



Hướng dẫn ngắn gọn về Lý thuyết và Thực hành Scrum

Phiên bản 2.0

Pete Deemer

GoodAgile

www.goodagile.com

Gabrielle Benefield

Evolve

www.evolvebeyond.com

Craig Larman

www.craiglarman.com

Bas Vodde

Odd-e

www.odd-e.com

Biên dịch: Nguyễn Khắc Nhật | Biên tập: Nguyễn Việt Khoa | Hiệu đính: Dương

Trọng Tấn

HỌC VIỆN AGILE



InfoQ
ENTERPRISE SOFTWARE
DEVELOPMENT SERIES

Nhấn gửi bạn đọc: Có nhiều tài liệu mô tả ngắn gọn Scrum trực tuyến, cuốn sách này hướng tới việc cung cấp mức độ chi tiết hơn về các kỹ thuật trong Scrum. Nó không dự định trở thành bước cuối cùng trong một quá trình đào tạo Scrum; các nhóm đang dự định áp dụng Scrum được khuyến cáo nên tự trang bị cho mình kiến thức bằng việc đọc kĩ các cuốn *Agile Project Management with Scrum (Quản lý Dự án Linh hoạt với Scrum)* của Ken Schwaber hoặc cuốn *Succeeding with Agile (Thành công với Agile)* của Mike Cohn và tận dụng ưu thế của rất nhiều khóa đào tạo và huấn luyện Scrum hiện đang có; mọi chi tiết đầy đủ có tại ScrumAlliance.org. Chúng tôi xin gửi lời cảm ơn tới Ken Schwaber, Tiến sĩ Jeff Sutherland và Mike Cohn vì những đóng góp quý giá.

Phiên bản mới nhất của cuốn Scrum Primer này có thể tìm tại:

http://www.infoq.com/minibooks/Scrum_Primer.

Các bản dịch có thể tìm tại: <http://www.scrumprimer.org/>

© 2012 Pete Deemer, Gabrielle Benefield, Craig Larman, Bas Vodde

Phương pháp Phát triển Truyền thống và hơn thế nữa

Phương pháp phát triển truyền thống với các nhóm *đơn chức năng*, vòng phản hồi chậm hoặc quá thừa, lập kế hoạch chi tiết theo lối ước đoán, và một luồng tuần tự đi từ phân tích cho đến kiểm thử đã không còn được thành công cho lắm trong thế giới nhiều biến động như ngày nay. Phương pháp này gây chậm trễ trong phản hồi, học tập và thu lại lợi nhuận trên đầu tư (ROI) do sự thiếu vắng của một phần mềm chạy tốt thực sự cho đến giai đoạn cuối của chu trình. Điều này dẫn đến thiếu minh bạch, thiếu khả năng cải tiến, giảm sự linh hoạt và tăng rủi ro trong cả hoạt động kinh doanh cũng như kỹ thuật.

Một giải pháp thay thế bao gồm các nhóm *liên chức năng* sử dụng mô hình phát triển lặp đã tồn tại trong nhiều thập kỷ nhưng đã không được sử dụng rộng rãi như mô hình truyền thống.

Scrum kết hợp các khái niệm phát triển-sản phẩm đã được thừa nhận vào trong một khung làm việc đơn giản, bao gồm: đội nhóm thực sự, nhóm liên chức năng, nhóm tự quản, các chu trình phản hồi ngắn khép kín, và giảm thiểu chi phí thay đổi. Các khái niệm này làm tăng tính linh hoạt (agility) và phản hồi, cho phép thu lại lợi nhuận trên đầu tư sớm hơn, và giảm thiểu rủi ro.

Tổng quan

Scrum là một khung phát triển trong đó các nhóm liên chức năng phát triển các sản phẩm hoặc dự án theo hình thức lặp và tăng trưởng. Scrum tổ chức quá trình phát triển thành các chu trình làm việc gọi là **Sprint**. Mỗi chu trình không kéo dài quá bốn tuần (phổ biến nhất là hai tuần) và diễn ra liên tiếp nhau mà không bị gián đoạn. Các Sprint được *đóng khung thời gian*, tức là chúng kết thúc vào một ngày nhất định cho dù công việc đã được hoàn thành hết hay chưa và *không được phép kéo dài thêm*. Thông

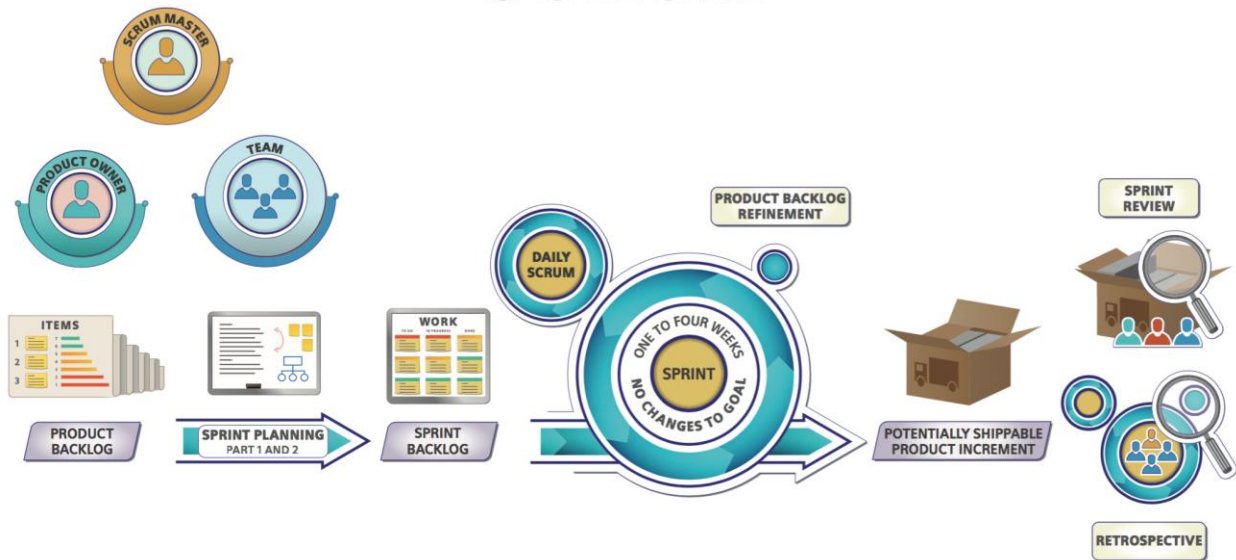
thường thì các Nhóm Scrum¹ chọn một độ dài Sprint và sử dụng nó cho đến khi họ có thể cải tiến và sử dụng chu trình ngắn hơn. Bắt đầu mỗi Sprint, một Nhóm *liên chức năng* được gọi là **Nhóm Phát triển**² (Development Team, bao gồm khoảng bảy người³) lựa chọn **các hạng mục** (yêu cầu khách hàng) từ một danh sách ưu tiên. Nhóm Phát triển thống nhất một mục tiêu chung mà họ tin rằng có thể chuyển giao được vào cuối Sprint, một thứ gì đó phải thấy được và thực sự được “hoàn thành”. Trong suốt Sprint, không có hạng mục mới nào được thêm vào; Scrum để dành các thay đổi cho Sprint *tiếp theo*, còn đối với Sprint ngắn hiện tại thì cố gắng tập trung vào một mục tiêu nhỏ, rõ ràng và tương đối ổn định. Hằng ngày Nhóm Phát triển họp với nhau trong một khoảng thời gian ngắn nhằm thanh tra tiến độ và điều chỉnh các bước cần thiết tiếp theo để hoàn thành công việc còn lại. Vào cuối Sprint, Nhóm Phát triển rà soát Sprint với các bên liên quan và trình bày phần mình đã xây dựng được. Mọi người nhận được các phản hồi và có thể đưa vào trong Sprint tiếp theo. Scrum nhấn mạnh vào sản phẩm chạy tốt ở cuối Sprint mà thực sự đã được “hoàn thành”; trong trường hợp của phần mềm, điều này có nghĩa là một hệ thống đã được tích hợp, thực hiện kiểm thử đầy đủ, viết tài liệu cho người dùng cuối, và có khả năng chuyển giao. Các vai trò, tạo tác và sự kiện chính được tóm lược trong Hình 1.

¹ Nhóm Scrum: là nhóm liên chức năng gồm toàn bộ Product Owner, Scrum Master và Nhóm Phát triển. Xem chi tiết trong phần “Các vai trò trong Scrum” trong tài liệu này. [*Chú thích của người dịch - ND*]

² Tài liệu Scrum Primer gốc chọn dùng từ Team để chỉ Development Team. Người dịch chọn cách dùng “Nhóm Phát triển” cho rõ nghĩa, và thống nhất với cách dùng thuật ngữ của các tác giả Scrum trong Scrum Guide [ND]

³ Về độ lớn của Nhóm Scrum, các tài liệu khác nhau sử dụng những con số khác nhau, ví dụ ScrumGuide của hai tác giả Scrum lưu ý số thành viên trong đội phát triển nên nằm trong khoảng 3-9. Nhiều chuyên gia Scrum khác đồng tình với cách lựa chọn kích thước nhóm như trong tài liệu này: 7, cộng trừ 2. [ND]

SCRUM



Hình 1: Tổng quan Scrum

Một nội dung chủ đạo trong Scrum là “thanh tra và thích nghi”. Do quá trình phát triển chắc chắn liên quan đến cả việc học tập, sáng tạo và sự bất ngờ, Scrum nhấn mạnh vào việc sử dụng các bước phát triển ngắn, thanh tra cả sản phẩm đạt được và sự hiệu quả của các kỹ thuật hiện tại, rồi sau đó điều chỉnh các mục tiêu của sản phẩm cũng như các kỹ thuật của quy trình. *Và cứ lặp lại mãi như vậy.*

Các Vai trò trong Scrum

Trong Scrum, có ba vai trò: Product Owner, Nhóm Phát triển, và ScrumMaster. Tất cả hợp thành Nhóm Scrum.

Product Owner chịu trách nhiệm tối ưu hóa lợi nhuận trên đầu tư (ROI - Return On Investment) thông qua việc xác định các tính năng của sản phẩm, chuyển thành một danh sách ưu tiên, quyết định hạng mục nào sẽ nằm phía trên của danh sách để đưa vào Sprint tiếp theo, và liên tục tái-sắp xếp và làm mịn danh sách này. Nếu là sản phẩm thương mại, Product Owner chịu trách nhiệm về lợi nhuận cũng như tổn thất của sản phẩm.

Trong trường hợp của một ứng dụng nội bộ, Product Owner không chịu trách nhiệm về ROI theo nghĩa của sản phẩm thương mại (sẽ tạo ra thu nhập), mà chịu trách nhiệm tối ưu hóa ROI theo nghĩa lựa chọn các hạng mục có giá trị cao nhất cho mỗi Sprint. Trong thực tế, “giá trị” là một khái niệm mập mờ và việc đánh giá độ ưu tiên có thể bị ảnh hưởng bởi mong muốn làm hài lòng các khách hàng chủ chốt, gắn với các mục tiêu chiến lược, hạn chế rủi ro, cải tiến hay các yếu tố khác. Trong một số trường hợp, Product Owner cũng chính là khách hàng; điều này thường xảy ra đối với các sản phẩm nội bộ. Trong một số trường hợp khác, khách hàng có thể là hàng triệu người với sự đa dạng về nhu cầu, khi đó vai trò Product Owner là tương tự như vị trí Product Manager (Giám đốc Sản phẩm) hoặc Product Marketing Manager (Giám đốc Tiếp thị Sản phẩm) trong nhiều tổ chức phát triển sản phẩm. Tuy nhiên, Product Owner thường khác với một Giám đốc Sản phẩm truyền thống ở chỗ họ chủ động và thường xuyên trao đổi với Nhóm Phát triển, ưu tiên làm việc với tất cả các bên liên quan và rà soát kết quả của mỗi Sprint thay vì ủy quyền các quyết định phát triển cho một quản lý dự án. Một lưu ý quan trọng là trong Scrum có một và chỉ một người phục vụ với tư cách Product Owner và có thẩm quyền tối cao của vai trò này, người này chịu trách nhiệm về giá trị của công việc, mặc dù không nhất thiết phải làm việc một mình.

