

Một số tiến trình phần mềm

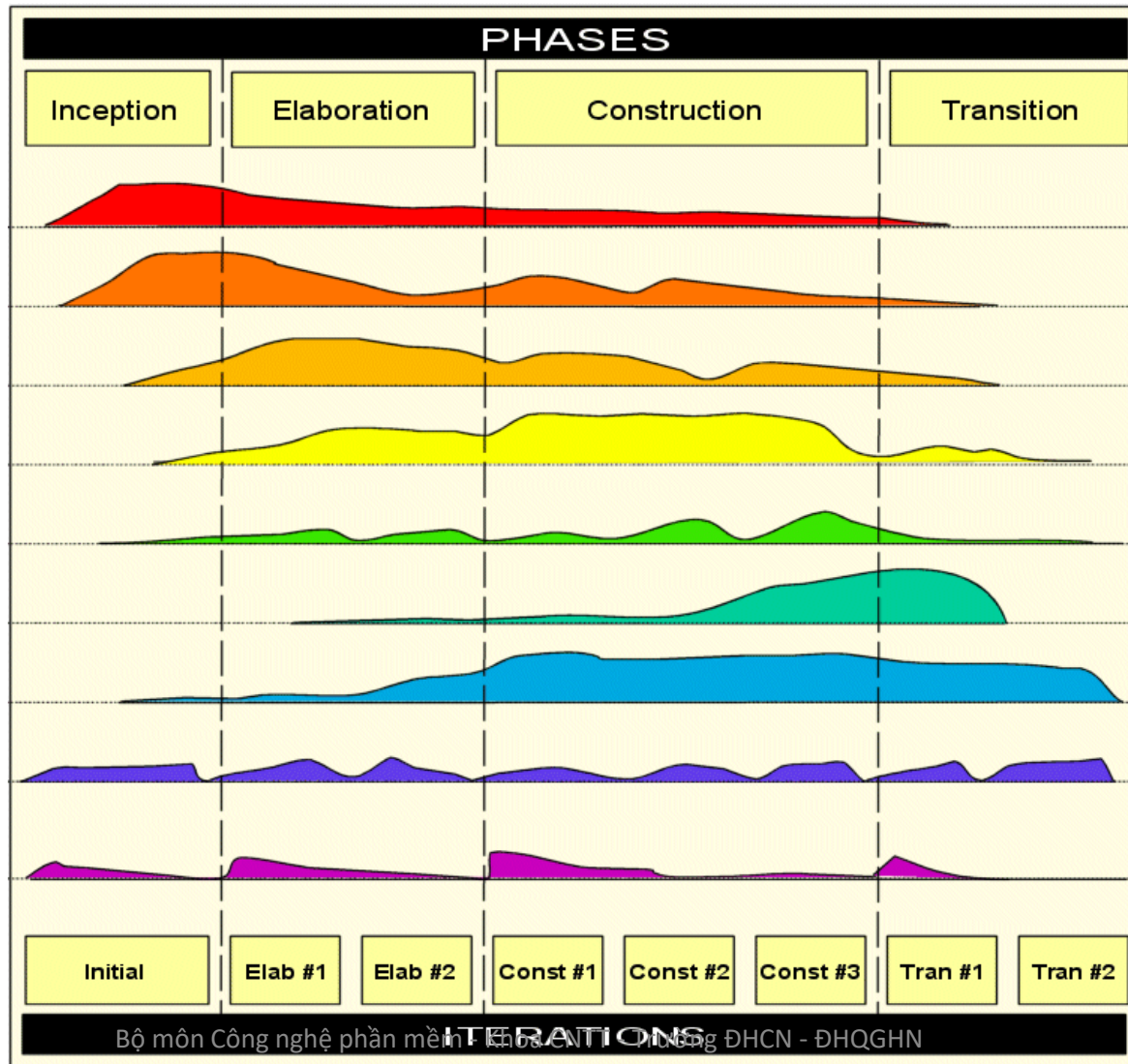
Các tiến trình phát triển phần mềm

1. RUP
2. Agile/XP, Scrum, ...

RUP – Rational Unified Process

- Rational Software, IBM
- Gắn với ngôn ngữ mô hình hóa thống nhất UML (Unified Modeling Language)
- Công cụ Rational Rose
- Kết hợp 3 mô hình tiến trình cơ bản: thác nước, tiến hóa và phát triển phần mềm dựa trên thành phần.
- Phân tách các pha (phases) với các bước (disciplines) phát triển
- Có pha chuyển giao, bố trí phần mềm trong môi trường khách hàng.

RUP – Rational Unified Process



RUP – Rational Unified Process

- 4 pha:
 - Inception: định nghĩa phạm vi dự án
 - Elaboration: lập kế hoạch, đặc tả các đặc tính và kiến trúc cơ bản
 - Construction: phát triển sản phẩm
 - Phát triển sản phẩm với nhiều lần lặp cho tới phiên bản bàn giao
 - Transition: chuyển giao sản phẩm cho người dùng.
 - Đào tạo, cài đặt, hỗ trợ người dùng

Inception

- Nhận ra các thực thể bên ngoài hệ thống: có thể là con người hoặc hệ thống khác, chúng sẽ tương tác với hệ thống.
- Định nghĩa những tương tác này
- Từ những thông tin trên, đánh giá mức độ đóng góp của hệ thống với nghiệp vụ
- Nếu đóng góp là không đáng kể thì dự án sẽ bị hủy bỏ.

Elaboration

- Định nghĩa mô hình yêu cầu:
 - Tập hợp các ca sử dụng (UML use cases) và mô tả chi tiết các ca sử dụng này
- Lên kế hoạch phát triển phần mềm
- Nhận ra các rủi ro

