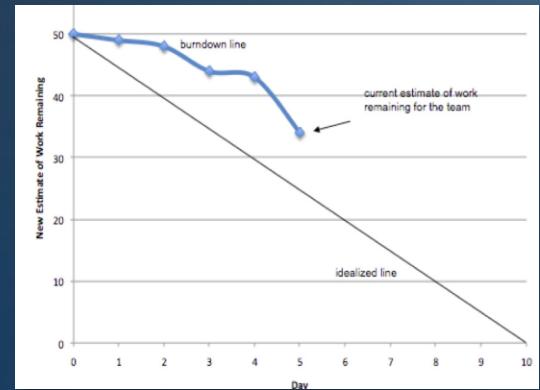




14 FPT-APTECH  
Hà Nội  
8:00 – 17:00  
Tháng Tư

# Căn bản Scrum



# Lời cảm ơn

Tài liệu này được làm ra và phát miễn phí cho người mới làm quen với Scrum. Trong tài liệu có sử dụng nhiều nguồn của Ken Schwaber, Jeff Sutherland, Mike Cohn, Bas Vodde và nhiều người khác nữa. Chúng tôi chân thành biết ơn họ, nhờ họ mà thế giới có được Scrum để làm việc và sống tốt hơn :-)

Nhóm Phát triển Nội dung HanoiScrum:  
**Dương Trọng Tân, Nguyễn Việt Khoa,  
Nguyễn Ngọc Tú & Phạm Anh Đói.**

# Instructors



## Dương Trọng Tấn

- Scrum Master & Agile Coach
- Làm việc tại khối giáo dục FPT
- Thành viên Ban điều hành HanoiScrum
- [tan@hanoiscrum.net](mailto:tan@hanoiscrum.net)



## Nguyễn Việt Khoa

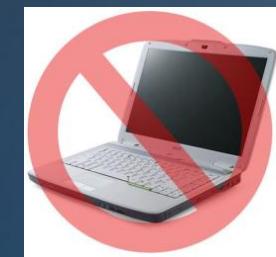
- Scrum Master & Senior Developer
- Làm việc tại FPT-Aptech Computer Education
- Thành viên Ban điều hành HanoiScrum
- [khoa@hanoiscrum.net](mailto:khoa@hanoiscrum.net)

# Khởi động



# Quy tắc khóa học

- Chỉ thảo luận về Scrum tiêu chuẩn, không bàn về ScrumBut
- Câu hỏi hơn là câu trả lời
- Cách học: làm bài tập, trao đổi và thảo luận
- Không máy tính
- Không điện thoại
- Đảm bảo thực hiện đầy đủ danh sách công việc
- Làm bài kiểm tra trên trang web: [scrum.org](http://scrum.org)



# Nội dung

- **Giới thiệu**
- **Lịch sử**
- **Khung làm việc Scrum**
- **Khái niệm “tự quản”**
- **Scrum Master**
- **Nhóm Phát triển**
- **Product Owner**
- **Sprint**
- **Định nghĩa Hoàn thành**
- **Product Backlog**
- **Lập kế hoạch Sprint và Sprint Backlog**
- **Họp Scrum Hằng ngày**
- **Sơ kết Sprint**
- **Cải tiến**
- **Những trở ngại với Scrum**
- **Các vấn đề kỹ thuật**
- **Áp dụng Scrum**
- **Case Study: Scrum tại ScrumLab**

# Thông điệp mở đầu

*"Phát triển phần mềm linh hoạt **không phải là viễn đạn bạc**, nhưng nó thực sự hữu ích. **Về mặt tổ chức**, agile đem lại giá trị và giảm thiểu chi phí; **về mặt kỹ thuật**, nó làm nổi bật tính hoàn hảo và giảm thiểu lỗi; **về mặt cá nhân**, nhiều người nhận thấy đây là cách làm việc mà họ thích thú."*

**James Shore**, tác giả The Art of Agile Development

*"Scrum works for idiots"*

**Ken Schwaber**, "tổ sư" Scrum

*Scrum đơn giản là một cách làm việc, cách sống khác, "tốt" hơn.*

**Tấn DT**

# Agile ở đâu?



Các địa điểm tổ chức  
**AgileTour**  
trên thế giới  
hội thảo mở về agile  
do cộng đồng tổ chức  
với ~7,500 người  
tham dự mỗi năm



|                                   |     |
|-----------------------------------|-----|
| 1. <a href="#">India</a>          | 100 |
| 2. <a href="#">Finland</a>        | 38  |
| 3. <a href="#">Sweden</a>         | 36  |
| 4. <a href="#">United States</a>  | 35  |
| 5. <a href="#">Netherlands</a>    | 34  |
| 6. <a href="#">United Kingdom</a> | 30  |
| 7. <a href="#">Denmark</a>        | 28  |
| 8. <a href="#">Australia</a>      | 28  |
| 9. <a href="#">Canada</a>         | 24  |
| 10. <a href="#">Belgium</a>       | 16  |

# LỊCH SỬ

# Scrum



Ảnh: *Greenandgoldrugby.com*

# Lịch sử Scrum (1)



Toyota Prius (XW10), ảnh: wikipedia

Ý tưởng cơ bản của Scrum lại có xuất xứ từ ngành công nghiệp ô tô.

(Xem: Toyota Product System - Phương thức Toyota, còn gọi là Lean Production-Sản xuất Tinh gọn)

# Lịch sử Scrum (2)

## Takeuchi và Nonaka

“The New New Product Development Game” của Takeuchi và Nonaka. Harvard Business Review, tháng Giêng 1986.

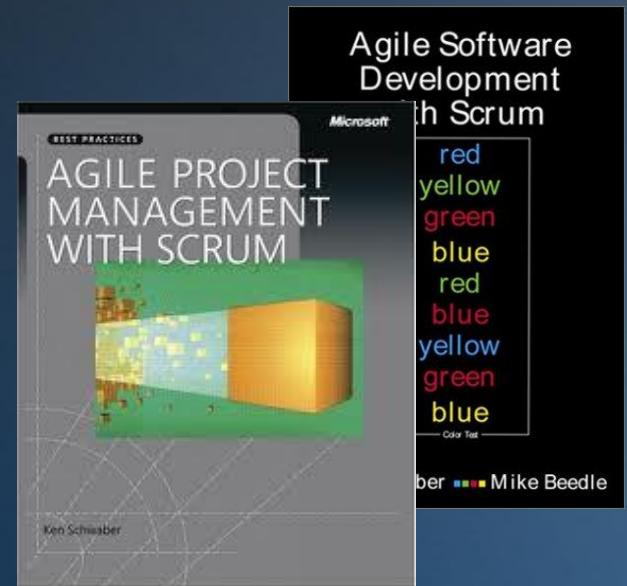
## Ken Schwaber

Cùng Sutherland thuyết trình về Scrum tại OOPSLA 96

## Jeff Sutherland

Lần đầu giới thiệu Scrum tại Tập đoàn Easel vào năm 1993

Cả hai cùng xây dựng định nghĩa Scrum tại Scrum.org

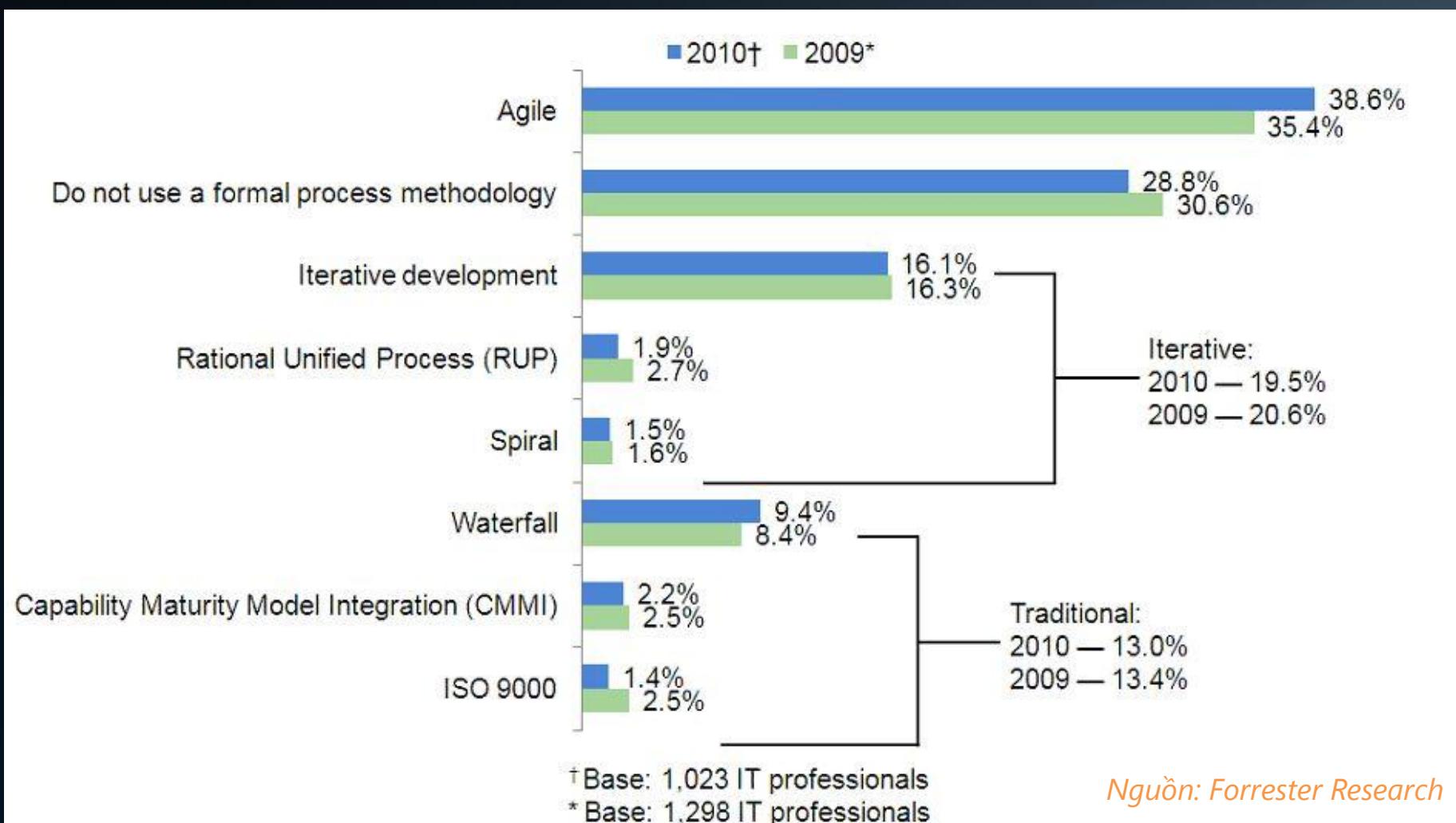


2001 “Tuyên ngôn Phát triển Phần mềm Linh hoạt” (Agile manifesto) và Liên minh Agile (Agile Alliance) ra đời

Năm 2002 Liên minh Scrum (Scrum Alliance) được thành lập



# Dùng phương pháp gì?



# Scrum đã được sử dụng cho

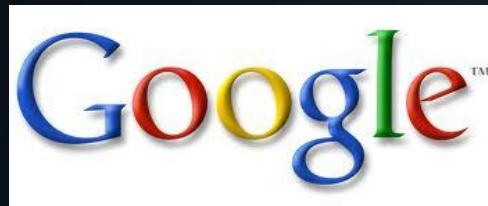
- Phần mềm Thương mại
- Phát triển Ứng dụng Nội bộ
- Phát triển theo đơn đặt hàng
- Các dự án mà giá đã được chốt
- Các ứng dụng Tài chính
- Các ứng dụng tuân thủ chuẩn ISO 9001
- Các hệ thống Nhúng
- Các hệ thống hoạt động 24x7 với yêu cầu 99.999% thời gian hoạt động.
- Joint Strike Fighter
- Phát triển Video game
- Hệ thống thiết yếu của cuộc sống
- Phần mềm Điều khiển-Vệ tinh
- Website
- Phần mềm cho thiết bị cầm tay
- Điện thoại di động
- Các ứng dụng chuyển mạng
- Các ứng dụng ISV
- Một số ứng dụng rất lớn
- Các chiến dịch Marketing
- Các dự án Học tập
- Quản lý các sự kiện Agile Tour

Nguồn: Mountain Goat Software, Scrum Alliance

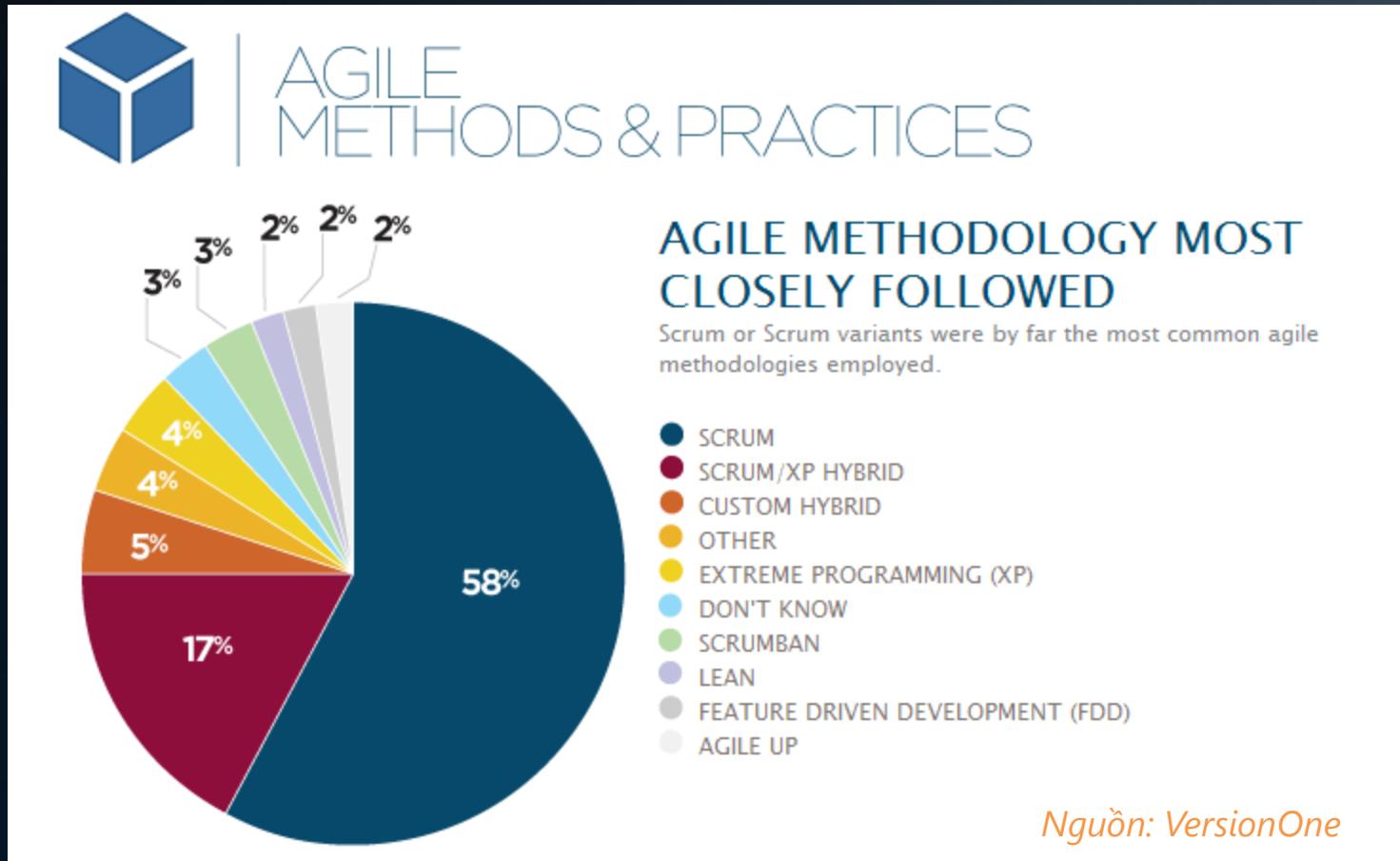
# Scrum là gì?

- **Khung làm việc linh hoạt** (agile framework) để quản lý các dự án phức tạp.
- Mang lại **giá trị cao nhất** trong **thời gian ngắn nhất**
- Các nhóm trong Scrum là **tự quản** (self-managing), tự tổ chức (self-organizing) và **liên chức năng** (cross-functional)
- Hoạt động theo nguyên lí thực nghiệm (empiricism)
- Gọn nhẹ và linh hoạt
- Dễ hiểu nhưng khó tinh thông

# Ai đã sử dụng Scrum?



# Scrum với các phương pháp agile khác



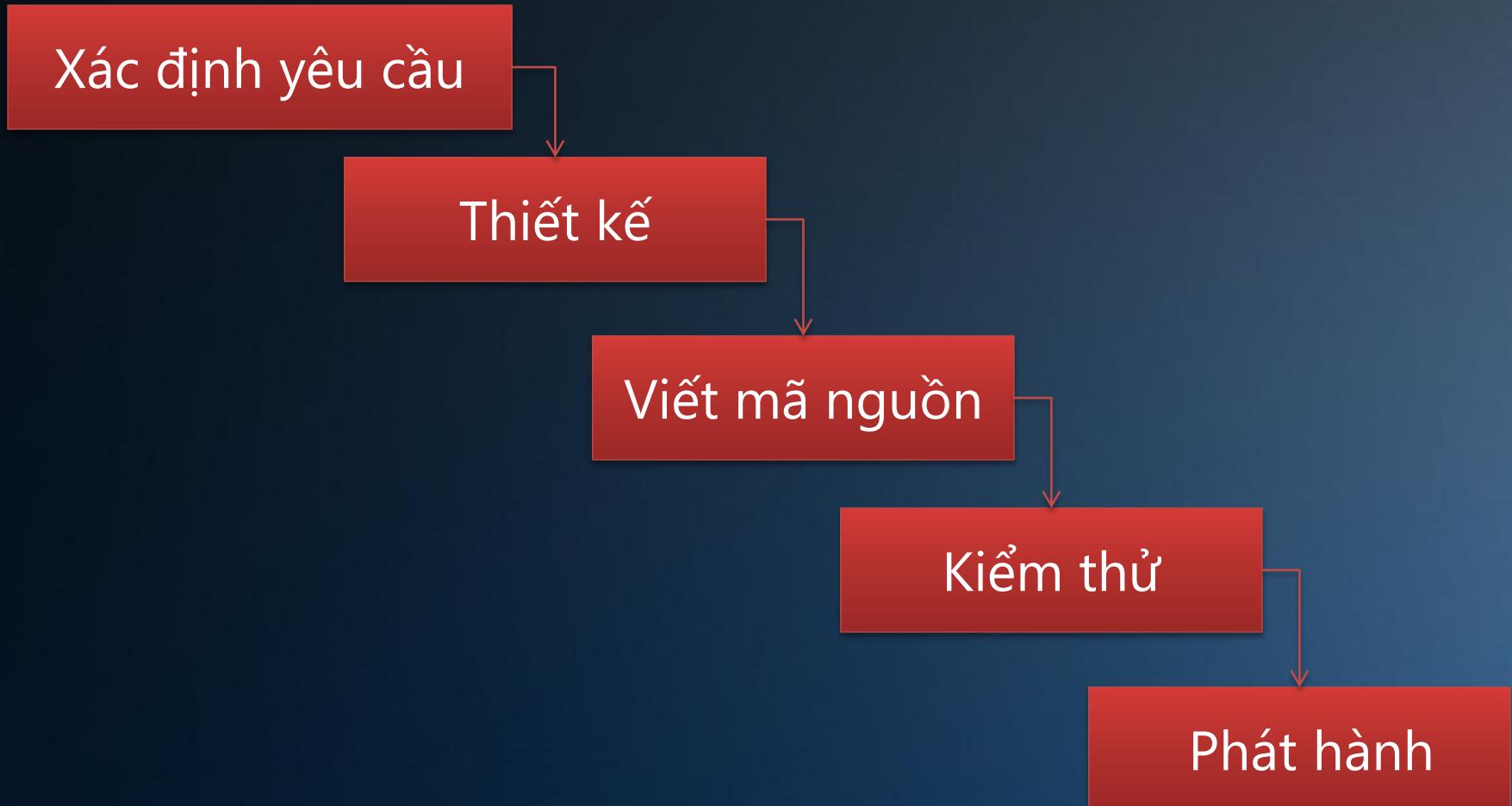
# Tại sao sử dụng Scrum?