Nhóm Phát triển(cách gọi khác: **Đội Phát triển**) xây dựng sản phẩm mà Product Owner yêu cầu, chẳng hạn một ứng dụng hoặc một trang web. Nhóm Phát triển trong Scrum là “liên chức năng”, có nghĩa là bao gồm tất cả các chuyên môn cần thiết để tạo được sản phẩm chuyển giao được sau mỗi Sprint. Nhóm Phát triển cũng là “tự tổ chức” (tự quản lý) với một mức độ tự chủ và tính trách nhiệm cao. Nhóm Phát triển quyết định số lượng các hạng mục (từ tập hợp do Product Owner đưa ra) để xây dựng trong một Sprint cũng như cách tốt nhất để đạt được mục tiêu đó.

Mỗi thành viên của Nhóm Phát triển đơn giản được gọi là một *thành viên nhóm*. Chú ý rằng không có các chức danh chuyên môn cố định trong một nhóm triển khai Scrum; không có chuyên gia phân tích nghiệp vụ, không có chuyên gia DBA, không có chuyên gia kiến trúc, không có trưởng nhóm,

không có chuyên gia thiết kế tương tác/UX, không có lập trình viên. Họ làm việc cùng nhau trong từng Sprint bằng bất cứ cách nào phù hợp để đạt được mục tiêu mà chính họ đã đưa ra.

Do chỉ có các *thành viên nhóm*, Nhóm Phát triển không chỉ là liên chức năng mà còn thể hiện việc *học tập lẫn nhau*: mỗi người sở hữu một thế mạnh riêng, nhưng vẫn tiếp tục học các chuyên môn khác. Mỗi người sẽ có các kỹ năng chính, kỹ năng phụ thứ hai và thậm chí là thứ ba. Điều này có nghĩa là họ “đi đến đâu có công việc”; các cá nhân nhận lấy các nhiệm vụ không quen thuộc lắm để giúp hoàn thành một hạng mục yêu cầu. Ví dụ, một người với kỹ năng chính là thiết kế tương tác có thể có kỹ năng phụ là kiểm thử tự động; một người với kỹ năng chính là viết tài liệu kỹ thuật cũng có thể trợ giúp với phân tích và lập trình.

Nhóm Phát triển trong Scrum có 7 (cộng trừ 2) người, và đối với sản phẩm phần mềm thì Nhóm Phát triển có thể bao gồm các cá nhân với kỹ năng phân tích, phát triển, kiểm thử, thiết kế giao diện, thiết kế cơ sở dữ liệu, thiết kế kiến trúc, viết tài liệu, và những kỹ năng khác. Nhóm phát triển sản phẩm đồng thời cung cấp ý tưởng cho Product Owner về cách để tạo ra sản phẩm tốt. Trong Scrum, Nhóm Phát triển sẽ trở nên năng suất và hiệu quả nhất khi các thành viên đều dành toàn bộ 100% để làm việc cho một sản phẩm trong suốt Sprint; Nhóm Phát triển tránh làm việc đa nhiệm giữa nhiều sản phẩm hoặc dự án để loại bỏ sự thất thoát tốn kém do phân tâm và chuyển đổi bối cảnh. Các nhóm ổn định thường có với năng suất cao hơn, do vậy nên tránh thay đổi thành viên Nhóm Phát triển. Các nhóm phát triển sản phẩm với quá nhiều người được tổ chức thành nhiều Nhóm Phát triển, mỗi nhóm tập trung vào một số tính năng của sản phẩm, với sự phối hợp nỗ lực chặt chẽ giữa các nhóm. Do mỗi nhóm thường làm tất cả mọi việc (lập kế hoạch, phân tích, lập trình, và kiểm thử) cho một tính năng hoàn chỉnh với khách hàng là trọng tâm (customer-centric) nên các Nhóm Phát triển còn được gọi là *Nhóm tính năng (feature team)*.

ScrumMaster giúp nhóm học và áp dụng Scrum để đạt được giá trị thương mại. ScrumMaster làm tất cả những gì trong thẩm quyền của mình

để giúp Nhóm Phát triển, Product Owner và tổ chức trở nên thành công. ScrumMaster *không* phải là người quản lý của các thành viên Nhóm Phát triển, cũng không phải là quản lý dự án, trưởng nhóm, hay đại diện của nhóm. Thay vào đó, ScrumMaster *phục vụ* Nhóm Phát triển; là người giúp loại bỏ các trở ngại, bảo vệ Nhóm Phát triển khỏi các tác động từ bên ngoài, và trợ giúp Nhóm Phát triển ứng dụng các kỹ thuật phát triển hiện đại. ScrumMaster cũng dạy, huấn luyện và hướng dẫn Product Owner, Nhóm Phát triển và phần còn lại của tổ chức trong việc áp dụng thành thạo Scrum. ScrumMaster là một *huấn luyện viên* và cũng là một *giáo viên*. ScrumMaster đảm bảo rằng tất cả mọi người (bao gồm cả Product Owner và những người trong ban quản trị) hiểu các nguyên lý và kỹ thuật của Scrum, và họ sẽ hỗ trợ dẫn dắt tổ chức vượt qua những thay đổi thường là rất khó khăn nhưng cần thiết để đạt được thành công với phát triển linh hoạt. Do Scrum giúp phát hiện nhiều trở ngại và nguy cơ ảnh hưởng đến hiệu quả của Nhóm Phát triển và Product Owner, một điều quan trọng đó là phải có ScrumMaster làm việc hăng hái để giúp giải quyết những vấn đề đó, nếu không Nhóm Phát triển và Product Owner sẽ khó mà thành công được. Cần phải có một ScrumMaster làm việc toàn thời gian, mặc dù đối với một Nhóm Phát triển nhỏ hơn thì có thể cử một thành viên đảm nhận vai trò này (khi đó sẽ đảm nhiệm ít công việc thường ngày hơn). Một ScrumMaster giỏi có thể xuất thân từ bất cứ nền tảng hoặc chuyên môn nào: Kỹ thuật, Thiết kế, Kiểm thử, Giám đốc Sản phẩm, Quản lý Dự án, hoặc Quản lý Chất lượng.

ScrumMaster và Product Owner không thể là cùng một người bởi vì trọng tâm công việc của họ quá khác nhau và việc kết hợp chúng lại thường dẫn đến sự nhầm lẫn và xung đột. Một kết quả không may mắn thường xảy ra khi kết hợp hai vai trò này là một Product Owner có phong cách quản lý vật (micro-managing), trái ngược với các nhóm tự quản lý mà Scrum yêu cầu. Không giống với một quản lý truyền thống, ScrumMaster không chỉ cho mọi người phải làm gì, hoặc gán công việc, mà tạo điều kiện cho quy trình, hỗ trợ Nhóm Phát triển để chính họ tự tổ chức và tự quản lý. Nếu trước đó ScrumMaster đã từng làm ở vị trí quản lý của Nhóm Phát triển

thì sẽ cần phải thay đổi đáng kể tư duy và phong cách tương tác của mình để Nhóm Phát triển có thể thành công với Scrum.

Lưu ý: không tồn tại vai trò của quản lý dự án trong Scrum. Bởi vì nó là không cần thiết; các nhiệm vụ truyền thống của một quản lý dự án đã được phân chia cho ba vai trò khác nhau trong Scrum. Phần lớn trách nhiệm này được chuyển cho Nhóm Phát triển và Product Owner hơn là cho ScrumMaster. Triển khai Scrum kèm với một quản lý dự án là biểu hiện của việc hiểu sai cơ bản Scrum và thường dẫn đến sự xung đột trách nhiệm, không rõ ràng về quyền hạn, và các kết quả kém tối ưu. Đôi khi một quản lý dự án (hoặc cựu quản lý dự án) có thể chuyển sang vai trò của ScrumMaster, nhưng sự thành công của việc này là phụ thuộc rất lớn vào cá nhân và sự hiểu biết của người ấy về sự khác nhau cơ bản giữa hai vai trò này, về cả các nhiệm vụ hằng ngày và tư duy cần thiết để thành công. Một cách tốt để hiểu thông suốt vai trò của ScrumMaster và bắt đầu phát triển các kỹ năng cốt lõi để thành công đó là tham gia khóa đào tạo ScrumMaster của ScrumAlliance.

Ngoài ba vai trò kể trên, còn có các bên liên quan khác đóng góp vào sự thành công của sản phẩm, bao gồm các nhà quản lý, khách hàng và người dùng cuối. Một vài người liên quan chẳng hạn như những người quản lý chức năng (ví dụ, người quản lý kỹ thuật) có thể thấy vai trò của họ nếu còn giá trị thì cũng thay đổi khi áp dụng Scrum. Ví dụ:

- họ hỗ trợ Nhóm Phát triển bằng cách tôn trọng các quy tắc và tinh thần của Scrum
- họ giúp loại bỏ các trở ngại mà Nhóm Phát triển và Product Owner chỉ ra
- họ sẵn sàng đóng góp chuyên môn và kinh nghiệm của mình

Trong Scrum, các cá nhân này thay thế thời gian đóng vai trò “bảo mẫu” trước đây của mình (gán công việc, ghi nhận báo cáo tiến độ, và các hình thức quản lý vật khác) bằng thời gian với vai trò “quân sự” và “người phục vụ” của Nhóm Phát triển (hướng dẫn, huấn luyện, trợ giúp loại bỏ trở ngại, trợ giúp giải quyết vấn đề, cung cấp các đóng góp mang tính sáng tạo, và

hướng dẫn phát triển các kỹ năng của các thành viên Nhóm Phát triển). Trong việc dịch chuyển này, các nhà quản lý có thể cần phải thay đổi phong cách quản trị; chẳng hạn, sử dụng cách hỏi Socratic để giúp Nhóm Phát triển tìm ra giải pháp của một vấn đề, thay vì chỉ đơn giản quyết định một giải pháp và gán nó cho Nhóm Phát triển.

Product Backlog

Khi một nhóm dự định chuyển sang Scrum thì trước khi có thể bắt đầu Sprint đầu tiên họ cần có một **Product Backlog**. Đây là một danh sách ưu tiên (được sắp xếp 1, 2, 3, ...) của các tính năng hướng-khách hàng (customer-centric).

Product Backlog tồn tại (và tiến hóa) trong suốt vòng đời của sản phẩm; nó cũng là lộ trình của sản phẩm (**Hình 2 và 3**). Tại bất cứ thời điểm nào, Product Backlog cũng là một góc nhìn duy nhất và mang tính quyết định của “tất cả mọi thứ có thể được Nhóm Phát triển hoàn thành theo thứ tự ưu tiên”. Chỉ tồn tại duy nhất một Product Backlog cho một sản phẩm; có nghĩa là Product Owner cần phải đưa ra các quyết định đánh giá ưu tiên xuyên suốt toàn bộ phạm vi, đại diện cho quyền lợi của các bên liên quan (bao gồm cả Nhóm Phát triển).

	Ước lượng công việc của Sprint...								
Độ ưu tiên	Hạng mục	Chi tiết (Wiki URL)	Kích thước Ước tính Ban đầu	1	2	3	4	5	6
1	Là một khách hàng, tôi muốn đưa một cuốn sách vào giỏ hàng (xem phác thảo giao diện trên trang wiki)	...	5						
2	Là một khách hàng, tôi muốn xóa một cuốn sách khỏi giỏ hàng	...	2						

3	Cải thiện hiệu năng xử lý giao dịch (xem các thông số hiệu năng mong muốn trong wiki)	...	13						
4	Nghiên cứu giải pháp để tăng tốc việc xác nhận thẻ tín dụng (xem các thông số hiệu năng mong muốn trong wiki)	...	20						
5	Cập nhật tất cả các máy dịch vụ lên Apache 2.2.3	...	13						
6	Chẩn đoán và sửa lỗi của đoạn mã xử lý hóa đơn (bugzilla ID 14823)	...	3						
7	Là một người mua hàng, tôi muốn tạo và lưu lại một danh sách yêu thích	...	40						
8	Là một người mua hàng, tôi muốn thêm và xóa các hạng mục trong danh sách yêu thích của mình	...	20						

Hình 2. Product Backlog



Hình 3. Quản lý trực quan: Các hạng mục Product Backlog trên tường

Product Backlog bao gồm nhiều *hạng mục* khác nhau, chủ yếu là các tính năng mới cho khách hàng (“cho phép tất cả người dùng thêm sách vào trong giỏ hàng”), và còn có các mục tiêu cải tiến kỹ thuật *trọng yếu* (ví dụ, “viết lại hệ thống từ C++ sang Java”), các mục tiêu cải tiến (ví dụ, “tăng tốc độ các kiểm thử”), các công việc nghiên cứu (“nghiên cứu giải pháp để tăng tốc độ xác nhận thẻ tín dụng”), và còn có thể là các lỗi đã được phát hiện (“chẩn đoán và sửa các lỗi của của đoạn mã xử lý hóa đơn”) nếu chỉ có ít lỗi. (Một hệ thống với quá nhiều lỗi thì thường có một hệ thống theo dõi lỗi riêng biệt.)