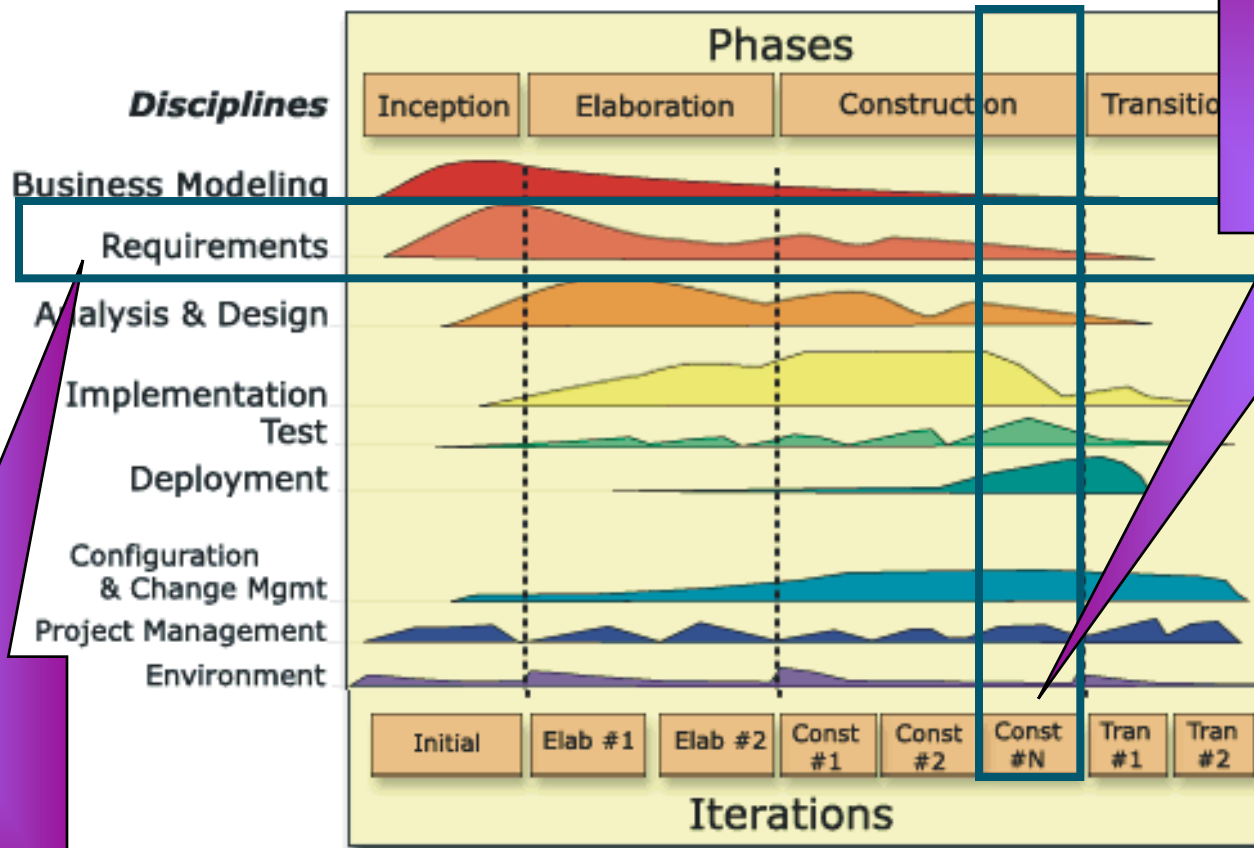
Construction

- Bao gồm các hoạt động thiết kế, lập trình và kiểm thử.
- Các phần của hệ thống được phát triển song song và được tích hợp ở pha này.
- Kết quả: chương trình vận hành được và các tài liệu sẵn sàng cho việc chuyển giao phần mềm.

Transition

- Chuyển giao phần mềm: di chuyển phần mềm từ môi trường phía đội phát triển sang môi trường vận hành thực tế của khách hàng.
- Đảm bảo phần mềm vận hành tốt trên môi trường của khách hàng.

RUP



In an iteration, you walk through all disciplines.

Disciplines group activities logically.

Business Modelling

- Các tiến trình nghiệp vụ được mô hình hóa, sử dụng các ca sử dụng nghiệp vụ (business use cases)
 - BPMN – Business Process Modeling Notation

Requirements

- Các tác nhân (actors) được nhận ra
- Các ca sử dụng được mô tả chi tiết để mô hình hóa (thể hiện) các yêu cầu của hệ thống.

Analysis and Design

- Mô hình thiết kế được tạo bằng các mô hình thành phần (component models), mô hình đối tượng (object models) và mô hình tuần tự (sequence models).

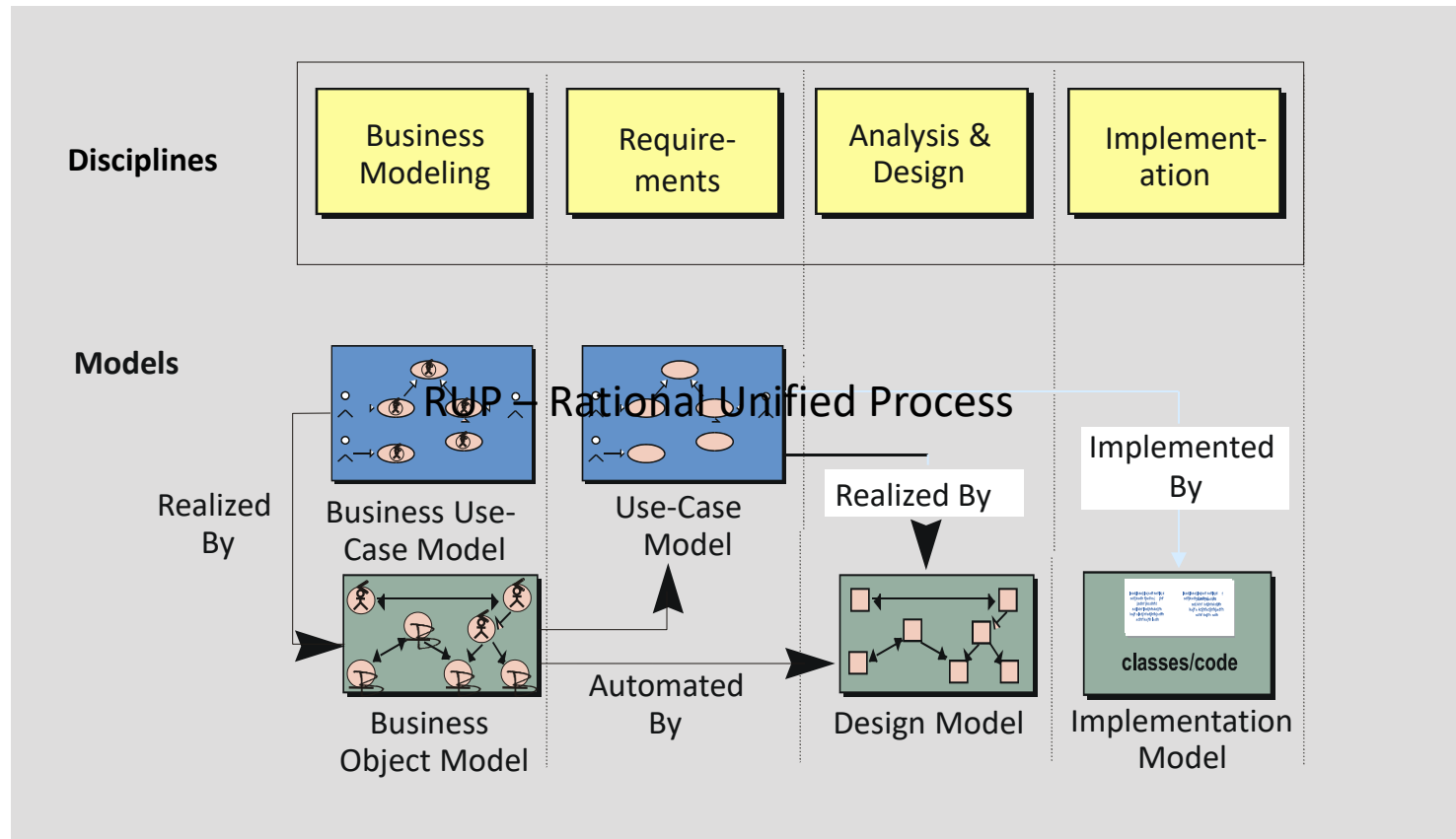
Implementation

- Các thành phần của hệ thống được cài đặt
- Phát sinh mã tự động từ thiết kế có thể được áp dụng ở đây.

