. Bởi vì điều đó không cần thiết; trách nhiệm truyền thống của một quản lý dự án được chia và phân công lại giữa ba vai trò của Scrum. Đôi khi người Quản lý Dự án có thể vào vai trò của Scrum Master, nhưng cần biết kết hợp các yếu tố để thành công – có sự khác nhau cơ bản giữa hai vai trò này, cả trong trách nhiệm hàng ngày và tư duy. Một cách để hiểu rõ vai trò và phát triển kỹ năng cơ bản để trở thành một Scrum Master thành công, là tham gia khóa đào tạo Scrum Master của Liên minh Scrum (Scrum Alliance).

Bên cạnh ba vai trò này, còn có những người khác đóng góp cho sự thành công của sản phẩm, bao gồm những người quản lý chức năng (ví dụ người quản lý kỹ thuật). Mặc dù vai trò của họ bị thay đổi trong Scrum, nhưng họ vẫn còn lại những giá trị của mình. Ví dụ:

- giúp đỡ Nhóm Phát triển bằng việc tôn trọng quy tắc và tinh thần của Scrum
- giúp đỡ tháo gỡ các trở ngại mà Nhóm Phát triển và Product Owner chỉ ra
- sẵn sàng đóng góp với chuyên môn và kinh nghiệm

Trong Scrum, những cá nhân này thay thời gian họ sử dụng trước kia như “người giữ trẻ” (phân việc, lấy báo cáo trạng thái, và các dạng khác của quản lý vi mô) bằng thời gian như người “quân sư” và “phục vụ” cho Nhóm Phát triển (hướng dẫn, huấn luyện, tháo gỡ trở ngại, giúp giải quyết vấn đề, đóng góp sáng tạo, và hướng dẫn phát triển kỹ năng của các thành viên). Trong hoàn cảnh này, những người quản lý có thể cần thay đổi cách quản lý của họ; ví dụ sử dụng cách hỏi kiểu Socrate để giúp Nhóm Phát triển tìm ra giải pháp của vấn đề, thay vì chỉ đơn giản quyết định cách giải quyết và gán cho Nhóm Phát triển.

## Bắt đầu với Scrum

Bước đầu tiên trong Scrum là Product Owner làm rõ tầm nhìn của sản phẩm. Cuối cùng đưa ra danh sách các chức năng được sàng lọc và sắp thứ tự gọi là Product Backlog. Backlog này tồn tại (và thay đổi) trong suốt vòng đời của sản phẩm; và là bản đồ chỉ đường (**Hình 2**). Tại bất kỳ thời điểm nào, Product Backlog là duy nhất, xác định “mọi thứ phải được thực hiện bởi Nhóm Phát triển, theo thứ tự ưu tiên”. Chỉ một Product Backlog tồn tại, nghĩa là Product Owner được yêu cầu ưu tiên hóa toàn bộ hạng mục công việc, thể hiện quyền lợi của các bên liên quan và được chi phối bởi nhóm phát triển.

					Ước lượng mới về khối lượng công việc còn lại theo Sprint...					
Hạng mục	Chi tiết (wiki URL)	Độ ưu tiên	Ước tính Giá trị	Ước tính khối lượng công việc ban đầu	1	2	3	4	5	6
Là người mua, tôi muốn cho một quyền sách vào giỏ hàng (xem giao diện phác thảo trên trang wiki)	...	1	7	5						
Là người mua, tôi muốn xóa bỏ	...	2	6	2						



một quyển sách trong giỏ hàng.										
Cải tiến tốc độ xử lý giao dịch (xem số đo yêu cầu trên wiki)	...	3	6	13						
Nghiên cứu giải pháp tăng tốc độ xác nhận thẻ tín dụng (xem số đo yêu cầu trên wiki)	...	4	6	20						
Nâng cấp máy chủ lên bản Apache 2.2.3	...	5	5	13						
Chuẩn đoán và sửa lỗi về xử lý đơn hàng (bugzilla ID 14823)	...	6	2	3						
Là người mua hàng, tôi muốn tạo và lưu lại danh sách yêu cầu.	...	7	7	40						
Là người mua hàng, tôi muốn thêm và xóa hạng mục khỏi danh sách yêu cầu.	...	8	4	20						

*Hình 2: Product Backlog*

Product Backlog trên bao gồm một loạt các hạng mục, tức chức năng cơ bản cho khách hàng mới (“cho phép người dùng thêm sách vào giỏ hàng”), và cả những yêu cầu cải tiến kỹ thuật (“làm lại phần xử lý giao dịch để có thể mở rộng nó”), việc thăm dò hoặc nghiên cứu (“tìm giải pháp để tăng tốc độ kiểm tra thẻ tín dụng”), và có thể là các lỗi đã được phát hiện (“chuẩn đoán và sửa lỗi xử lý yêu cầu”) nếu có ít vấn đề (Một hệ thống có nhiều lỗi thường có một hệ thống quản lý lỗi riêng). Bằng cách nào đó Product Backlog nên rõ ràng, mạch lạc và bền vững, dù Use Cases hoặc “user stories” thường được dùng để mô tả các hạng mục về giá trị của chúng cho người dùng sản phẩm.

Một phần của Product Backlog dành cho bản phát hành (release) hiện thời được gọi là **Release Backlog**, và thông thường, đây là phần quan tâm chính của Product Owner.

Product Backlog liên tục được cập nhật bởi Product Owner để phản ánh thay đổi trong yêu cầu của khách hàng, các ý tưởng mới, nhu cầu cạnh tranh, khó khăn kỹ thuật, và nhiều yếu tố khác. Nhóm Phát triển cung cấp cho Product Owner ước tính khối lượng công việc cho mỗi hạng mục trong Product Backlog. Thêm vào đó, Product Owner có trách nhiệm gán ước tính giá trị cho từng hạng mục. Product Owner thường không quen với công việc này. Do đó, Scrum Master có thể giúp Product Owner học cách làm. Với hai loại ước lượng này (khối lượng công việc và giá trị) và có thể thêm cả ước lượng rủi ro, Product Owner sắp xếp Backlog theo thứ tự ưu tiên (thường làm với Release Backlog) để tối đa hóa lợi nhuận trên vốn đầu tư (chọn hạng mục giá trị cao và khối lượng công việc thấp), hoặc để giảm các rủi ro chính. Như sẽ thấy, ước tính khối lượng công việc và giá trị có thể được làm mới ở mỗi Sprint khi mọi người tìm hiểu; do đó việc ưu tiên lại hạng mục diễn ra liên tục khi mà Product Backlog ngày càng phát triển.

Scrum không cung cấp kỹ thuật để biểu diễn và ưu tiên các hạng mục trong Product Backlog và cũng không cung cấp kỹ thuật ước tính. Một kỹ thuật phổ biến là ước tính theo kích thước tương đối (dựa trên nỗ lực, độ phức tạp, và tính bất định) sử dụng đơn vị “story point” hoặc đơn giản là “point” (điểm).

Với thời gian, Nhóm Phát triển sẽ biết cần bao nhiêu điểm để làm một Sprint, ví dụ cần trung bình 26 điểm mỗi Sprint. Với thông tin này họ có thể đưa ra được ngày hoàn thành tất cả các chức năng, hoặc bao nhiêu chức năng có thể hoàn thành trong một thời gian cố định, nếu tiếp

tốc độ trung bình và không có gì thay đổi. Giá trị này được gọi là “tốc độ” của nhóm. Tốc độ được biểu diễn cùng đơn vị khi ước tính kích cỡ của các hạng mục trong Product Backlog.

Các hạng mục trong Product Backlog có thể thay đổi đáng kể về kích cỡ và khối lượng công việc. Hạng mục lớn có thể được chia nhỏ trong buổi Làm mịn Product Backlog hoặc buổi họp Kế hoạch Sprint (Sprint Planning meeting), và hạng mục nhỏ có thể nhóm lại. Các hạng mục trong Product Backlog cho các Sprint sắp tới nên đủ nhỏ và gọn để Nhóm hiểu được, cho phép các cam kết đưa ra trong cuộc họp Kế hoạch Sprint trở nên có ý nghĩa; gọi là phạm vi “triển khai được”.

