

# Ví Dụ về Danh Sách Kiểm Tra cho các ScrumMaster

Michael James

([mj4scrum@gmail.com](mailto:mj4scrum@gmail.com))

14/09/2007

(Chỉnh sửa 02/02/2016)

Người dịch: Nguyễn Bảo Minh

([baominh.nguyen.295@gmail.com](mailto:baominh.nguyen.295@gmail.com))

## Một người trợ giúp toàn lực?

Có thể chỉ cần một ScrumMaster cũng đủ để điều phối hai hoặc ba nhóm cùng một lúc. Nếu bạn hài lòng với việc hạn chế vai trò của mình trong những việc như tổ chức các cuộc họp, áp đặt các khung thời gian, và giải quyết những khó khăn được báo cáo rõ ràng, thì bạn có thể hoàn thành vai trò này mà không cần phải hoàn toàn tập trung hết khả năng của mình. Nhóm vẫn có thể vượt được hạn mức, những mục tiêu mà công ty đặt ra trước khi có Scrum, và có lẽ cũng không có hậu quả nghiêm trọng gì xảy ra.

Nhưng nếu bạn có thể gợi mở cho nhóm về một thời gian tuyệt vời khi tất cả có thể hoàn thành được những việc mà không ai từng nghĩ là có thể, trong một tổ chức đã qua quá trình chuyển đổi, thì bạn sẽ được coi là một ScrumMaster *tuyệt vời*.

Một ScrumMaster thực sự có thể chỉ điều phối *một* nhóm một lúc.

Chúng tôi khuyến nghị rằng nên có một ScrumMaster tâm huyết cho một nhóm khoảng bảy thành viên vào lúc mới bắt đầu.

Nếu bạn vẫn chưa hiểu hết những việc cần phải làm, hãy cố gắng tìm hiểu nhiều hơn về Product Owner của bạn, nhóm của bạn, các hoạt động kỹ thuật trong nhóm, và toàn thể tổ chức bên ngoài nhóm của bạn. Tuy không có một giải pháp toàn diện nào cho tất cả mọi người, nhưng tôi có thể kể ra ở đây những điều mà tôi thường thấy các ScrumMaster bỏ qua. Hãy đánh dấu mỗi ô bằng  $\checkmark$ ,  $\Delta$ ,  $?$ , hoặc N/A như miêu tả ở trang cuối.

## Phần I – Product Owner của tôi đang thực hiện công việc như thế nào?

Các ScrumMaster cải thiện hiệu quả công việc của Product Owner bằng việc giúp họ tìm cách duy trì Product Backlog và Kế Hoạch Phát Hành. (Cần lưu ý rằng Product Owner là người chịu trách nhiệm cho backlog ưu tiên).

- Product Backlog có được sắp xếp theo hiểu biết cập nhật nhất của họ không?
- Các yêu cầu và mong muốn từ tất cả các bên liên quan có được ghi nhận trong Product Backlog không? Ghi nhớ: backlog là *quan trọng*.
- Khối lượng của Product Backlog có thể quản lý được không? Để duy trì số lượng item trong giới hạn có thể quản lý được, hãy giữ những hạng mục chi tiết ở trên cùng, còn các hạng mục trừu tượng nói chung ở dưới. Sẽ phản tác dụng nếu chúng ta phân tích quá sâu vào những hạng mục ở phía trên của Product Backlog. Những yêu cầu đặt ra sẽ thay đổi trong những cuộc trao đổi liên tục giữa sản phẩm đang phát triển và các bên liên quan hoặc khách hàng.
- Có yêu cầu nào (đặc biệt là những yêu cầu ở phía trên cùng của Product Backlog) có thể được thể hiện tốt hơn bằng những user story độc lập, có thể thương lượng được, có thể đánh giá được, có thể ước lượng được, nhỏ, và có thể kiểm thử được?