Các hạng mục Product Backlog được mô tả theo bất cứ cách nào miễn là đảm bảo rõ ràng và nhất quán. Trái ngược với một hiểu nhầm khá phổ biến, Product Backlog *không* chứa các “user story”; nó đơn giản chỉ là chứa các *hạng mục*. Các hạng mục này có thể được thể hiện dưới dạng user story, user case, hoặc bất cứ hình thức ghi nhận yêu cầu nào mà nhóm thấy phù hợp. Nhưng cho dù sử dụng hình thức nào đi nữa thì lượng lớn các hạng mục phải tập trung vào chuyển giao giá trị cho khách hàng.

Một Product Backlog tốt cần đạt được tiêu chí DEEP...

Detailed appropriately (Chi tiết hợp lý). Các hạng mục với độ ưu tiên cao cần được làm mịn và chi tiết hơn so với các hạng mục với độ ưu tiên thấp, bởi vì chúng được lựa chọn để thực hiện trước. Ví dụ, 10% trên cùng của Backlog có thể bao gồm các hạng mục rất nhỏ và được phân tích kỹ, và 90% còn lại thì ít chi tiết hơn.

Estimated (Được ước lượng). Các hạng mục dành cho bản phát hành hiện tại cần phải được ước lượng, và hơn nữa cần phải xem xét tái-ước lượng trong mỗi Sprint khi mà mọi người học và có xuất hiện thêm thông tin mới. Nhóm Phát triển cung cấp cho Product Owner *khối lượng công việc* ước tính của từng hạng mục trong Product Backlog, và cũng có thể bao gồm cả ước tính *rủi ro kỹ thuật*. Product Owner và các bên liên quan về nghiệp vụ khác cung cấp thông tin về giá trị của yêu cầu sản phẩm, có thể bao gồm lợi nhuận thu được, chi phí được giảm thiểu, rủi ro kinh doanh, tầm quan trọng đối với một số bên liên quan, và nhiều thứ khác.

Emergent (Tiến hóa). Đáp lại việc học tập và biến đổi, Product Backlog thường xuyên được làm mịn. Trong mỗi Sprint, các hạng mục có thể được thêm vào, xóa đi, hoặc thay đổi, chia nhỏ và thay đổi độ ưu tiên. Do vậy, Product Backlog liên tục được cập nhật bởi Product Owner để thể hiện các thay đổi trong nhu cầu của khách hàng, các ý tưởng hoặc hiểu biết mới, động thái của các đối thủ cạnh tranh, các rào cản kỹ thuật vừa xuất hiện, v.v...

Prioritized (Sắp xếp theo độ ưu tiên). Các hạng mục ở phía trên của Product Backlog được đánh độ ưu tiên hoặc *sắp xếp* theo trật tự 1-N. Nhìn chung, các hạng mục có độ ưu tiên cao nhất cần phải chuyển giao phần lớn *lợi ích tương xứng với đồng tiền mà bạn đã bỏ ra (bang for your buck)*: nhiều lợi ích (giá trị thương mại) với ít tiền nhất (chi phí). Một lí do khác có thể dẫn đến tăng độ ưu tiên của một hạng mục là nhằm *giải quyết sớm các rủi ro lớn trước khi chúng tấn công bạn*.

Phương pháp phát triển truyền thống thường không nhấn mạnh vào việc chuyển giao lợi ích cao nhất tương xứng với đồng tiền bỏ ra, nhưng đây lại chính là nội dung chủ đạo của Scrum, do vậy Product Owner cần phải học cách để định giá phần lợi ích của “giá trị thương mại”. Đây là một thứ mà ScrumMaster có thể giúp Product Owner học. Vậy “giá trị thương mại” có nghĩa là gì? Một số nhóm phát triển sản phẩm sử dụng một đơn vị đơn giản là ước tính điểm-giá trị tương đối cho mỗi hạng mục Product Backlog. Đơn vị này tổng hợp các “phỏng đoán” của nhiều yếu tố bao gồm mức tăng doanh thu, mức giảm chi phí, ý muốn của các bên liên quan, tạo sự khác biệt trên thị trường, v.v.. Một vài nhóm thì lại đầu tư cho một hạng mục cụ thể bằng một hoặc nhiều khách hàng trả tiền cho việc phát triển nó rồi sau đó sử dụng doanh thu chính xác (trong ngắn hạn) của hạng mục đó như là đại diện của giá trị. Đối với một số nhóm khác thì việc ước tính dựa trên giá trị của một hạng mục-cụ thể này là quá chi tiết và không tập trung; họ sử dụng một phương pháp rộng hơn đó là dựa-trên-kết-quả-kinh-doanh (“tăng lượng đăng ký thêm 10% cho đến ngày 1 tháng 9”) trong đó giá trị chỉ được chuyển giao khi nhiều hạng mục có-đóng-góp-cho-kết-quả được chuyển giao cùng nhau. Trong trường hợp này, Product Owner cần

phải xác định phần tăng trưởng tiếp theo của Sản phẩm Hữu dụng Tối thiểu (Minimum Viable Product).

Để ước tính khối lượng công việc, một kỹ thuật phổ biến đó là ước tính theo kích thước tương đối (hệ số nỗ lực, độ phức tạp, và tính bất định) sử dụng đơn vị “story point” hoặc đơn giản là “point”.

Đây chỉ là những gợi ý; Scrum không quy định kỹ thuật để mô tả hoặc đánh độ ưu tiên của các hạng mục trong Product Backlog và nó cũng không quy định các kỹ thuật hay đơn vị ước tính.

Một kỹ thuật được sử dụng phổ biến trong Scrum là theo dõi lượng công việc hoàn thành trong mỗi Sprint; ví dụ, trung bình hoàn thành 26 điểm (point) trong một Sprint. Với thông tin này, nếu giá trị trung bình được giữ nguyên và không có gì khác thay đổi, người ta có thể trù liệu được ngày phát hành mà tất cả các tính năng đều được hoàn thành, hoặc tính xem bao nhiêu tính năng có thể được hoàn thành vào một ngày xác định nào đó. Giá trị trung bình này được gọi là “tốc độ” (velocity). Tốc độ được biểu diễn cùng đơn vị với ước tính kích thước của hạng mục Product Backlog.

Các hạng mục trong Product Backlog có thể dao động đáng kể về kích thước hay khối lượng công việc. Các hạng mục lớn được chia thành các hạng mục nhỏ hơn trong phiên Làm mịn Product Backlog (Product Backlog Refinement) hoặc trong buổi Lập kế hoạch Sprint (Sprint Planning Meeting). Các hạng mục nhỏ có thể được hợp nhất lại với nhau. Các hạng mục Product Backlog dành cho một vài Sprint tiếp theo cần phải được làm mịn và đủ nhỏ để Nhóm Phát triển hiểu đúng và giúp các dự báo được đưa ra trong buổi Lập kế hoạch Sprint trở nên có nghĩa; kích thước nhỏ này được xem là “có thể thực hiện được”.

Những cải tiến kỹ thuật lớn mà tiêu tốn nhiều thời gian và tiền bạc thì cần phải đưa vào Product Backlog vì chúng có thể là một khoản đầu tư phát sinh trong kinh doanh mà cuối cùng thì người Product Owner có-định-hướng-thương-mại cũng phải bỏ ra. Chú ý rằng trong Scrum thì Nhóm Phát triển có quyền độc lập trong việc quyết định số lượng hạng mục Product Backlog được đưa vào trong một Sprint, do vậy họ hoàn toàn tự

do trong việc chọn lấy các hạng mục cải tiến kỹ thuật nhỏ bởi vì chúng được xem như là một phần chi phí thông thường trong kinh doanh và việc này là cần thiết để các nhà phát triển có thể làm tốt công việc của mình. Điều này có nghĩa là, trong mỗi Sprint, *phần lớn* thời gian của Nhóm Phát triển là dành cho các mục tiêu của Product Owner chứ không phải là các công việc kỹ thuật nội bộ.

Một trong những ngộ nhận về Scrum là nó ngăn cản viết các đặc tả chi tiết; trong thực tế, Product Owner và Nhóm Phát triển quyết định mức độ chi tiết cần thiết, và điều này cũng thay đổi tùy theo từng hạng mục Backlog, phụ thuộc vào hiểu biết của Nhóm Phát triển và một số yếu tố khác. Hãy chỉ ra điều gì là quan trọng trong khoảng không gian nhỏ nhất có thể. Nói cách khác, không nên mô tả tất cả mọi chi tiết có thể có của một hạng mục mà chỉ nên làm rõ những gì cần thiết đủ để hiểu tốt, sau đó tăng cường thông qua việc trao đổi liên tục giữa Nhóm Phát triển và Product Owner và các bên liên quan. Các Hạng mục Product Backlog có độ ưu tiên thấp và sẽ không được thực hiện trong thời gian sắp tới thường được để ở dạng “thô” (lớn, với các yêu cầu có ít chi tiết). Các Hạng mục Product Backlog có độ ưu tiên lớn, được làm mịn và sẽ được triển khai sớm thường có nhiều chi tiết hơn.

Định nghĩa Hoàn thành

Kết quả của mỗi Sprint được gọi chính thức là Phần tăng trưởng Sản phẩm Có khả năng Chuyển giao được (Potentially Shippable Product Increment). Trước khi bắt đầu Sprint đầu tiên, Product Owner, Nhóm Phát triển và ScrumMaster cần phải rà soát lại những thứ cần làm để biến một hạng mục Product Backlog trở thành có thể chuyển giao được. Tất cả các hoạt động cần thiết để chuyển giao sản phẩm cần được đưa vào trong định nghĩa của Có khả năng Chuyển giao được (Potentially Shippable) và do đó cần phải được hoàn thành trong Sprint.

Không may, khi các Nhóm Phát triển bắt đầu sử dụng Scrum, họ thường không thể đạt được mục tiêu chuyển giao một Phần tăng trưởng Có khả

năng Chuyển giao được sau mỗi Sprint. Điều này thường là do nhóm thiếu tự động hóa hoặc không đủ liên chức năng (ví dụ, những người viết tài liệu kỹ thuật không có mặt trong Nhóm Phát triển). Theo thời gian, Nhóm Phát triển cần phải cải thiện để có khả năng chuyển giao một Phần tăng trưởng Có khả năng Chuyển giao được vào cuối mỗi Sprint, nhưng để bắt đầu thì họ cần tạo một cơ sở khởi đầu với những khả năng hiện có của mình. Những cơ sở này được ghi vào trong Định nghĩa Hoàn thành.

Trước Sprint đầu tiên, Product Owner và Nhóm Phát triển cần thống nhất một Định nghĩa Hoàn thành, nó là một tập con của các hoạt động cần thiết để tạo ra một Phần tăng trưởng Sản phẩm Có khả năng Chuyển giao được (đối với một Nhóm Phát triển tốt thì hai tập này là giống nhau). Nhóm Phát triển sẽ lập kế hoạch công việc của Sprint dựa theo Định nghĩa Hoàn thành này.

Một Product Owner tốt sẽ luôn cố gắng để Định nghĩa Hoàn thành bám sát nhất có thể với Khả năng Chuyển giao được vì nó sẽ giúp tăng tính minh bạch trong phát triển và giảm thiểu *chậm trễ cũng như rủi ro*. Nếu Định nghĩa Hoàn thành không giống với Khả năng Chuyển giao được thì công việc sẽ bị trì hoãn đến trước ngày phát hành gây ra *rủi ro và chậm trễ*. Công việc bị chậm trễ đôi khi còn được gọi là *công việc chưa hoàn thành (undone work)*.

Một Nhóm Scrum cần phải liên tục cải tiến, điều này được phản ánh vào trong việc mở rộng Định nghĩa Hoàn thành.