Một trong những điều người ta thường ngộ nhận là Scrum không cho phép bạn viết đặc tả yêu cầu chi tiết; trong thực tế Product Owner và Nhóm Phát triển quyết định mức độ chi tiết cần thiết, và điều này thay đổi ở các hạng mục khác nhau, tùy thuộc vào sự thông suốt của Nhóm, và các yếu tố khác. Xác định những điều quan trọng trong khoảng thời gian nhỏ nhất có thể – nói cách khác không mô tả chi tiết tất cả các hạng mục, chỉ đủ để hiểu. Hạng mục có mức ưu tiên thấp, được làm sau và thường vẫn ở dạng “thô” và lớn, có ít yêu cầu chi tiết hơn. Hạng mục nhỏ và ưu tiên cao để làm ngay được mô tả chi tiết hơn.

## Họp Kế hoạch Sprint

Họp Kế hoạch Sprint (Sprint Planning Meeting) được tổ chức khi bắt đầu mỗi Sprint. Buổi họp này được chia thành hai phần nhỏ, phần đầu được gọi là Họp Kế hoạch Sprint Phần một (gọi tắt là Phần một).

Trong Phần một, Product Owner và Nhóm Phát triển (với sự hỗ trợ của Scrum Master) xem xét các hạng mục có độ ưu tiên cao trong Product Backlog mà Product Owner muốn triển khai trong Sprint này. Họ thảo luận mục tiêu và bối cảnh của các hạng mục ưu tiên này trong Product Backlog, làm cho Nhóm Phát triển hiểu thấu những suy nghĩ của Product Owner. Product Owner và Nhóm Phát triển cũng sẽ xem xét “Định nghĩa Hoàn thành” (đã được thiết lập từ trước) mà tất cả hạng mục phải tuân thủ, như “Hoàn thành nghĩa là mã lệnh được làm theo chuẩn, được kiểm thử và triển khai với phương pháp Test-Driven-Development (TDD), được kiểm thử tự động 100 phần trăm, được tích hợp và làm tài liệu”. Phần một tập trung vào việc hiểu Product Owner muốn *điều gì*. Theo quy tắc của Scrum kết thúc Phần một, Product Owner (người luôn bận rộn) có thể rời buổi họp mặc dù họ vẫn *phải* sẵn sàng hiện diện (ví dụ qua điện thoại) trong buổi họp kế tiếp. Tuy nhiên Product Owner có thể tham gia Phần hai...

Họp Kế hoạch Sprint Phần hai tập trung vào việc lập kế hoạch công việc chi tiết để *làm thế nào* triển khai các hạng mục Nhóm Phát triển đã quyết định. Nhóm chọn các hạng mục từ Product Backlog đã cam kết hoàn thành trong Sprint, bắt đầu từ trên cùng (các hạng mục có độ ưu tiên cao nhất đối với Product Owner) đến cuối danh sách. Một hoạt động then chốt trong Scrum là: Nhóm quyết định cam kết sẽ hoàn thành bao nhiêu công việc, thay vì được phân công bởi Product Owner. Sự cam kết sẽ trở nên tin cậy hơn vì Nhóm làm dựa trên sự phân tích và lên kế hoạch của họ, chứ không phải được quyết định bởi người khác. Trong khi Product Owner không có quyền kiểm soát các cam kết từ phía Nhóm Phát triển, nhưng anh ta biết rằng các hạng mục được chọn bắt đầu từ trên cùng của Product Backlog – nói cách khác, đó là những hạng mục mà anh ta đánh giá là quan trọng nhất. Nhóm Phát triển có khả năng vận động, đề xuất cho các

hạng mục ở phía dưới của danh sách; điều này thường xảy ra khi Nhóm Phát triển và Product Owner nhận thấy điều gì đó ở những hạng mục ưu tiên thấp nhưng lại dễ dàng ăn khớp và hợp lý với các hạng mục ưu tiên cao.

Họp Kế hoạch Sprint thường diễn ra trong nhiều giờ, nhưng không quá tám tiếng cho một Sprint bốn tuần - Nhóm Phát triển nghiêm túc đưa ra một cam kết để hoàn thành công việc, và để thành công cam kết phải được suy xét cẩn thận. Nhóm có thể bắt đầu Họp Kế hoạch Phần hai bằng việc ước lượng mỗi thành viên dành bao nhiêu thời gian cho các công việc trong Sprint - nghĩa là thời gian làm việc hàng ngày trừ đi thời gian dành cho họp, email, ăn trưa, và những thứ khác. Phần lớn mọi người có từ 4 đến 6 tiếng một ngày cho Sprint. Đây là năng lực sản xuất của Nhóm Phát triển cho Sprint sắp tới. Xem Hình 3.

<b>Độ dài Sprint</b>	2 tuần
<b>Số ngày làm việc</b>	8 ngày

<b>Thành viên</b>	<b>Số ngày có mặt trong Sprint*</b>	<b>Số giờ có mặt trong ngày</b>	<b>Tổng số giờ có mặt</b>
Tracy	8	4	32
Sanjay	7	5	35
Philip	8	4	32
Jing	6	5	30

\* Trừ những ngày nghỉ và không có mặt tại văn phòng

*Hình 3: Ước tính số giờ có mặt*

Một khi năng lực sản xuất đã được xác định, Nhóm Phát triển tính toán số hạng mục trong Product Backlog mà họ có thể hoàn thành, và cách họ sẽ tiến hành để hoàn thành các hạng mục đó. Thường việc này bắt đầu với việc thảo luận về thiết kế trên bảng trắng. Khi đã hiểu thiết kế tổng thể, Nhóm Phát triển phân tách các hạng mục trong Product Backlog thành công việc cụ thể. Nhóm bắt đầu với hạng mục đầu tiên trong Product Backlog – nói cách khác, đó là hạng mục được Product Owner đặt độ ưu tiên cao nhất - và cùng nhau chia nhỏ thành những tác vụ riêng lẻ, và đưa vào một tài liệu gọi là **Sprint Backlog** (Hình 4). Như đã đề cập, Product Owner phải sẵn sàng (kể cả qua điện thoại) để làm sáng tỏ các vấn đề nếu cần. Nhóm Phát triển sẽ tiến hành tuần tự như vậy từ trên xuống dưới của Product Backlog, đến khi sử dụng hết năng lực ước tính. Cuối buổi họp, Nhóm Phát triển sẽ đưa ra danh sách tất cả các công việc và ước tính khối lượng của chúng (thường tính bằng tiếng hoặc một phần của ngày).

Scrum khuyến khích những thành viên đa kỹ năng, hơn là chỉ “làm việc theo chức danh” theo kiểu “tester” thì chỉ làm mỗi việc kiểm thử. Nói cách khác, các thành viên của Nhóm Phát triển “xông đến nơi có việc” và tham gia khi có thể. Nếu có nhiều tác vụ liên quan đến kiểm thử, tất cả các thành viên có thể trợ giúp. Điều này không hàm ý rằng mọi người đều đa năng; không có vấn đề khi trong Nhóm Phát triển có người đặc biệt giỏi về kiểm thử (cũng như việc khác), nhưng các thành viên cùng làm việc và học được từ người khác các kỹ năng mới. Cho nên, trong quá trình thiết lập và ước tính công việc ở buổi Họp Kế hoạch không cần thiết – và cũng không thích hợp – cho bất kỳ ai xung phong làm tất cả các công việc được cho là “họ làm tốt nhất”. Tốt

hơn là chỉ tình nguyện làm một việc tại một thời điểm, sau đó chọn một công việc mới và hãy xem xét chọn những tác vụ giúp mình có thể học tập, nghiên cứu (có thể bằng cách làm việc cùng với những chuyên gia trong nhóm).

Tuy vậy, có khi chỉ duy nhất một thành viên trong nhóm có thể làm được một công việc đặc thù nào đó bởi vì để người khác học được kĩ năng đó thì mất quá nhiều thời gian hoặc không thể học được - ví dụ John là người duy nhất có khả năng mỹ thuật để vẽ những bức tranh đẹp chẳng hạn. Thế nhưng những thành viên khác trong Nhóm Phát triển không thể đứng nhìn nếu cuộc sống của họ phụ thuộc vào điều đó. Trong trường hợp hiếm gặp này - và nếu không phải là hiếm hoặc ngày càng ít xảy ra khi Nhóm Phát triển học hỏi, thì có điều gì đó không ổn – mọi người cần phải yêu cầu John lập một kế hoạch khả thi để hoàn thành các cộng việc vẽ vời này trong khuôn khổ một Sprint ngắn.