- Hoạt động hướng giá trị (Value-Oriented)
  - ROI tốt
- Định hướng khách hàng (Customer-Centric)
  - Tăng độ hài lòng
- Giảm thiểu các “**món nợ kĩ thuật**”
- Chất lượng sản phẩm cao
- Giảm thiểu rủi ro khi ứng dụng gấp vần đề
- Tăng năng suất lao động
- Phát triển bền vững (sustainable development)
  - “NO OT”
  - Vui vẻ hơn, nhân văn hơn

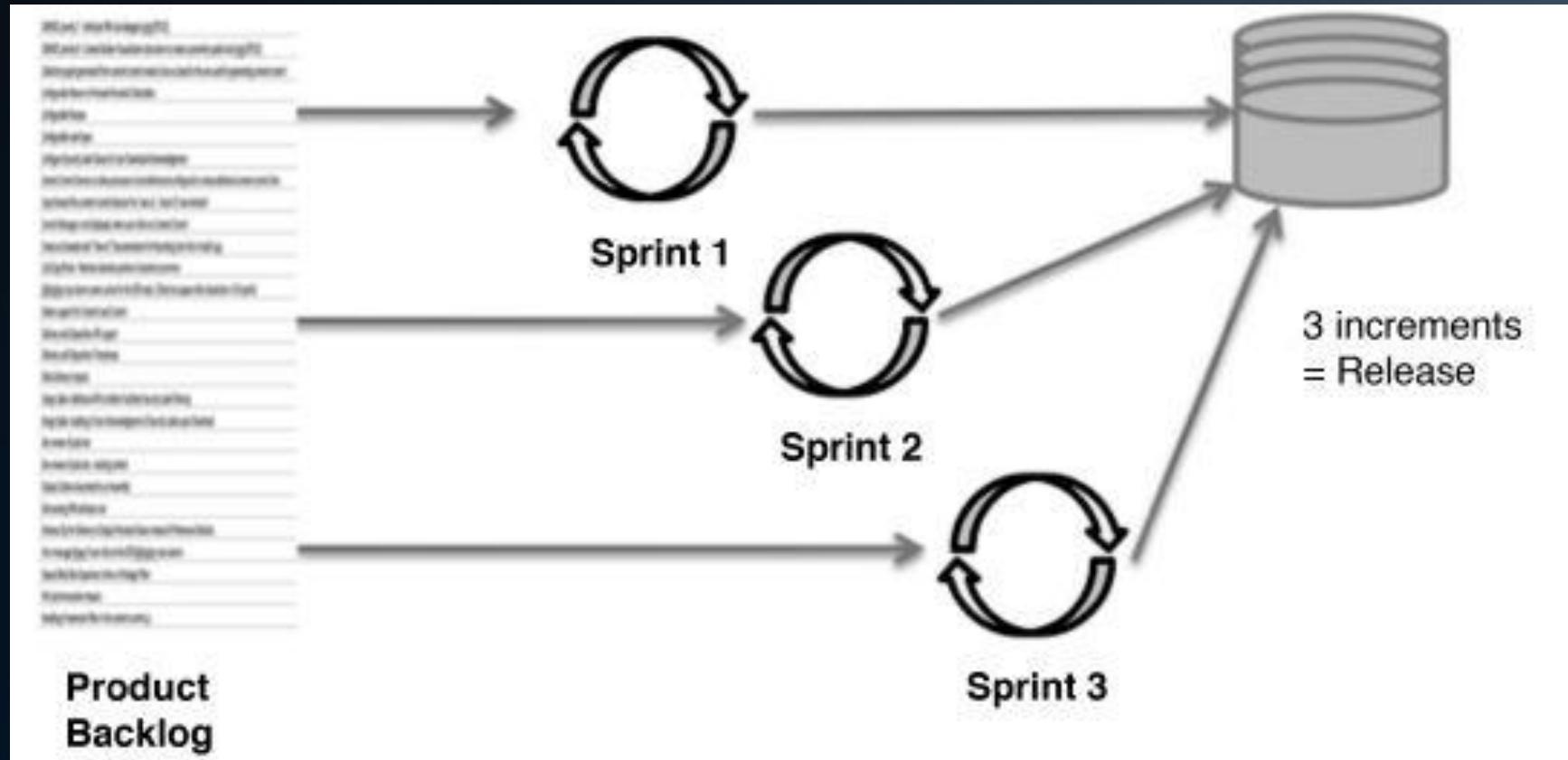
# WATERFALL VÀ AGILE

# Mô hình Truyền thống

## Thác nước|Plan-Driven, Predictive



# Tiếp cận tăng trưởng



Ảnh: Ken Schwaber & Jeff Sutherland

# Tuyên ngôn Phát triển Phần mềm Linh hoạt

Chúng tôi đã phát hiện ra cách phát triển phần mềm tốt hơn bằng cách thực hiện nó và giúp đỡ người khác thực hiện.

Qua công việc này, chúng tôi đã đi đến việc đánh giá cao:

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

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

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

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

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

# 12

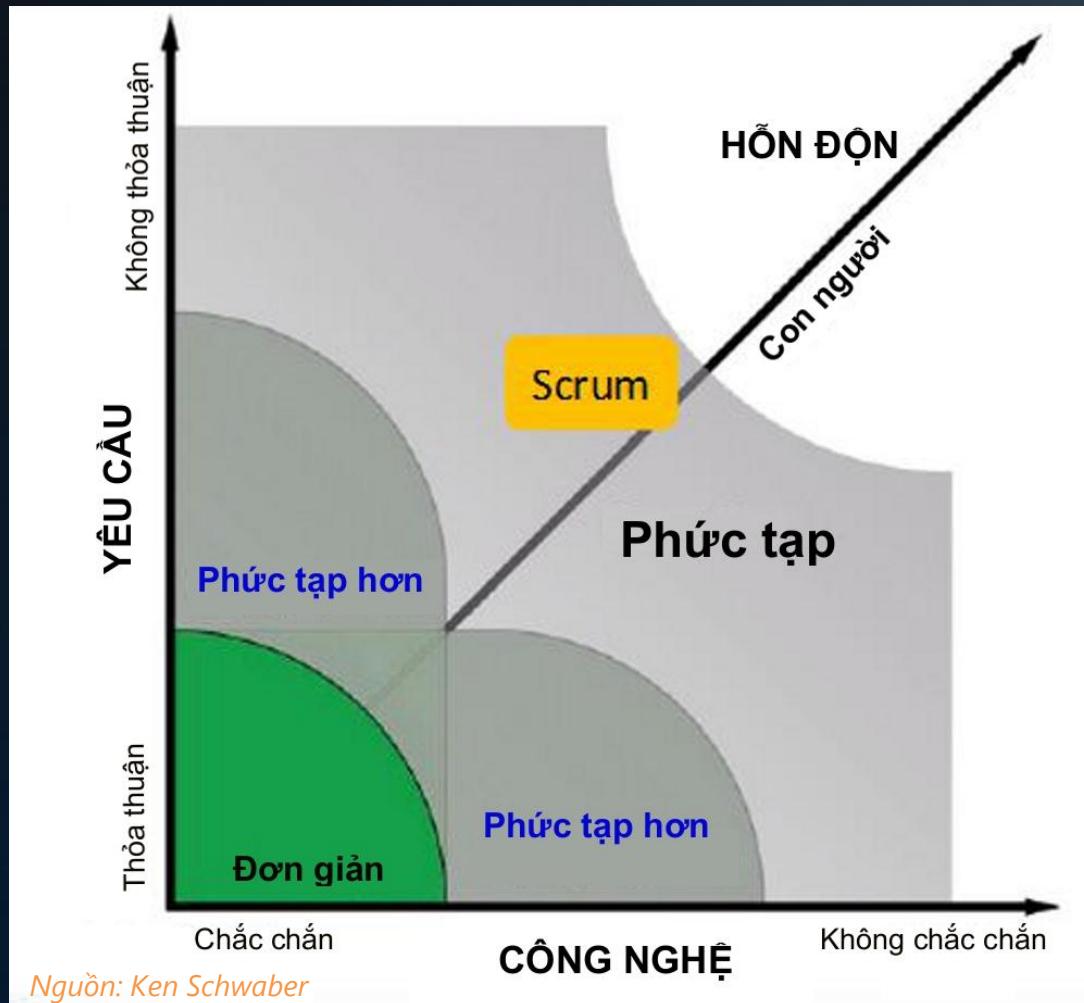
## nguyên tắc

phía sau  
Tuyên ngôn Agile

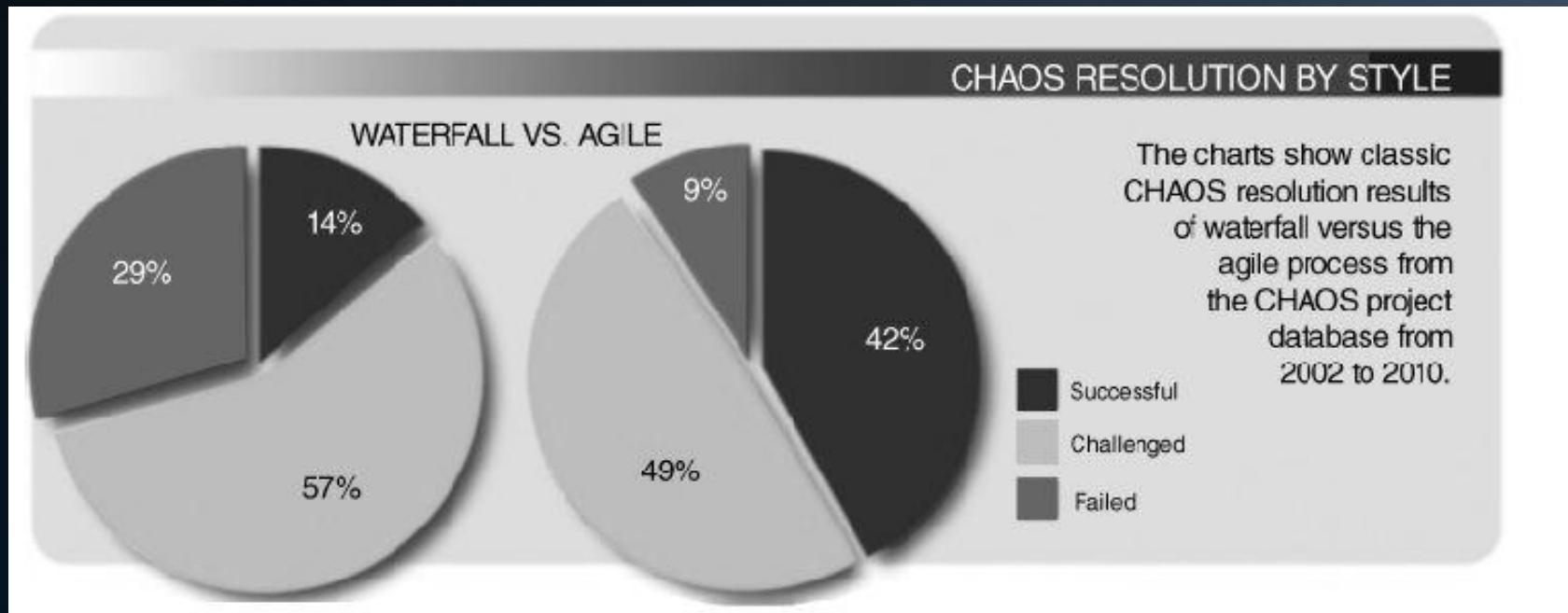
<http://agilemanifesto.org>

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

# Độ phức tạp



# Tiếp cận đúng để thành công



Nguồn: Standish Group's 2001 CHAOS Report

# Video

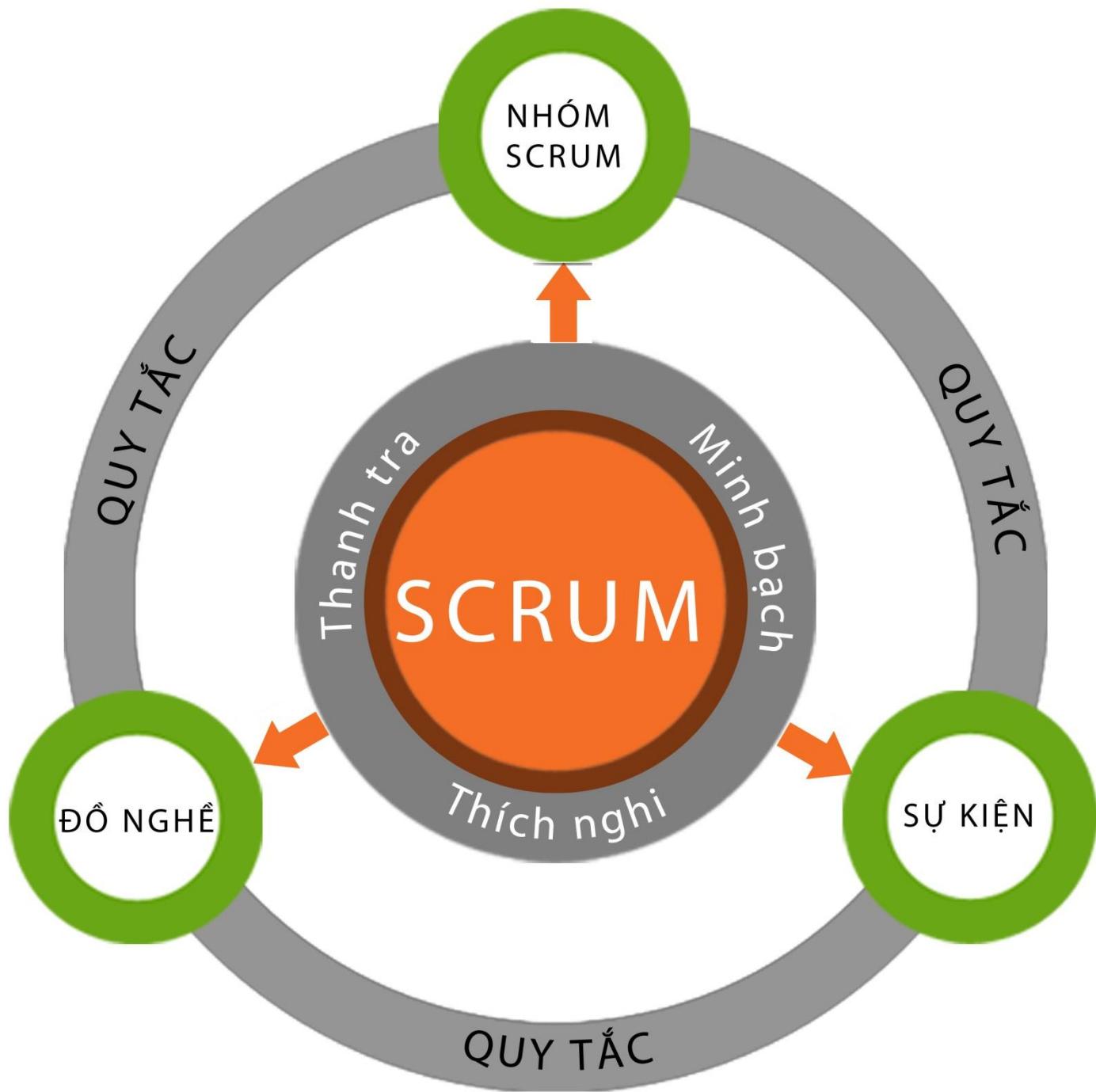
## Agile và Waterfall: Chuyện về hai Nhóm Phát triển



<http://www.youtube.com/watch?v=gDDO3ob-4ZY&feature=related>

# KHUNG LÀM VIỆC SCRUM

## SCRUM FRAMEWORK



# Ba trụ cột của Scrum



Ba chân của Scrum tạo thành cơ chế quản lý tiến trình thực nghiệm

# Khung làm việc Scrum

| Role   | Artifact   | Event  |
|--|--|--|
| <ul style="list-style-type: none"><li>• Product Owner</li><li>• Nhóm Phát triển</li><li>• Scrum Master</li></ul> | <ul style="list-style-type: none"><li>• Gói tăng trưởng</li><li>• Product Backlog</li><li>• Sprint Backlog</li></ul> | <ul style="list-style-type: none"><li>• Sprint</li><li>• Họp kế hoạch Sprint</li><li>• Họp Scrum Hằng ngày</li><li>• Sơ kết Sprint</li><li>• Họp Cải tiến Sprint</li></ul> |