Testing

- Kiểm thử được thực hiện lặp đi lặp lại cùng với hoạt động cài đặt.
- Kiểm thử mức hệ thống được thực hiện sau khi đã hoàn thành việc cài đặt.

Disciplines Produce Models



6 nguyên tắc thực hành của RUP

1. Phát triển phần mềm lặp đi lặp lại
 - Lên kế hoạch cho các bước lặp theo độ ưu tiên của yêu cầu.

6 nguyên tắc thực hành của RUP

1. Phát triển phần mềm lặp đi lặp lại
2. Quản lý yêu cầu
 - Làm tài liệu các yêu cầu
 - Lưu giữ vết của sự thay đổi yêu cầu
 - Phân tích tầm ảnh hưởng của sự thay đổi yêu cầu trước khi chấp nhận sự thay đổi đó.

6 nguyên tắc thực hành của RUP

1. Phát triển phần mềm lặp đi lặp lại
2. Quản lý yêu cầu
3. Sử dụng kiến trúc dựa trên thành phần
 - Cấu trúc hệ thống theo các thành phần.

6 nguyên tắc thực hành của RUP

1. Phát triển phần mềm lặp đi lặp lại
2. Quản lý yêu cầu
3. Sử dụng kiến trúc dựa trên thành phần
4. Mô hình phần mềm trực quan
 - Sử dụng UML để thể hiện đầy đủ khía cạnh tĩnh và động của hệ thống.

6 nguyên tắc thực hành của RUP

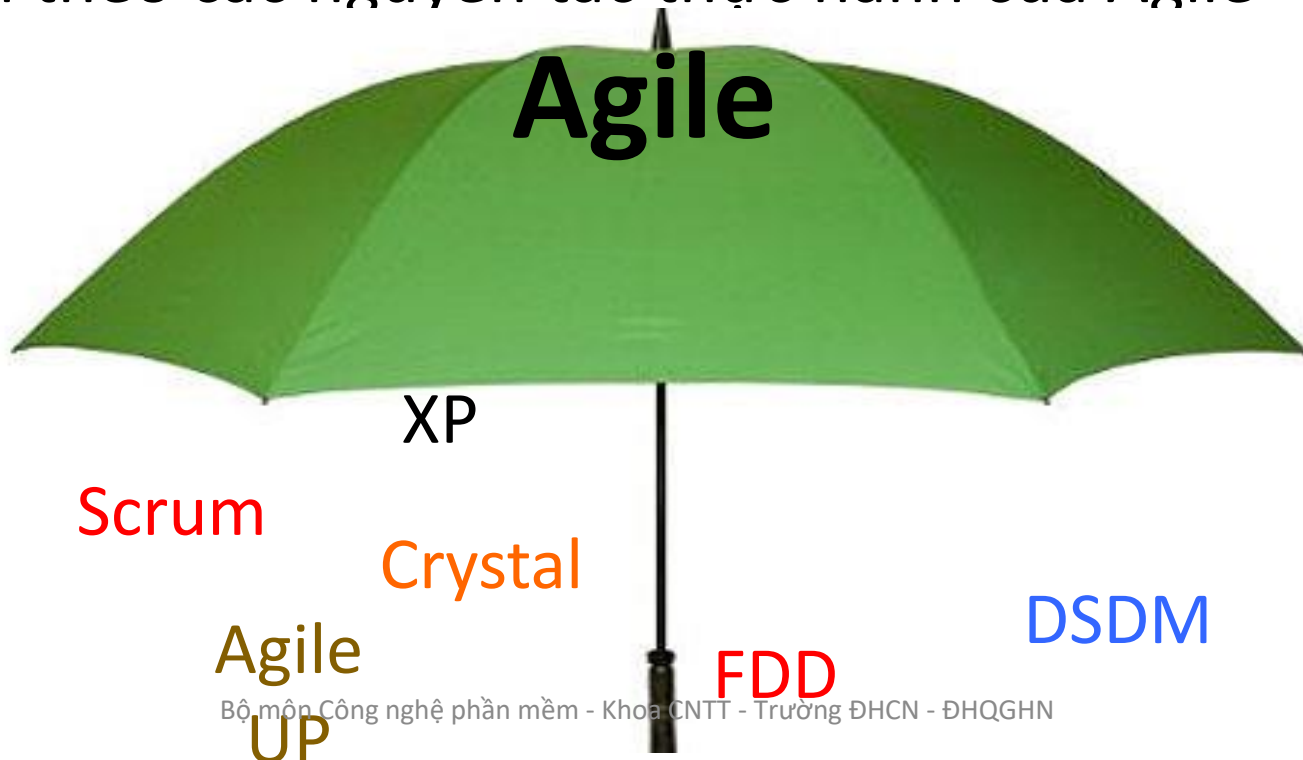
1. Phát triển phần mềm lặp đi lặp lại
2. Quản lý yêu cầu
3. Sử dụng kiến trúc dựa trên thành phần
4. Mô hình phần mềm trực quan
5. Kiểm chứng chất lượng phần mềm
 - Đảm bảo phần mềm đáp ứng chuẩn chất lượng.

6 nguyên tắc thực hành của RUP

1. Phát triển phần mềm lặp đi lặp lại
2. Quản lý yêu cầu
3. Sử dụng kiến trúc dựa trên thành phần
4. Mô hình phần mềm trực quan
5. Kiểm chứng chất lượng phần mềm
6. Kiểm soát sự thay đổi cho phần mềm
 - Sử dụng hệ thống quản lý thay đổi
 - Áp dụng các thủ tục, công cụ quản lý cấu hình hệ thống.

Agile method/ Scrum, XP,...