Lập kế hoạch Sprint

Tóm lược: Một buổi chuẩn bị cho Sprint, thường được chia làm hai phần (phần một là “làm gì” và phần hai là “làm như thế nào”).

Thành phần tham dự:

Phần Một: Product Owner, Nhóm Phát triển, ScrumMaster.

Phần Hai: Nhóm Phát triển, ScrumMaster, Product Owner (không bắt buộc nhưng cần phải sẵn sàng để trả lời các thắc mắc).

Thời gian: Mỗi phần được đóng khung trong một giờ tương ứng với mỗi tuần của Sprint.

Buổi **Lập kế hoạch Sprint** diễn ra vào đầu mỗi Sprint. Nó được chia làm hai phần nhỏ khác nhau, phần đầu tiên được gọi là **Lập kế hoạch Sprint Phần Một**.

Trong **Lập kế hoạch Sprint Phần Một**, Product Owner và Nhóm Phát triển rà soát lại những hạng mục có độ ưu tiên cao trong Product Backlog mà Product Owner muốn triển khai trong Sprint này. Thông thường, những hạng mục này đã được phân tích kỹ trong các Sprint trước (trong buổi Làm mịn Product Backlog), do vậy lúc này chỉ cần làm rõ những thắc mắc nhỏ còn lại. Trong phần này, Product Owner và Nhóm Phát triển thảo luận các mục tiêu và hiện trạng của các hạng mục có độ ưu tiên cao trong Product Backlog, giúp Nhóm Phát triển hiểu rõ mong muốn của Product Owner. Phần Một tập trung để hiểu Product Owner muốn những gì và *tại sao* lại cần chúng. Kết thúc Phần Một, Product Owner (người lúc nào cũng bận rộn) có thể rời đi nhưng *phải* luôn sẵn sàng (chẳng hạn là qua điện thoại) trong suốt Phần Hai của buổi Lập kế hoạch.

Trong Phần Một, Nhóm Phát triển và Product Owner có thể đưa ra **Mục tiêu Sprint**. Đây là một đoạn tóm lược lại mục tiêu của Sprint, lý tưởng nhất là nó có một nội dung cô đọng. Mục tiêu Sprint cũng giúp cho Nhóm Phát triển có được sự linh hoạt-trong-phạm vi liên quan đến những thứ mà thực tế họ chuyển giao được, bởi vì mặc dù họ có thể phải loại bỏ một số hạng mục (do Sprint được đóng khung thời gian), họ vẫn cần phải cam kết chuyển giao thứ gì đó thấy được và đã “hoàn thành” như trong tinh thần của Mục tiêu Sprint.

Các hạng mục được lựa chọn cho một Sprint nên có độ lớn bao nhiêu là vừa? Mỗi hạng mục cần phải được chia đủ nhỏ để con số ước tính là nhỏ hơn đáng kể so với toàn bộ Sprint. Một gợi ý phổ biến đó là một hạng mục được ước tính đủ nhỏ để toàn bộ Nhóm Phát triển có thể hoàn thành trong vòng một phần tư Sprint hoặc ít hơn.

Lập kế hoạch Sprint Phần Hai tập trung vào việc *làm thế nào* để triển khai các hạng mục mà Nhóm Phát triển đã quyết định lựa chọn. Nhóm Phát triển dự tính số lượng hạng mục⁴ mà họ có thể hoàn thành khi kết thúc Sprint, bắt đầu từ phần trên cùng của Product Backlog (nói cách khác là bắt đầu với các hạng mục có độ ưu tiên cao nhất đối với Product Owner) rồi lần lượt đi xuống phía dưới của danh sách. *Đây chính là một kỹ thuật cốt lõi trong Scrum: Nhóm Phát triển quyết định lượng công việc mà họ sẽ hoàn thành, thay vì được giao bởi Product Owner.* Điều này giúp việc dự báo trở nên đáng tin hơn vì Nhóm Phát triển dựa vào phân tích và kế hoạch của chính họ. Trong khi Product Owner không kiểm soát được lượng công việc mà Nhóm Phát triển sẽ nhận thì vẫn an tâm rằng các hạng mục đó được lấy ra từ phần trên của Product Backlog, tức là các hạng mục mà mình đã đánh giá là quan trọng nhất. Nhóm Phát triển có thể thuyết phục để lựa chọn các hạng mục ở xa phía dưới của danh sách; việc này thường xảy ra khi Nhóm Phát triển và Product Owner thấy rằng một vài hạng mục có độ ưu tiên thấp nhưng dễ dàng ăn khớp và phù hợp với các hạng mục có độ ưu tiên cao.

Buổi Lập kế hoạch Sprint thường kéo dài vài giờ đồng hồ nhưng không nhiều hơn bốn giờ đối với một Sprint hai tuần. Nhóm Phát triển đưa ra các dự báo nghiêm túc để hoàn thành công việc, việc này nhất thiết phải được suy tính cẩn thận để đảm bảo thành công. Phần Một và Phần Hai có khung thời gian bằng nhau; đối với một Sprint hai tuần thì mỗi phần kéo dài tối đa hai giờ đồng hồ.

Scrum không quy định chính xác cách thực hiện Phần Hai của buổi Lập kế hoạch Sprint. Một số nhóm sử dụng tốc độ từ các Sprint trước để định hướng khối lượng công việc định làm. Một số nhóm khác sử dụng một

⁴ Lưu ý: trong Scrum Guide, các tác giả có đề cập một ý “Nhóm Phát triển có quyền lựa chọn số lượng hạng mục để triển khai trong mỗi Sprint”, và đặt nó trong mô tả về Phần Một của Lập kế hoạch Sprint. Trong khi đó, ở đây các tác giả Scrum Primer ghi rõ *cách làm* để Nhóm Phát triển có thể lựa chọn ra số lượng đó, và đặt ở Phần Hai – vốn liên quan đến câu hỏi “Làm sao để đạt được Mục tiêu Sprint”. Như vậy không có xung đột nào về mặt ý nghĩa liên quan đến định nghĩa trong hai tài liệu. [ND]

phương pháp chi tiết hơn đó là bắt đầu bằng việc tính khả năng sản xuất của nhóm.

Khi sử dụng phương pháp tính khả năng sản xuất, trong Phần Hai của buổi Lập kế hoạch Sprint Nhóm Phát triển tính toán lượng thời gian mà mỗi thành viên dành cho các công việc của Sprint. Phần lớn các nhóm đều giả định rằng các thành viên chỉ có thể tập trung cho các công việc của Sprint từ 4-6 giờ mỗi ngày, thời gian còn lại là dành cho email, nghỉ trưa, facebook, họp hành, và uống cà phê. Khi đã xác định được khả năng sản xuất thì Nhóm Phát triển cần phải tính số lượng hạng mục Product Backlog mà mình có thể hoàn thành trong khoảng thời gian đó, và mình dự định sẽ hoàn thành chúng bằng cách nào. Việc này thường bắt đầu bằng một thảo luận về thiết kế trên một tấm bảng. Khi mọi người đã hiểu rõ thiết kế tổng quan thì Nhóm Phát triển phân tách các hạng mục Product Backlog thành các công việc chi tiết. Trước khi lựa chọn các hạng mục Product Backlog, Nhóm Phát triển có thể tập trung liệt kê các công việc của mục tiêu cải tiến đã được đưa ra từ buổi Cải tiến Sprint trước. Sau đó, Nhóm Phát triển lựa chọn hạng mục đầu tiên của Product Backlog - tức là hạng mục có độ ưu tiên cao nhất của Product Owner - rồi đi dần xuống phía dưới cho đến khi “đầy”. Với mỗi hạng mục, Nhóm Phát triển tạo một danh sách các công việc tương ứng. Các công việc này có thể là được phân rã từ các hạng mục Product Backlog hoặc là chính bản thân các hạng mục Product Backlog nếu chúng đủ nhỏ để chỉ cần thực hiện trong vài giờ. Danh sách công việc cần hoàn thành trong Sprint được gọi là **Sprint Backlog** (Hình 4 và Hình 5).

Ước tính Khối lượng công việc									
Còn lại vào cuối Ngày...									
Hạng mục Product Backlog	Công việc của Sprint	Người thực hiện	Ước tính khối lượng công việc ban đầu	1	2	3	4	5	6

Là khách hàng, tôi muốn thêm một cuốn sách vào trong giỏ hàng	Chỉnh sửa cơ sở dữ liệu		5						
	Tạo trang web (UI)		8						
	Tạo trang web (nghiệp vụ Javascript)		13						
	Viết kiểm thử chấp nhận tự động		13						
	Cập nhật trang trợ giúp khách hàng		3						
	...								
Cải thiện hiệu suất xử lý giao dịch	Hợp nhất mã DCP và hoàn thành kiểm thử ở mức tăng		5						
	Hoàn thành máy đặt hàng cho pRank		8						
	Thay đổi DCP và bộ đọc để sử dụng API http của pRank		13						

Hình 4. Ví dụ về một cách để tạo Sprint Backlog

Khi kết thúc buổi Lập kế hoạch Sprint, Nhóm Phát triển đưa ra một mục tiêu thực tế mà họ tin là có thể chuyển giao được vào cuối Sprint. Theo truyền thống, đây được gọi là Cam kết Sprint - nhóm cam kết làm những gì tốt nhất có thể để đạt được mục tiêu của mình. Không may, điều này đôi khi được diễn giải sai thành một lời thề viết-bằng-máu thay vì được hiểu là nhóm thực sự “hết mình vì nó”. Để tránh sự hiểu lầm này, mục tiêu-sprint cần được xem như là một “dự báo” được dùng để truyền đạt tới Product Owner.

Scrum khuyến khích các thành viên đa-kỹ năng, thay vì chỉ “làm đúng danh xưng của công việc”, chẳng hạn “kiểm thử viên” thì chỉ làm các công việc kiểm thử. Nói cách khác, các thành viên của Nhóm Phát triển “đi đến bất cứ đâu có việc” và trợ giúp hết sức mình. Nếu có quá nhiều công việc liên quan đến kiểm thử thì *toàn bộ* thành viên Nhóm Phát triển có thể hỗ trợ.

Điều này không có nghĩa là tất cả mọi người đều đa năng; một vài người giỏi kỹ năng kiểm thử (cũng như việc khác) nhưng các thành viên Nhóm Phát triển làm việc cùng nhau và học thêm các kỹ năng mới. Do vậy, khi thiết lập và ước lượng công việc trong Lập kế hoạch Sprint, mọi người không nhất thiết - và cũng không nên - lựa chọn các công việc mà “mình có thể làm tốt nhất”. Thay vào đó, tốt hơn hết là chỉ chọn một công việc tại một thời điểm, đồng thời cũng nên xem xét lựa chọn các công việc với mục đích thúc đẩy việc học (chẳng hạn bằng cách làm việc cùng với một chuyên gia). Đây cũng là một lý do để không gán trước công việc trong buổi Lập kế hoạch Sprint, thay vào đó nên để việc này diễn ra “khi cần thiết” trong suốt Sprint.

Tất cả những điều này ý nói rằng, *hiếm* khi *John* phải làm một công việc nhất định nào đó chỉ vì đối với những người khác thì việc này mất quá nhiều thời gian hoặc là họ không thể học được - chẳng hạn John là người duy nhất có kỹ năng nghệ thuật để vẽ tranh. Những thành viên khác của Nhóm Phát triển sẽ không thể vẽ nổi một “người que” (stick man) nếu họ cứ mãi phụ thuộc vào người khác. Trong trường hợp hiếm gặp này - còn nếu nó là không hiếm hoặc không trở nên ít hơn khi Nhóm học được nhiều hơn thì có nghĩa là có điều gì đó không ổn đang xảy ra - có thể sẽ phải đặt ra câu hỏi liệu tổng lượng công việc đã đưa vào kế hoạch mà *cần* John xử lý có thể làm xong trong Sprint ngắn này hay không.

Nhiều nhóm có Sprint Backlog dưới dạng một bảng công việc treo tường (thường được gọi là **Bảng Scrum**). Trong suốt Sprint, các công việc (viết trên các mảnh giấy dán - stickynote) được chuyển qua lại giữa các cột được dán nhãn là “Việc cần làm”, “Việc đang làm”, và “Hoàn thành”. Xem Hình 5.