Nhiều Nhóm Phát triển cũng sử dụng công cụ theo dõi công việc trực quan, dưới dạng bảng dán trên tường, nơi mà các công việc (được viết dưới dạng các mẫu ghi chú) dịch chuyển trong suốt Sprint qua các cột được dán nhãn “Chưa bắt đầu”, “Đang làm”, và “Hoàn thành”. Xem Hình 5.

				Ước tính mới về khối lượng công việc còn lại...					
Hạng mục trong Product Backlog	Công việc trong Sprint	Người thực hiện	Ước tính khối lượng công việc ban đầu	1	2	3	4	5	6
Là người mua tôi muốn cho sách vào giỏ hàng	sửa 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						

Hình 4: Sprint Backlog



*Hình 5: Quản lý trực quan - Công việc trong Sprint Backlog được dán lên tường*

Một vấn đề trụ cột của Scrum là khi nhóm đã cam kết, bất cứ hành động bổ sung, sửa đổi phải chuyển sang Sprint tiếp theo. Có nghĩa là nếu Sprint đang được triển khai, Product Owner quyết định có một hạng mục mới muốn Nhóm Phát triển làm, anh ta không thể thay đổi cho đến khi bắt đầu một Sprint mới. Nếu có yếu tố bên ngoài xuất hiện làm thay đổi đáng kể quyền ưu tiên, và Nhóm Phát triển sẽ lãng phí thời gian nếu tiếp tục, Product Owner và Nhóm Phát triển có thể kết thúc Sprint đó. Nhóm dừng làm việc và buổi Họp Kế hoạch Sprint khác được tổ chức để bắt đầu một Sprint mới. Việc này thường gây ra gián đoạn lớn trong công việc; làm nản lòng Product Owner và Nhóm Phát triển khi phải đưa ra quyết định.

Việc Nhóm phát triển được bảo vệ khỏi sự thay đổi mục tiêu trong suốt Sprint có các tác động mạnh mẽ và rất tích cực. Thứ nhất, Nhóm Phát triển làm việc khi biết chắc chắn rằng những cam kết không thay đổi, điều đó củng cố sự tập trung của Nhóm để đảm bảo hoàn thành công việc. Thứ hai, quy tắc đó rèn luyện cho Product Owner suy nghĩ thật sự thấu đáo những hạng mục ưu tiên trong Product Backlog để đưa cho Nhóm Phát triển trong một Sprint.

Product Owner đạt được hai thứ khi tuân thủ những quy tắc này của Scrum. Thứ nhất, họ có niềm tin rằng Nhóm Phát triển đã cam kết hoàn thành và hiểu rõ các công việc đã chọn. Qua thời gian Nhóm Phát triển sẽ dần có kỹ năng tốt khi chọn và thực hiện một cam kết. Thứ hai, Product Owner có thể thay đổi mọi thứ mình muốn trong Product Backlog trước khi bắt đầu Sprint tiếp theo. Tại thời điểm đó mọi việc như thêm, xóa, sửa, và đặt lại ưu tiên cho các hạng mục đều có thể thực hiện và được chấp nhận. Trong khi Product Owner không thể thay đổi các hạng mục đã chọn trong Sprint hiện tại, họ chỉ phải đợi trong khoảng thời gian của một Sprint hoặc ít hơn để thay đổi những gì họ muốn. Có sự kỳ thị nhất định về sự thay đổi - thay đổi hướng phát triển, yêu cầu, hay chỉ thuần túy thay đổi suy nghĩ của bạn - và đó có thể là lý do mà các Product Owner thường say mê về Scrum như các thành viên khác.

## Họp Scrum Hằng ngày (Daily Scrum)

Khi một Sprint được triển khai, Nhóm Phát triển phải thực thi một hoạt động quan trọng khác của Scrum: đó là **Họp Scrum Hằng ngày** (Daily Scrum). Đây là một cuộc họp ngắn (kéo dài 15 phút hoặc ít hơn), được tiến hành hằng ngày tại một thời điểm đã được ấn định trước. Mọi thành viên của Nhóm Phát triển phải có mặt. Để đảm bảo tính ngắn gọn, yêu cầu mọi thành viên phải đứng trong suốt thời gian diễn ra cuộc họp. Họp Scrum Hằng ngày là cơ hội để Nhóm Phát triển đồng bộ hóa các công việc của họ và trình bày với các thành viên khác những trở ngại của mình. Trong buổi Họp Scrum Hằng ngày, từng thành viên trong Nhóm Phát triển chỉ báo cáo ba vấn đề sau với các thành viên khác của Nhóm Phát triển: (1) Những gì mình đã làm kể từ buổi họp trước đến nay; (2) những gì mình sẽ làm từ giờ tới buổi họp tiếp theo; (3) những trở ngại hoặc khó khăn mình gặp phải trong quá trình làm việc. Lưu ý rằng buổi họp Scrum Hằng ngày không phải là một cuộc họp mà ở đó thành viên phải báo cáo với người quản lý; mà là khoảng thời gian để một Nhóm Phát triển tự-tổ-chức chia sẻ với nhau những gì đang triển khai, và để giúp họ phối hợp với nhau trong công việc. Một số thành viên đưa ra các trở ngại, và Scrum Master là người chịu trách nhiệm giúp đỡ các thành viên của Nhóm Phát triển giải quyết các trở ngại đó. Các thành viên không được thảo luận trong suốt cuộc Họp Scrum Hằng ngày, mà chỉ trả lời ba câu hỏi nêu trên; nếu cần thì sẽ thảo luận trong một cuộc họp khác ngay sau cuộc Họp Scrum Hằng ngày, mặc dù Scrum không yêu cầu ai phải tham dự cuộc họp này. Cuộc họp sau đó là thường là sự kiện để Nhóm Phát triển thích nghi với những thông tin mà họ nghe được trong cuộc Họp Scrum Hằng ngày: nói cách khác, đây là lúc thực hiện một chu kỳ thanh tra và thích nghi. Scrum khuyến cáo rằng sẽ không có bất cứ một nhà quản lý hoặc người có chức vụ nào tham dự cuộc Họp Scrum Hằng ngày. Sự có mặt của họ sẽ làm cho Nhóm Phát triển cảm thấy mình “bị giám sát” – với áp lực mình phải báo cáo tiến độ Hằng ngày (kỳ vọng không thực tế), bị ức chế với về việc báo cáo các vấn đề khó khăn – và sẽ dẫn đến xu hướng làm giảm tính chất tự quản của Nhóm Phát triển, và gây ra tình trạng quản lý vi mô. Sẽ có lợi hơn khi các bên liên quan tiếp cận Nhóm Phát triển sau cuộc họp, và trợ giúp họ giải quyết các trở ngại làm Nhóm Phát triển bị chậm tiến độ.

## Cập nhật Sprint Backlog và Biểu đồ Sprint Burndown

Trong Scrum, Nhóm Phát triển hoạt động theo cơ chế tự quản, và để đảm bảo việc thành công nhóm phải biết được những gì mình đang làm. Hằng ngày, các thành viên của Nhóm Phát triển cập nhật thời gian còn lại phải làm để hoàn tất công việc lên **Sprint Backlog** (Hình 6). Dựa vào thông tin này một người sẽ tính toán lại tổng thời gian còn lại của cả Nhóm và vẽ lên một **Biểu đồ Sprint Burndown** (Hình 7). Mỗi ngày, biểu đồ sẽ cho thấy ước lượng khối lượng công việc (tính theo số giờ làm việc) mà nhóm còn phải làm cho tới khi kết thúc nhiệm vụ của mình. Lý tưởng nhất, đó là một *đồ thị đi xuống*, đó là quỹ đạo để đạt tới “số giờ còn lại bằng 0” ở ngày làm việc cuối cùng của Sprint. Do đó, biểu đồ này được gọi là biểu đồ *Burndown*. Và đôi khi biểu