- Agile: tập các nguyên tắc thực hành cho phát triển phần mềm
- Scrum, XP, ... : các quy trình phát triển phần mềm tuân theo các nguyên tắc thực hành của Agile



Agile method – khung cảnh ra đời

- Doanh nghiệp vận hành trong môi trường thay đổi nhanh chóng và toàn cầu hóa.
- Đáp ứng các cơ hội mới, vấn đề thị trường, sự thay đổi điều kiện tài chính.
- Phần mềm là một phần vận hành của doanh nghiệp
- Phần mềm mới cần được phát triển nhanh để đáp ứng các cơ hội của doanh nghiệp.

Agile method – khung cảnh ra đời

- Định nghĩa một tập yêu cầu cố định cho một phần mềm là điều không thể.
- Các yêu cầu ban đầu chắc chắn sẽ thay đổi
- Khi yêu cầu thay đổi, thiết kế và cài đặt phải được làm lại và được kiểm thử lại
 - Phá vỡ lịch trình và chi phí dự kiến
- Tiến trình phát triển nhanh (rapid development process) phù hợp với các phần mềm nghiệp vụ
 - Dễ thích nghi với sự thay đổi yêu cầu.

Đặc trưng của Tiến trình phát triển nhanh

- Các hoạt động đặc tả, thiết kế, cài đặt được thực hiện xen kẽ.
- Không có đặc tả hệ thống chi tiết
- Tài liệu thiết kế được tối giản hoặc được phát sinh tự động bởi công cụ.
- Tài liệu yêu cầu người dùng chỉ mô tả các đặc điểm quan trọng nhất của hệ thống.

Đặc trưng của Tiến trình phát triển nhanh

- Hệ thống được phát triển với nhiều phiên bản.
- Người dùng và nhà đầu tư tham gia vào việc đặc tả và đánh giá mỗi phiên bản.
- Người dùng và nhà đầu tư có thể đề xuất sự thay đổi yêu cầu hoặc yêu cầu mới cho phiên bản tiếp theo.

Đặc trưng của Tiến trình phát triển nhanh

- Giao diện người dùng thường được phát triển nhanh
 - Sử dụng công cụ phát triển tương tác
 - Vẽ, kéo, thả, đặt các biểu tượng trên công cụ
 - Phát sinh tự động giao diện cho ứng dụng Web hoặc ứng dụng chạy trên nền hệ điều hành cụ thể.

Agile methods

- Phát triển tăng dần
- Phiên bản mới được làm trong vòng 2-3 tuần
- Khách hàng tham gia vào tiến trình phát triển
 - Phát biểu yêu cầu
 - Xếp thứ tự ưu tiên các yêu cầu
 - Phản hồi nhanh về sự thay đổi, bổ sung yêu cầu cho phiên bản tiếp theo.
- Hạn chế tài liệu, thay vào đó là các trao đổi thường xuyên giữa các thành viên đội phát triển với nhau và với khách hàng.

Agile manifesto – Tuyên ngôn của Agile

- *Individuals and interactions* over *processes and tools*
- *Working software* over *comprehensive documentation*
- *Customer collaboration* over *contract negotiation*
- *Responding to change* over *following a plan*

Just-in-time | Tức thì

				New Estimates of Effort Remaining as of Day...					
Product Backlog Item	Sprint Task	Volunteer	Initial Estimate of Effort	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								
	update buyer help webpage								
	...								
Improve transaction processing performance	merge DCP code and complete layer								
	complete machine order for pRank								
	change DCP and reader to use pRank								



Khó khăn khi áp dụng Agile methods

- Khách hàng có thể tham gia toàn bộ thời gian với đội phát triển không?
- Các thành viên trong đội có thể cộng tác với nhau tốt đẹp không?
- Xếp thứ tự ưu tiên cho các yêu cầu là khó, đặc biệt khi mà có nhiều nhà đầu tư
- Dưới áp lực về lịch chuyển giao sản phẩm, đội phát triển khó xây dựng được phần mềm có cấu trúc tốt và đơn giản, dễ hiểu.
- Khó khăn khi thay đổi văn hóa làm việc từ quy trình phát triển truyền thống sang Agile.
- Tightly-integrated teams (đội phát triển cộng tác chặt chẽ với nhau): nếu thành viên đội rời nhóm?

Khả năng áp dụng Agile method

- Các sản phẩm phần mềm vừa và nhỏ được phát triển để bán.
- Phần mềm phức tạp?
- Phần mềm có yêu cầu về tính an toàn, an ninh cao?

Agile methods and software maintenance

- Hầu hết các tổ chức phát triển phần mềm tiêu tốn phần lớn thời gian vào giai đoạn bảo trì phần mềm hơn là phát triển phần mềm mới:
 - Agile method có hỗ trợ bảo trì?
 - Đề cao việc viết mã có cấu trúc tốt và dễ đọc, đầu tư công sức vào cải tiến mã.
 - Tối thiểu hóa tài liệu trong Agile method
 - Mô tả yêu cầu theo cách phi hình thức và không tạo tài liệu yêu cầu nhất quán.

eXtreme Programming

XP – lập trình cực đoan

Extreme programming

- Được sử dụng rộng rãi nhất trong Agile methods
- Phát triển tăng dần, với nhiều phiên bản
- Khách hàng tham gia toàn bộ thời gian với đội phát triển
- Lập trình đôi
- Chấp nhận sự thay đổi ở các phiên bản

Requirements

- Khách hàng hoặc người dùng là một phần của đội phát triển
 - Ra quyết định về yêu cầu
- Yêu cầu người dùng được biểu thị bằng kịch bản (scenarios) or câu chuyện người dùng (**user stories**).
 - Viết trên các tấm thẻ
 - Đội phát triển phân chia yêu cầu thành các nhiệm vụ (implementable tasks).
 - Tasks là cơ sở để ước lượng và lập lịch
- Khách hàng chọn các stories sẽ được thực thi ở phiên bản tiếp theo
 - Dựa trên độ ưu tiên và ước lượng lịch trình.