# Sự hỗn độn



Ảnh: VnExpress.net

# Bài tập: Mệnh lệnh và Điều khiển

- **Nhiệm vụ:** “Sản xuất” 100 “bước đi chất lượng”
- **Thời gian:** 2 phút
- **Không gian:** Giới hạn bởi các bàn làm việc
- Quy tắc:
  1. Thành lập nhóm **3 người**
  2. Các vai trò: Quản lý, Công nhân và Quản lý Chất lượng (QA)
  3. Người Quản lý điều khiển Công nhân bằng các mệnh lệnh: TIẾN, DỪNG, TRÁI, PHẢI, NHANH, CHẬM
  4. Quản lý không được chạm vào Công nhân
  5. QA đi theo Công nhân và đảm bảo mọi bước chân được thực hiện đều có “chất lượng cao”

# Bài tập: Tự quản

- Giữ nguyên các nhóm như trên
- Mọi thành viên đều tham ra vào việc sản xuất và có trách nhiệm thực hiện các bước đi có chất lượng
- Mục tiêu: gấp đôi sản lượng trong cùng khoảng thời gian

# Ba vai trò của Nhóm Scrum

Nhóm  
Phát triển

- Tự quản lý
- Tạo ra phần tăng trưởng “hoàn thành”

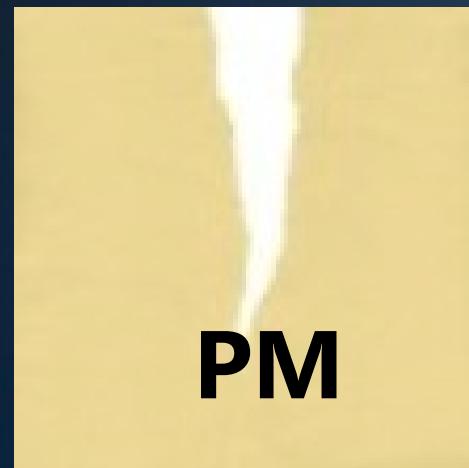
Product  
Owner (PO)

- Tối ưu giá trị của sản phẩm
- Quản lý Product Backlog

Scrum  
Master

- Quản lý tiến trình thực hành Scrum
- Tháo gỡ khó khăn

# Bài tập: Quản lý Dự án – anh là ai?



# SCRUM MASTER

# Trách nhiệm của Scrum Master

- Chịu trách nhiệm về Scrum
- Là người **lãnh đạo**
- Cũng là **đ thầy tớ** của
  - Product Owner
  - Nhóm Phát triển
  - Tổ chức

# Scrum Master phục vụ Product Owner

- Tìm kiếm các kĩ thuật để quản lý hiệu quả Product Backlog;
- Giao tiếp tích cực với Nhóm Phát triển về tầm nhìn, mục đích, và các hạng mục của Product Backlog;
- Dạy cho Nhóm Phát triển biết cách tạo ra các hạng mục Product Backlog thật rõ ràng và đơn giản;
- Hiểu rõ việc lập kế hoạch dài hạn sản phẩm trong một môi trường thực nghiệm;
- Hiểu rõ và thực hành sự linh hoạt (agility); và,
- Thúc đẩy các sự kiện Scrum theo yêu cầu hoặc khi cần thiết.

*Trích "Hướng dẫn Scrum"*

# Scrum Master phục vụ Nhóm Phát triển

- Huấn luyện Nhóm Phát triển cách tự tổ chức và làm việc liên chức năng;
- Dạy và lãnh đạo Nhóm Phát triển cách tạo ra các sản phẩm có giá trị cao;
- Loại bỏ các trở lực trong quá trình tác nghiệp của Nhóm Phát triển;
- Thúc đẩy các sự kiện Scrum theo yêu cầu hoặc khi cần thiết; và,
- Huấn luyện Nhóm Phát triển trong trường hợp tổ chức chưa có hiểu biết về Scrum.

*Trích "Hướng dẫn Scrum"*

# Scrum Master phục vụ Tổ chức

- Lãnh đạo và huấn luyện tổ chức trong việc áp dụng Scrum;
- Lập kế hoạch triển khai Scrum trong phạm vi tổ chức;
- Giúp đỡ nhân viên và các bên hữu quan hiểu và sử dụng được Scrum cũng như quá trình phát triển sản phẩm thực nghiệm (empirical product development);
- Tạo ra sự thay đổi làm tăng năng suất của Nhóm Scrum; và,
- Làm việc với các Scrum Master khác để gia tăng hiệu quả của việc áp dụng Scrum trong tổ chức của mình.

*Trích "Hướng dẫn Scrum"*

# Scrum Master, Là và Không là

## Là

- Lãnh đạo-Đầy tớ
- Cố vấn
- Nhân tố thay đổi
- Người sửa lỗi
- “Chó chăn cừu”

## Không là

- Quản lý Dự án
- Nắm quyền đối với nhóm
- Kiến trúc sư trưởng

# Một ngày của Scrum Master

- Tìm kiếm các cải tiến:
  - Product Owner đang làm việc thế nào?
  - Nhóm Phát triển đang làm việc thế nào?
  - Các kỹ thuật đang được dùng thế nào?
  - Tổ chức đang làm việc ra sao?
  - Ai cần được huấn luyện về cái gì?

*Xem danh mục kiểm tra đầy đủ ở Phụ lục*

# Hỏi để “thanh tra và thích nghi”

- Tôi nhận thấy <tình huống>, chúng ta sẽ làm gì?
- Tôi quan sát thấy <tình huống>, nó có quan trọng không?
- Tôi thấy <cảm giác>, bạn có thấy điều đó?
- Chúng ta sẽ cố tìm lý do của <tình trạng>?
- Bạn nghĩ chúng ta cần làm gì?
- Ai có ý tưởng gì về <tình trạng>?
- Điều này có hiệu quả không?
- Bạn đã quyết định điều gì?
- Bạn nên làm gì?



Câu hỏi Socratic là gì? 43

# Cộng tác nhóm



Quản lý Dự án (PM) điều phối

Scrum Master tạo điều kiện

# Chuyển đổi cách quản lý

## Truyền thống

- Người quản lý ra lệnh và đảm bảo mệnh lệnh được thực thi
- Người quản lý ra quyết định
- Người quản lý giới hạn thông tin và tài nguyên đối với nhân viên



## Scrum

- Mọi người quyết định việc cần làm và cách thức triển khai
- Nhóm ra quyết định
- Minh bạch về thông tin

# PRODUCT OWNER

CHỦ SẢN PHẨM

# Product Owner

- Định nghĩa các hạng mục trong **Product Backlog** (*các tính năng, các bản vá lỗi, v.v.*)
- Quyết định ngày và nội dung của bản phát hành
- **Sắp xếp** các hạng mục trong Product Backlog (PBI) để tối ưu hóa mục tiêu và nhiệm vụ
  - Trách nhiệm để **tối ưu hóa lợi nhuận** (ROI)
- Duy trì sự hiện diện và nội dung của Product Backlog
- Chấp nhận hoặc từ chối kết quả công việc
- **Tham gia** tích cực vào tiến trình phát triển
- Phải có **tâm nhìn** cho sản phẩm

# NHÓM PHÁT TRIỂN

# Nhóm Phát triển

- **Tự tổ chức**
  - Xác định công việc và gán việc
  - Lý tưởng là không có chức danh, nhưng rất ít khi xảy ra
- **Liên chức năng**
  - Không có vai trò chức năng (kiểm thử, lập trình viên, thiết kế, v.v.)
- Trách nhiệm cung cấp phần sản phẩm tăng trưởng có khả năng chuyển giao được
- Không nhiều hơn 9
- Để Nhóm trưởng thành và năng suất
  - Thành viên nên cố định
  - Thành viên nên làm việc toàn thời gian
- Bảo trì Sprint Backlog thường xuyên

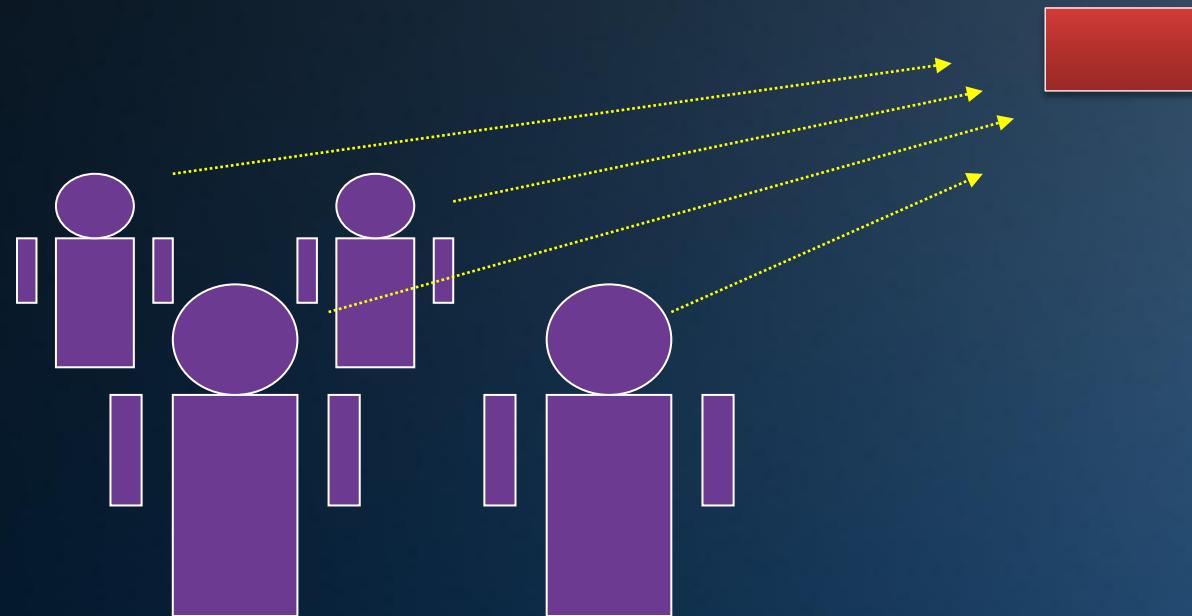


URL: T3DS.com/247423

The3dStudio.com

# Nhóm cộng tác

- Hoạt động hướng vào mục đích chung, không hướng vào công việc của từng cá nhân
- Giới hạn lượng việc-đang-làm



# So sánh cách tổ chức nhóm

| Nhóm tự tổ chức   | Nhóm chức năng  |
|---|---|
| <ul style="list-style-type: none"><li>Hướng khách hàng</li><li>Nhóm đa kỹ năng</li><li>Ít mô tả công việc</li><li>Thông tin được chia sẻ rộng rãi</li><li>Ít cấp quản lý</li><li>Tập trung vào nghiệp vụ sản phẩm</li><li>Chia sẻ mục tiêu</li><li>Có vẻ như hỗn độn</li><li>Nhấn mạnh mục đích cần đạt được</li><li>Cam kết của thành viên cao</li><li>Cải tiến liên tục</li><li>Tự kiểm soát</li><li>Nền tảng là các Giá trị\Nguyên tắc</li></ul> | <ul style="list-style-type: none"><li>Hướng quản lý</li><li>Nhóm của các chuyên gia độc lập</li><li>Nhiều mô tả công việc</li><li>Thông tin bị giới hạn</li><li>Nhiều cấp quản lý</li><li>Tập trung vào chức năng\phòng ban</li><li>Cô lập mục tiêu</li><li>Có vẻ được tổ chức tốt</li><li>Nhấn mạnh giải quyết vấn đề</li><li>Cam kết quản lý cao</li><li>Cải tiến tăng dần</li><li>Quản lý kiểm soát</li><li>Nền tảng là các chính sách\thủ tục</li></ul> |

# Ra quyết định

- Ra quyết định kiểu gì?
  - Đồng thuận
  - Bỏ phiếu
  - Chuyên gia quyết định
- Nếu nhóm không có một thỏa thuận về cách ra quyết định, thì nhóm có thể phải đổi mặt với nhiều vấn đề sau

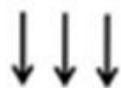
# Video

## Một ngày của Nhóm Scrum



# Quy trình Scrum

Thông tin từ người dùng cuối, khách hàng, Nhóm phát triển và các bên liên quan



Product Owner

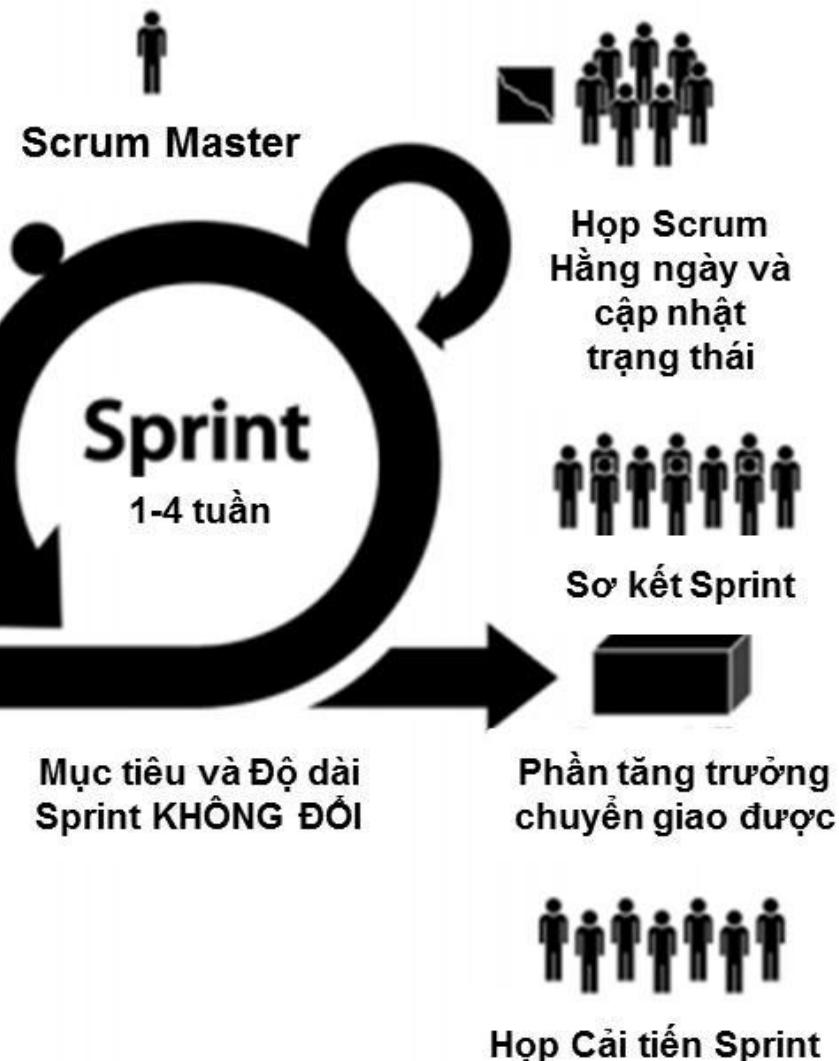
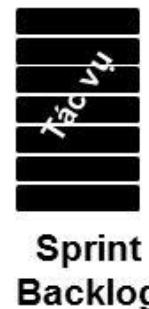


Làm mịn Product Backlog

Nhóm phát triển

Hợp Kế hoạch Sprint

Nhóm phát triển  
chọn lượng  
công việc để làm  
trong Sprint tới



# Sprint

- Phân đoạn ngắn để tạo ra phần chức năng hoàn chỉnh
- Ngắn hơn 30 ngày
- Mỗi Sprint đều có Mục tiêu (Sprint Goal)
- Giữ độ dài Sprint không đổi để tạo **nhịp đập** cho nhóm
- Sản phẩm được thiết kế, lập trình và kiểm thử trong Sprint
- Sprint càng ngắn, chi phí quản lí càng lớn

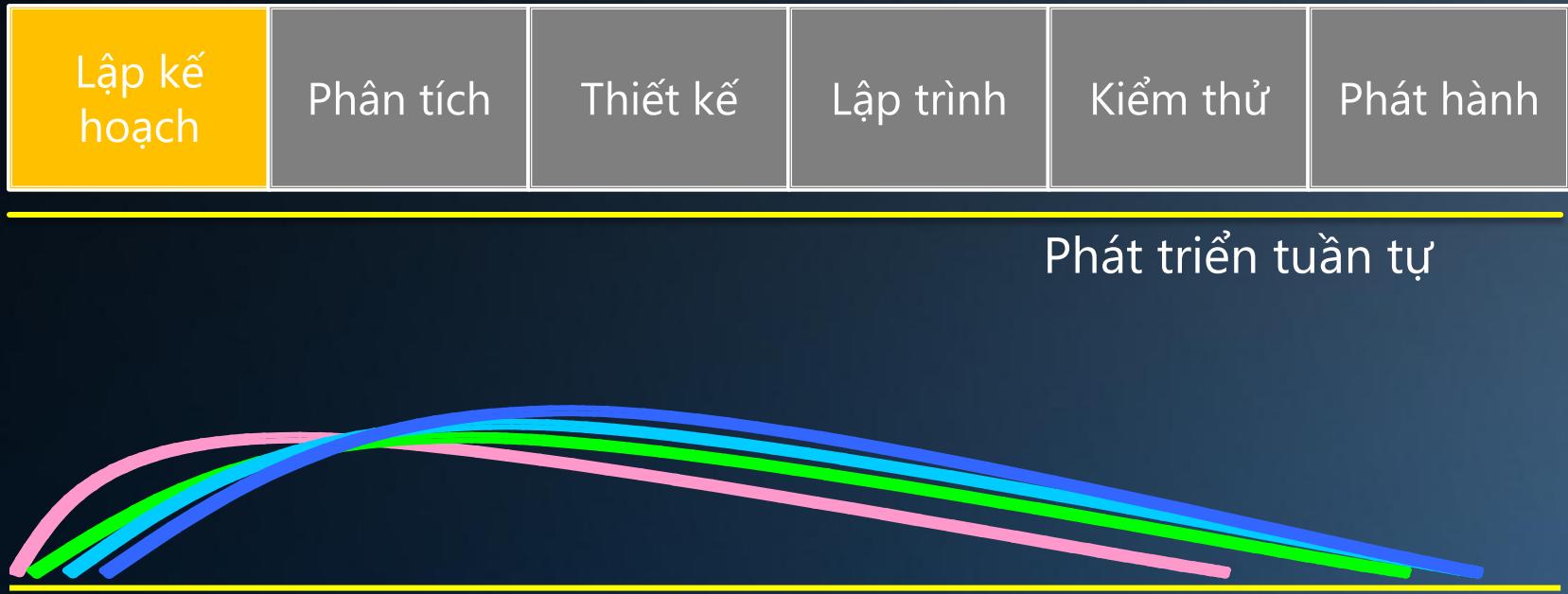
# Rủi ro được giảm thiểu trong Sprint

Cùng công việc, nhưng được tổ chức khác và với ít yêu cầu hơn.