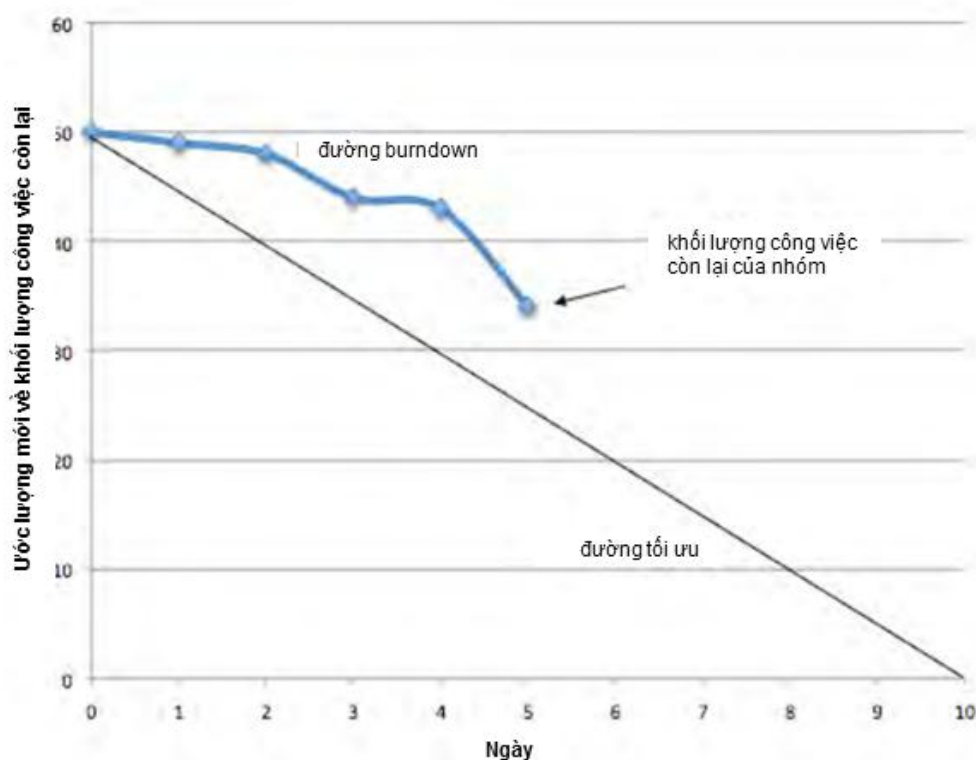
đồ cho thấy công việc đang diễn ra tốt đẹp, thường thì không được như vậy; đó mới thực sự phản ánh thực tế của quá trình phát triển sản phẩm. Điều quan trọng là biểu đồ *burndown* cho thấy tiến độ hướng tới mục tiêu của Nhóm Phát triển, biểu đồ không hiển thị lượng thời gian đã làm (thứ mà thực tế chẳng hề ảnh hưởng tới *tiến độ* công việc) mà cho thấy khối lượng *công việc còn lại trong tương lai* – đó mới là điều cản trở Nhóm Phát triển đạt tới mục tiêu của mình. Nếu đường đi của *burndown* không hướng xuống để cho thấy công việc sẽ hoàn thành khi mà Sprint sắp kết thúc, khi đó Nhóm Phát triển cần phải điều chỉnh, chẳng hạn như giảm bớt phạm vi công việc hoặc tìm ra cách thức làm việc hiệu quả hơn trong khi vẫn duy trì được nhịp độ làm việc ổn định.

Trong khi biểu đồ Sprint Burndown được tạo và thể hiện bằng cách sử dụng bảng tính điện tử, nhiều Nhóm Phát triển tìm cách tăng hiệu quả của chúng bằng cách vẽ biểu đồ trên giấy và dán lên tường ở phòng làm việc của mình, và cập nhật hằng ngày bằng cách dùng bút vẽ lên đó; giải pháp “low-tech/high-touch” này nhanh, đơn giản, và thường rõ ràng hơn so với biểu đồ trên máy tính.

				Ước lượng thời gian còn lại sau mỗi ngày làm việc (giờ)					
Hạng mục Product Backlog	Tên công việc	Người thực hiện	ước lượng ban đầu (giờ)	1	2	3	4	5	6
Là khách hàng, tôi muốn đưa một cuốn sách vào giỏ hàng	Chỉnh sửa CSDL	Sanjay	5	4	3	0	0	0	
	Tạo trang web (UI)	Jing	3	3	3	2	0	0	
	Tạo trang web (với Javascript)	Tracy & Sam	2	2	2	2	1	0	
	Viết acceptance test tự động	Sarah	5	5	5	5	5	0	
	Cập nhật trang trợ giúp khách hàng	Sanjay & Jing	3	3	3	3	3	0	
	...								
Tăng hiệu suất xử lý các giao dịch	Kết hợp mã nguồn DCP và hoàn chỉnh test layer-level		5	5	5	5	5	5	
	Hoàn chỉnh chức năng sắp xếp dành cho pRank		3	3	8	8	8	8	

	Thay đổi DCP và đọc các sử dụng API của pRank http		5	5	5	5	5	5	
	...	...	...						
		Tổng số (giờ)	50	49	48	44	43	34	

Hình 6. Cập nhật hằng ngày Khối lượng công việc còn lại trên Sprint Backlog



Hình 7. Biểu đồ Sprint Burndown

## Làm mịn Product Backlog

Một điều ít được biết đến, nhưng quan trọng trong Scrum đó là phương châm sử dụng năm đến mười phần trăm thời gian của mỗi Sprint để Nhóm Phát triển làm mịn (hoặc “tút” lại) Product Backlog. Công việc này bao gồm việc phân tích lại các yêu cầu một cách chi tiết, phân nhỏ các hạng mục lớn, ước tính khối lượng công việc của các hạng mục mới và các hạng mục còn lại. Scrum không nói đến công việc như thế nào được gọi là hoàn thành, nhưng một kỹ thuật thường được sử dụng cho việc này đó là tổ chức một cuộc họp tập trung trước khi kết thúc Sprint, nhờ vậy mà Nhóm Phát triển và Product Owner có thể tập trung cho công việc của mình mà không bị gián đoạn. Với một Sprint hai tuần, năm phần trăm thời gian của Sprint được sử dụng, nghĩa là mỗi Sprint sẽ có nửa ngày để làm mịn Product Backlog. Các hoạt động làm mịn không chỉ dành cho các hạng mục đã được chọn cho Sprint hiện tại mà còn dành cho



các hạng mục sẽ làm trong tương lai, nhiều khả năng là trong một hoặc hai Sprint tiếp theo. Với thực tế này, buổi họp Kế hoạch Sprint diễn ra tương đối đơn giản bởi vì Product Owner và Nhóm Scrum bắt đầu kế hoạch mới khi đã tường tận mọi vấn đề, phân tích chi tiết và ước lượng cẩn thận đối với các hạng mục công việc. Một biểu hiện cho thấy việc làm mìn không được thực hiện (hoặc thực hiện không tốt) đó là buổi họp Kế hoạch Sprint sẽ bị cuốn vào các câu hỏi quan trọng, phát hiện hoặc nhận thấy những điều mập mờ và cảm thấy chưa đầy đủ; công việc lập kế hoạch khi đó thường lấn cả vào Sprint, đây là điều không ai mong muốn cả.

## Kết thúc Sprint

Một trong những nguyên lý cốt lõi của Scrum là thời gian của một Sprint không bao giờ được mở rộng – Sprint phải kết thúc vào đúng ngày đã được ấn định trước bất kể Nhóm Phát triển có hoàn thành cam kết của họ hay không. Nhóm Phát triển thường cam kết quá nhiều ở một vài Sprint đầu và không thể hoàn thành được các cam kết của mình. Đôi khi để sửa chữa những sai lầm này họ lại cam kết quá ít, và thường kết thúc sớm hơn dự kiến (trong trường hợp này Nhóm có thể yêu cầu Product Owner cho triển khai thêm một số hạng mục của Product Backlog). Tuy nhiên, thường thì từ Sprint thứ ba hoặc thứ tư trở đi Nhóm Phát triển sẽ xác định được năng lực sản xuất của mình (phần lớn thời gian), và việc hoàn thành mục tiêu Sprint của họ sẽ đáng tin cậy hơn. Các Nhóm Phát triển được khuyến khích chọn khoảng thời gian cho các Sprint của mình (thường là hai tuần) và không thay đổi nó. Điều này sẽ giúp Nhóm Phát triển học được cách xác định được những gì có thể thực hiện, và giúp họ trong việc ước tính thời gian và lập kế hoạch phát hành dài hạn (longer-term release planning). Việc này cũng giúp Nhóm phát triển đạt được nhịp điệu cho công việc của mình; thường được gọi là “nhịp tim” của Nhóm Phát triển.