<sup>1</sup> <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>

- Bạn đã giảng giải cho Product Owner của bạn về nợ kỹ thuật và làm sao để tránh nó chưa? Một trong những giải pháp có thể là thêm việc viết kiểm thử tự động và tải cấu trúc vào định nghĩa "hoàn thành" cho mỗi backlog item.
- Backlog có phải là một *biểu đồ thông tin*, mà các bên liên quan có thể lập tức nhận thấy được không?
- Nếu bạn đang sử dụng một công cụ tự động trong quản lý backlog, mọi người có biết làm sao để sử dụng nó dễ dàng không? Các công cụ quản lý tự động cũng dẫn tới nguy cơ của việc trở thành sự *tắc nghẽn thông tin* nếu thiếu đi sự truyền tải thông tin chủ động từ Scrum Master.
- Bạn có thể giúp truyền tải thông tin bằng các bản in cho mỗi người không?
- Bạn có thể giúp truyền tải thông tin bằng cách tạo ra các biểu đồ to và rõ ràng không?
- Bạn đã giúp Product Owner sắp xếp các hạng mục backlog vào các thời điểm phát hành thích hợp hoặc trong những nhóm ưu tiên chưa?
- Có phải mỗi người trong đội đều nhận thức được liệu kế hoạch phát hành còn phù hợp với thực tế không? Bạn có thể thử cho mọi người xem Biểu đồ tương quan sản phẩm / phát hành với thời gian thực hiện<sup>2</sup> sau khi các hạng mục đã được xác nhận "hoàn thành" trong mỗi cuộc họp Sprint Review. Biểu đồ thể hiện tỷ lệ các Product Backlog Items đã được hoàn thành và những hạng mục mới được thêm vào để cho phép phát hiện sớm những biến động trong quy mô hoặc kế hoạch thực hiện.
- Product Owner của bạn đã điều chỉnh kế hoạch phát hành sau buổi họp Sprint Review gần nhất chưa? Chỉ có số ít Product Owner đã bàn giao sản phẩm được kiểm thử đầy đủ đúng hạn lại *sắp xếp lại* kế hoạch phát hành vào mỗi Sprint. Điều này có thể sẽ khiến cho một vài sản phẩm cần được phát hành sau vì có những việc quan trọng hơn được phát hiện ra.

## Phần II - Nhóm của bạn đang triển khai công việc như thế nào?

Mặc dù việc cộng tác với các thành viên nhóm trong công việc của họ là một việc đáng được khuyến khích như là một cách nêu gương với cả nhóm, nhưng cũng có khả năng bạn sẽ bị đắm chìm trong những công việc kỹ thuật. Hãy xem xét những nhiệm vụ chính của bạn đối với nhóm:

- Nhóm của bạn hoạt động có *trôi chảy* không? Một vài đặc điểm của tình trạng trôi chảy<sup>3</sup>:
- Mục tiêu rõ ràng (những mong đợi và quy định phải rõ ràng, và mục tiêu có thể đạt được, phù hợp với kỹ năng và khả năng của mỗi người).
  - Tập trung và có trọng điểm, tập trung cao độ vào một lĩnh vực nhất định cần được chú ý.
  - Không còn cảm giác tự ti, đề cao hành động và nhận thức.
  - Phản hồi trực tiếp và ngay lập tức (nhanh chóng nhìn nhận thấy các thành công và thất bại của một chuỗi hoạt động, nhờ thế có thể điều chỉnh hành vi nếu cần thiết).
  - Cân bằng giữa cấp độ khả năng và thử thách (hoạt động đưa ra không quá khó cũng không quá dễ).
  - Mỗi người đều có khả năng tự kiểm soát trong mỗi tình huống hay hoạt động.
  - Mỗi hoạt động đều hiển nhiên đem lại kết quả, vì thế không cần quá nhiều cố gắng trong hành động.
- Các thành viên nhóm có thích nhau không, có thư giãn cùng nhau không, và có vui mừng trước thành công của những thành viên khác không?

<sup>2</sup> Mike Cohn, *Agile Estimation and Planning*. (2005).

<sup>3</sup> Mihaly Csikszentmihalyi, *Flow: The Psychology of Optimal Experience* (1990).