Scenarios

Scenario: Transferring money to a savings account

Given my Current account has a balance of 1000.00

And my Savings account has a balance of 2000.00

When I transfer 500.00 from my Current account to my Savings account

Then I should have 500.00 in my Current account

And I should have 2500.00 in my Savings account

User stories

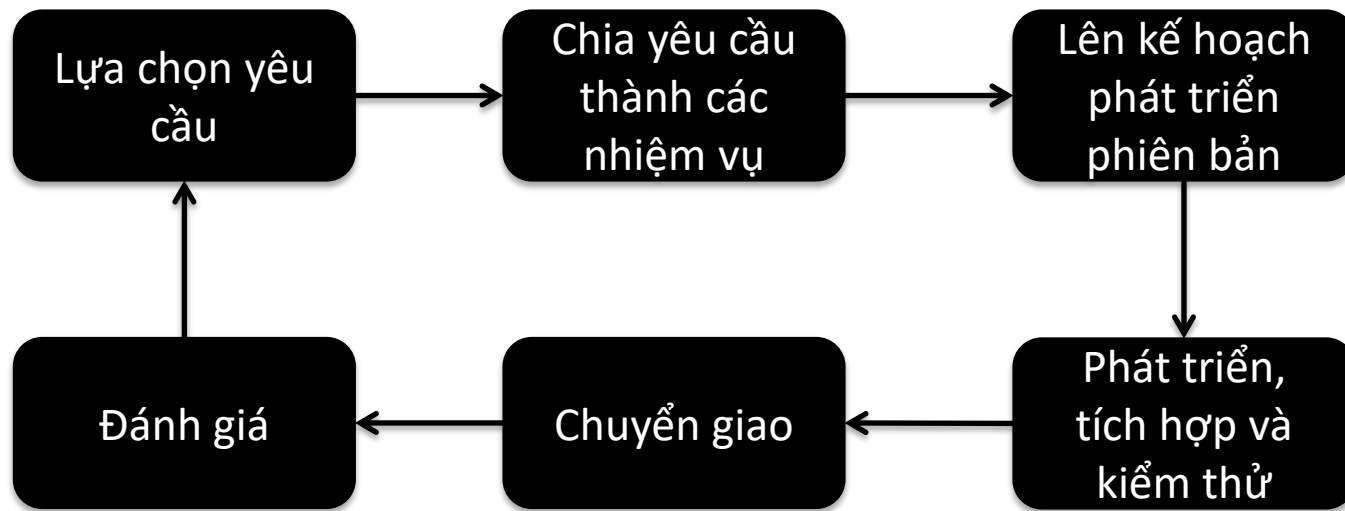
- Template

"As a <role>, I want <goal/desire> so that <benefit>"

- Examples

As a customer representative, I want to search for my customers by their first and last name.

Các bước trong vòng đời phát triển và chuyển giao các phiên bản với XP



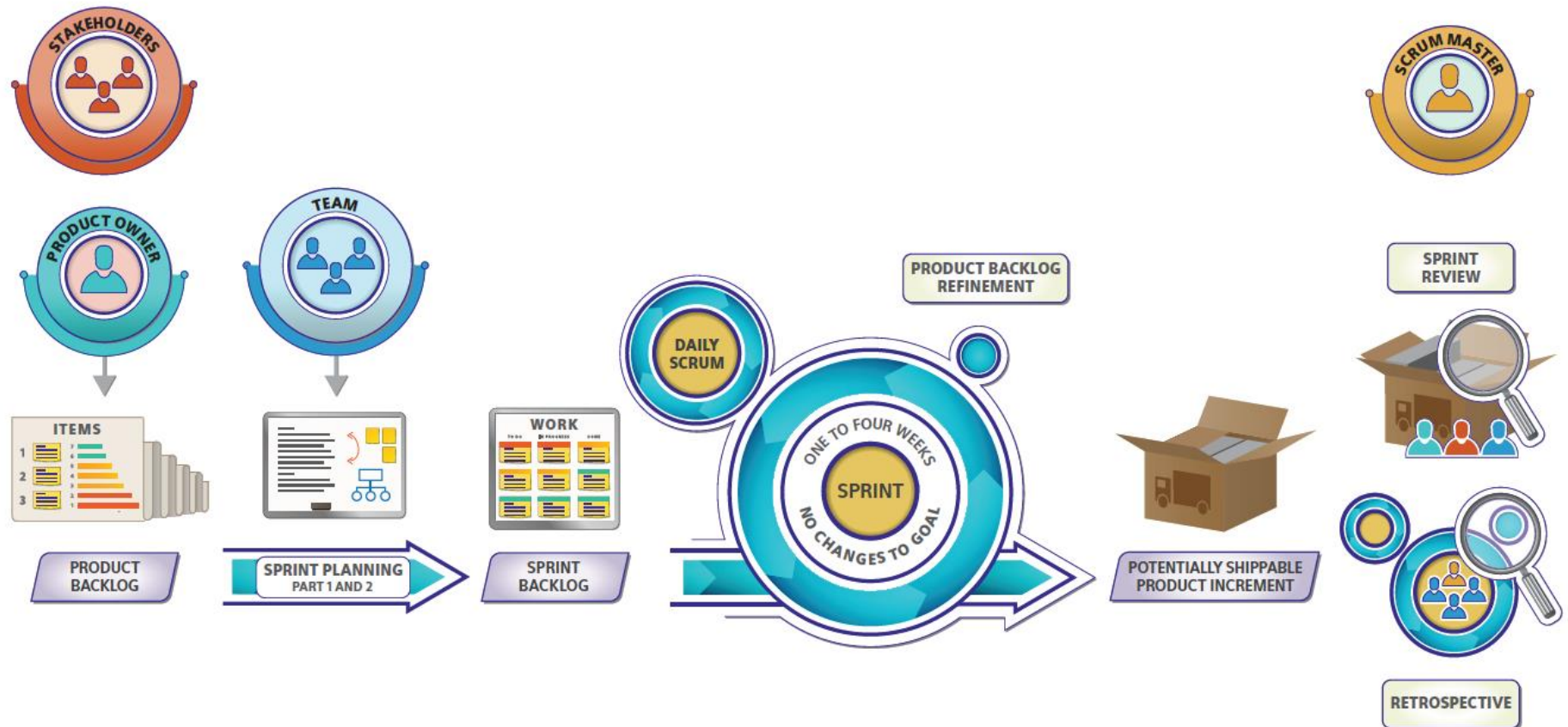
Pair programming



- Lập trình viên làm việc theo cặp (không cố định)
 - Ngồi với nhau trên một máy, cùng viết mã.
 - Cùng sở hữu và có trách nhiệm với mọi đoạn mã nguồn và chia sẻ tri thức trong toàn đội.
 - Mọi lập trình viên đều hiểu bất kỳ đoạn mã nào -> thuận lợi cho việc cải tiến mã nguồn.
- Hiệu suất cho lập trình đôi là tương đương như làm việc độc lập.