## Họp Sơ kết Sprint

Sau mỗi Sprint sẽ có một buổi Họp Sơ kết Sprint(Sprint Review) để Nhóm Phát triển và Product Owner xem xét lại Sprint vừa diễn ra. Buổi họp này thường bị gọi sai là buổi “demo” (buổi giới thiệu sản phẩm). Tên “demo” không mô tả được ý nghĩa thực của cuộc họp này. Một trong những ý tưởng chủ đạo của Scrum là *thanh tra và thích nghi* (inspect và adapt) để xem xét, phân tích mọi việc đang diễn ra và hành động dựa trên những phản hồi lặp đi lặp lại. Họp Sơ kết Sprint là một hoạt động để thanh tra và thích nghi dành cho *sản phẩm*. Đó là cơ hội để Product Owner tìm hiểu những gì đã được làm về sản phẩm và Nhóm Phát triển (xem xét lại Sprint); và là cơ hội để Nhóm Phát triển biết về các yêu cầu của Product Owner và thị trường. Bởi thế, yếu tố quan trọng nhất của buổi Sơ kết là một *cuộc trao đổi* chất lượng giữa Nhóm Phát triển và Product Owner để xem xét tình hình hiện thời, đưa ra các lời khuyên,v.v. Giới thiệu sản phẩm

đã hoàn thành trong Sprint là một phần của buổi họp, nhưng nếu tập trung vào giới thiệu và ít trao đổi thì buổi họp sẽ thiếu sự cân bằng.

Một điều hữu ích – nhưng thường bị bỏ qua – trong các hướng dẫn Scrum đó là trách nhiệm của Scrum Master đối với việc đảm bảo mọi thành viên hiểu rõ “Định nghĩa Hoàn thành” đối với sản phẩm hoặc bản phát hành. Anh ta cần đảm bảo nhóm không giới thiệu hoặc thảo luận các Hạng mục của Product Backlog chưa “hoàn thành” theo “Định nghĩa Hoàn thành”. Các Hạng mục chưa “hoàn thành” sẽ được đưa trở lại Product Backlog và sẽ được Product Owner đánh giá lại độ ưu tiên. Bằng cách này, sự minh bạch về công việc làm cho Nhóm Phát triển không thể giả mạo chất lượng bằng cách giới thiệu sản phẩm làm việc tốt, nhưng thực tế mã nguồn kém chất lượng và chưa được kiểm thử.

Tham gia buổi họp gồm Product Owner, Nhóm Phát triển, Scrum Master, và có thể có thêm khách hàng, các bên liên quan, các chuyên gia, các nhà quản lý hoặc bất kỳ ai có quan tâm. Phần giới thiệu trong buổi Họp Sơ kết Sprint không phải là phần “trình diễn” của Nhóm Phát triển – không dùng phần mềm trình chiếu. Theo hướng dẫn của Scrum không nên dành nhiều hơn 30 phút để chuẩn bị cho buổi sơ kết, trừ khi có sai sót trong công việc của Nhóm Phát triển. Đó đơn giản chỉ là một bản giới thiệu về những gì đã được làm. Mọi người có mặt đều có thể hỏi và đóng góp thêm.

## Họp Cải tiến Sprint

Buổi Họp Sơ kết Sprint tập trung vào việc thanh tra và thích nghi hướng đến *sản phẩm*. Trong khi đó, buổi họp Cải tiến Sprint được tiến hành ngay sau buổi Sơ kết lại hướng việc thanh tra và thích nghi vào *quy trình*. Đây là một thực tế mà nhiều nhóm bỏ qua, và thật không may, đó lại là cơ chế chủ đạo mang lại sự minh bạch vốn có trong Scrum, điều sẽ đem đến cơ hội cải tiến và chuyển thành kết quả. Đây là cơ hội cho Nhóm Scrum thảo luận xem điều gì làm được và chưa làm được, và đồng ý với các thay đổi thử nghiệm. Thành phần tham dự buổi họp gồm Nhóm Phát triển, Scrum Master và Product Owner có thể tham dự nhưng không bắt buộc. Scrum Master có thể đóng vai trò làm điều phối viên trong buổi họp, nhưng tốt hơn là một người trung lập nắm giữ vai trò này; một cách tiếp cận tốt cho Scrum Master để giúp các thành viên cải tiến lẫn nhau, là cho phép sự trao đổi chéo giữa các thành viên trong đội.

Có rất nhiều cách để tiến hành buổi Họp Cải tiến Sprint, cuốn Agile Retrospectives (được viết bởi Derby và Larsen năm 2006) có liệt kê nhiều kỹ thuật hữu ích. Cách đơn giản để tổ chức buổi thảo luận là vẽ lên bảng hai cột, một cột là “Điều đã làm tốt” và một cột là “Điều cần làm tốt hơn” - và sau đó đi quanh phòng, mỗi thành viên sẽ thêm vào một hoặc nhiều ý vào cột tương ứng. Khi các mục lặp lại, chúng sẽ được đánh dấu vào bên cạnh, vì vậy làm nổi bật các ý kiến phổ biến. Sau đó đội sẽ tìm kiếm nguyên nhân cơ bản, và đồng thuận trên một số lượng nhỏ

các thay đổi để cải tiến trong Sprint sắp tới, cùng với sự cam kết sẽ xem xét lại chúng trong buổi Họp Cải tiến Sprint tiếp theo.

Một kinh nghiệm hữu ích tại cuối buổi Họp Cải tiến Sprint là cho đội đánh nhãn các mục trên, nhãn là “C” nếu nguyên nhân tại Scrum (nói cách khác, nếu không làm theo Scrum thì điều này không xảy ra), nhãn là “E” nếu bị phát hiện bởi Scrum (nói cách khác, nó sẽ xảy ra nếu có hay không có Scrum, nhưng Scrum làm cho đội phát hiện ra nó), nhãn là “U” nếu nó chẳng liên quan gì đến Scrum (ví dụ như vấn đề thời tiết). Đội có thể tìm thấy nhiều “C” trên cột “Điều đã làm tốt” và nhiều “E” trên cột “Điều cần làm tốt hơn”; đây là một tin tốt lành, thậm chí nếu “Điều cần làm tốt hơn” là một danh sách dài, bởi vì bước đầu tiên để giải quyết các vấn đề cơ bản là khiến chúng lộ diện, và Scrum là một chất xúc tác mạnh mẽ cho điều đó.

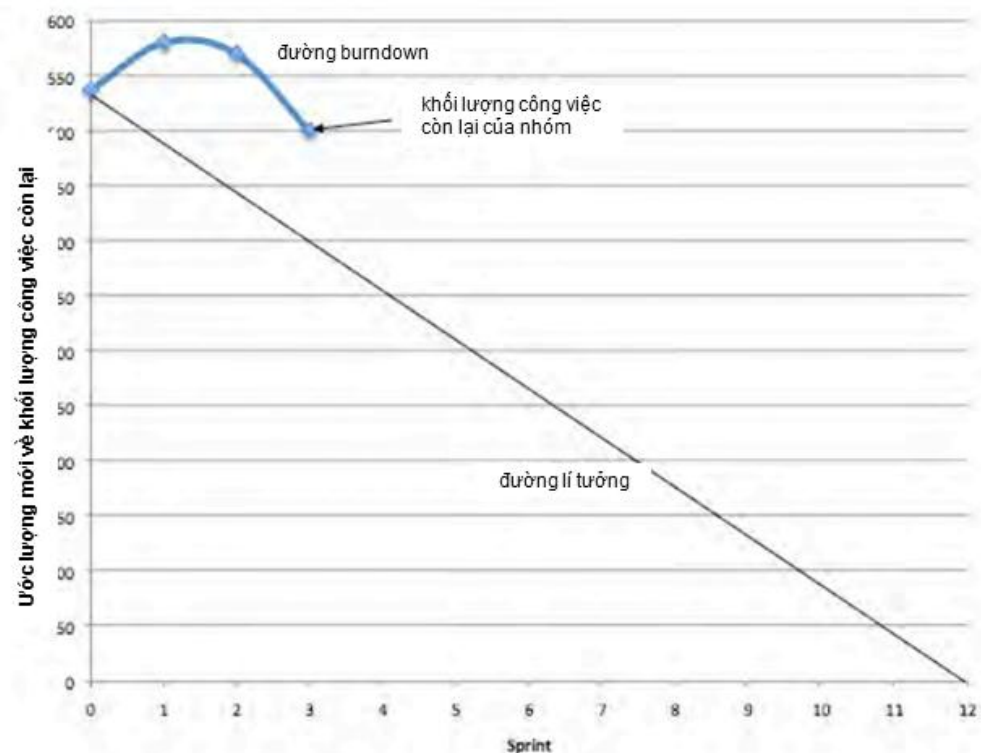
## Cập nhật Release Backlog và Biểu đồ Release Burndown