Những lãng phí lớn sẽ bị phát lộ và có thể được xóa bỏ một cách hệ thống.



# Tuần tự & Chồng lắp



Nhóm Scrum làm mỗi thứ một ít ở mọi thời điểm với cơ chế **one-piece flow** (luồng một sản phẩm)

Nguồn: "The New New Product Development Game" của Takeuchi và Nonaka. *Harvard Business Review*, tháng Giêng 1986.

# Không thay đổi trong suốt Sprint



- Cần tạo **không gian tự chủ** cho nhóm
- Tránh quản lí vi mô (micro-manage)
- Scrum Master – “chó chăn cừu”: “giữ sói tránh xa đàn cừu”
- Tránh thay đổi: Mục tiêu Sprint, thành viên Nhóm, chất lượng mục tiêu, phạm vi của tính năng

# ĐỊNH NGHĨA HOÀN THÀNH

DEFINITION OF DONE

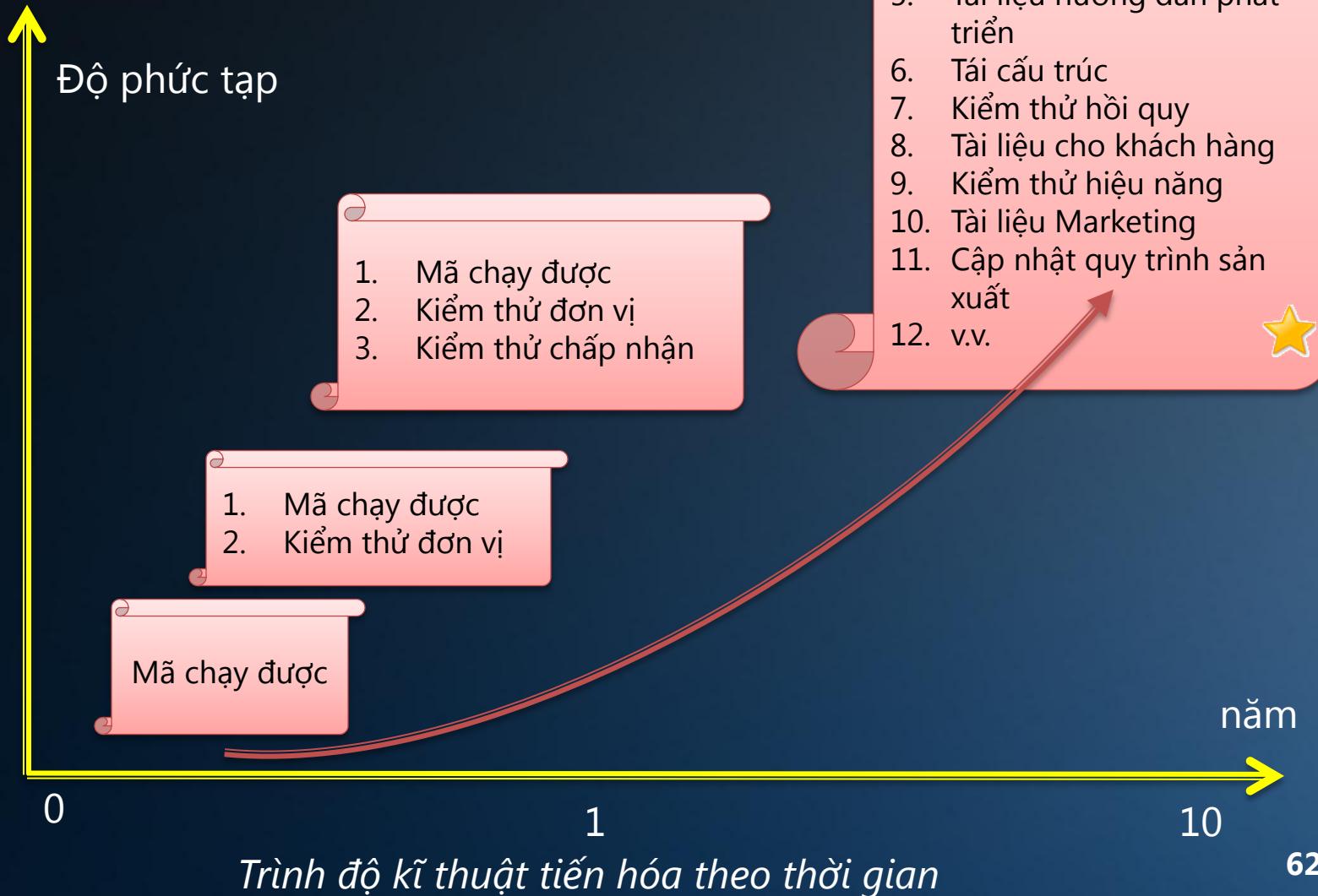
# Sản phẩm hoàn chỉnh có thể chuyển giao được là gì?



# Định nghĩa “Hoàn thành”

- Nhóm Phát triển cùng Product Owner định nghĩa thế nào là “hoàn thành” cho các công việc cần làm
- Công cụ để đảm bảo chất lượng
- Công cụ để tự tổ chức công việc
- “Định nghĩa Hoàn thành” phản ánh **trình độ kỹ thuật** của nhóm

# Quá trình tiến hóa của “Hoàn thành”



Kích não:

Có khó khăn gì trong việc mở rộng “Định nghĩa Hoàn thành”?

# CÁC SỰ KIỆN SCRUM

# Các Sự kiện Scrum

## Họp Kế hoạch Sprint

- Thiết lập Mục tiêu Sprint
- Lựa chọn các việc cho Sprint và làm rõ cách để đạt được chúng.

## Scrum Hàng ngày

- Đồng bộ hóa công việc
- Cập nhật kế hoạch và tiến độ

## Sơ kết Sprint

- Sơ kết về các việc đã hoàn thành trong Sprint
- Kiểm tra có đạt Mục tiêu Sprint không

## Họp Cải tiến Sprint

- Rà soát quy trình
- Tìm kiếm sự cải tiến

## Sprint

- Hành động để đạt được Mục tiêu Sprint
- Design, Code, Review, Test ...

# Họp Kế hoạch Sprint

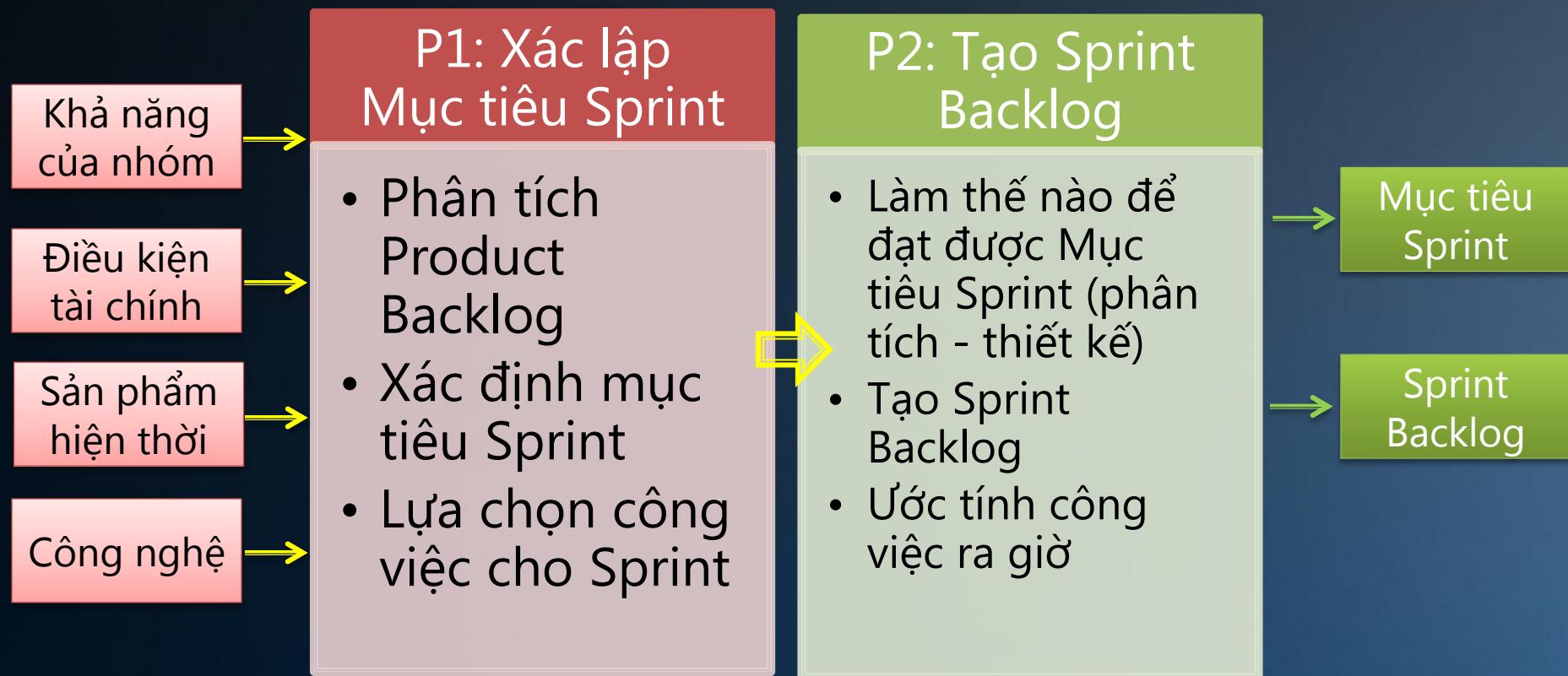
- Nhóm Scrum chọn các hạng mục có độ ưu tiên cao nhất từ Product Backlog để làm trong Sprint dựa theo năng lực (capacity) của nhóm
- Tạo ra Sprint backlog :
  - Công việc cụ thể được xác định và với ước lượng (1-16 giờ) thời gian hoàn tất
- Toàn bộ Nhóm Scrum cùng cộng tác

Là người lên kế hoạch  
nghỉ mát, Tôi muốn xem  
ảnh của khách sạn để từ  
đó có thể quyết định đặt  
khách sạn này.



| Việc cần làm                        |
|-------------------------------------|
| Tạo bản mẫu (8 hours)               |
| Viết các kịch bản kiểm thử(4)       |
| Viết code giao diện (4)             |
| Viết lớp PhotoCollection (6)        |
| Cập nhật các kiểm thử hiệu năng (4) |

# Hợp Kế hoạch Sprint



# Lập kế hoạch tức thời(JIT)

## Lập kế hoạch tiên lượng

Tất cả các hoạt động được hoạch định ngay từ đầu



## Lập kế hoạch thực nghiệm

Lập kế hoạch tức thời và liên tục, hoạch định lại dựa trên cơ sở thanh tra–thích nghi thường xuyên.



# Hợp Scrum Hằng ngày

- Duy trì mỗi ngày **15 phút** để **đồng bộ hóa công việc** và **lập kế hoạch tức thời**
- Một cơ chế “**thanh tra và thích nghi**” của nhóm tự quản.
- **3 câu hỏi:**
  - **Đã làm được gì kể từ lần họp trước?**
  - **Sẽ làm gì từ giờ cho tới lần họp tiếp theo?**
  - **Có khó khăn gì trong công việc?**
- Các thành viên trong nhóm sẽ báo cáo cho nhau, **KHÔNG** báo cáo cho Scrum Master
- Cố định thời gian, địa điểm; nên đứng khi họp.

# Truyện vui: Gà và Lợn



- Người có trách nhiệm mới ra quyết định
- Giữ “gà” tránh xa “lợn”



# Hoạt động: Scrum from hell

*Photo: <http://rosamundwo.com/wp-content/uploads/2010/05/Standup-meeting.jpg>*

# Scrum Hàng ngày có hiệu quả?

Thanh tra để thích nghi:

- Nhóm có tự quản lý?
- Nhóm có chia sẻ công việc?
- Các báo cáo của nhóm có rõ ràng?
- Các nhiệm vụ quá lớn?
- Nó có lâu quá không?
- Có trở ngại nào được tìm ra không?
- Có vướng mắc nào được tháo gỡ không?
- Có hành động cải tiến nào không?



Scrum Master

# Sau buổi Họp Scrum Hằng ngày

- Họp Scrum Hằng ngày không giải quyết các vấn đề,
- Nó là cơ chế “thanh tra-thích nghi”
  - Phải bám đuổi Scrum Hằng ngày để “thích nghi”
  - Các hành động bám đuổi: họp, huấn luyện, thảo luận, xem xét, v.v.
- Scrum Master trợ giúp nhóm tháo gỡ những trở ngại
- Nhóm cập nhật sau Họp Scrum Hằng ngày:
  - Sprint Backlog với các tác vụ và đánh giá mới
  - Danh mục các vấn đề <Impediment Backlog>
  - Biểu đồ Sprint Burndown

# Sơ kết Sprint

- Nhóm Phát triển trình bày những hạng mục đã “**hoàn thành**” của Product Backlog cho Product Owner và các bên liên quan
- Khung thời gian: 4 giờ
- Thành phần: Nhóm Scrum (pig) + các bên liên quan(chicken)
- **Không trình bày những tính năng chưa “hoàn thành”**
- Các phản hồi được đưa ra – Product Backlog có thể được đánh giá lại độ ưu tiên
- Đây không phải buổi DEMO, chuẩn bị **ít hơn 30 phút**
- Product Owner nên sử dụng kĩ thuật kiểm thử chấp nhận để đánh giá các tính năng



# Kiểm thử chấp nhận

## Acceptance Testing

- Người sử dụng (hoặc khách hàng) chấp nhận kết quả làm việc của Nhóm Phát triển
  - Khách hàng là bên viết các bài test này.
- Các test được áp dụng đối với tất cả các **logic nghiệp vụ quan trọng**
- Có thể tự động hóa được thông qua các hệ thống trợ giúp
- Công cụ:
  - Framework for Integrated Test (<http://FIT.c2.com> )
  - FitNesse (<http://www.fitnesse.org/> )
  - Các hệ CI

# Ví dụ về kiểm thử chấp nhận

Là người mua sách, Tôi muốn tìm kiếm sách bằng cách nhập các thông tin như tác giả, tiêu đề, mã ISBN với bất kỳ sự kết hợp nào

ATs

- Sử dụng tên tác giả, tiêu đề của một sách đã tồn tại.
- Sử dụng tên tác giả và tiêu đề với một sách chưa tồn tại.
- Thủ tìm kiếm theo ISBN

Là người mua sách, Tôi muốn thay đổi số lượng của bất kì một sản phẩm nào trong giỏ hàng, Thiết lập số lượng hoặc xóa một sản phẩm khỏi giỏ hàng.

ATs

- Thêm một sản phẩm hết hàng vào giỏ hàng.
- Đặt một cuốn sách chưa xuất bản vào giỏ hàng.
- Đặt một sách có trong kho vào giỏ hàng
- Đặt thêm cuốn sách đó vào giỏ hàng.

# Hợp Cải tiến Sprint

- **Dừng và nhìn lại, tìm kiếm các cải tiến và xây dựng tổ chức học tập**
- Khung thời gian: 3 giờ
- Thành phần: Scrum Master + Nhóm Phát triển
  - Product Owner có thể tham dự <chicken>
- Câu hỏi:
  - **Đã làm tốt những gì?**
  - **Phải cải thiện những gì?**
- Scrum Master trợ giúp nhóm tìm hiểu, không đưa ra câu trả lời

# Kích náo:

## Cần rà soát những gì trong buổi Họp Cải tiến Sprint?



# Hướng dẫn cải tiến

- Kiểm tra các hành động trước đó. Nếu chưa hoàn thành thì sẽ truy xét lại.
- Chỉ lựa chọn một vài hành động thực sự để làm.
- Tập trung với thái độ xây dựng
  - “Chúng ta có thể làm gì?”
- Lên kế hoạch cho hành động cải tiến tiếp theo
- Quy tắc: Plan>Do>Check>Act

# Cần rà soát những gì?

- Các hành động
  - Liệu chúng ta có quá ít hành động?
  - Liệu chúng có hữu dụng?
  - Liệu chúng có khả thi?
- Có phải nghĩa “hoàn thành” bị nới rộng?
- Có cập nhật lại bản thỏa thuận làm việc của nhóm?
- Chúng ta có cần
  - thêm công việc vào Sprint backlog?
  - thêm mục nào vào Product backlog không?

*Theo Bas Vodde*

# Quy tắc Làm việc

## Đội Z

\*\*\*\*\*

1. Thời gian họp hàng ngày: 9:00
2. Phạt đến muộn: 2 \$
3. Mọi người đều tích hợp hàng ngày
4. Tái cấu trúc lại mã bẩn
5. Hỏi nếu không rõ
6. Sử dụng Lập trình cặp và TDD
7. Thực hiện đúng chuẩn viết mã
8. Kiểm tra lại Định nghĩa Hoàn thành trước khi commit.

Đã ký

# Các sự kiện khác (ngoài Scrum)

- Họp Kế hoạch Phát hành
- Họp làm mịn Product Backlog
- Họp viết User story
- Refactoring Dojo
- Coding Dojo
- Họp Rà soát mã nguồn (Code Review)
- v.v.