Hình 5. Quản lý Trực quan - Các công việc của Sprint Backlog dán trên tường

Một trong các trụ cột của Scrum là một khi Nhóm Phát triển đã thiết lập mục tiêu cho Sprint thì bất cứ sự bổ sung hay thay đổi nào cũng phải được lùi lại cho đến Sprint tiếp theo. Điều này có nghĩa là nếu đến giữa Sprint mà Product Owner quyết định Nhóm phải triển khai một hạng mục mới thì không thể thay đổi ngay mà cần chờ đến khi bắt đầu Sprint tiếp theo. Nếu một yếu tố từ bên ngoài xuất hiện dẫn đến sự thay đổi đáng kể độ ưu tiên, và Nhóm Phát triển sẽ lãng phí thời gian của họ nếu tiếp tục làm việc thì Product Owner hoặc Nhóm Phát triển có thể kết thúc Sprint. Nhóm ngừng làm việc và một buổi Lập kế hoạch Sprint mới được triển khai để bắt đầu Sprint mới. Sự gián đoạn gây ra bởi việc này thường là rất lớn, đủ để khiến Product Owner và Nhóm Phát triển phải cân nhắc kỹ càng trước khi đi đến các quyết định gay cấn như vậy.

Có một tác động mạnh mẽ và tích cực xuất phát từ việc Nhóm Phát triển được bảo vệ khỏi các thay đổi mục tiêu trong suốt Sprint. Thứ nhất, Nhóm

Phát triển bắt đầu làm việc khi biết chắc chắn tuyệt đối rằng mục tiêu của họ không thay đổi, điều này giúp Nhóm Phát triển tập trung hơn vào việc hoàn thành nhiệm vụ. Thứ hai, nó buộc Product Owner phải suy nghĩ thật kỹ về các hạng mục được đánh giá ưu tiên trong Product Backlog và chuyển cho Nhóm Phát triển thực hiện trong Sprint.

Khi tuân thủ các quy tắc này của Scrum thì Product Owner được hưởng lợi ở hai điểm. Thứ nhất, anh ta có thể tin tưởng rằng Nhóm Phát triển đã cam kết làm việc hết mình để hoàn thành tập các công việc thực tế và rõ ràng mà họ đã chọn. Theo thời gian, một Nhóm Phát triển có thể trở nên thành thạo trong việc lựa chọn và chuyển giao một dự báo thực tế. Thứ hai, Product Owner có thể thực hiện bất cứ sự thay đổi nào mình muốn với Product Backlog trước khi bắt đầu Sprint tiếp theo. Ở thời điểm đó, mọi sự bổ sung, loại bỏ, sửa đổi, và tái-đánh-giá-ưu-tiên đều có thể thực hiện và được chấp nhận. Trong khi Product Owner không được phép thay đổi các hạng mục đang được triển khai trong Sprint hiện tại thì anh ta cũng chỉ phải chờ đợi trong khoảng thời gian của một Sprint hoặc ít hơn để đưa vào bất cứ sự thay đổi nào mà mình muốn. Có nhiều dấu hiệu dẫn đến sự thay đổi - thay đổi hướng phát triển, thay đổi yêu cầu, hay đơn giản chỉ là thay đổi suy nghĩ - và có thể đây là lí do khiến các Product Owner thường say mê Scrum như bất cứ ai khác.

Scrum Hằng ngày

Tóm lược: Cập nhật và phối hợp giữa các thành viên Nhóm Phát triển.

Thành phần tham dự: Nhóm Phát triển bắt buộc có mặt; Product Owner không bắt buộc. ScrumMaster thường có mặt để đảm bảo Nhóm Phát triển tiến hành tốt sự kiện này.

Thời gian: Tối đa 15 phút.

Một khi Sprint bắt đầu, Nhóm Phát triển thực hiện một kỹ thuật cốt lõi khác của Scrum đó là: **Scrum Hằng ngày**. Đây là một buổi trao đổi ngắn (15 phút hoặc ít hơn) diễn ra hằng ngày vào một thời điểm ấn định sẵn. Tất cả các thành viên của Nhóm Phát triển đều tham dự. Để đảm bảo ngắn gọn,

tất cả mọi người đều nên đứng. Đây chính là cơ hội để Nhóm đồng bộ công việc và thông báo cho nhau biết về các trở ngại gặp phải. Trong buổi Scrum Hằng ngày, từng thành viên một lần lượt *trình bày với mọi người* ba thứ: (1) Mình đã làm được gì kể từ buổi gặp mặt trước?; (2) Mình sẽ làm những gì từ bây giờ đến buổi gặp mặt tiếp theo?; và (3) Mình gặp phải trở ngại gì?. Lưu ý rằng buổi Scrum Hằng ngày không phải là một buổi báo cáo tiến độ cho một người quản lý; mà nó là lúc để các thành viên của Nhóm Phát triển tự tổ chức có thể chia sẻ với nhau về tình hình công việc, giúp họ phối hợp tốt hơn. Một thành viên ghi lại các khó khăn, sau đó ScrumMaster chịu trách nhiệm giúp các thành viên Nhóm Phát triển giải quyết chúng. Không nên có hoặc nếu có thì chỉ nên hạn chế các cuộc thảo luận chi tiết trong suốt buổi Scrum Hằng ngày. Nội dung chính của buổi trao đổi là *trình bày* câu trả lời cho ba câu hỏi ở trên. Nếu cần thiết phải thảo luận thì nên tổ chức ngay sau khi kết thúc buổi Scrum Hằng ngày, chia thành một hoặc nhiều các nhóm thảo luận cùng lúc, tuy nhiên, trong Scrum thì không ai phải bắt buộc tham gia các cuộc họp này cả. Các cuộc thảo luận này chính là để một vài thành viên hoặc toàn bộ Nhóm Phát triển thích nghi với các thông tin nhận được trong buổi Scrum Hằng ngày: nói cách khác, đây cũng chính là một chu trình thanh tra và thích nghi nữa của Scrum. Đối với những Nhóm Phát triển mới sử dụng Scrum, *không* nên để những người quản lý hay những người ở vị trí có quyền lực tham dự buổi Scrum Hằng ngày. Nó sẽ làm cho Nhóm Phát triển cảm thấy “bị giám sát” - tạo ra áp lực để báo cáo tiến độ tốt hơn hằng ngày (đây là kỳ vọng không thực tế) và ngại đưa ra các vấn đề - và nó gây nguy hại đến tính tự quản của Nhóm Phát triển, dẫn đến tình trạng quản lý vật (micro-management). Sẽ là có ích hơn nếu các bên liên quan đến gặp Nhóm Phát triển sau buổi trao đổi này và cung cấp sự giúp đỡ cần thiết để loại bỏ những trở ngại đang làm chậm tiến độ của Nhóm Phát triển.

Theo dõi tiến độ trong suốt Sprint

Nhóm trong Scrum hoạt động theo cơ chế tự quản, và để đảm bảo thành công thì họ cần phải nắm được tình hình công việc của mình. Hằng ngày,

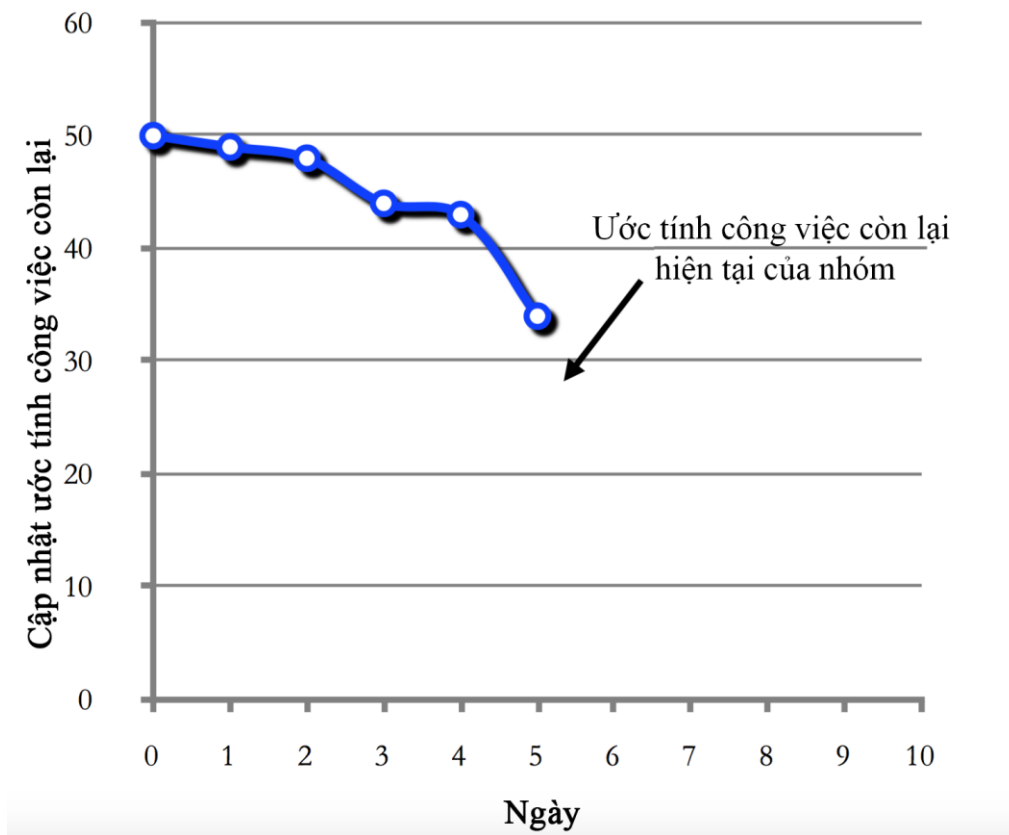
các thành viên Nhóm Phát triển cập nhật ước tính khối lượng công việc còn lại để hoàn tất công việc mình đang làm trên **Sprint Backlog** (Hình 6). Ai đó sẽ cộng toàn bộ lượng công việc của còn lại cho cả nhóm rồi vẽ lên **Biểu đồ Sprint Burndown** (Hình 7 và Hình 8). Biểu đồ này thể hiện con số ước tính cập nhật hằng ngày về khối lượng công việc còn lại cho đến khi Nhóm Phát triển hoàn tất toàn bộ. Trong trường hợp lý tưởng, nó sẽ là một đồ thị *đi xuống* theo một quỹ đạo đạt tới “công việc còn lại bằng 0” vào ngày cuối cùng của Sprint. Chính vì điều này nên nó được gọi là biểu đồ *burndown*. Đôi khi nó nhìn có vẻ ổn, nhưng thường thì không được như vậy; đây chính là thực tế của việc phát triển sản phẩm. Điều quan trọng nhất ở đây là nó cho Nhóm biết tiến độ của mình hướng đến mục tiêu chung, không phải bằng tổng số giờ *đã bỏ ra* trong quá khứ (một thông số không có ý nghĩa xét về mặt *tiến độ*) mà bằng tổng số giờ còn lại *cần phải làm trong tương lai* - đây cũng chính là khoảng cách mà Nhóm Phát triển cần phải vượt qua để đi đến đích. Nếu đường đồ thị không đi xuống hướng đến điểm hoàn thành ở gần cuối Sprint thì Nhóm Phát triển cần phải điều chỉnh, chẳng hạn như thu nhỏ phạm vi công việc hoặc tìm cách để làm việc hiệu quả hơn trong khi vẫn giữ nhịp độ ổn định.

Mặc dù biểu đồ Sprint Burndown có thể được tạo và hiển thị trên một bảng tính điện tử, có nhiều Nhóm cảm thấy hiệu quả hơn khi vẽ trên một tờ giấy dán trên tường ở nơi làm việc rồi cập nhật bằng bút viết; giải pháp “công nghệ-thấp/tiếp xúc-cao” (low-tech/high-touch) này khá nhanh, đơn giản và thường trực quan hơn so với một biểu đồ trên máy tính.