Scrum

Các bước thực hiện

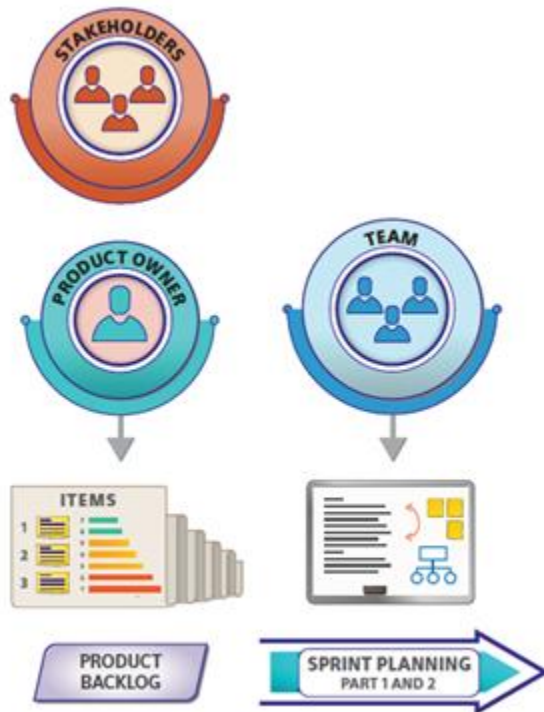


Các bước thực hiện



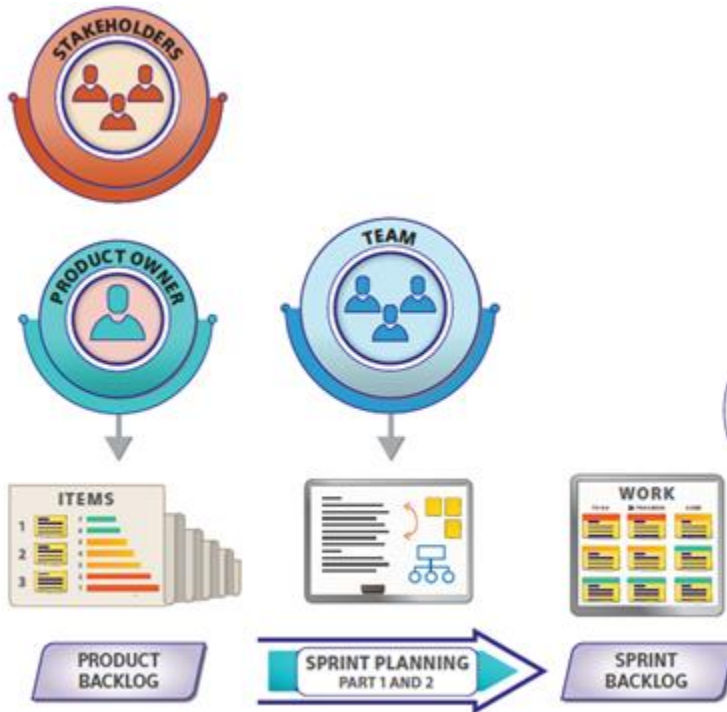
- Stakeholders, product owner đề xuất yêu cầu phát triển phần mềm
- Các yêu cầu cho phần mềm được phát biểu dưới dạng product backlog

Các bước thực hiện



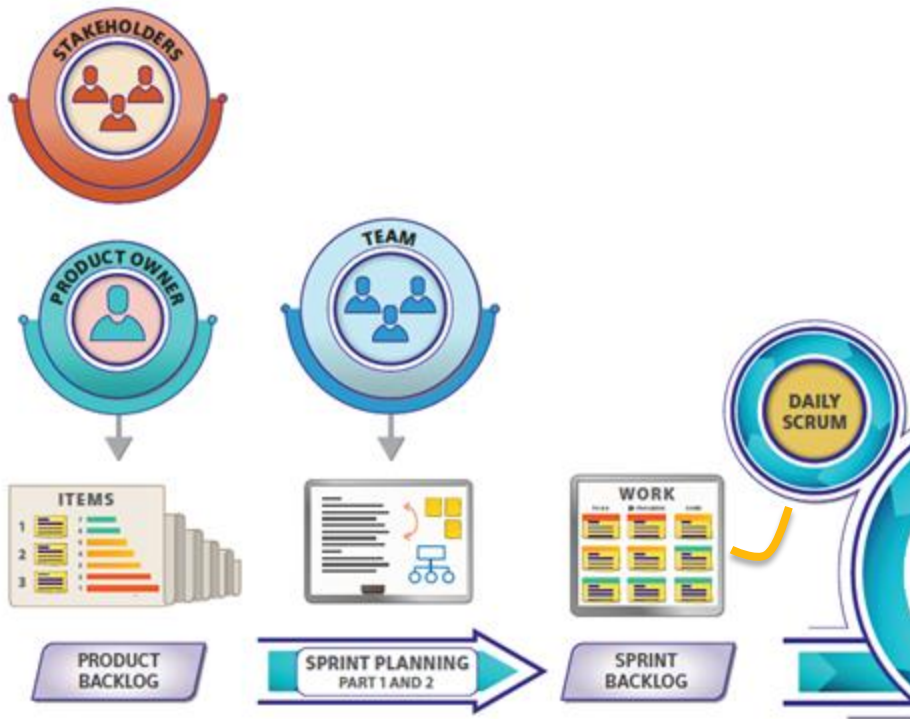
- Team (đội phát triển) ngồi với nhau để lập kế hoạch cho Sprint
 - Xác định các yêu cầu sẽ được hiện thực trong tháng/tuần tới
 - Mục tiêu: một vài chức năng hoàn chỉnh để chuyển giao vào cuối tháng/tuần tới.

Các bước thực hiện



- Kết quả của buổi họp kế hoạch
 - Mục tiêu
 - Những việc cần làm trong tháng/tuần tới (sprint backlog)