# ĐỒ NGHỀ SCRUM

# Các công cụ <đồ nghề> cơ bản

## Product Backlog

- Danh sách ưu tiên mô tả các tính năng và kết quả của sản phẩm
- Có thể chỉnh sửa được

## Sprint Backlog

- Danh sách các công việc cần hoàn thành trong Sprint
- Được cập nhật hàng ngày

## Biểu đồ Burndown

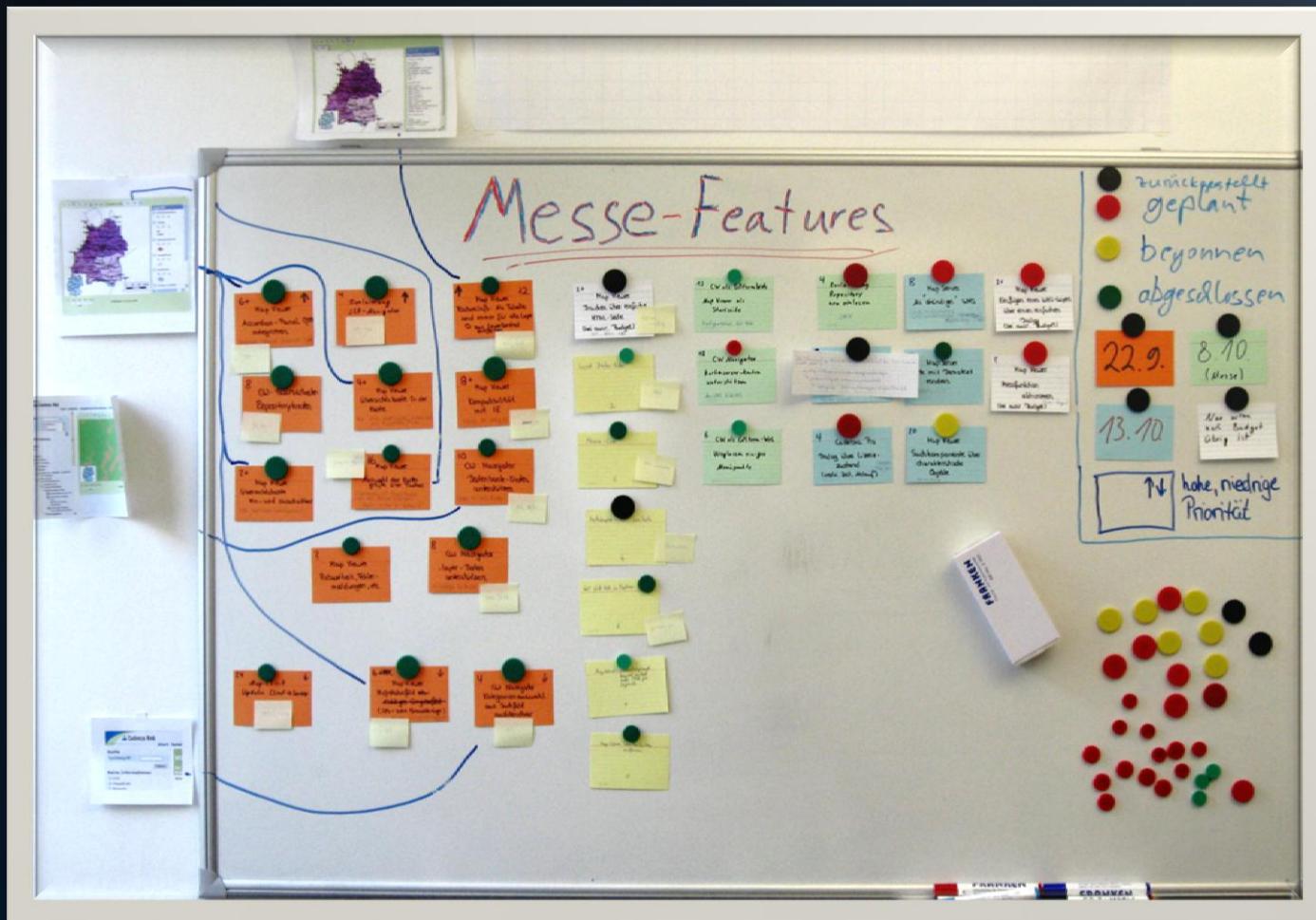
- Hiển thị xu hướng về “thời gian còn lại để hoàn tất công việc”
- Cho biết tiến độ hướng đến Mục tiêu

# Product Backlog

| Item  | Details<br>(wiki URL) | Priority | Estimate of Value | Initial Estimate of Effort | New Estimates of Effort Remaining as of Sprint... |   |   |   |   |   |
|---|-----------------------|----------|-------------------|----------------------------|---|---|---|---|---|---|
|   |                       |          |                   |                            | 1   | 2 | 3 | 4 | 5 | 6 |
| As a buyer, I want to place a book in a shopping cart (see UI sketches on wiki page)                  | ...                   | 1        | 7                 | 5                          |   |   |   |   |   |   |
| As a buyer, I want to remove a book in a shopping cart  | ...                   | 2        | 6                 | 2                          |   |   |   |   |   |   |
| Improve transaction processing performance (see target performance metrics on wiki)                   | ...                   | 3        | 6                 | 13                         |   |   |   |   |   |   |
| Investigate solutions for speeding up credit card validation (see target performance metrics on wiki) | ...                   | 4        | 6                 | 20                         |   |   |   |   |   |   |
| Upgrade all servers to Apache 2.2.3   | ...                   | 5        | 5                 | 13                         |   |   |   |   |   |   |
| Diagnose and fix the order processing script errors ( <a href="#">bugzilla ID 14823</a> )             | ...                   | 6        | 2                 | 3                          |   |   |   |   |   |   |
| As a shopper, I want to create and save a wish list   | ...                   | 7        | 7                 | 40                         |   |   |   |   |   |   |
| As a shopper, I want to add or delete items on my wish list   | ...                   | 8        | 4                 | 20                         |   |   |   |   |   |   |

- Các yêu cầu về sản phẩm
- Các hạng mục có giá trị với người dùng và khách hàng
- Được đặt độ ưu tiên và bảo trì bởi Product Owner

# Product Backlog: Bảng tính năng



Nguồn: [http://img151.imagevenue.com/img.php?image=28385\\_messefeatures\\_122\\_458lo.jpg](http://img151.imagevenue.com/img.php?image=28385_messefeatures_122_458lo.jpg)

# Các đặc điểm của Product Backlog

- Mỗi sản phẩm chỉ có một Product Backlog
- Danh sách các tính năng (chức năng & phi chức năng)
- Nổi bật, ưu tiên hóa và được ước tính
- Chi tiết hơn với các hạng mục có độ ưu tiên cao hơn
- Product Owner xác định độ ưu tiên cho các hạng mục
- **Luôn luôn hiện diện** cho các bên
- Có nguồn gốc từ Kế hoạch Kinh doanh hoặc Tuyên bố Tầm nhìn
- *Các nội dung tùy chọn:*
  - Rủi ro, kiểm thử, các sản phẩm phụ thuộc, người hiểu rõ về một hạng mục, v.v..

# Làm mịn Product Backlog

- Phân tách các hạng mục của backlog thành các nhóm **Thực thi** và **Chờ** trong Product Backlog
- Những hạng mục thuộc nhóm Thực thi sẽ được triển khai
- Các hạng mục Chờ sẽ được xác định lại độ ưu tiên và tiếp tục làm mịn sau đó

## Thực thi

1. -----
2. -----
3. -----

## Chờ

- 
- 
- 
- 
-

# Sprint Backlog

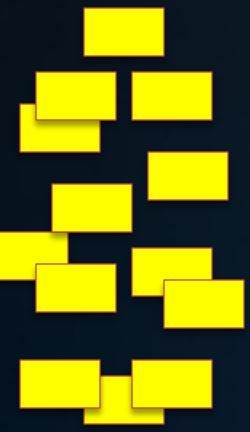
| Product Backlog Item                                  | Sprint Task                                   | Volunteer | Initial Estimate of Effort | New Estimates of Effort Remaining as of Day... |   |   |   |   |   |
|---|---|-----------|----------------------------|--|---|---|---|---|---|
|   |   |           |                            | 1  | 2 | 3 | 4 | 5 | 6 |
| As a buyer, I want to place a book in a shopping cart | modify database                               |           | 5                          |  |   |   |   |   |   |
|   | create webpage (UI)                           |           | 8                          |  |   |   |   |   |   |
|   | create webpage (Javascript logic)             |           | 13                         |  |   |   |   |   |   |
|   | write automated acceptance tests              |           | 13                         |  |   |   |   |   |   |
|   | update buyer help webpage                     |           |                            |  |   |   |   |   |   |
|   | ...   |           |                            |  |   |   |   |   |   |
| Improve transaction processing performance            | merge DCP code and complete layer-level tests |           |                            |  |   |   |   |   |   |
|   | complete machine order for pRank              |           |                            |  |   |   |   |   |   |
|   | change DCP and reader to use pRank http API   |           |                            |  |   |   |   |   |   |

- Công cụ lên kế hoạch và theo dõi trong một Sprint
- Được bảo trì bởi Nhóm Phát triển

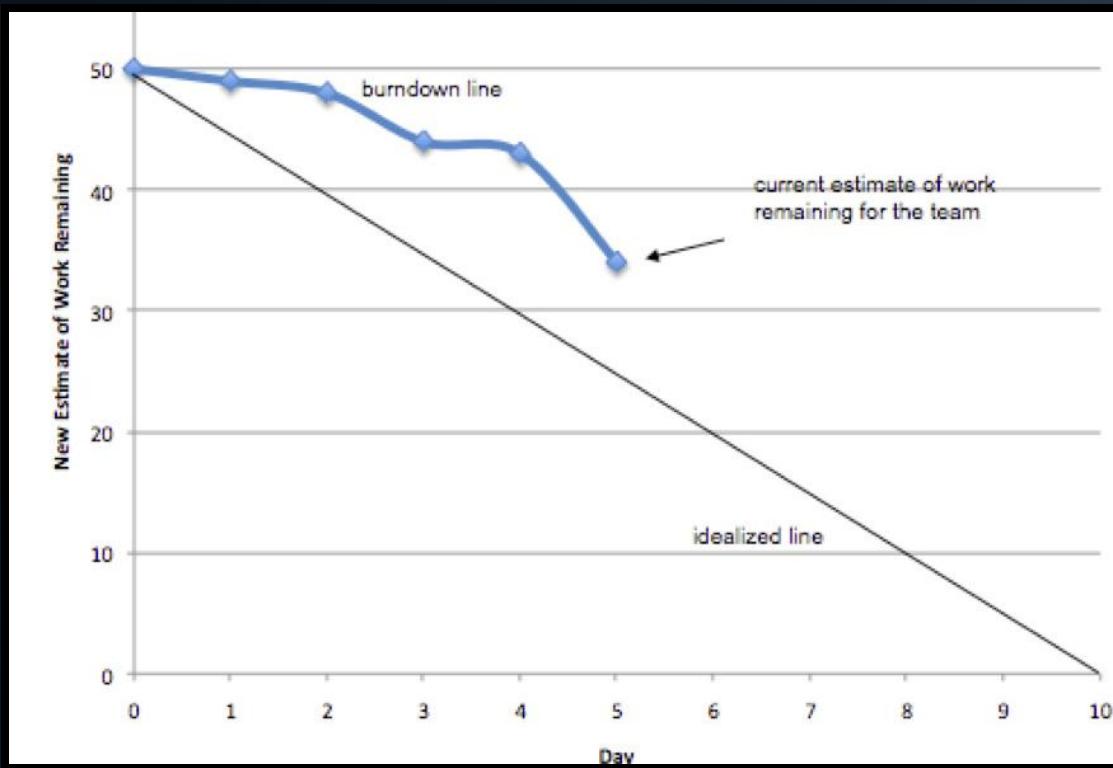


Bảng Công việc – Sprint Backlog hiệu quả

# Bảng Công việc

| Việc cần làm   | Việc đang triển khai   |                        |     |                        | Hoàn thành |
|--|------------------------|------------------------|-----|------------------------|------------|
|  | Bước 1                 | Bước 2                 | ... | Bước n                 |            |
|  | Hàng đợi<br>Đang xử lý | Hàng đợi<br>Đang xử lý | ... | Hàng đợi<br>Đang xử lý |            |

# Biểu đồ Burndown



- Thể hiện **tiến độ** hướng tới Mục tiêu Sprint
- Điều **quan trọng** là còn bao nhiêu việc phải làm để hoàn tất công việc, chứ không phải là đã bỏ ra bao nhiêu công sức trong quá khứ

# Bảo trì Sprint Backlog và Biểu đồ Burndown

- Các nhà phát triển tự nguyện chọn công việc để làm
  - Không ai “gán” việc cho họ
  - Công việc được ước tính theo giờ (từ 1-16 giờ)
- Tùy ý thêm, bớt hoặc thay đổi Sprint Backlog
  - Người có “vai trò” chủ đạo và người có “vai trò” tình nguyện
- Nếu công việc không rõ ràng, tạo ra một hạng mục Sprint Backlog có khối lượng lớn về thời gian và chia nhỏ dần
- Ước tính khối lượng công việc được cập nhật hằng ngày vào Sprint Backlog
- Cập nhật lại Burndown hằng ngày (ngay sau Hợp Scrum Hằng ngày)

# Ước tính Sprint Backlog

- Sprint luôn cho thấy **khối lượng công việc <ước tính> còn lại** để hoàn thành Mục tiêu Sprint
- Sprint Backlog KHÔNG phải là công cụ để kiểm soát thời gian
- **Timesheet** KHÔNG có ý nghĩa gì trong Scrum
- **Năng suất** được đo bằng việc đạt tới Mục tiêu Sprint, nó hướng tới kết quả chứ không bởi ta bỏ ra bao nhiêu nguồn lực

# Tốc độ (velocity)

- Khối lượng công việc được hoàn tất trong một Sprint
  - hoặc trên một tuần
- Căn cứ để Product Owner tiên đoán thời điểm phát hành
- Căn cứ để Nhóm Phát triển xác định công việc khối lượng cho Sprint
- Trong cùng sản phẩm, với chất lượng đảm bảo, tốc độ tăng *có thể* phản ánh năng suất lao động
- Các yếu tố ảnh hưởng: độ xác định, thay đổi công nghệ, tăng trưởng kĩ năng, động lực, cam kết, các loại waste v.v.

# Công cụ khác

- Biểu đồ Burndown cho sản phẩm hay bản Phát hành
- Biểu đồ burnup (cho tính năng)
- Biểu đồ năng suất
- Danh mục Vấn đề
- Các hệ thống Phần mềm cộng tác\quản lí dự án linh hoạt (Redmine, TFS, JIRA, Assembla.com v.v.)
- v.v

# Thảo luận:

Có những khó khăn nào cản trở  
thực hiện Scrum?



# CÁC VẤN ĐỀ KĨ THUẬT

# User Story

- User story là các yêu cầu linh hoạt (agile requirement)
- Đảm bảo sự cân bằng của các bên tham gia phát triển sản phẩm: khách hàng, người dùng và nhà phát triển
  - Thể hiện bằng một ngôn ngữ hướng-người-dùng và các nhà phát triển có thể hiểu được
- Hướng tới người dùng và nghiệp vụ, không chứa các đặc tính về hệ thống

| <b>Item</b>   |
|---|
| As a buyer, I want to place a book in a shopping cart (see UI sketches on wiki page)                  |
| As a buyer, I want to remove a book in a shopping cart  |
| Improve transaction processing performance (see target performance metrics on wiki)                   |
| Investigate solutions for speeding up credit card validation (see target performance metrics on wiki) |

# INVEST – các tiêu chuẩn cho một user story tốt

**Independent – Độc lập**

**Negotiable – Đàm phán được**

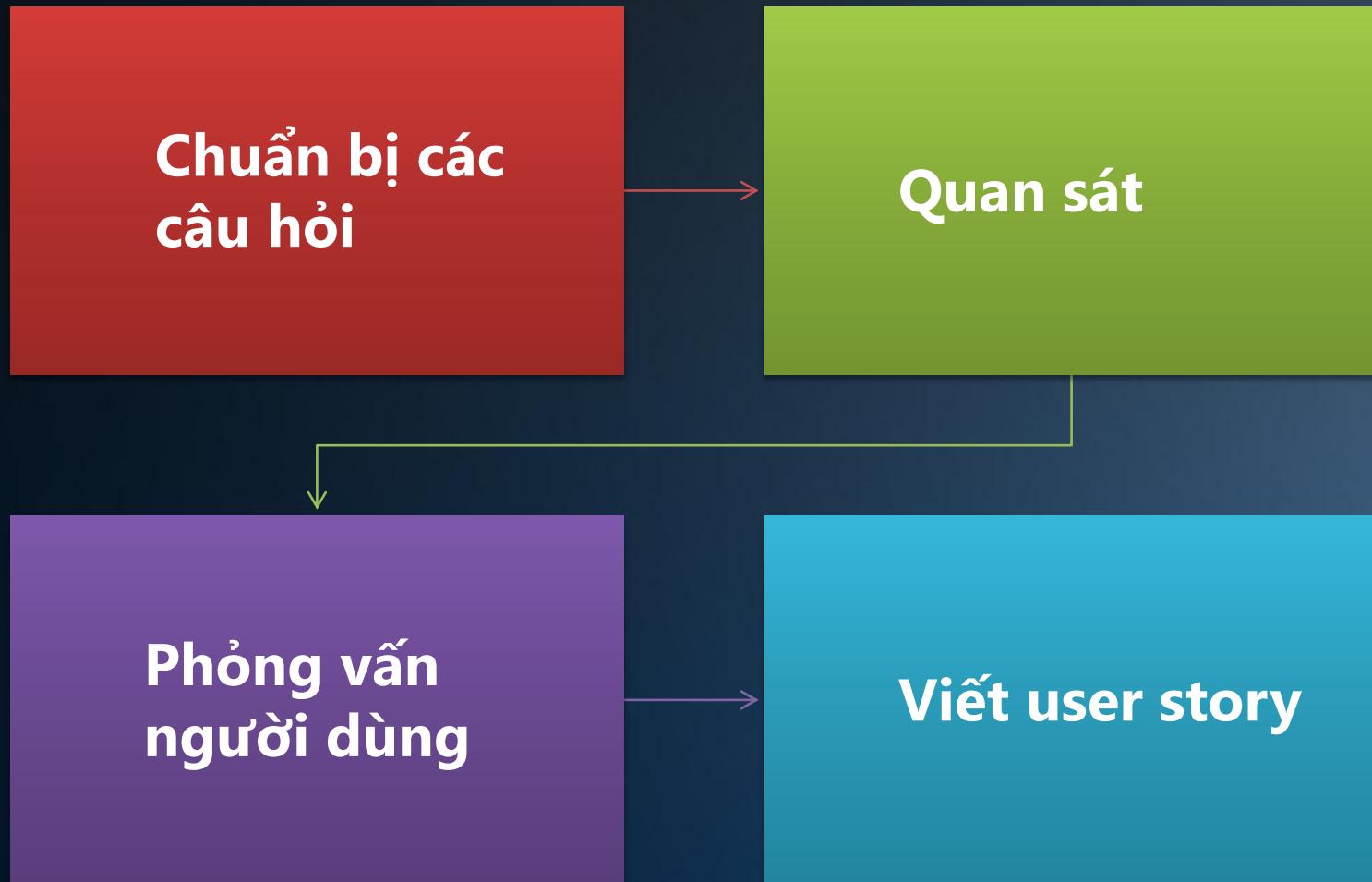
**Valuable – Đáng giá**

**Estimatable – Ước tính được**

**Sized appropriately – Kích thước phù hợp**

**Testable – Kiểm thử được**

# Tập hợp các user story



# Hợp xây dựng user story

- Người tham gia: nhà phát triển, người dùng, khách hàng, thành phần khác (được gọi là những “chú lợn”)
- Thảo luận để đưa ra các story
- Mục tiêu là viết được **càng nhiều story càng tốt**
  - Một số sẽ “sẵn sàng triển khai”
  - Một số khác sẽ là “epic”
- Không xác định độ ưu tiên trong buổi hội thảo này
- **Product Owner và những người có liên quan được tham gia nhưng không bắt buộc.**

# Cộng tác nhóm: Viết user story

Là <người dùng cụ thể\vai trò>,  
Tôi muốn <làm gì đó>  
để <phục vụ mục đích gì đó>.