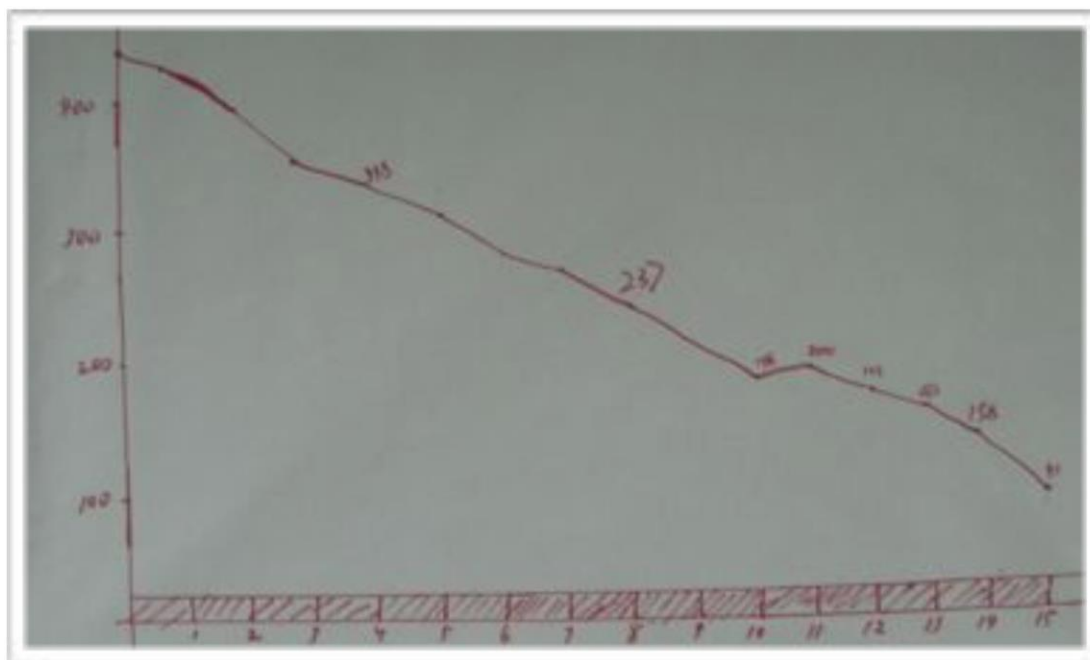
Ước tính Khối lượng công việc Còn lại vào cuối Ngày...									
Hạng mục Product Backlog	Công việc của Sprint	Người thực hiện	Ước tính khối lượng công việc ban đầu	1	2	3	4	5	6
Là khách hàng, tôi	Chỉnh sửa cơ sở dữ liệu	Sanja y	5	4	3	0	0	0	

muốn thêm một cuốn sách vào trong giỏ hàng	Tạo trang web (UI)	Jing	3	3	3	2		0	
	Tạo trang web (nghiệp vụ Javascript)	Tracy & Sam	2	2	2	2	1	0	
	Viết kiểm thử chấp nhận tự động	Sarah	5	5	5	5	5	0	
	Cập nhật trang trợ giúp khách hàng	Sanja y & Jing	3	3	3	3	3	0	
	...								
Cải thiện hiệu suất xử lý giao dịch	Hợp nhất mã DCP và hoàn thành kiểm thử ở mức tăng		5	5	5	5	5	5	
	Hoàn thành máy đặt hàng cho pRank		3	3	8	8	8	8	
	Thay đổi DCP và bộ đọc để sử dụng API http của pRank		5	5	5	5	5	5	

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



Hình 7. Biểu đồ Sprint Burndown



Hình 8. Quản lý Trực quan: Biểu đồ Sprint Burndown vẽ bằng tay

Làm mịn Product Backlog

Tóm lược: Chia nhỏ các hạng mục lớn, phân tích các hạng mục, tái-ước lượng, tái-đánh giá độ ưu tiên của các hạng mục dành cho các Sprint sắp tới.

Thành phần tham dự: Nhóm Phát triển; Product Owner sẽ tham dự toàn bộ hoạt động này nếu anh ta là một chuyên gia có thể trợ giúp Nhóm Phát triển trong việc làm mịn chi tiết, nếu không thì chỉ cần tham dự một lúc để định hướng và giúp tái-đánh giá độ ưu tiên; những người nào am hiểu các yêu cầu và có thể trợ giúp Nhóm Phát triển; ScrumMaster sẽ tham dự ở những phiên đầu tiên để huấn luyện cho nhóm cách làm việc hiệu quả, còn lại thì không cần thiết phải tham dự.

Thời gian: Thông thường không kéo dài hơn 10% thời gian của Nhóm Phát triển dành cho Sprint, đôi khi cần nhiều hơn đối với các hạng mục “khủng”. Ví dụ, đối với một Sprint hai tuần thì có thể dành một ngày để làm mịn.

Một khuyến nghị giá trị nhưng ít được biết đến trong Scrum đó là nên dành một phần thời gian của các Sprint cho việc làm mịn (hay “chải chuốt”) Product Backlog nhằm phục vụ cho các Sprint sắp tới. Việc này bao gồm phân tích chi tiết các yêu cầu, phân tách các hạng mục lớn thành các hạng mục nhỏ hơn, ước tính các hạng mục mới, và ước tính lại các hạng mục đã có. Scrum không đề cập đến cách để thực hiện việc này, nhưng một kỹ thuật thường xuyên được sử dụng đó là tổ chức một phiên làm việc tập trung ở giai đoạn giữa hoặc cuối Sprint, nó sẽ giúp cho Nhóm Phát triển và Product Owner cùng các bên liên quan khác có thể tham dự mà không bị gián đoạn công việc.

Hoạt động làm mịn này *không* phải là dành cho các hạng mục đã được lựa chọn cho Sprint hiện tại; mà nó được dành cho các hạng mục dùng trong tương lai, thường thì trong một hay hai Sprint tiếp theo. Khi áp dụng kỹ thuật này, buổi Lập kế hoạch Sprint sẽ trở nên tương đối đơn giản bởi vì Product Owner và Nhóm Phát triển bắt đầu lập kế hoạch với tập các hạng mục rõ ràng, đã được phân tích kỹ và ước tính cẩn thận. Một dấu

hiệu cho thấy phiên làm mịn này không được thực hiện (hoặc không được thực hiện tốt) là buổi Lập kế hoạch Sprint có tương đối nhiều câu hỏi, khám phá, hoặc nhầm lẫn và có cảm giác chưa đầy đủ; do đó buổi lập kế hoạch công việc thường lấn sang thời gian của Sprint, đây là một hệ quả không mong muốn.

Sơ kết Sprint

Tóm lược: Thanh tra và thích nghi liên quan đến phần tăng trưởng của chức năng sản phẩm.

Thành phần tham dự: Nhóm Phát triển, Product Owner, ScrumMaster. Các bên liên quan khác được Product Owner mời nếu phù hợp.

Thời gian: Đóng khung trong vòng một giờ tương ứng với một tuần làm việc của Sprint.

Sau khi Sprint kết thúc thì diễn ra buổi **Sơ kết Sprint** (Sprint Review), đây là buổi mà mọi người rà soát lại kết quả của Sprint. Những người có mặt ở sự kiện này là Product Owner, các thành viên Nhóm Phát triển, và ScrumMaster, ngoài ra có thêm khách hàng, người dùng, các bên liên quan, các chuyên gia, các lãnh đạo, và bất cứ ai quan tâm. Đối với một Sprint hai tuần thì thời gian tối đa là hai giờ đồng hồ. Bất cứ ai có mặt đều có thể tự do đưa ra câu hỏi và đóng góp ý kiến.

Buổi Sơ kết Sprint thường bị gọi sai là buổi “demo”, nó không bao hàm được ý nghĩa thực sự của sự kiện này. Một ý tưởng chủ đạo trong Scrum là *thanh tra và thích nghi*. Nó nhằm xem xét và học từ những gì đang xảy ra rồi cải tiến dựa trên phản hồi, chu trình này được lặp đi lặp lại như vậy. Sơ kết Sprint là một hoạt động thanh tra và thích nghi cho *sản phẩm*. Đây là lúc mà Product Owner tìm hiểu để nắm tình hình của sản phẩm và của Nhóm Phát triển (đúng vậy, một buổi sơ kết của Sprint); còn đối với Nhóm thì họ tìm hiểu để nắm được tình hình của Product Owner và của thị trường. Theo đó, một thành tố quan trọng của buổi Sơ kết Sprint là sự *trao đổi* giữa Nhóm Phát triển và Product Owner để tìm hiểu về tình hình, ghi nhận các khuyến nghị, v.v.. Hoạt động Sơ kết Sprint chắc chắn phải bao

gồm việc trực tiếp sử dụng phần mềm mà nhóm đã xây dựng trong suốt Sprint, nhưng nếu tập trung quá nhiều vào phần mềm mà không có trao đổi thì sẽ dẫn đến mất cân bằng.

Thời gian dành cho “trực tiếp sử dụng phần mềm” trong buổi Sơ kết Sprint không phải là một “buổi trình diễn” của Nhóm - không có các thiết bị trình chiếu. Nó có nghĩa là một buổi kiểm tra phần mềm thực tế đang chạy trực tiếp, ví dụ như trong một môi trường phát triển được cách ly. Có một máy tính hoặc nhiều hơn ở nơi diễn ra buổi Sơ kết Sprint để mọi người có thể trực tiếp kiểm tra và sử dụng phần mềm. Sẽ là rất tốt nếu có một phiên để người dùng thực sự và Product Owner chủ động tương tác với phần mềm, thay vì chỉ chờ xem một phiên trình diễn từ Nhóm Phát triển.

Cố gắng chỉ dành không quá 30 phút chuẩn bị cho buổi Sơ kết Sprint, nếu không thì có nghĩa là có điều gì đó không ổn.

Cải tiến Sprint

Tóm lược: Thanh tra và thích nghi liên quan đến quy trình và môi trường.

Thành phần tham dự: Nhóm Phát triển, ScrumMaster, Product Owner (không bắt buộc). Các bên liên quan khác có thể được Nhóm Phát triển mời, còn nếu không thì không được phép tham dự.

Thời gian: Đóng khung trong 45 phút tương ứng với một tuần làm việc của Sprint.

Buổi Sơ kết Sprint là để thực hiện thanh tra và thích nghi liên quan đến *sản phẩm*. Buổi **Cải tiến Sprint** diễn ra ngay sau Sơ kết Sprint là để thực hiện thanh tra và thích nghi liên quan đến *quy trình và môi trường*. Đây là cơ hội để Nhóm Phát triển thảo luận những thứ đang làm tốt và những thứ không làm tốt, từ đó thống nhất các thay đổi cần thực hiện. Đôi khi ScrumMaster có thể đóng vai trò là một người hỗ trợ hiệu quả cho buổi Cải tiến, nhưng tốt hơn là tìm một người khác từ bên ngoài để hỗ trợ hoạt động này; một cách làm tốt hay được dùng là ScrumMaster của Nhóm

Phát triển này hỗ trợ buổi cải tiến cho nhóm khác, điều này cũng giúp tạo ra trao-đổi-chéo giữa các Nhóm Phát triển.

Có nhiều kỹ thuật để tiến hành một buổi Cải tiến Sprint, cuốn sách *Agile Retrospectives* (do Derby và Larsen viết năm 2006) có cung cấp một danh sách rất hữu ích của các kỹ thuật này.

Nhiều nhóm cố giữ cho buổi cải tiến chỉ tập trung vào các *vấn đề*, việc này là rất không tốt. Nó có thể dẫn đến việc mọi người nghĩ về buổi cải tiến như một sự kiện chán nản và tiêu cực. Thay vào đó, hãy đảm bảo rằng tất cả các buổi Cải tiến cũng có tập trung vào những điểm mạnh và tích cực; có một vài cuốn sách về *khai thác điểm mạnh* (*Appreciative Inquiry*) cung cấp các mẹo chi tiết hơn về việc này.

Nếu các buổi cải tiến cứ sử dụng mãi một kỹ thuật thì có thể dẫn đến buồn chán, do đó nên sử dụng một vài kỹ thuật để thay đổi theo thời gian.

Bắt đầu Sprint tiếp theo

Ngay sau buổi Sơ kết Sprint, Product Owner có thể cập nhật Product Backlog với những thông tin mới - thêm các hạng mục mới, loại bỏ những hạng mục không còn phù hợp, hoặc chỉnh sửa những hạng mục đang có. Product Owner chịu trách nhiệm đảm bảo rằng mọi thay đổi này được thể hiện ở trong Product Backlog. Xem Hình 9 để tham khảo một ví dụ về việc cập nhật Product Backlog.