- Các thành viên nhóm có cùng nhau giữ cho mỗi người đều phải đạt được những tiêu chuẩn cao, và thúc đẩy mỗi người đều phát triển?
- Có vấn đề hoặc cơ hội nào mà nhóm đang không thảo luận cùng nhau vì những vấn đề hoặc cơ hội đó có thể gây ra tình trạng quá khó chịu trong nhóm không?<sup>4</sup>
- Bạn đã từng thử nhiều cách thức và địa điểm cho các buổi họp Sprint Retrospective chưa?<sup>5</sup>
- Nhóm có tập trung liên tục vào các mục tiêu Sprint không? Có thể bạn nên thực hiện một bước kiểm tra giữa Sprint để có thể tái rà soát các tiêu chuẩn chấp thuận của các hạng mục trong Product Backlog đã cam kết hoàn thành trong Sprint hiện tại.
- Bảng phân công nhiệm vụ của Sprint có phản ánh đúng những công việc mà nhóm đang thực sự làm không? Cần nhận thức rõ “vấn đề tiềm ẩn” của những công việc không được nêu ra và những nhiệm vụ có thể tốn hơn một ngày mới có thể hoàn thành. Những công việc không liên quan đến những cam kết trong Sprint sẽ là những trở ngại cho việc hoàn thành các cam kết đó.
- Nhóm của bạn có có khoảng 3-9 người với một tổ hợp đầy đủ các kỹ năng để xây dựng một sản phẩm hoàn chỉnh có tiềm năng có thể chuyển giao được không?
- Bảng phân công nhiệm vụ của nhóm bạn có cập nhật không?
- Các thành viên nhóm có biết về các công cụ tự quản lý của nhóm không, các công cụ đó có tiện dụng không?
- Những công cụ này có được những người ngoài nhóm tôn trọng đúng mức không? Sự giám sát quá mức đối với các hoạt động thường nhật của những người bên ngoài nhóm có thể hủy hoại sự minh bạch và cơ chế tự quản lý nội bộ trong nhóm.
- Các thành viên nhóm có tự nguyện nhận nhiệm vụ không?
- Sự cần thiết của việc hoàn trả nợ kỹ thuật, việc dần dần sẽ giúp cho việc lập trình trở nên dễ chịu hơn, đã được ghi nhận rõ ràng trong khái niệm *hoàn thành* chưa?
- Các thành viên nhóm có đang cùng nhau chịu trách nhiệm về mọi khía cạnh của sản phẩm chung (kiểm thử, tài liệu cho người dùng, v.v) mà không quan tâm đến chức danh của mỗi người không?

### Phần III – Các Hoạt Động Kỹ Thuật Diễn Ra Thế Nào?

- Hệ thống của bạn trong quá trình phát triển có nút bấm “Thúc đẩy kiểm thử”, cho phép bất kỳ ai (trong cùng nhóm hoặc khác nhóm) dễ dàng phát hiện ra khi họ vừa tạo ra một lỗi hồi quy (làm hỏng chức năng đã chạy tốt trước đây) không? Điểm đặc thù này có thể đạt được nhờ xUnit framework (JUnit, NUnit, v.v).
- Bạn có một cân bằng phù hợp giữa kiểm thử hệ thống tự động hai đầu (còn được gọi là “kiểm thử chức năng” và kiểm thử đơn vị tự động không?
- Nhóm có viết kiểm thử hệ thống và kiểm thử đơn vị bằng ngôn ngữ mà họ đang dùng trong phát triển hệ thống không? Việc cộng tác sẽ không được nâng cao bởi những ngôn ngữ kịch bản độc quyền hoặc các công cụ ghi chép và phát lại mà chỉ một bộ phận trong nhóm biết cách bảo trì.

<sup>4</sup> Marshall Rosenberg, *Nonviolent Communication: A Language of Life: Life-Changing Tools for Healthy Relationships* (2003). Xem xét thêm việc gọi thêm những người trợ giúp chuyên nghiệp có khả năng biến những cuộc đối thoại không thoải mái trở thành thoải mái.

<sup>5</sup> Derby/Larson *Agile Retrospectives: Making Good Teams Great* (2006).