Khi một số hạng mục hoàn thành, thêm mới, hoặc có ước tính mới, và một số khác bị xóa khỏi mục tiêu phát hành, Product Owner phải đảm bảo những thay đổi này được thể hiện trong Release Backlog (và rộng hơn là cả Product Backlog). Ngoài ra, Scrum còn có một biểu đồ Burndown để mô tả quá trình tới ngày phát hành. Biểu đồ **Release Burndown** giống như biểu đồ Sprint Burndown, nhưng được dùng cho các hạng mục ở mức cao hơn (yêu cầu khách hàng) thay vì các công việc trong Sprint. Scrum Master sẽ có cơ hội để hỗ trợ Product Owner nếu Product Owner mới không biết tại sao hoặc làm như thế nào để tạo biểu đồ Release Burndown. Hình 8 và Hình 9 là một ví dụ của Release Backlog và biểu đồ Release Burndown.

					Ước tính mới về khối lượng công việc còn lại					
Hạng mục	Chi tiết (wiki url)	Độ ưu tiên	Giá trị ước tính	Ước tính công việc ban đầu	1	2	3	4	5	6
Là khách hàng, tôi muốn thêm một cuốn sách vào giỏ hàng (xem giao diện trên wiki)		1	7	5	0	0	0			
Là khách hàng, tôi muốn xóa một cuốn sách khỏi giỏ hàng		2	6	2	0	0	0			
Cải thiện hiệu suất xử lý giao dịch (xem hiệu suất mục tiêu trên wiki)		3	6	13	13	0	0			
Nghiên cứu các giải pháp để tăng tốc độ xác nhận thẻ tín dụng (xem hiệu suất mục tiêu trên wiki)		4	6	20	20	20	0			
Nâng cấp tất cả các server thành Apache 2.2.3		5	5	13	13	13	13			
Tìm và sửa lỗi của script xử lý		6	2	3	3	3	3			

(buzilla ID 14823)										
Là khách hàng tôi muốn tạo ra và lưu một danh sách		7	7	40	40	40	40			
Là khách hàng tôi muốn thêm hoặc xóa các mục trong danh sách mong muốn của mình		8	4	20	20	20	20			
			Tổng	537	580	570	500			

Hình 8. Release Backlog (một phần của Product Backlog)



Hình 9. Biểu đồ Release Burndown

## Khởi động Sprint tiếp theo

Ngay sau buổi Họp Sơ kết Sprint, Product Owner nên cập nhật Product Backlog khi có thay đổi. Ngay lúc này Product Owner và Nhóm Phát triển đã sẵn sàng để bắt đầu một Sprint khác. Không có thời gian nghỉ giữa các Sprint - Các Nhóm Phát triển thường Họp Cải tiến Sprint vào buổi chiều và họp Lập kế hoạch Sprint vào sáng hôm sau (hoặc đầu tuần tới).

Một trong những nguyên tắc của phương pháp phát triển linh hoạt là “tiến độ ổn định”, và chỉ với thời gian làm việc chính quy, mức độ hợp lý giúp Nhóm Phát triển tiếp tục chu trình này một cách vô hạn.

## Sprint Phát hành

Sẽ là hoàn hảo nếu ở cuối mỗi Sprint, sản phẩm có khả năng chuyển giao được, không cần thêm việc nào khác như kiểm thử hay tài liệu. Điều này nghĩa là *mọi việc được hoàn thành* sau mỗi Sprint; bạn có thể chuyển giao hoặc cài đặt ngay sau buổi Họp Sơ kết Sprint. Điều đó có nghĩa là mỗi phần tăng trưởng là một phần hoàn chỉnh của sản phẩm cuối cùng và hoàn toàn minh bạch với Product Owner và các bên có liên quan. Họ biết chính xác mình đang ở đâu cuối mỗi Sprint.

Tuy nhiên, nhiều tổ chức yếu trong việc thực thi, công cụ và cơ sở hạ tầng nên không thể đạt được điều này hoặc vì xảy ra tình huống ít nghiêm trọng hơn (như “máy bị hỏng”). Trong trường hợp này, sẽ còn lại một số công việc, như kiểm thử môi trường phát triển, và do đó cần một “Sprint Phát hành” (Release Sprint) để xử lý các công việc tồn đọng.

Chú ý rằng sự cần thiết của Sprint Phát hành là tín hiệu của sự yếu kém; lý tưởng nhất là không cần Sprint Phát hành. Khi cần thiết, các Sprint còn tiếp tục tới khi Product Owner quyết định là sản phẩm gần như sẵn sàng để phát hành. Khi đó sẽ có một Sprint Phát hành để chuẩn bị ra mắt. Nếu Nhóm dùng các kỹ thuật như liên tục tái cấu trúc và tích hợp, và kiểm thử một cách hiệu quả trong mỗi Sprint, thì công việc tiền phát hành hoặc việc cần thiết khác sẽ rất ít.

## Lập kế hoạch Phát hành & Làm mịn Product Backlog ban đầu

Cách lập kế hoạch phát hành dài hạn trong mô hình lập là câu hỏi đôi khi xuất hiện. Có hai trường hợp cần xem xét: (1) cho lần phát hành đầu tiên của sản phẩm mới, và (2) cho lần phát hành tiếp theo của sản phẩm đã có.

Trường hợp sản phẩm mới, hoặc *sản phẩm đã tồn tại muốn áp dụng Scrum*, phải làm mịn Product Backlog trước khi tiến hành Sprint đầu tiên. Việc làm mịn Product Backlog là của Product Owner và Nhóm Phát triển nhằm tạo ra một Product Backlog hợp lý. Việc làm mịn có thể mất một vài ngày hoặc một tuần, và cần cả một buổi thảo luận về tầm nhìn để có thể phân tích chi tiết các yêu cầu, và ước lượng tất các hạng mục cho lần phát hành đầu tiên.

Điều bất ngờ trong Scrum, với sản phẩm được phát triển theo Product Backlog đã chính thức hóa thì việc lập kế hoạch cho lần phát hành tới không cần quá chi tiết. Tại sao? Bởi Product Owner và Nhóm Phát triển sẽ phải làm mịn Product Backlog ở mỗi Sprint (năm hoặc mười phần trăm mỗi Sprint) một cách liên tục để chuẩn bị cho sản phẩm tương lai. Cách thức *phát triển sản phẩm liên tục* là cần thiết để phòng ngừa sự ngắt quãng trong các giai đoạn chuẩn bị-thực thi-kết thúc thường thấy ở chu trình phát triển tuần tự truyền thống.

Trong buổi họp làm mịn Product Backlog ban đầu và các buổi làm mịn Backlog ở mỗi Sprint, Nhóm Phát triển và Product Owner sẽ lập kế hoạch phát hành, làm mịn các ước lượng, độ ưu tiên, và các nội dung mà họ hiểu rõ hơn.

Một số lần phát hành được ấn định thời gian cụ thể; ví dụ: “Chúng ta sẽ phát hành phiên bản 2.0 của dự án vào hội chợ thương mại diễn ra ngày 10 tháng 11.” Trong trường hợp này, Nhóm Phát triển sẽ thực hiện một số Sprint (và xây dựng các tính năng) phụ thuộc vào thời gian mà họ có. Những sản phẩm khác cần các tính năng cơ bản nhất định trước khi được gọi là hoàn thành và sản phẩm sẽ không thể ra mắt tới khi những yêu cầu này hoàn tất mà không phụ thuộc vào thời gian để hoàn thành chúng. Do Scrum nhấn mạnh tới việc tạo ra các tính năng có thể chuyển giao được sau mỗi Sprint, nên Product Owner có thể phải chọn các tính năng để tạo ra bản phát hành tạm thời. Từ đó cho phép người dùng biết các lợi ích của sản phẩm khi hoàn thành.

Khách hàng không biết sản phẩm sẽ như thế nào, nên phải tập trung tạo và làm mịn kế hoạch để họ biết hướng của sản phẩm khi phát hành, và làm rõ những tính năng chính sẽ có (ví dụ: quy mô so với tiến độ). Hãy coi đó là lộ trình giúp bạn tới đích, còn con đường và các quyết định chính xác của bạn sẽ được xác định khi bạn đi.