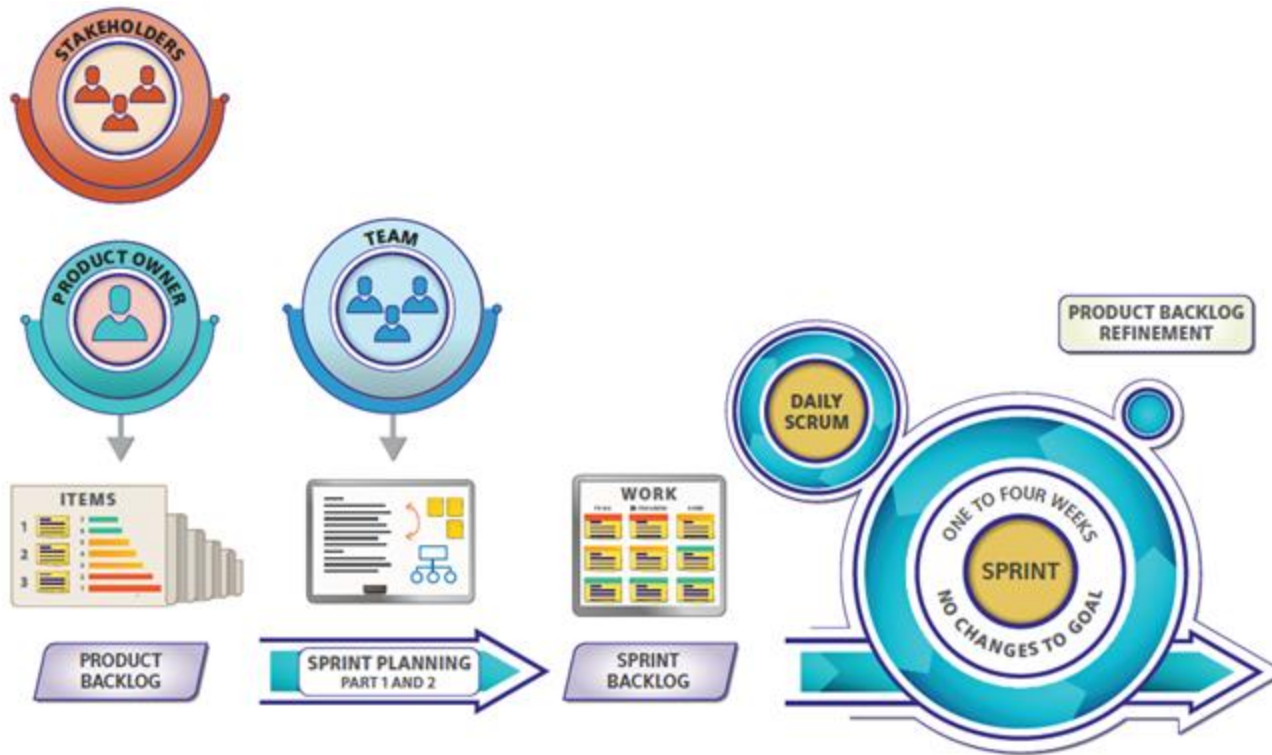
Các bước thực hiện



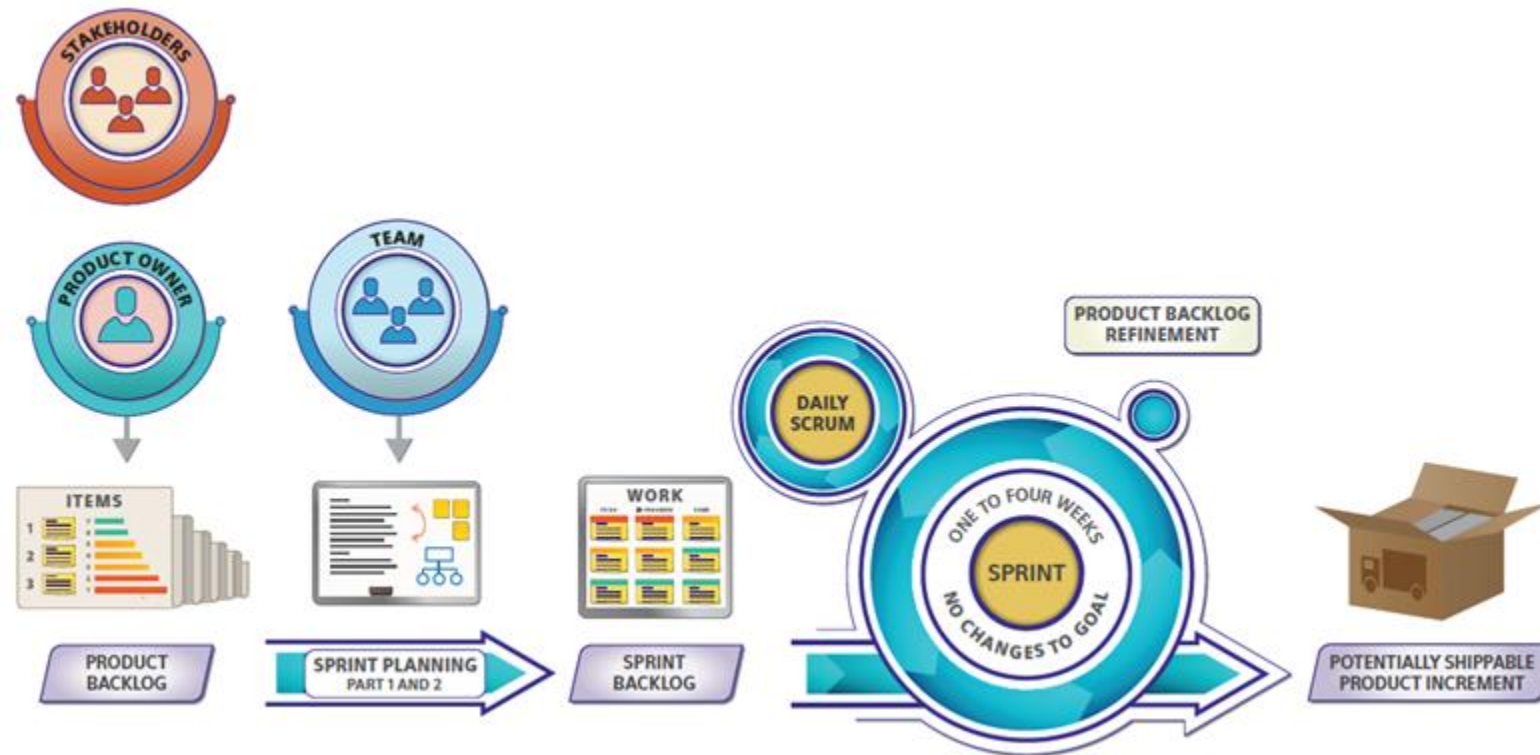
- Thành viên đội làm việc mình được giao và cộng tác chặt chẽ với các thành viên khác
- Họp hàng ngày: nắm tiến độ, phát hiện trở ngại nếu có.
- Có thể thêm việc, bớt việc: cập nhật bản kế hoạch.

Các bước thực hiện

- Cho đến hết khung thời gian của Sprint (1 tháng/tuần)