- Nhóm của bạn có phát hiện ra phạm vi không rõ rệt giữa kiểm thử hệ thống và kiểm thử đơn vị<sup>6</sup> không?
- Máy chủ của tự động tích hợp liên tục<sup>7</sup> có phát ra âm thanh báo động khi ai đó vừa tạo ra một lỗi hồi quy không? Vòng lặp phản hồi này có được giảm xuống theo giờ hoặc phút không? ("Biên dịch (build) hàng ngày chỉ dành cho những người nhút nhát." -- Kent Beck)
- Tất cả kiểm thử có được hợp nhất trên kết quả ở máy chủ tích hợp liên tục không?
- Các thành viên trong nhóm có tìm ra được niềm vui của việc không ngừng thiết kế và tái cấu trúc<sup>8</sup>, thay vì Thiết Kế Trước, Lập Trình Sau hay không? Tái cấu trúc có một khái niệm chặt chẽ: thay đổi cấu trúc bên trong mà không thay đổi hành vi bên ngoài. Tái cấu trúc nên được thực hiện vài lần một giờ, mỗi khi xuất hiện mã nguồn trùng lặp, logic điều kiện phức tạp (dễ dàng nhận thấy khi mã nguồn lù vào quá nhiều hoặc có nhiều phương thức dài), các định danh được đặt tên khó hiểu, quá nhiều kết hợp giữa các đối tượng, v.v. Bạn chỉ có thể tái cấu trúc một cách tự tin với tỷ lệ kiểm thử tự động. Bỏ qua tái cấu trúc sẽ làm cho việc thay đổi sản phẩm khó khăn hơn trong tương lai, đặc biệt là khi bạn khó khăn trong việc tìm được những nhà phát triển giỏi sẵn sàng làm việc với mã nguồn được viết rối rắm.
- Định nghĩa hoàn thành cho từng Product Backlog Item của bạn có bao gồm tỷ lệ kiểm thử tự động toàn bộ (cho tất cả các trường hợp) và tái cấu trúc không? Hiểu biết về Kiểm Thử Dẫn Dắt Phát Triển (TDD) sẽ làm gia tăng khả năng đạt được điều này.
- Các thành viên của nhóm có lập trình cặp trong phần lớn thời gian hay không? Lập trình cặp có thể làm tăng vọt khả năng bảo trì của mã nguồn và giảm tỷ lệ lỗi. Điều đó thử thách giới hạn của mọi người và đôi khi dường như cần nhiều thời gian (nếu chúng ta đo bằng số dòng mã nguồn thay vì chức năng có thể bàn giao). Hãy làm gương bằng cách bắt đầu làm việc mỗi ngày với các thành viên. Một vài thành viên sẽ bắt đầu lựa chọn cách làm việc như thế này.

#### Phần IV – Cơ chế hoạt động của tổ chức?

- Việc trao đổi giữa các nhóm có được thực hiện đầy đủ không? "Scrum của các Scrum" chỉ là một cách để đạt được điều này, và hiếm khi là cách tốt nhất.
- Các nhóm có khả năng độc lập để tạo ra các cơ chế làm việc, thậm chí là mở rộng các ranh giới kiến trúc?<sup>10</sup>
- Các ScrumMaster của bạn có tổ chức các cuộc họp với nhau để lập nên danh sách những khó khăn của công ty không?
- Vào thời điểm thích hợp, các khó khăn của công ty có được trình bày với giám đốc phát triển không? Chi phí liên quan có thể định lượng dưới dạng tiền bạc, thời gian bị lãng phí, chất lượng bị sụt giảm hoặc các cơ hội kinh doanh bị bỏ lỡ không? (Rút kinh nghiệm từ sai lầm của Ken Schwaber: "Một ScrumMaster chết là một ScrumMaster vô dụng."<sup>11</sup>)

<sup>6</sup> <http://blogs.collab.net/agile/junit-is-not-just-for-unit-testing-anymore>

<sup>7</sup> <http://www.martinfowler.com/articles/continuousIntegration.html>

<sup>8</sup> Martin Fowler, *Refactoring: Improving the Design of Existing Code* (1999).

<sup>9</sup> Đọc <http://less.works/less/framework/coordination-and-integration.html> cho các lựa chọn khác.

<sup>10</sup> <http://FeatureTeamPrimer.org/>

<sup>11</sup> Ken Schwaber, *Agile Project Management with Scrum* (2004)

Công ty của bạn có phải là một trong số ít các công ty mà con đường phát triển sự nghiệp tương ứng với những mục tiêu chung của các nhóm không? Trả lời “không” nếu tồn tại động cơ<sup>12</sup> khiến ưu tiên việc làm lập trình hoặc các công việc kiến trúc hơn là kiểm thử, kiểm thử tự động, hoặc viết tài liệu cho người dùng.

Công ty của bạn có được công nhận là một trong những nơi làm việc tuyệt vời nhất hoặc là công ty hàng đầu trong lĩnh vực hoạt động bởi các tạp chí chuyên ngành hoặc các nguồn độc lập khác hay không?