## Hoạt động: Ước lượng linh hoạt

# Lập trình cặp

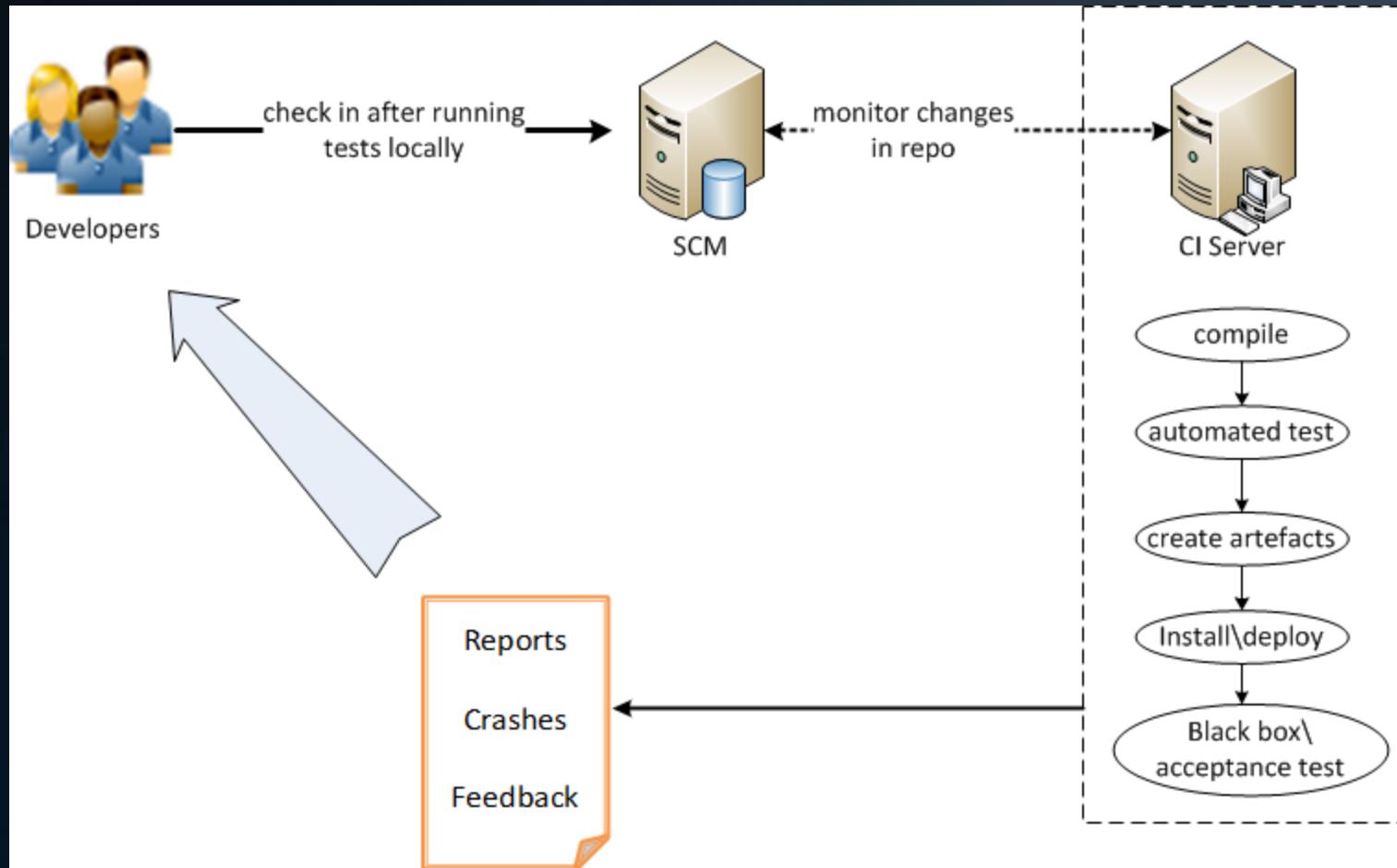
- **Hai nhà phát triển cùng chia sẻ một vấn đề, với một máy tính, một bàn phím và với mục tiêu: giải quyết vấn đề đó.**
- Sử dụng sự ĐỒNG THUẬN, nhưng thông qua TRANH LUẬN!
- Một ví dụ về “luồng một sản phẩm”
- Chậm hơn nhưng hiệu quả hơn & chất lượng hơn
- **Có 2 vai trò: Người lái (Driver) và Hoa tiêu (Navigator):**
  - Người lái không quan tâm tới bức tranh toàn cảnh
  - Người lái nên “**rời bàn phím trong giây lát**”
  - Hoa tiêu có xu hướng sử dụng tư duy mẫu trong giải quyết vấn đề



# Tích hợp Liên tục

- **Tích hợp Liên tục** (Continuous integration - CI) triển khai *liên tục* các tiến trình để đảm bảo việc quản lý chất lượng — từng phần nhỏ hiệu quả, áp dụng thường xuyên.
- Được hỗ trợ bởi một hệ thống CI với rất nhiều các kiểm thử tự động, build và các thành phần khác.
- Lợi ích:
  - Tăng cường sự minh bạch
  - Tăng cường sự hợp tác và truyền thông
  - Cho phép mọi người cùng làm việc trên một mã nguồn

# Hệ thống CI



# Thiết kế tiến hóa

Thiết kế  
những thứ  
phức tạp có  
khả năng linh  
hoạt trên  
giấy hoặc  
công cụ

Thiết kế tiến  
hóa

Luôn có  
những thứ  
thay đổi bất  
ngờ

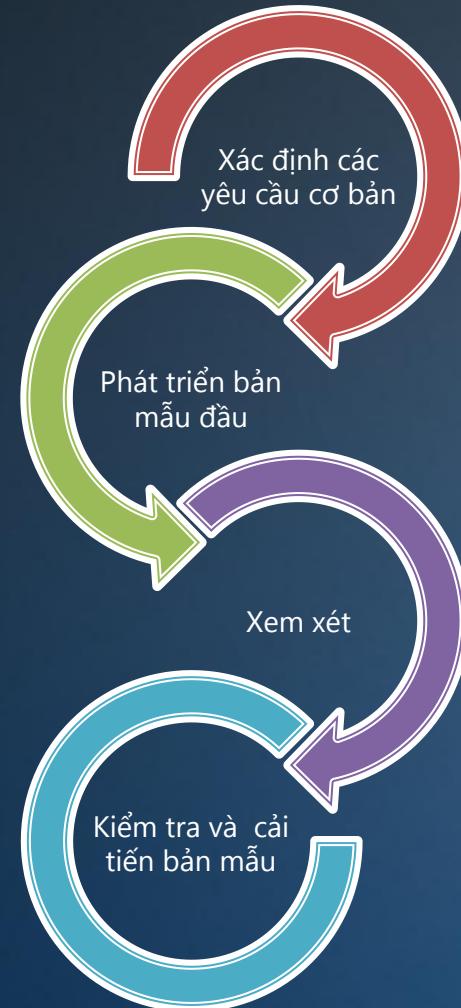
Luôn có  
những thứ  
thay đổi bất  
ngờ

Nhiều phức tạp  
hơn mức cần  
thiết. Khó khăn  
trong việc bảo  
trì

Dễ dàng thích  
nghi. ID **thay**  
**đổi dễ dàng.**  
Giảm bớt phức  
tạp

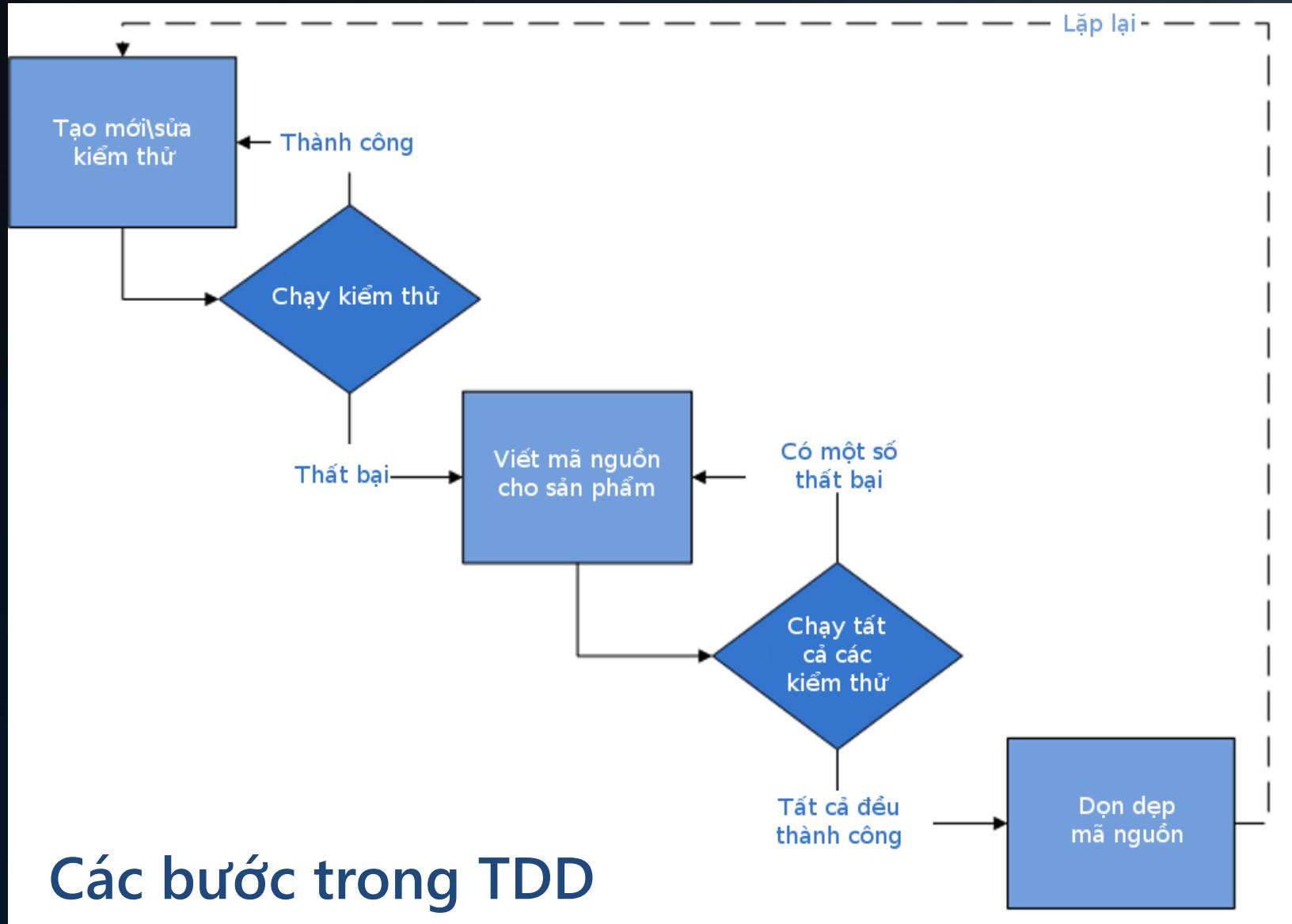
# Làm bản mẫu

- Việc bản mẫu có sớm sẽ giúp người dùng dễ hình dung ra sản phẩm sau khi hoàn thành
- Khuyến khích sự **tham gia tích cực** của người dùng và nhà phát triển
- Tăng tốc độ phát triển hệ thống



# Phát triển Hướng-Kiểm-thử (TDD)

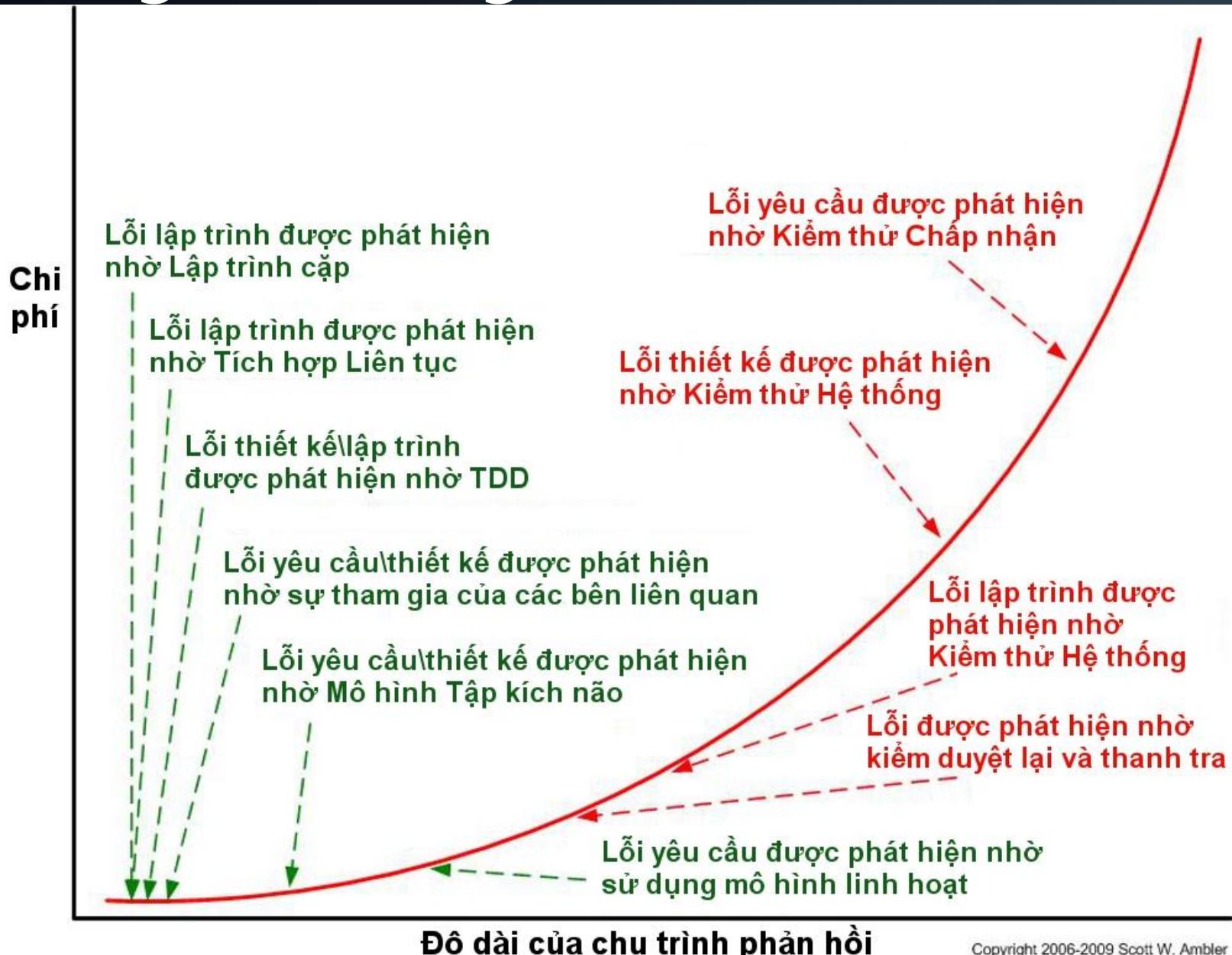
- Không tiến hành viết mã nguồn cho tới khi các kiểm thử đã được thiết kế hoàn chỉnh!
- Chiến lược
  - **Làm cho Thất bại**
    - Không có mã nguồn nào mà không có kiểm thử thất bại
  - **Làm cho Thành công**
    - Đơn giản nhất có thể
  - **Làm cho Tốt hơn**
    - Cải tiến (mã nguồn, thiết kế, kiểm thử, tài liệu)
  - **Tin tưởng vào việc kiểm thử**



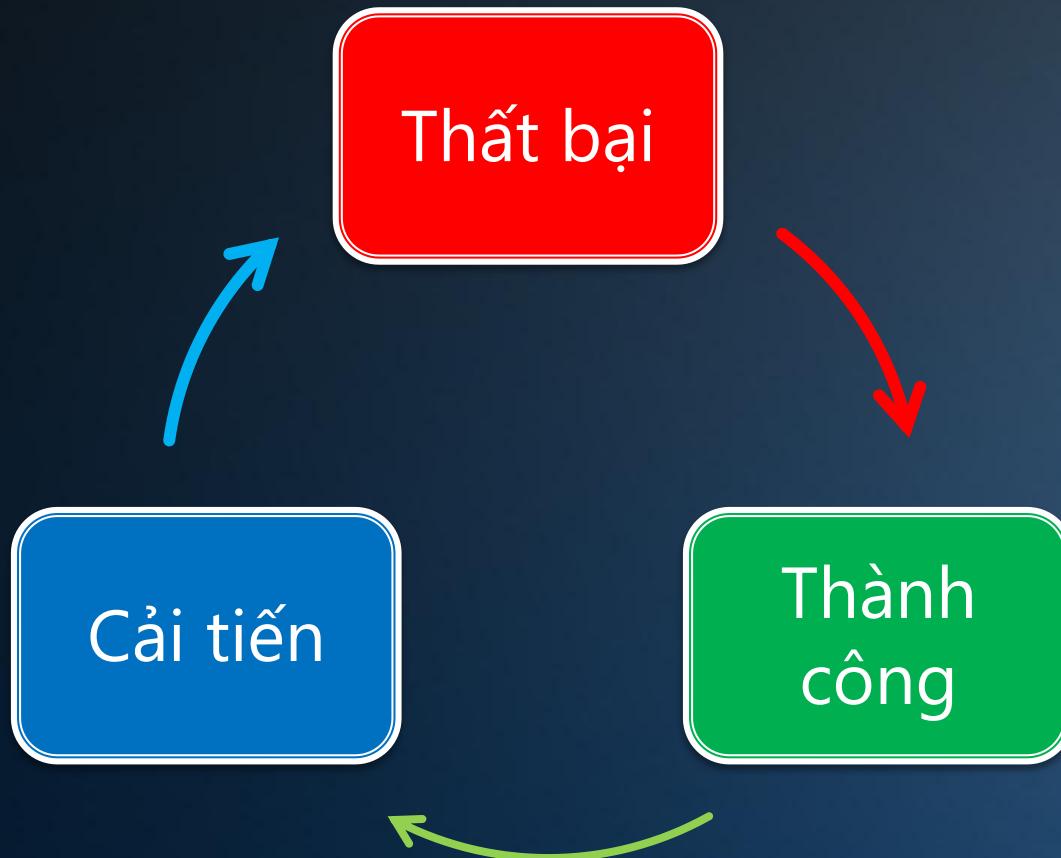
## Các bước trong TDD

Nguồn ảnh: [Excirial](http://upload.wikimedia.org/wikipedia/en/9/9c/Test-driven_development.PNG) ([http://upload.wikimedia.org/wikipedia/en/9/9c/Test-driven\\_development.PNG](http://upload.wikimedia.org/wikipedia/en/9/9c/Test-driven_development.PNG))