Hầu hết các Product Owner sẽ có một phương án phát hành. Ví dụ, Product Owner sẽ quyết định một ngày phát hành và làm việc với Nhóm Phát triển để ước lượng các hạng mục của Release Backlog để có thể phát hành đúng ngày dự kiến. Trong tình huống “giá cố định / ngày cố định / sản phẩm cố định”, ví dụ trong một hợp đồng phát triển – thì một vài tham số này phải có bộ đệm để đề phòng sự thay đổi và các điều bất trắc ; về điều này, Scrum không khác gì so với các cách tiếp cận khác.

## **Tập trung vào ứng dụng hoặc sản phẩm**

Với đặc điểm hướng ứng dụng hoặc sản phẩm – cả thương mại lẫn sử dụng nội bộ trong tổ chức – Scrum đưa nhóm tránh xa khỏi mô hình cũ lấy dự án làm *trung tâm* sang mô hình *phát triển ứng dụng/sản phẩm liên tục*. Không còn một dự án với phần bắt đầu, phần giữa và kết thúc. Vì thế không cần người quản lý dự án theo kiểu truyền thống. Và thay vào đó cần một Product Owner ổn định và một Nhóm tự quản bền vững cộng tác với nhau trong một chuỗi “vô hạn” các Sprint, cho đến khi sản phẩm hoặc ứng dụng ngừng phát triển. Tất cả công việc quản lý cần thiết cho “dự án” được điều khiển bởi Nhóm Phát triển và Product Owner – là một khách hàng nội bộ hay đến từ bộ phận quản lý sản phẩm. “Dự án” không được điều khiển bởi người quản lý IT hoặc ai đó từ Phòng Quản lý Dự án.

Scrum cũng được sử dụng cho những *dự án* thật sự mà chỉ diễn ra một lần (thay vì tạo ra và phát triển các ứng dụng lâu dài); trong trường hợp này Nhóm Phát triển và Product Owner làm công việc quản lý dự án.

Sẽ ra sao nếu không có đủ việc từ một hay nhiều ứng dụng đang tồn tại để đảm bảo cho một Nhóm bền vững? Trong trường hợp này, Nhóm Phát triển có thể chọn những hạng mục từ một

ứng dụng cho một Sprint, và chọn các hạng mục của ứng dụng khác trong Sprint sau; tình huống như vậy thì Sprint thường khá ngắn, chỉ khoảng một tuần.

Thỉnh thoảng, không có đủ việc cho mọi người thậm chí với giải pháp đã có, và Nhóm Phát triển có thể chọn các hạng mục của nhiều ứng dụng trong cùng một Sprint; tuy nhiên, phải thận trọng với cách này vì có thể sẽ xảy ra trường hợp nhiều công việc không được triển khai hiệu quả. Một mẫu hình sản xuất trong Scrum cho Nhóm Phát triển là chỉ *tập trung* vào một ứng dụng hay sản phẩm trong một Sprint.

## Những thách thức phổ biến

Scrum không chỉ là những điều được đúc rút từ thực tiễn – hơn thế, và quan trọng hơn nó là một khung làm việc mang đến sự minh bạch, và một cơ chế cho phép “thanh tra và thích nghi”. Scrum thực hiện điều này bằng cách phát lộ ra các chức năng hỗn độn và những trở ngại ảnh hưởng đến hiệu suất làm việc của Product Owner và Nhóm Phát triển, để họ có thể giải quyết chúng. Ví dụ, Product Owner có thể không thực sự hiểu rõ về thị trường, các tính năng, hoặc cách để ước tính các giá trị có liên quan. Hoặc Nhóm Phát triển có thể thiếu kỹ năng trong việc ước tính nội lực của mình hoặc công việc phát triển.

Scrum sẽ giúp nhanh chóng phát hiện ra các yếu điểm đó. Scrum không giải quyết các vấn đề về phát triển; nó làm cho các vấn đề này hiển hiện ra như một vấn đề nhức nhối, và cung cấp một khung làm việc để mọi người phát hiện ra cách giải quyết những vấn đề đó trong thời gian ngắn, mà chỉ cần thực hiện những cải tiến nhỏ.

Giả sử Nhóm Phát triển thất bại trong việc chuyển giao những gì họ đã cam kết trong Sprint đầu tiên do yếu kém trong việc phân tích và kỹ năng ước tính. Đối với Nhóm Phát triển, họ cảm thấy như là một thất bại. Nhưng trong thực tế, kinh nghiệm này là bước đầu tiên cần thiết để họ thực tế hơn và thận trọng hơn với những cam kết của mình. Mô hình này của Scrum giúp làm rõ các chức năng hỗn độn, cho phép Nhóm Phát triển thực hiện điều gì đó với vấn đề này, nó là một cơ chế cơ bản mang lại những lợi ích rất quan trọng với những Nhóm Phát triển có kinh nghiệm trong việc sử dụng Scrum.

Một sai lầm phổ biến nảy sinh khi triển khai Scrum trong thực tế đó là thay đổi Scrum, đây chính là một thách thức. Ví dụ, Nhóm Phát triển khi gặp khó khăn trong việc thực hiện những cam kết của họ trong một Sprint nào đó họ có thể đi đến quyết định kéo dài thời gian triển khai Sprint, điều này làm Sprint không bao giờ trở hện – và với quy trình này chắc chắn rằng sẽ không bao giờ Nhóm học được cách làm việc tốt hơn với khoảng thời gian đã ước tính và quản lý thời gian của mình. Với cách thức này, nếu không có sự huấn luyện và hỗ trợ của một Scrum Master có kinh nghiệm, tổ chức có thể thay đổi Scrum thành một hình ảnh phản chiếu các khuyết điểm và sự xung đột chức năng của chính họ, đồng thời hủy hoại những lợi ích thực sự mà Scrum đem lại: Làm lộ rõ các mặt tốt và xấu, và cung cấp cho tổ chức những lựa chọn để nâng họ lên một tầm cao mới.

Một sai lầm phổ biến khác đó là việc cho rằng một hoạt động nào đó trong Scrum không được khuyến khích hoặc bị ngăn cấm chỉ vì Scrum không đưa ra yêu cầu bắt buộc về nó. Ví dụ, Scrum

không quy định Product Owner thiết lập một chiến lược dài hạn cho sản phẩm của mình; cũng không yêu cầu các kỹ sư tìm kiếm lời khuyên từ những người có kinh nghiệm hơn về những vấn đề kỹ thuật phức tạp. Scrum để mặc cho mỗi cá nhân có liên quan đưa ra những quyết định đúng đắn; và nhiều trường hợp, cả hai thực thể đó (cùng với nhiều thứ khác) đều là những lời khuyên tốt.

Trái lại, có những điều cần cảnh giác khi các nhà quản lý áp đặt Scrum vào Nhóm Phát triển của họ; Scrum mang lại cho Nhóm Phát triển không gian riêng và các công cụ để họ tự quản, và việc có các quyết định áp đặt từ trên xuống sẽ không đem lại sự thành công. Cách tiếp cận tốt hơn với một Nhóm Phát triển khi mới bắt đầu nghiên cứu về Scrum là từ một đơn vị ngang hàng hoặc nhà quản lý, triển khai huấn luyện một cách chuyên nghiệp và toàn diện, và sau đó đưa ra các quyết định để một Nhóm triển khai thực tế trong một giai đoạn nhất định; cuối giai đoạn đó, Nhóm Phát triển sẽ đánh giá những trải nghiệm của mình và quyết định việc có tiếp tục triển khai hay không.

Một điều chắc chắn đó là trong Sprint đầu tiên thường có nhiều thách thức với Nhóm Phát triển, các lợi ích của Scrum có xu hướng hiện diện ở cuối Sprint, nhiều Nhóm Scrum mới thường kêu lên:

“Scrum khó thật, nhưng chắc chắn là nó tốt hơn hẳn tất cả những gì mà chúng ta đã làm trước đây!”



## **Phụ Lục: Thuật ngữ**

### **Burndown**

Là biểu đồ thể hiện xu hướng của công việc còn lại theo thời gian trong một Sprint, một Bản phát hành (release), hoặc Sản phẩm. Dữ liệu được lấy từ Sprint Backlog và Product Backlog, với công việc còn lại được theo dõi trên trục tung và khoảng thời gian (các ngày trong một Sprint, hoặc nhiều Sprint) theo dõi trên trục hoành.