Ước lượng công việc của Sprint...									
Độ ưu tiên	Hạng mục	Chi tiết (Wiki URL)	Kích thước Ước tính Ban đầu	1	2	3	4	5	6
1	Là một khách hàng, tôi muốn đưa một cuốn sách vào giỏ hàng (xem phác thảo giao diện trên trang wiki)	...	5	0	0	0			

2	Là một khách hàng, tôi muốn xóa một cuốn sách khỏi giỏ hàng	...	2	0	0	0			
3	Cải thiện hiệu năng xử lý giao dịch (xem các thông số hiệu năng mong muốn trong wiki)	...	13	13	0	0			
4	Nghiên cứu giải pháp để tăng tốc việc xác nhận thẻ tín dụng (xem các thông số hiệu năng mong muốn trong wiki)	...	20	20	20	0			
5	Cập nhật tất cả các máy dịch vụ lên Apache 2.2.3	...	13	13	13	13			
6	Chẩn đoán và sửa lỗi của đoạn mã xử lý hóa đơn (bugzilla ID 14823)	...	3	3	3	3			
7	Là một người mua hàng, tôi muốn tạo và lưu lại một danh sách yêu thích	...	40	40	40	40			
8	Là một người mua hàng, tôi muốn thêm và xóa các hạng mục trong danh sách yêu thích của mình	...	20	20	20	20			
...					
			537	58	57	5			
				0	0	0			
						0			

Hình 9. Cập nhật Product Backlog

Không có thời gian nghỉ giữa các Sprint - các Nhóm Phát triển thường chuyển từ buổi Cải tiến Sprint chiều hôm trước sang buổi Lập kế hoạch Sprint ngay sáng hôm sau (hoặc sau cuối tuần).

Một trong những nguyên lý của phát triển linh hoạt đó là “nhịp độ ổn định”, và để đảm bảo được các chu trình này tiếp tục mãi thì chỉ có cách là Nhóm dành thời gian làm việc ở mức độ hợp lý. Năng suất sẽ tăng theo thời gian bằng việc cải tiến các kỹ thuật làm việc của Nhóm Phát triển, và loại bỏ

các trở ngại ảnh hưởng đến năng suất, chứ không phải bằng cách làm việc quá sức hay thỏa hiệp về chất lượng.

Các Sprint sẽ được tiếp tục mãi cho đến khi Product Owner quyết định rằng sản phẩm đã sẵn sàng để phát hành. Tầm nhìn hoàn hảo của Scrum là sản phẩm có thể được chuyển giao vào mỗi cuối Sprint, điều này có nghĩa là không cần làm những công việc để tổng kết, chẳng hạn như là kiểm thử hoặc viết tài liệu. Nó hàm ý là tất cả *mọi thứ* đã được *hoàn thiện* toàn bộ trong mỗi Sprint; thực tế là bạn có thể chuyển giao hoặc triển khai sản phẩm ngay sau buổi Sơ kết Sprint. Tuy nhiên, nhiều tổ chức đang sở hữu những kỹ thuật phát triển, công cụ và cơ sở hạ tầng rất kém, dẫn đến việc không đạt được tầm nhìn hoàn hảo này và cần đến một “Sprint Phát hành” để giải quyết các công việc tồn đọng. Nếu phải nhờ đến một “Sprint Phát hành” thì cần xem đó là một thảm họa và nhiệm vụ của tổ chức là phải cải thiện các kỹ thuật để điều này không còn xảy ra nữa.

Quản lý Phát hành

Một câu hỏi thỉnh thoảng được đưa ra là làm thế nào để lập kế hoạch phát hành dài hạn khi sử dụng mô hình phát triển lặp. Có hai trường hợp cần xem xét: (1) một sản phẩm mới với bản phát hành đầu tiên, và (2) một sản phẩm đã tồn tại với bản phát hành mới.

Trong trường hợp của một sản phẩm mới, hoặc là *một sản phẩm đã tồn tại nhưng vừa mới áp dụng Scrum*, cần phải tiến hành làm mịn Product Backlog ban đầu trước khi khởi động Sprint đầu tiên, đây chính là lúc để Product Owner và Nhóm Phát triển định hình một Product Backlog phù hợp. Việc này có thể mất vài ngày hoặc một tuần, nó bao gồm một phiên làm việc (đôi khi được gọi là buổi Xây dựng Product Backlog Ban đầu hoặc Lập kế hoạch Phát hành), một số phân tích chi tiết các yêu cầu, và ước tính tất cả các hạng mục đã được xác định cho lần phát hành đầu tiên.

Một điều ngạc nhiên là trong Scrum, đối với một sản phẩm đã tồn tại và đã có Product Backlog thì không cần tới bất cứ một sự kiện lớn và đặc biệt nào để lập kế hoạch phát hành cho lần phát hành kế tiếp. Tại sao? Tại vì

Product Owner và Nhóm Phát triển đã triển khai làm mịn Product Backlog trong từng Sprint (năm hoặc mười phần trăm thời gian của mỗi Sprint), liên tục chuẩn bị cho tương lai. Hình thức *phát triển sản phẩm liên tục* này tránh được sự cần thiết của các giai đoạn rời rạc là chuẩn bị-thực thi-kết thúc như được nhìn thấy trong vòng đời của phương pháp phát triển tuần tự truyền thống.

Trong phiên làm mịn Product Backlog ban đầu và trong các buổi làm mịn Product Backlog ở mỗi Sprint, Nhóm Phát triển và Product Owner sẽ tiến hành lập kế hoạch phát hành, làm mịn các ước tính cũng như độ ưu tiên và các nội dung khác khi mà họ đã học được những thứ mới.

Một số bản phát hành được định hướng theo ngày; ví dụ: “Chúng ta sẽ phát hành phiên bản 2.0 của dự án ở một hội chợ thương mại vào ngày 10 tháng 11”. Trong tình huống này, Nhóm Phát triển sẽ cố gắng hoàn thành càng nhiều Sprint (và xây dựng càng nhiều tính năng) càng tốt trong khoảng thời gian mình có. Một vài sản phẩm khác thì lại yêu cầu một số tính năng nhất định phải được xây dựng xong trước khi được coi là hoàn thành và sản phẩm sẽ không được tung ra cho đến khi các yêu cầu này được thỏa mãn, cho dù có phải chờ đợi lâu. Bởi vì Scrum nhấn mạnh vào việc làm ra mã có khả năng chuyển giao được ở mỗi Sprint, Product Owner có thể lựa chọn đưa ra các bản phát hành tạm thời để giúp khách hàng sớm hưởng lợi từ các công việc đã hoàn tất.

Bởi vì khách hàng không có khả năng biết trước mọi thứ, nên hướng sự tập trung vào việc tạo và tinh chỉnh một kế hoạch cho phép một hướng phát hành rộng hơn, và nói rõ hướng để đưa ra những quyết định lựa chọn trong tương lai (ví dụ, quy mô so với lịch trình). Hãy coi đó như một lộ trình giúp bạn định hướng về phía đến đích cuối cùng; còn con đường chính xác mà bạn sẽ đi cũng như các quyết định mà bạn đưa ra thì sẽ được xác định dọc theo hành trình.

Đích đến quan trọng hơn là hành trình.

Phần lớn các Product Owner lựa chọn một hình thức phát hành cụ thể. Chẳng hạn, họ xác định một ngày phát hành, rồi sau đó trao đổi với Nhóm

Phát triển để ước tính số hạng mục Product Backlog có thể hoàn thành vào ngày đó. Các hạng mục được dự kiến có mặt trong bản phát hành hiện tại đôi khi còn được gọi là *hạng mục phát hành*. Trong các trường hợp đòi hỏi một cam kết “giá cố định / ngày cố định / sản phẩm cố định” - ví dụ, phát triển theo hợp đồng - thì một hoặc một số trong các tham số này phải để một phần dự trữ dành cho sự bất ổn và thay đổi; về mặt này thì Scrum không khác so với các phương pháp khác.

Tập trung vào Ứng dụng hoặc Sản phẩm

Đối với các sản phẩm hoặc ứng dụng - kể cả thương mại lẫn sử dụng nội bộ trong một tổ chức - Scrum chuyển từ mô hình cũ là hướng-dự án sang mô hình *phát triển ứng dụng/sản phẩm liên tục*. Không còn các dự án với các giai đoạn bắt đầu, ở giữa, và kết thúc. Do đó, cũng không còn vai trò quản lý dự án truyền thống. Thay vào đó, cần một Product Owner ổn định và một Nhóm tự quản bền vững, hợp tác với nhau trong một chuỗi “vô hạn” các Sprint với độ dài cố định cho đến khi sản phẩm hoặc ứng dụng được ngừng hỗ trợ. Tất cả những công việc cần thiết liên quan đến quản lý “dự án” đều được xử lý bởi Nhóm Phát triển và Product Owner - người trong vai trò là một khách hàng nội bộ hoặc đến từ bộ phận Quản lý Sản phẩm. Nó không được quản lý bởi một quản lý Công nghệ Thông tin hoặc ai đó đến từ Phòng Quản lý Dự án.

Scrum cũng có thể được dùng cho các *dự án* thực sự thuộc dạng giải pháp chỉ dùng một lần (thay vì làm việc để tạo ra hay cải tiến các ứng dụng lâu dài); trong trường hợp này cũng vậy, Nhóm Phát triển và Product Owner đảm nhiệm công việc quản lý dự án.

Điều gì xảy ra nếu không có đủ lượng công việc mới từ một hoặc nhiều ứng dụng đang có để đảm bảo sự tồn tại bền vững của Nhóm cho từng ứng dụng? Trong trường hợp này, một Nhóm bền vững ổn định có thể lấy các hạng mục từ một ứng dụng cho một Sprint, sau đó lấy các hạng mục từ ứng dụng khác cho Sprint tiếp theo; trong tình huống này các Sprint thường ngắn, chẳng hạn một tuần.

Đôi khi, không cả có đủ công việc mới cho giải pháp ở trên, Nhóm Phát triển có thể lấy các hạng mục từ *một vài* ứng dụng cho một Sprint; tuy nhiên, hãy thận trọng với cách làm này bởi vì nó có thể dẫn đến năng suất thấp do làm việc đa nhiệm giữa nhiều ứng dụng. Một nội dung cơ bản để đảm bảo năng suất trong Scrum là để cho Nhóm chỉ *tập trung* vào một sản phẩm hoặc ứng dụng trong một Sprint.

Các thử thách Thường gặp

Scrum không chỉ là một tập hợp cụ thể các kỹ thuật - mà quan trọng hơn, nó là một khung làm việc thúc đẩy tính minh bạch, và là một cơ chế cho phép “thanh tra và thích nghi”. Scrum hoạt động dựa trên việc làm rõ những bất ổn và trở ngại gây ảnh hưởng đến hiệu quả của Product Owner và Nhóm Phát triển, nhờ vậy mà chúng có thể được xử lý tốt. Ví dụ, Product Owner có thể không hiểu rõ về thị trường, các tính năng, hoặc không biết cách ước tính giá trị thương mại tương đối của các tính năng này. Hoặc Nhóm Phát triển không nắm rõ kỹ năng để ước tính khối lượng công việc hoặc tiến hành phát triển.

Khung làm việc Scrum giúp nhanh chóng phát hiện ra các yếu điểm này. Scrum không giải quyết các vấn đề của việc phát triển; mà giúp chúng hiện lên một cách nhứt nhối, và cung cấp một nền tảng để mọi người có thể tìm ra cách giải quyết các vấn đề trong những chu trình ngắn với các thử nghiệm cải tiến nhỏ.

Giả sử Nhóm Phát triển thất bại trong việc chuyển giao những gì mà họ đã dự báo trong Sprint đầu tiên do yếu kém về kỹ năng phân tích và ước lượng công việc. Họ sẽ cảm thấy đây là một thất bại. Nhưng trong thực tế, kinh nghiệm này là một bước khởi đầu cần thiết để giúp họ trở nên thực tế và thận trọng hơn trong các dự báo của mình. Việc Scrum giúp làm rõ các bất ổn, cho phép Nhóm tìm cách giải quyết chúng là cơ chế căn bản tạo nên những lợi ích to lớn mà Nhóm sử dụng Scrum trải qua

Một lỗi hay mắc phải đó là khi gặp phải khó khăn với một kỹ thuật Scrum nào đó thì lại cố gắng chỉnh sửa Scrum. Ví dụ, Các nhóm gặp khó khăn

trong việc chuyển giao sản phẩm thì có thể quyết định cho phép kéo dài thời gian của Sprint để không bao giờ bị thiếu thời gian nữa - và quá trình này dẫn đến việc họ sẽ không bao giờ phải học cách ước lượng và quản lý thời gian tốt hơn. Bằng cách này, khi không có sự huấn luyện và hỗ trợ của một ScrumMaster có kinh nghiệm, các tổ chức có thể biến đổi Scrum thành một bản sao của những yếu điểm và bất ổn của chính họ, và phá hủy lợi ích thực sự mà Scrum mang lại đó là: Làm rõ những mặt tốt và những mặt xấu, và trao cho tổ chức cơ hội để nâng họ lên một tầm cao mới.