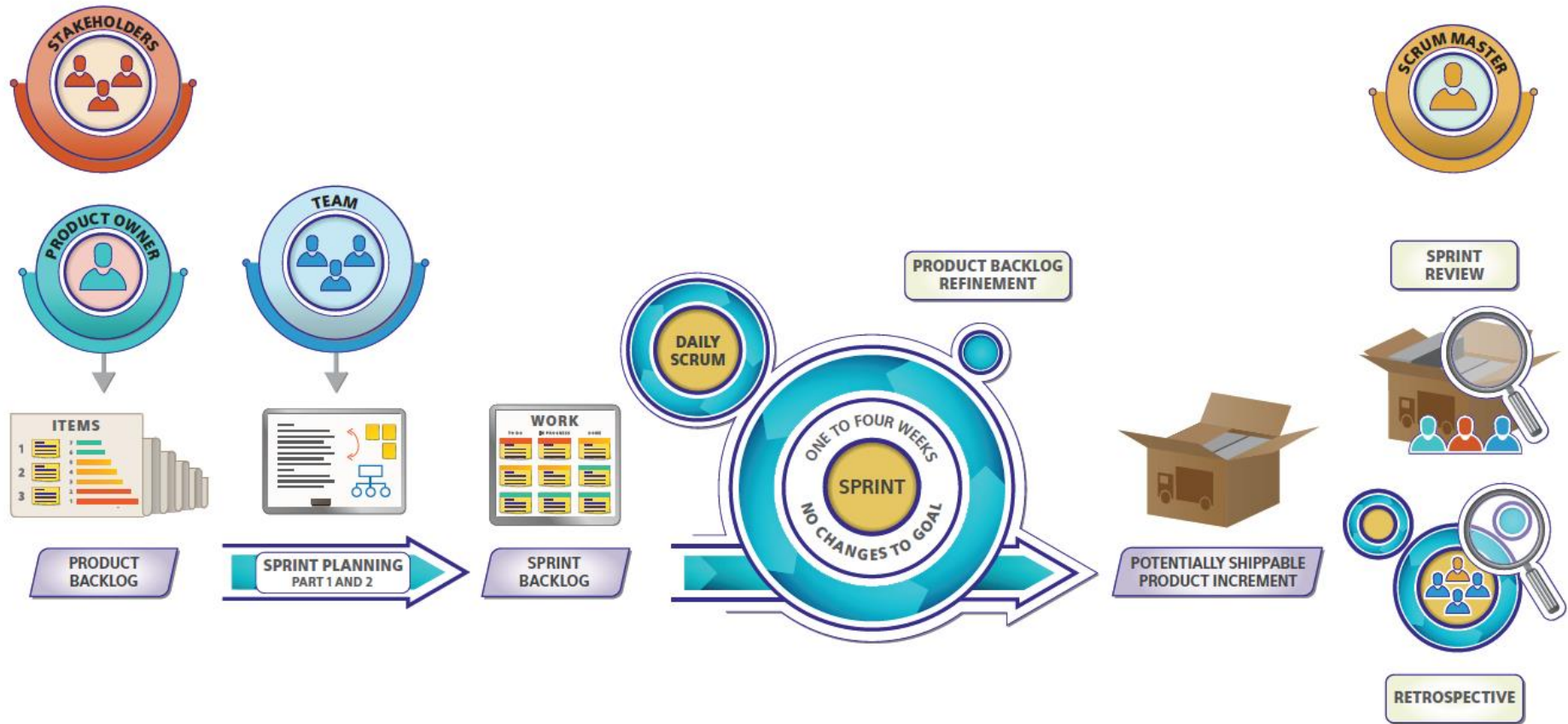


Các bước thực hiện



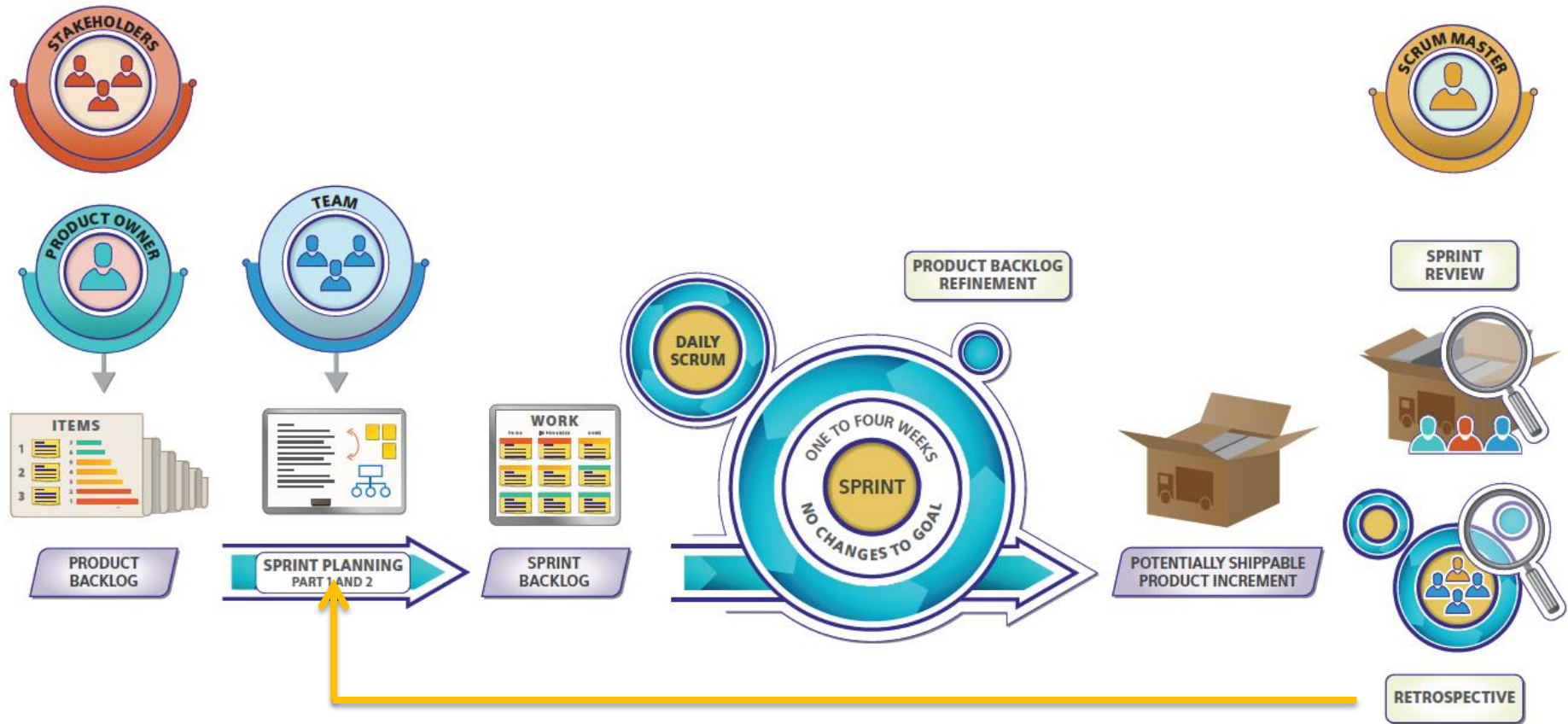
- Cho đến hết khung thời gian của Sprint (1 tháng/tuần)
→ phần mềm chạy tốt ra đời

Các bước thực hiện



- Cùng ngồi với nhau để xem lại
 - Đội phát triển đã làm việc tốt trong Sprint vừa rồi?
 - Tìm ra điều cần cải tiến

Các bước thực hiện



- Rồi lại quay lại để bắt đầu Sprint tiếp theo

Tools

- Use online tool
 - www.agilebench.com/
 - www.agile-tools.net

Tổng kết

- RUP
- Agile
 - XP
 - Scrum