# Cái giá của bug



# Ba nhân tố RGB



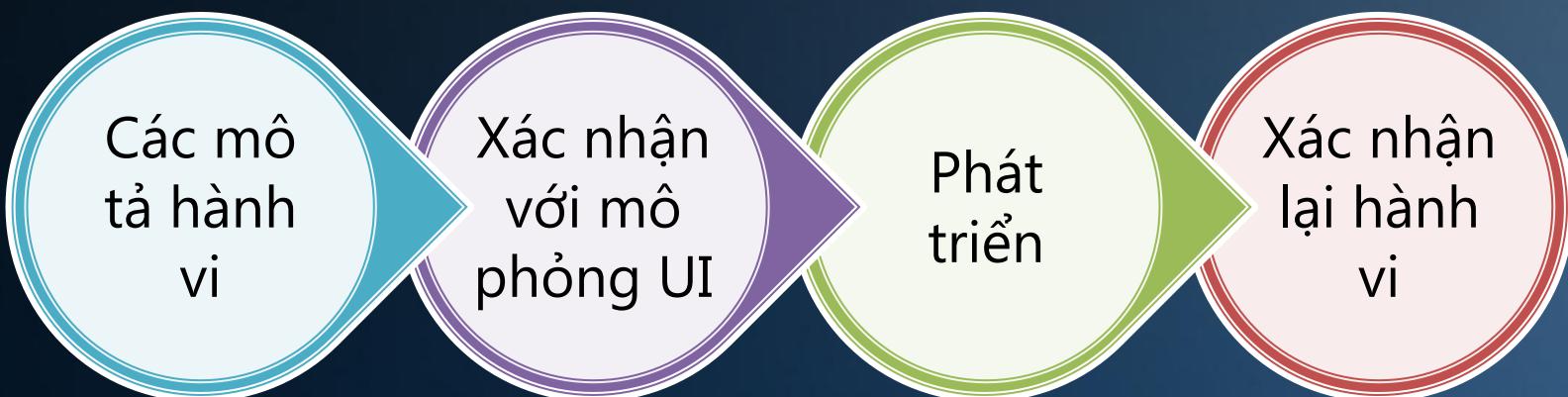
# Phát triển Hướng Kiểm thử Chấp nhận (ATDD)

- **Một kỹ thuật dành cho tự động kiểm thử chấp nhận**
- **Chiến lược 3D**
  - Thảo luận (**Discuss**) trong hội thảo về xác định yêu cầu
    - Để xây dựng các thư viện kiểm thử
  - Phát triển (**Develop**) với sự đồng thuận
    - Để tạo ra nhiều tính năng đạt kiểm thử hơn
  - Cung cấp (**Deliver**) các chấp nhận
    - Để đạt được định nghĩa hoàn thành, cần sự chấp nhận của người dùng



# Phát triển Hướng-Hành-vi (BDD)

- Phát triển phần mềm được chỉ dẫn trực tiếp bởi các mô tả hành vi và các tính năng (và mô phỏng).
- Hơi giống với ATDD, nhưng khác về tư duy
- Chất lượng phần mềm cao hơn, tính tự tổ chức tốt hơn



# Tái cấu trúc mã nguồn

## Code Refactoring

- Bạn thực hành “viết mã một ít, sửa lỗi một ít”  
=> kết quả là code bẩn và thiết kế tồi.
- Tái cấu trúc để code tốt hơn, có thiết kế tốt hơn, vẫn giữ nguyên chức năng của hệ thống
  - “tốt hơn” nghĩa là gì?
- Giữ cho mã nguồn:
  - Dễ bảo trì
  - Dễ mở rộng
  - Tính gắn kết cao (High Cohesion)
  - Ít phụ thuộc (Low Coupling)
  - Loại bỏ trùng lặp

# Các kỹ thuật tái cấu trúc

- **Trừu tượng hóa (Abstraction)**
  - Bao gói các trường
  - Dùng kiểu khái quát (generic)
  - Thay thế mã kiểm tra (check) với State/Strategy
  - Thay thế các điều kiện bằng đa hình
- **Phân tách mã**
  - Tạo mới phương thức, thay thế một phần lớn mã trong một phương thức bằng phương thức khác
  - Tạo thêm lớp mới
- **Chuẩn hóa mã**
  - Chuyển phương thức hoặc trường
  - Đổi tên phương thức, trường
  - Đẩy lên lớp cấp trên hoặc lớp cha
  - Đẩy xuống lớp cấp dưới hoặc lớp con

# TỔ CHỨC

# Tổ chức

- Scrum sẽ xung đột vai trò\ tổ chức:
  - Quản lý Dự án
  - Phòng ban Chức năng: Đảm bảo Chất lượng, Kiểm thử, Phát triển, v.v..
  - Các vai trò khác sẽ thay đổi
- Vai trò\sự nghiệp của từng cá nhân thay đổi
  - Các chức danh mới: Huấn luyện viên agile, lãnh đạo agile, Scrum Master, Product Owner

# Phạm vi áp dụng Scrum

Pilot

- Thủ để biết Scrum vận hành thế nào
- Thời gian ngắn
- Để ra quyết định dùng Scrum hay không

Project

- Sử dụng tư duy Scrum là chủ yếu
- Đầu tư ngắn hạn, thu lợi nhanh
- Không bền vững

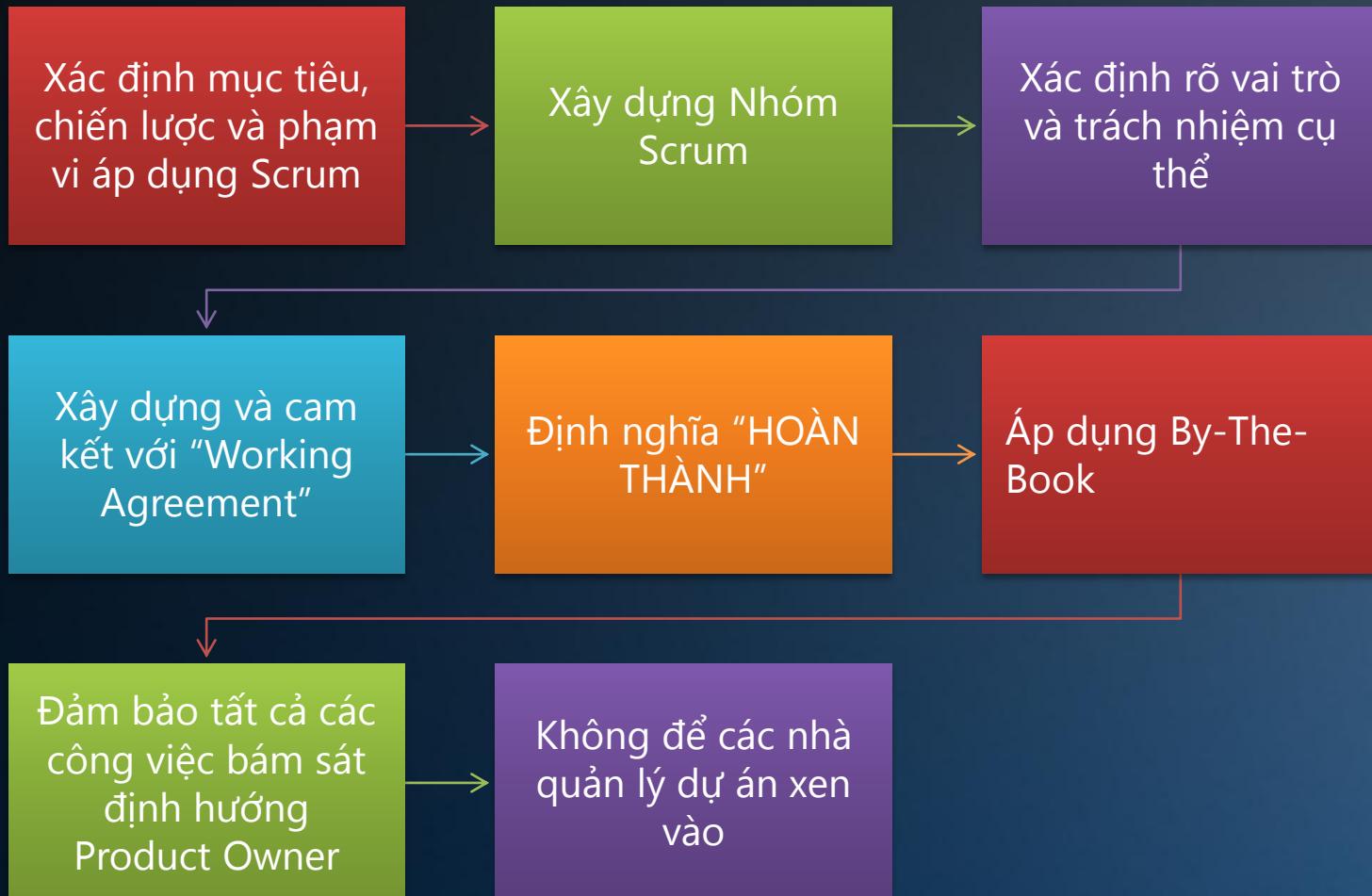
Studio

- Xây dựng “nhà máy” sản xuất phần mềm
- 100% Scrum

Enterprise

- Mở rộng hoạt động
- Scrum thành văn hóa công ty

# Bắt đầu với Scrum



# Đường đến ScrumBut

- Không hiểu rõ mục đích của Scrum
  - “Viên đạn bạc”?
- Che dấu những vấn đề hoặc ngăn cản sự thay đổi của tổ chức
- Chỉnh sửa do bối cảnh cụ thể
  - Điều này rất phổ biến, nhưng phải giữ được sự cân bằng và tinh thần Scrum

# Đưa Scrum vào tổ chức của bạn

- Có được **sự tin tưởng** của tổ chức trước khi bắt họ phải **thay đổi**
  - Để thay đổi tổ chức, hãy cho họ thấy *Bằng chứng* về Siêu-Hiệu suất khi sử dụng Scrum.
  - “Hãy làm cho Scrum trở thành một loại vi rút trong tổ chức”
- Luôn luôn tập trung vào sự thay đổi và **tiến bộ** của bản thân thay vì **đổ lỗi** cho tổ chức
  - Hãy suy nghĩ “theo lối tinh gọn” (**lean thinking**)

# Thảo luận:

Cần làm gì để tạo  
môi trường Scrum  
trong công ty?



# Những trở ngại với Scrum

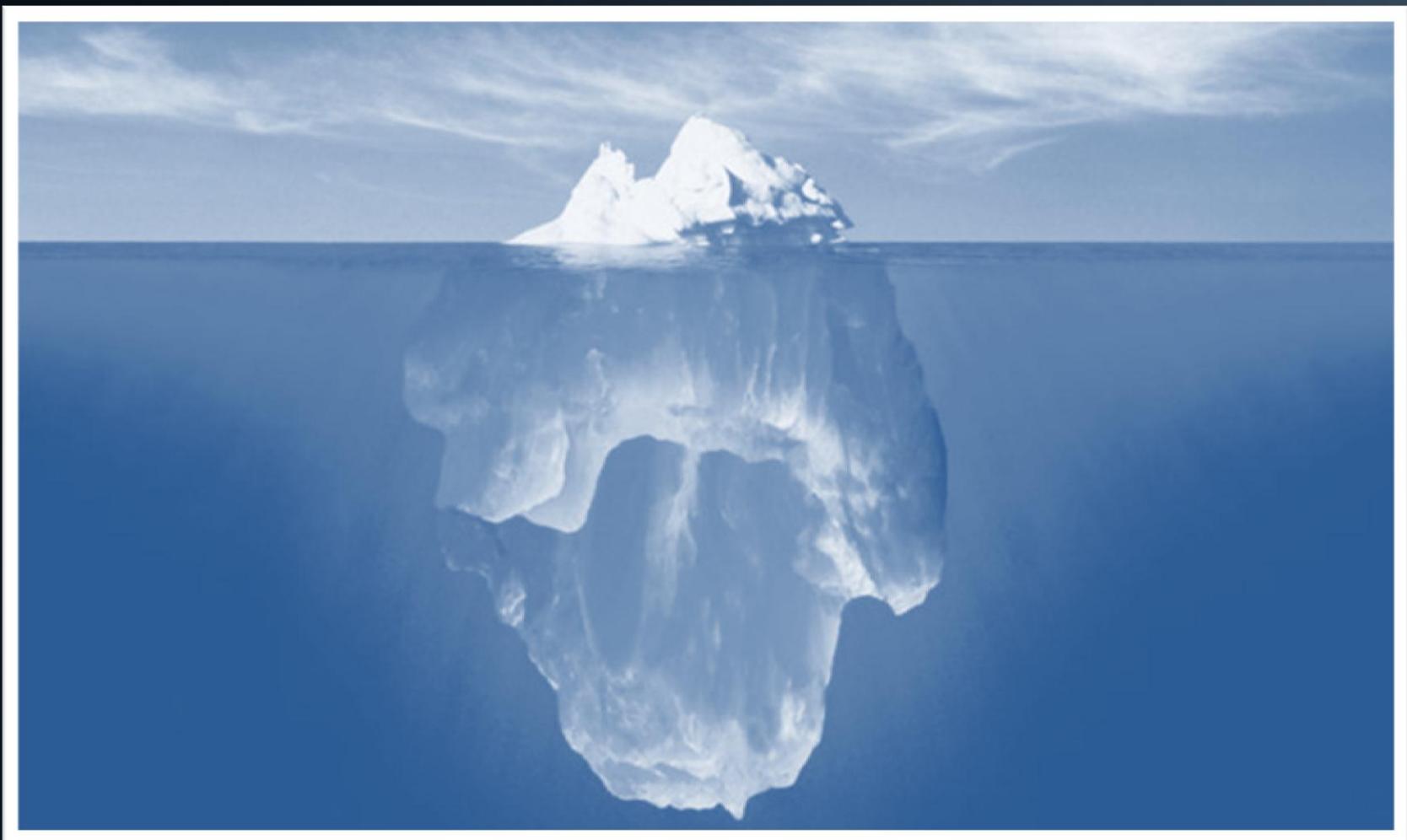
- Theo **Bas Vodde**:
  - Những ảo tưởng về mệnh lệnh và điều khiển
  - Vẫn giữ các lề thói cũ
  - Tính “hiển nhiên” của ScrumBut
  - Trông chờ phép màu nào đó
  - Thiếu sự minh bạch
  - “Quán tính Waterfall”

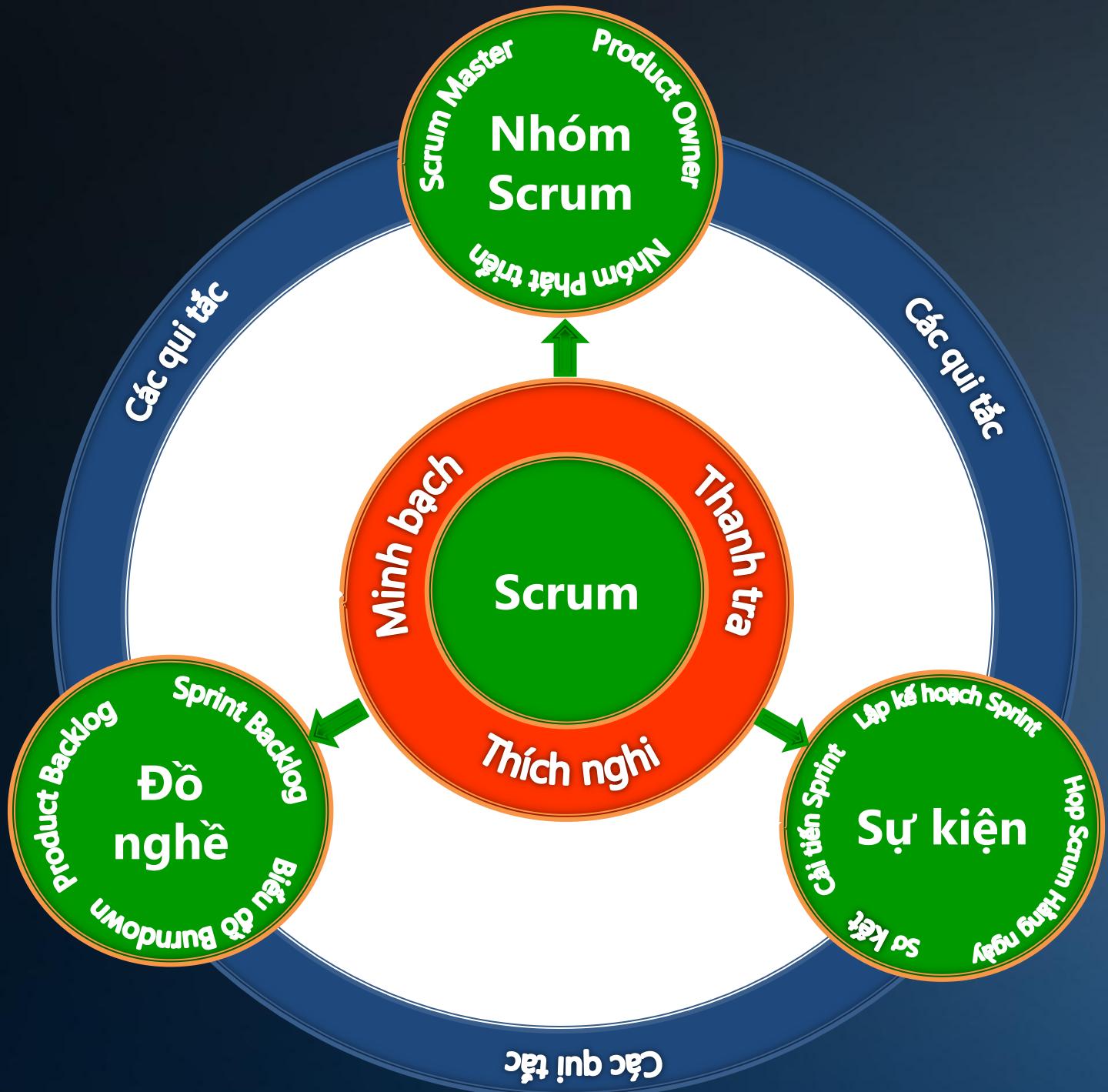
# Nguyên nhân thất bại

- Sử dụng không hiệu quả hoạt động cải tiến
- Không có khả năng lôi kéo tất cả mọi người cùng tham gia vào lập kế hoạch
- Không chú ý tới cơ sở hạ tầng cần thiết
- Scrum Master tồi
- Product Owner không giữ được sự nhất quán
- Thất bại trong việc thúc đẩy hoạt động kiểm thử
- Khôi phục lại khuôn mẫu trước đây
- Chỉ quan tâm tới "cam kết sổ sách" từ phía quản lý điều hành
- Nhóm thiếu thẩm quyền và khả năng ra quyết định
- Không có người chịu trách nhiệm truyền đạt khi tiến hành làm việc phân tán
- Văn hóa của tổ chức không hỗ trợ việc học tập
- Từ chối chấp nhận một cách gay gắt