### **Chicken - Gà**

Một người nào đó quan tâm đến dự án nhưng không có trách nhiệm trong dự án (không giữ các vai trò Nhóm Phát triển, Product Owner hay ScrumMaster).

### **Daily Scrum – Hợp Scrum Hằng ngày**

Một cuộc họp ngắn được tổ chức hàng ngày của mỗi Nhóm Phát triển, trong thời gian đó, các thành viên của nhóm rà soát công việc của mình, đồng bộ hóa công việc với người khác và báo cáo tiến độ công việc và các trở ngại để Scrum Master loại bỏ chúng. Đội sản xuất và ScrumMaster có thể phải tiến hành các hoạt động tiếp nối (follow-up) Daily Scrum để thích ứng với tình hình sắp tới tối ưu hóa Sprint.

### **Done – Hoàn thành**

Định nghĩa về sự hoàn thành (Done Definition, hay gọi tắt là Done – Hoàn thành) được đồng thuận giữa tất cả các bên và phù hợp với tiêu chuẩn, quy ước của tổ chức cũng như các chỉ dẫn khác. Khi một công việc được ghi nhận là “hoàn thành” tại cuộc họp Sơ kết Sprint, nó phải phù hợp với định nghĩa về sự hoàn thành này.

### **Estimated Work Remaining (Sprint Backlog Items) - Ước tính khối lượng công việc còn lại (Các hạng mục Sprint Backlog)**

Số giờ mà một thành viên của nhóm ước lượng để thực hiện một công việc nào đó. Ước lượng này được cập nhật hằng ngày khi thành viên đó thực hiện công việc trong Sprint Backlog. Đây là con số ước tính tổng số giờ còn lại để hoàn thành công việc, không kể đến số lượng người thực hiện công việc hay số giờ đã bỏ ra cho công việc ấy trong quá khứ.

### **Increment – Phần tăng trưởng**

Chức năng của sản phẩm được phát triển bởi Đội sản xuất trong mỗi Sprint, cái mà có thể chuyển giao được hoặc có thể được sử dụng bởi chủ sở hữu sản phẩm hay các bên liên quan.

### **Increment of Potentially Shippable Product Functionality – Phần tăng trưởng chuyển giao được**

Một phần nhỏ nhưng hoàn chỉnh của sản phẩm tổng thể hoặc hệ thống có thể được sử dụng bởi chủ sở hữu sản phẩm hoặc các bên liên quan.

### **Sprint – Phân đoạn nước rút**

Một phân đoạn (iteration) trong Scrum; là một chu kỳ phát triển nhằm tạo ra các phần tăng trưởng cho hệ thống. Sprint thường kéo dài từ một tuần một tháng. Trong suốt dự án, thời gian cho một Sprint là cố định. Từ “sprint” có nghĩa là giai đoạn nước rút, ám chỉ sự gấp gáp và tập trung cao độ trong khoảng thời gian ngắn để làm việc.

### **Pig – lợn**

Một người nào đó thực hiện một trong ba vai trò của Scrum (Nhóm Phát triển, Product Owner hoặc ScrumMaster), người đã có những cam kết và có quyền để thực hiện cam kết của mình.

### **Product Backlog**

Một danh sách ưu tiên của các yêu cầu với thời gian ước tính để biến chúng thành các tính năng hoàn chỉnh của sản phẩm. Các hạng mục được ưu tiên hơn trong Product Backlog được ước lượng cẩn thận hơn, và thường chính xác hơn các phần khác. Danh sách này có thể thay đổi khi có sự thay đổi trong điều kiện kinh doanh hoặc công nghệ.

### **Product Backlog Item – Hạng mục Product Backlog**

Một yêu cầu chức năng hay phi chức năng, và các vấn đề được sắp độ ưu tiên. Độ chính xác của ước lượng phụ thuộc vào tầm quan trọng và độ chi tiết của hạng mục đó. Phần có độ ưu tiên cao nhất sẽ được chọn trong Sprint tiếp theo để làm việc.

### **Product Owner – Chủ Sản phẩm**

Một trong ba vai trò của Nhóm Scrum. Là người chịu trách nhiệm quản lý Product Backlog để tối đa hóa giá trị của dự án. Chủ sở hữu sản phẩm có trách nhiệm đại diện cho lợi ích của tất cả mọi người có quyền lợi trong dự án và sản phẩm được tạo ra sau dự án.

### **Scrum Master**

Là một trong ba vai trò của Nhóm Scrum; là người chịu trách nhiệm cho quy trình Scrum, bao gồm việc triển khai đúng quy tắc, và tối đa hóa lợi ích từ Scrum.

### **Sprint Backlog**

Danh sách các nhiệm vụ xác định công việc của nhóm trong một Sprint. Danh sách này được cập nhật trong suốt Sprint. Nhóm Phát triển tự quản lý Sprint Backlog bằng việc cập nhật trạng thái thực thi các nhiệm vụ với người chịu trách nhiệm và thời gian còn lại để hoàn tất công việc tính tới thời điểm cập nhật.

### **Sprint Backlog Task – Công việc trong Sprint Backlog**

Một trong những nhiệm vụ mà nhóm hoặc một thành viên xác định theo yêu cầu để chuyển đổi các yêu cầu trong Product Backlog thành chức năng của hệ thống.

### **Sprint Planning meeting – Hợp Kế hoạch Sprint**

Cuộc họp diễn ra trong phạm vi tám giờ đồng hồ để khởi động mỗi Sprint. Hợp lập kế hoạch được chia thành hai phân đoạn bốn giờ. Trong bốn giờ đầu tiên Product Owner trình bày các yêu cầu có độ ưu tiên cao nhất cho nhóm phát triển. Nhóm Phát triển và Product Owner sẽ kết hợp để xác định số lượng các hạng mục sẽ được chọn để tiến hành phát triển trong Sprint tới. Sau bốn giờ đầu tiên, Nhóm Phát triển và Product Owner xác định mục tiêu của Sprint sắp tới. Trong suốt bốn giờ thứ hai của cuộc họp, Nhóm lập kế hoạch để đạt được mục đích của Sprint bằng các kĩ thuật phân tích và thiết kế cần thiết, và sau đó ghi lại chi tiết công dưới dạng một bản kế hoạch có tên Sprint Backlog.

### **Sprint Retrospective meeting - Hợp Cải tiến Sprint**

Trong phạm vi ba giờ, ScrumMaster sẽ tổ chức cho nhóm thực hiện công việc khảo sát lại toàn bộ quy trình làm việc của Sprint vừa qua để tìm ra các cải tiến trong Sprint tới, nhằm mang lại hiệu suất cao hơn cho nhóm.

### **Sprint Review meeting - Hợp Sơ kết Sprint**

Với khung thời gian bốn giờ, được tiến hành sau khi Sprint kết thúc, trong buổi họp này Nhóm Phát triển cùng với Product Owner và các bên liên quan đánh giá lại công việc của Sprint vừa qua. Bắt đầu bằng việc Nhóm Phát triển trình bày các phần việc đã hoàn tất trong Product Backlog, thảo luận về các cơ hội, hạn chế và phát hiện các vấn đề, và thảo luận về những gì tốt nhất có thể thực hiện tiếp theo (có khả năng dẫn tới thay đổi Product Backlog). Chỉ những chức năng hoàn thành mới được phép trình bày.

### **Stakeholder – người liên quan**

Người có sự quan tâm đến kết quả của một dự án (có tài trợ cho nó, sẽ sử dụng, hoặc sẽ bị ảnh hưởng bởi sản phẩm tạo ra từ dự án).

### **Team –Nhóm Phát triển (còn gọi là Đội Sản xuất)**

Một nhóm liên chức năng (cross-functional) tự quản lý để phát triển một gói tăng trưởng của sản phẩm sau mỗi Sprint.

### **Time box – Khung thời gian**

Là khoảng thời gian tối đa dành cho một hoạt động nào đó trong Scrum. Ví dụ, Daily Scrum là cuộc họp có khung thời gian là mười lăm phút và nó phải kết thúc trước phút thứ mười năm. Đối với các cuộc họp, nó có thể ngắn hơn. Đối với các Sprint, đòi hỏi phải chính xác.