Bạn có đang tạo ra một *tổ chức học tập* hay không?

## Kết Luận

Nếu bạn có thể kiểm tra hầu hết các mục này và vẫn còn chút thời gian, tôi muốn lắng nghe đóng góp từ bạn.

Không có một công thức đóng khuôn nào cho sự thông minh của con người. Bản danh sách này liệt kê những điểm có, hoặc không, có ích với tình huống của bạn.

Một khi bạn bắt đầu nhận ra rằng điều bạn làm có thể tạo ra sự khác biệt, bạn có thể thấy e ngại khi thực hiện nó. Đó là dấu hiệu cho thấy rằng bạn đang đi đúng hướng.

---

<sup>12</sup> Alfie Kohn, *Punished By Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes* (1999)

## Khó Khăn Của Công Ty

Vấn đề bề ngoài:

Nguyên Nhân Cốt Lõi (Dùng năm câu hỏi “Tại sao?”):

Ảnh Hưởng Công Việc:

Ảnh Hưởng Cảm Xúc:

Yêu cầu rõ ràng, thực hiện được:

---

## Khó Khăn Của Công Ty

Vấn đề bề ngoài:

Nguyên Nhân Cốt Lõi (Dùng năm câu hỏi “Tại sao?”):

Ảnh Hưởng Công Việc:

Ảnh Hưởng Cảm Xúc:

Yêu cầu rõ ràng, thực hiện được:

## Khó Khăn Của Công Ty

Vấn đề bề ngoài:

Nguyên Nhân Cốt Lõi (Dùng năm câu hỏi “Tại sao?”):

Ảnh Hưởng Công Việc:

Ảnh Hưởng Cảm Xúc:

Yêu cầu rõ ràng, thực hiện được:

---

## Khó Khăn Của Công Ty

Vấn đề bề ngoài:

Nguyên Nhân Cốt Lõi (Dùng năm câu hỏi “Tại sao?”):

Ảnh Hưởng Công Việc:

Ảnh Hưởng Cảm Xúc:

Yêu cầu rõ ràng, thực hiện được:

## Khó Khăn Của Công Ty

Vấn đề bề ngoài:

Nguyên Nhân Cốt Lõi (Dùng năm câu hỏi “Tại sao?”):

Ảnh Hưởng Công Việc:

Ảnh Hưởng Cảm Xúc:

Yêu cầu rõ ràng, thực hiện được:

---

## Khó Khăn Của Công Ty

Vấn đề bề ngoài:

Nguyên Nhân Cốt Lõi (Dùng năm câu hỏi “Tại sao?”):

Ảnh Hưởng Công Việc:

Ảnh Hưởng Cảm Xúc:

Yêu cầu rõ ràng, thực hiện được:



# HƯỚNG DẪN

Nếu bạn nhận được danh sách kiểm tra này dưới dạng một bài tập huấn luyện và công ty của bạn hiện tại (hoặc gần nhất) đã từng thử qua bất cứ mô hình nào như Scrum, hãy áp dụng điều này cho những gì bạn quan sát được ở đó. Đánh dấu mỗi mục với một trong các ký hiệu sau:

- √ (cho “làm tốt”)
- Δ (cho “có thể cải thiện và tôi biết bắt đầu thế nào”)
- ? (cho “có thể cải thiện, nhưng làm thế nào?”)
- N/A (cho “không áp dụng được” hoặc “sẽ không có lợi”)

Ngược lại, nếu công ty của bạn hiện tại (hoặc gần nhất) chưa từng thử qua bất cứ mô hình nào như Scrum, đánh dấu mỗi mục với một trong các ký hiệu sau:

- √ (cho “làm tốt” hoặc “sẽ dễ dàng làm tốt”)
- Δ (cho “sẽ là một thử thách và tôi biết bắt đầu thế nào”)
- ? (cho “sẽ là một thử thách và tôi không biết bắt đầu thế nào”)
- N/A (cho “không áp dụng được” hoặc “sẽ không có lợi”)

Khi tất cả các mục đã được đánh dấu, hãy trình bày từ 2 đến 6 khó khăn của công ty trong mẫu khai Khó Khăn Của Công Ty được đính kèm, không quan trọng việc chúng được lấy ra từ danh sách kiểm tra này hay không. Hãy chọn những trở ngại mà bạn có ít nhất 1% hy vọng thay đổi.