*Jean Tabaka*

# Tảng băng trôi Scrum







# HỎI & ĐÁP

# Tài nguyên và Tham khảo

- Pete Deemer, Gabrielle Benefield, Craig Larman & Bas Vodde, **Scrum Primer** ver. 1.2
- Ken Schwaber & Jeff Sutherland, Scrum Guide, Scrum.org
- Jean Tabaka, **Twelve ways agile adoption failed**, Better Software, Nov. 2007,  
[www.stikymind.com](http://www.stikymind.com) (<http://www.nxtbook.com/nxtbooks/sqe/bettersoftware1107/>)
- Mountain Goats, Scrum Overview,  
(<http://www.mountaingoatsoftware.com/scrum/overview> )
- MoutainGoats, Planning Poker Estimating in details  
(<http://store.mountaingoatsoftware.com/pages/planning-poker-in-detail>)
- Scrum Alliance , <http://www.scrumalliance.org/blog/129-pmi-develops-agile-certification>
- Bill Wake, **INVEST in Good Stories, and SMART Tasks**,  
<http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>
- A gallery of team rooms and charts, <http://xp123.com/articles/a-gallery-of-team-rooms-and-charts/>

# Cộng đồng Agile\Scrum

- Definitive Scrum, Training: [www.Scrum.org](http://www.Scrum.org)
- Scrum Alliance: [www.scrumalliance.org](http://www.scrumalliance.org)
- Agile Alliance: [www.agilealliance.org](http://www.agilealliance.org)
- Hà Nội: [www.hanoiscrum.net](http://www.hanoiscrum.net)
- Hồ Chí Minh City: [www.agilevietnam.org](http://www.agilevietnam.org)

# PHỤ LỤC A: DANH MỤC KIỂM TRA DÀNH CHO SCRUM MASTER

Tải tệp tin:

[http://www.scrummasterchecklist.org/pdf/scrummaster\\_checklist09.pdf](http://www.scrummasterchecklist.org/pdf/scrummaster_checklist09.pdf)  
<http://hanoiscrum.net/hnsrum/resource/139>

# Phụ lục B: *East2West* (1)

***East2West Tours and Travels*** là một công ty trẻ và năng động được điều hành bởi các cá nhân cung cam kết cung cấp những dịch vụ đạt chuẩn cao nhất trong ngành du lịch. Chúng tôi đặt sự tập trung vào các dịch vụ dành cho cá nhân, sự quan tâm tới từng chi tiết nhỏ, các hành trình được đổi mới và sự chăm sóc tuyệt vời dành cho khách hàng. Tuy nhiên, trong những năm qua chúng tôi gặp phải một vài truy vấn hoặc tình huống khó, đó là nhưng thứ mà chúng tôi không thể giải quyết và hướng tới việc tiến bộ trong tương lai. Chúng tôi mong chờ dịch vụ của các bạn 7 ngày/tuần. Khả năng cạnh tranh của chúng tôi cao, và nếu bạn đang tìm kiếm một công ty du lịch không chỉ cung cấp được các dịch vụ hàng đầu, mà còn có thể dự đoán và hiểu rõ nhu cầu của bạn, bạn không cần phải tìm kiếm đâu xa.

# Phụ lục B: *East2West* (2)

- Các hệ thống hiện nay tại ***East2West Tours and Travels*** với các hướng dẫn đăng ký các dịch vụ dựa trên lợi nhuận nhất định như: **Dịch vụ du lịch trọn gói và Dịch vụ cho thuê xe hơi**.
  - Một tour trọn gói là một kế hoạch du lịch tới một địa điểm định trước. Chúng tôi nhóm các kế hoạch này thành những gói như:
    - Các tour đã được sắp đặt địa điểm
    - Tiện nghi
    - Đồ ăn
    - Du lịch bản địa
  - Mỗi gói được đặc trưng bởi sự cố định về thời gian và một mức giá. Các thành phần không bị tính giá thêm.
  - Chúng tôi đã làm trong lĩnh vực này 10 năm. Kinh nghiệm cho phép chúng tôi tạo ra các gói đó, có thể đã trở thành phổ biến. Chúng tôi cũng có các dữ liệu được lấy từ các hồ sơ trong quá khứ.
  - Các gói có thể:
    - Các tour tìm hiểu về một quốc gia
    - Các hoạt động như bộ hành, cắm trại hoặc săn bắn
    - Các địa điểm lịch sử
  - Trong hệ thống hiện tại, khi một khách hàng đến với chúng tôi để tìm kiếm một gói đầy đủ cho kỳ nghỉ, chúng tôi đưa cho họ những gói sẵn có của mình. Khách hàng lựa chọn và chúng tôi tiến hành đặt cho họ gói đó. Thông thường, mức giá trọn gói được xác định trên cơ sở số người. Do đó, số tiền phải trả được tính tùy thuộc vào số vé họ đặt. Vì vậy, hệ thống hướng dẫn là rất chậm, khi hiển thị tài liệu quảng cáo cho các gói khác nhau, nhập dữ liệu khách hàng và ghi nhận các dữ liệu bán hàng thường tốn nhiều thời gian và cần nhiều sổ sách. Vậy nên, một đề xuất là xây dựng hệ thống thay thế hệ thống cũ để giúp các tiến trình xử lý nhanh hơn.

# Phụ lục B: *East2West* (3)

## • Dịch vụ Thuê xe hơi

- Thứ hai, chúng tôi cũng cho thuê xe hơi. Đôi khi các khách hàng sẵn sàng với kế hoạch du lịch của họ và chỉ cần sự chuyên chở từ chúng tôi.

Chúng tôi có nhiều loại xe hơi như Honda, Hyundai, Tata, Maruti và Toyota và một số có kích cỡ nhỏ, kích cỡ lớn, hạng sang và xe tải hạng nhỏ. Khách hàng sẽ lựa chọn dựa trên các yếu tố:

- Mẫu xe
- Loại xe
- Số ghế
- Có lái xe\ tự lái
- Có điều hòa nhiệt độ hoặc không
- Ngày thuê và ngày trả xe

- Với hệ thống hiện tại, khi khách hàng tới thuê xe, một danh sách xe dựa theo yêu cầu của khách hàng sẽ được đưa ra. Khách hàng lựa chọn và chúng tôi đặt chiếc xe đó cho họ. Tìm chiếc xe phù hợp và ghi nhận các thông tin bán hàng là tốn kém thời gian và cần nhiều sổ sách.
- Hai dịch vụ trên đều có thể lập hóa đơn. Tuy nhiên có thể hủy bỏ. Việc bồi thường do hoàn trả được tính toán dựa vào số ngày còn lại trước khi ngày thực tế sử dụng dịch vụ.
- Các quy tắc tính tiền hoàn trả như sau:

- Trước 1 ngày – 75%
- 2 ngày – 80%
- 3 ngày – 85%
- 4 ngày – 90%
- 5 ngày – 95%

## • Thông tin trợ giúp

- Đây là dịch vụ thứ ba, dịch vụ này phi lợi nhuận, nhưng chúng tôi coi nó như một dịch vụ tạo ra cơ hội thu lợi nhuận cho chúng tôi. Chúng tôi có một cơ sở dữ liệu rất lớn và phong phú theo từng ngày về khách sạn và vé máy bay. Chúng tôi phải tổ chức chúng thông thạo. Khách hàng khi đã quyết định được địa điểm du lịch, họ sẽ rất hứng thú khi tìm thấy những thông tin chi tiết về các khách sạn, các chuyến bay đến địa điểm đó. Dịch vụ trợ giúp thông tin của chúng tôi sẽ giúp họ truy xuất tới các thông tin như khách sạn, mức giá của họ, nhưng tiện ích mà họ cung cấp, các món ăn đặc biệt và các chuyến bay tới thành phố đó, lịch trình cũng như giá vé. Các thông tin này được cung cấp rất phổ biến và không lập hóa đơn. Trong hệ thống hiện tại, các thông tin này được lưu trữ trong các tài liệu quảng cáo và các hồ sơ tĩnh. Do đó, chúng tôi lập kế hoạch để lưu trữ và thu thập tự động các thông tin đó.

# Phụ lục B: *East2West* (4)

Hệ thống đề xuất phát triển có thể thực hiện các chức năng sau:

- **Du lịch trọn gói**
  - Tìm kiếm và hiển thị thông tin chi tiết về các gói du lịch có sẵn theo các tiêu chí được cung cấp bởi khách hàng
  - Đặt tour du lịch mà khách hàng đã lựa chọn
  - Hủy bỏ tour du lịch khi cần
- **Cho thuê xe hơi**
  - Tìm kiếm và hiển thị các mẫu xe, loại xe và giá thuê sẵn có phù hợp với các tiêu chí đưa ra bởi khách hàng. Các tiêu chí có thể là mẫu xe, loại xe, yêu cầu về điều hòa nhiệt độ, yêu cầu về lái xe, số ghế, ngày thuê và ngày trả xe.
  - Đặt chiếc xe mà khách hàng đã lựa chọn
  - Hủy bỏ đặt thuê xe khi cần
- **Thông tin trợ giúp**
  - Tìm kiếm và hiển thị các khách sạn có trong một thành phố nào đó với mức giá được đưa ra bởi khách hàng.
  - Tìm kiếm và hiển thị các thông tin về những chuyến bay sẵn có với đích đến và ngày khởi hành mà khách hàng chỉ định
- **Phân tích thông tin bán hàng**
  - Thực hiện phân tích thông tin bán hàng của dịch vụ cho thuê xe bằng cách tính toán doanh thu dựa trên các yếu tố: Mẫu xe, Loại xe, Phạm vi thuê xe của khách hàng, Giá thuê, Xe hơi có điều hòa, Xe hơi không điều hòa, Thời kỳ trong năm.
  - Phân tích bán hàng trên dịch vụ tour trọn gói bằng cách tính toán doanh thu theo các yếu tố: Thời gian của gói du lịch, Loại gói, Giá, Phạm vi đặt tour của khách hàng, Nhóm khách hàng, Thời kỳ trong năm.

# Phụ lục C: Scrum trên một trang giấy

## Các thành phần trong tiến trình Scrum

### Các vai trò



Product Owner:  
Thiết lập Độ ưu tiên



ScrumMaster:  
Quản lý tiến trình, tháo gỡ trở ngại

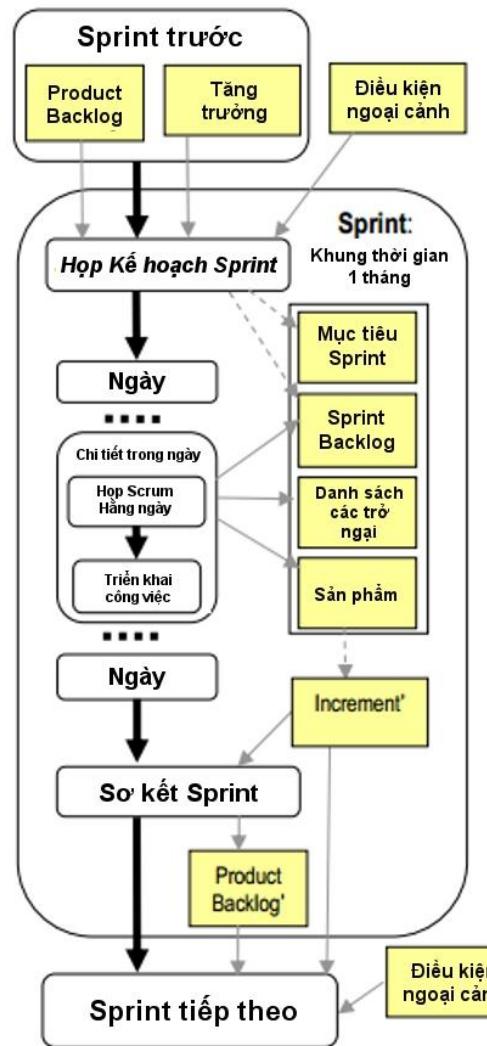


Nhóm Phát triển:  
Phát triển sản phẩm



Các bên liên quan:  
Quan sát và góp ý

### Tiến trình Sprint



### Các cuộc họp quan trọng

#### Sơ kết Sprint

- ScrumMaster tổ chức; tất cả phải có mặt. 1/2 ngày
- Không cần hình thức, mang lại thông tin
- Nghị trình:
  1. Demo và thảo luận về những phần tăng trưởng
  2. Xác định những điều cần cải tiến
  3. Thông báo về buổi Sơ kết Sprint tiếp theo

#### Hợp Kế hoạch Sprint

- ScrumMaster tổ chức; tất cả phải có mặt. 1/2 ngày
- Nội dung: Product Backlog, những tăng trưởng mới nhất, các điều kiện về công nghệ và nghiệp vụ
- Sản phẩm: Mục tiêu Sprint, Sprint Backlog
- Nghị trình:
  1. Product Owner trình bày các hạng mục ứng viên cho việc triển khai.
  2. Nhóm ước tính các hạng mục và dự toán việc triển khai.
  3. Nhóm chọn các hạng mục có độ ưu tiên cao nhất từ Product Backlog.
  4. Product Owner xác định Mục tiêu Sprint.
  5. Nhóm đưa các hạng mục vào Sprint Backlog mới

#### Hợp Scrum Hàng ngày

- Được tổ chức bởi ScrumMaster; tất cả có mặt; Những người liên quan là quan sát viên
- Thời gian là 15 phút, tổ chức hàng ngày vào cùng một thời điểm
- Mỗi thành viên trả lời 3 câu hỏi:
  1. Bạn đã làm gì ngày hôm qua?
  2. Bạn sẽ làm gì ngày hôm nay?
  3. Bạn gặp những khó khăn gì?
- Nhóm cập nhật Sprint Backlog; ScrumMaster cập nhật Danh sách các trở ngại

#### Hủy bỏ Sprint (hiếm khi xảy ra)

ScrumMaster kêu gọi nếu cảm thấy Nhóm không thể hoàn thành Mục tiêu Sprint.

VD: những thay đổi lớn về môi trường, không có khả năng xác định các trở ngại, sai lầm nghiêm trọng trong ước tính

### Các đồ nghề quan trọng

#### Product Backlog

- Danh sách các yêu cầu và vấn đề cần giải quyết
- Được sở hữu bởi Product Owner
- Bất cứ ai cũng có thể bổ sung thêm
- Chỉ Product Owner mới được đặt độ ưu tiên

#### Mục tiêu Sprint

- Một câu tổng kết
- Được đưa ra bởi Product Owner
- Một thỏa thuận giữa Nhóm và Product Owner
- Hoan nghênh các giải pháp sáng tạo

#### Sprint Backlog

- Danh sách các công việc
- Được sở hữu bởi Nhóm Phát triển
- Chỉ Nhóm mới được phép thay đổi

#### Danh sách các trở ngại

- Các trở ngại, khó khăn và quyết định chờ xử lý
- Sở hữu bởi ScrumMaster
- Được cập nhật hàng ngày

#### Tăng trưởng

- Phiên bản của sản phẩm
- Chức năng chuyển giao được (đã kiểm thử, đã viết tài liệu, v.v.)

#### Phản hồi trực quan

- "Bộ bức xạ thông tin"
- VD: Biểu đồ Burndown, sơ đồ kiến trúc

# Phụ lục D: Danh mục kiểm tra chất lượng triển khai Scrum

## Scrum tốt

- Ước tính được cập nhật hằng ngày
- Mọi người đều có mặt tại buổi Họp Scrum Hằng ngày
- Mọi người hỗ trợ lẫn nhau
- Mọi người yêu cầu trợ giúp
- Mọi người trình bày vấn đề với Nhóm và cùng nhau giải quyết chúng
- Có nhiều trao đổi và tương tác
- Nhiều vấn đề ngớ ngẩn được đưa ra bởi Nhóm

## Scrum tồi

- Sprint yêu cầu nhiều công việc hơn so với kế hoạch
- Thành viên Nhóm báo cáo một hạng mục vượt quá hai ngày so với ước tính mà không có ai chú ý hoặc quan tâm
- Không có tương tác nào khác ngoài buổi Họp Scrum Hằng ngày
- Product Owner không có mặt khi cần tham vấn
- Những phiền nhiễu từ bên ngoài
- Che dấu các công việc còn tồn đọng
- Chấp nhận hiện trạng
- Thất bại trong việc cung cấp Gói tăng trưởng Chuyển giao được sau mỗi Sprint

# Phụ lục E: Mở rộng quy mô

- Mở rộng quy mô thành Nhóm của Nhóm
- Các yếu tố mở rộng:
  - Loại ứng dụng
  - Kích cỡ Nhóm
  - Nhóm phân tán
  - Thời gian thực hiện dự án
- Scrum đã từng được triển khai cho một dự án với hơn 500 thành viên tham gia

# Phụ lục F: Scrum phân tán

- **Các loại Nhóm Scrum phân tán:**
  - **Scrum cô lập:** Nhóm bị cô lập về mặt địa lý
  - **Scrum phân tán của Scrum:** Nhóm Scrum bị cô lập về địa lý và được tích hợp bởi Scrum của Scrum
  - **Scrum tích hợp hoàn toàn:** Nhóm Scrum liên chức năng với các thành viên phân tán ở các khu vực địa lý khác nhau.
- **Khó khăn để chia sẻ các giá trị và mục tiêu**
- **[www.distributedscrumprimer.com](http://www.distributedscrumprimer.com)**

# Sách