Một lỗi nữa cũng hay mắc phải đó là cho rằng một kỹ thuật nào đó không được khuyến khích hoặc cấm sử dụng chỉ bởi vì Scrum không yêu cầu nó một cách cụ thể. Ví dụ, Scrum không yêu cầu Product Owner phải đưa ra một chiến lược dài hạn cho sản phẩm của mình; nó cũng không yêu cầu các kỹ sư phải tham khảo ý kiến của những kỹ sư giàu kinh nghiệm hơn khi đối mặt với các vấn đề kỹ thuật phức tạp. Scrum dành quyền quyết định việc này cho những cá nhân liên quan; và trong phần lớn các trường hợp thì cả hai kỹ thuật này (cùng với nhiều kỹ thuật khác) đều được thực hiện tốt.

Một điều nữa cần phải thận trọng đó là khi những người quản lý áp đặt Scrum lên các Nhóm của mình; Scrum hướng đến việc trao cho Nhóm không gian và công cụ để họ tự quản lý, việc ép buộc từ bên trên xuống không phải là một công thức cho thành công. Một cách tốt có thể là bắt đầu cho một Nhóm học Scrum từ một đồng nghiệp hoặc một người quản lý, họ sẽ được đào tạo chuyên môn đầy đủ, rồi sau đó đi đến quyết định với tư cách là một Nhóm để tuân thủ chặt chẽ những kỹ thuật đã được dạy trong một khoảng thời gian nhất định; khi kết thúc giai đoạn này, Nhóm sẽ đánh giá trải nghiệm của mình và quyết định liệu có tiếp tục hay không.

Một tin tốt đó là trong khi Sprint đầu tiên thường là rất khó khăn đối với Nhóm, thì các lợi ích của Scrum sẽ được bộc lộ ngay sau đó làm cho nhiều Nhóm Scrum mới phải thốt lên: “Scrum khó thật, nhưng chắc chắn nó tốt hơn rất nhiều so với những thứ mà chúng tôi đã làm trước đó!”.

Phụ lục A: Tham khảo Thêm

Có khá nhiều tài liệu về Scrum. Trong phần tham khảo này, chúng tôi muốn liệt kê một vài tài liệu trực tuyến và một số cuốn sách.

Tài liệu trực tuyến:

- [The Lean Primer - An introduction to Lean Thinking, an important influence to Scrum. http://www.leanprimer.com](http://www.leanprimer.com)
- [The Distributed Scrum Primer - Additional tips for teams who aren't co-located. http://www.goodagile.com/distributedscrumprimer/](http://www.goodagile.com/distributedscrumprimer/)
- [The ScrumMaster Checklist - A list of question that good ScrumMasters use. http://www.scrummasterchecklist.org/](http://www.scrummasterchecklist.org/)
- [Feature Team Primer - Scaling Scrum with Feature Teams, http://www.featureteams.org](http://www.featureteams.org)
- [The Agile Atlas - Core Scrum. ScrumAlliance description of Scrum. http://agileatlas.org/atlas/scrum](http://agileatlas.org/atlas/scrum)
- [Scrum Guide - Scrum.org description of Scrum. http://www.scrum.org/Scrum-Guides](http://www.scrum.org/Scrum-Guides)
- [Agile Contracts Primer - How to make Scrum-friendly contracts. http://www.agilecontracts.org/](http://www.agilecontracts.org/)

Sách:

- Leading Teams - Richard Hackman
- Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum - Craig Larman, Bas Vodde
- Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum - Craig Larman, Bas Vodde
- Agile Project Management with Scrum - Ken Schwaber
- Succeeding with Agile: Software Development using Scrum - Mike Cohn

Phụ lục B: Thuật ngữ

Burndown

Xu hướng của công việc còn lại xuyên suốt thời gian của một Sprint, một Bản phát hành, hoặc một Sản phẩm. Nguồn dữ liệu thô được lấy từ Sprint Backlog và Product Backlog, khối lượng công việc còn lại được thể hiện ở trục dọc (trục tung) còn thời gian (các ngày của Sprint, hoặc các Sprint) được thể hiện ở trục ngang (trục hoành).

Daily Scrum - Scrum Hằng ngày

Một buổi trao đổi ngắn của từng Nhóm Phát triển diễn ra hằng ngày để các thành viên Nhóm thanh tra công việc của mình, đồng bộ công việc và tiến độ, trình bày các trở ngại gặp phải để ScrumMaster có thể loại bỏ chúng. Có thể diễn ra các cuộc thảo luận ngay sau đó để thích nghi phần công việc tiếp theo nhằm tối ưu hóa Sprint.

Development Team - Nhóm Phát triển

Một trong ba vai trò trong Nhóm Scrum gồm ScrumMaster, Product Owner và Development Team.

Done - Hoàn thành

Khái niệm hoàn thành được thống nhất bởi tất cả các bên và tuân thủ các tiêu chuẩn, quy ước, và chỉ dẫn của tổ chức. Khi một thứ được coi là “hoàn thành” trong buổi Sơ kết Sprint thì nó phải tuân thủ được định nghĩa đã thống nhất này.

Estimated Work Remaining (Sprint Backlog items) - Công việc Còn lại Được ước lượng (các hạng mục Sprint Backlog)

Số giờ còn lại mà một thành viên Nhóm Phát triển ước tính dành để làm việc trên một hạng mục nào đó. Ước tính này được cập nhật vào mỗi cuối ngày khi hạng mục đó đã được triển khai. Con số này là tổng lượng công việc còn lại được ước tính chứ không phụ thuộc vào số người sẽ tham gia làm việc.

Increment - Phần tăng trưởng

Tính năng của sản phẩm được Nhóm Phát triển xây dựng trong từng Sprint có khả năng chuyển giao được hoặc sử dụng bởi các bên liên quan của Product Owner.

Increment of Potentially Shippable Product Functionality - Phần tăng trưởng Tính năng Sản phẩm Có khả năng Chuyển giao được

Một lát cắt hoàn chỉnh của tổng thể sản phẩm hoặc hệ thống mà Product Owner hoặc các bên liên quan có thể sử dụng nếu họ muốn triển khai.

Sprint

Một phân đoạn, hoặc một chu trình lặp với các công việc giống nhau để sản xuất ra phần tăng trưởng của sản phẩm hoặc hệ thống. Sprint không được phép kéo dài hơn một tháng và thường thì dài hơn một tuần. Độ dài của Sprint được cố định trong suốt quá trình phát triển, và tất cả các nhóm cùng tham gia làm việc trên một sản phẩm hoặc hệ thống thì sử dụng chung chu trình có cùng độ dài.

Product Backlog

Một danh sách ưu tiên chứa các yêu cầu với thời gian ước tính cần thiết để phát triển thành tính năng sản phẩm. Các hạng mục ở phía trên của Product Backlog có độ ưu tiên cao hơn thì được ước tính chính xác hơn. Danh sách này tiến hóa và thay đổi khi các điều kiện kinh doanh hoặc công nghệ thay đổi.

Product Backlog Item - Hạng mục Product Backlog

Các yêu cầu chức năng, phi chức năng, và các vấn đề được đánh giá độ ưu tiên theo mức độ quan trọng đối với nghiệp vụ kinh doanh và sự phụ thuộc lẫn nhau, chúng cũng được ước lượng. Độ chính xác của việc ước tính phụ thuộc vào độ ưu tiên và mức chi tiết của hạng mục Product Backlog, các hạng mục có độ ưu tiên cao nhất có thể được lựa chọn cho Sprint tiếp theo sẽ khá chi tiết và chính xác.

Product Owner

Là người chịu trách nhiệm quản lý Product Backlog với mục tiêu tối ưu hóa giá trị của sản phẩm. Product Owner cũng là người đại diện cho quyền lợi của tất cả những ai có liên quan đến dự án và sản phẩm được tạo ra.

Scrum

Không phải là một từ viết tắt mà là những cơ chế trong trò chơi bóng bầu dục nhằm đưa quả bóng đã ra ngoài quay trở lại sân.

ScrumMaster

Là người chịu trách nhiệm về quy trình Scrum, đảm bảo nó được triển khai đúng và tối ưu hóa các lợi ích mà nó mang lại.

Sprint Backlog

Danh sách công việc của Nhóm Phát triển trong một Sprint. Nó thường được phân tách thành một tập các nhiệm vụ chi tiết hơn. Danh sách này tiến hóa trong suốt buổi Lập kế hoạch Sprint và có thể được Nhóm cập nhật trong suốt Sprint thông qua việc loại bỏ một số hạng mục hoặc thêm các công việc mới khi cần thiết. Từng hạng mục công việc trong Sprint Backlog được theo dõi trong suốt Sprint và hiển thị khối lượng công việc ước tính còn lại.

Sprint Backlog Task - Công việc trong Sprint Backlog

Là một trong số các công việc mà Nhóm Phát triển hoặc thành viên của Nhóm xác định là cần phải thực hiện để biến các hạng mục Product Backlog thành chức năng hệ thống.

Sprint Planning - Lập kế hoạch Sprint

Một sự kiện được đóng khung trong bốn giờ đồng hồ (đối với một Sprint hai tuần) để khởi động một Sprint. Sự kiện này được chia làm hai phần, mỗi phần kéo dài hai giờ và cũng được đóng khung như nhau. Trong phần đầu tiên, Product Owner trình bày với Nhóm các hạng mục Product Backlog có độ ưu tiên cao nhất. Nhóm Phát triển và Product Owner hợp tác để giúp Nhóm Phát triển xác định số lượng hạng mục Product Backlog

mà họ có thể chuyển thành tính năng sản phẩm trong Sprint sắp diễn ra. Trong phần thứ hai, Nhóm Phát triển lên kế hoạch để thực hiện nhiệm vụ đã chọn thông qua việc thiết kế và phân tách các công việc nhằm biết được cách để đạt Mục tiêu Sprint.

Sprint Retrospective - Cải tiến Sprint

Một sự kiện được hỗ trợ bởi ScrumMaster để toàn bộ Nhóm Phát triển thảo luận về Sprint vừa kết thúc nhằm tìm ra những thay đổi để có thể làm cho Sprint tiếp theo trở nên thú vị và năng suất hơn.

Sprint Review - Sơ kết Sprint

Một sự kiện được đóng khung trong hai giờ đồng hồ (đối với một Sprint hai tuần) diễn ra ở cuối mỗi Sprint để Nhóm Phát triển phối hợp với Product Owner và các bên liên quan nhằm thanh tra kết quả của Sprint. Nó thường bắt đầu bằng việc rà soát lại những hạng mục Product Backlog đã được hoàn thành, một phiên thảo luận về các cơ hội, hạn chế và rủi ro, và một phiên thảo luận về những thứ tốt nhất mà chúng ta nên làm tiếp theo (dẫn đến thay đổi trên Product Backlog). Chỉ những tính năng của sản phẩm đã được hoàn thành thì mới được trình diễn.

Stakeholder - Bên liên quan

Những người có quyền lợi trong kết quả của dự án, có thể là vì họ đã đầu tư cho nó, hoặc sẽ dùng nó, hoặc nó sẽ ảnh hưởng đến họ.

Team – Nhóm (cách gọi tắt của Development Team trong tài liệu Scrum Primer)

Một đội liên chức năng bao gồm những người chịu trách nhiệm tự quản lý để phát triển một phần tăng trưởng sản phẩm trong mỗi Sprint.

Time box - Khung thời gian

Một khoảng thời gian không được phép kéo dài thêm để thực hiện một sự kiện hoặc hoạt động nào đó. Ví dụ, một buổi Scrum Hằng ngày được đóng khung trong mười lăm phút và bắt buộc kết thúc sau mười lăm phút bất kể

kết quả như thế nào. Đối với các sự kiện, nó có thể ngắn hơn. Còn đối với các Sprint thì chúng phải có độ dài chính xác như thế.