TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

NHẬP MÔN CNPM

Nội dung / Chương 4: Phát triển Agile

Thông tin GV

Phương pháp phát triển linh hoạt

(Agile Development)

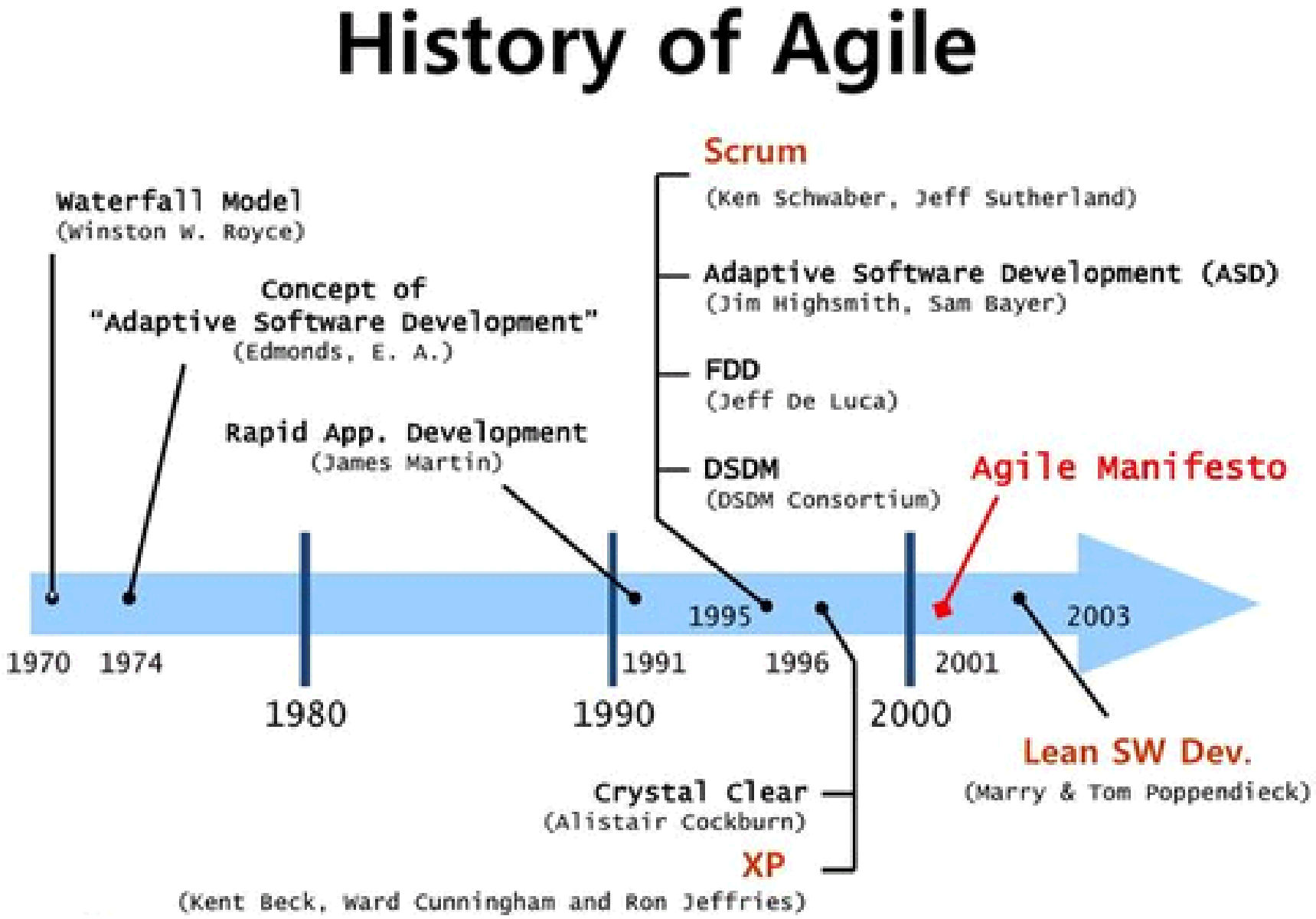
- **Agile software development (phương thức phát triển phần mềm linh hoạt) bắt đầu xuất hiện vào đầu những năm 90 với mục tiêu là phần mềm phải có khả năng biến đổi, phát triển và tiến hóa theo thời gian mà không cần phải làm lại từ đầu.**
- **Phương thức này tập chung vào tính đơn giản: tạo ra một phần mềm thật đơn giản đáp ứng đúng yêu cầu của khách hàng hôm nay và sẵn sàng cho những thay đổi vào ngày mai.**
- **Phương pháp Agile cố gắng cực tiểu hoá rủi ro bằng cách phát triển phần mềm trong những khung thời gian ngắn.**
  - **Mỗi bước lặp (iteration) giống như phát triển một phần mềm hoàn chỉnh cũng gồm có lấy yêu cầu, làm phân tích thiết kế, code, test, viết tài liệu.**

Phương pháp phát triển linh hoạt

(Agile Development)

- **Tập trung vào bước của chu trình phát triển và hạn chế việc sử dụng mô hình hóa, xây dựng tài liệu của các bước trung gian**
- **Phát triển ứng dụng đơn giản, lặp đi lặp lại**
- **Sử dụng eXtreme Programming (XP)**

Lịch sử của Agile



Tuyên ngôn dành cho phát triển phần mềm theo quy trình Agile

"Chúng tôi đã phát hiện ra cách tốt hơn để phát triển phần mềm bằng cách làm nó và giúp đỡ người khác làm việc đó. Thông qua việc này chúng tôi đã đem đến những giá trị:

- *Các cá nhân và sự tương tác hơn là các quá trình và công cụ*
- *Phần mềm hoạt động được hơn là các tài liệu đầy đủ*
- *Cộng tác với khác hàng hơn là thương lượng hợp đồng.*
- *Phản hồi các thay đổi hơn là tuân thủ theo kế hoạch*

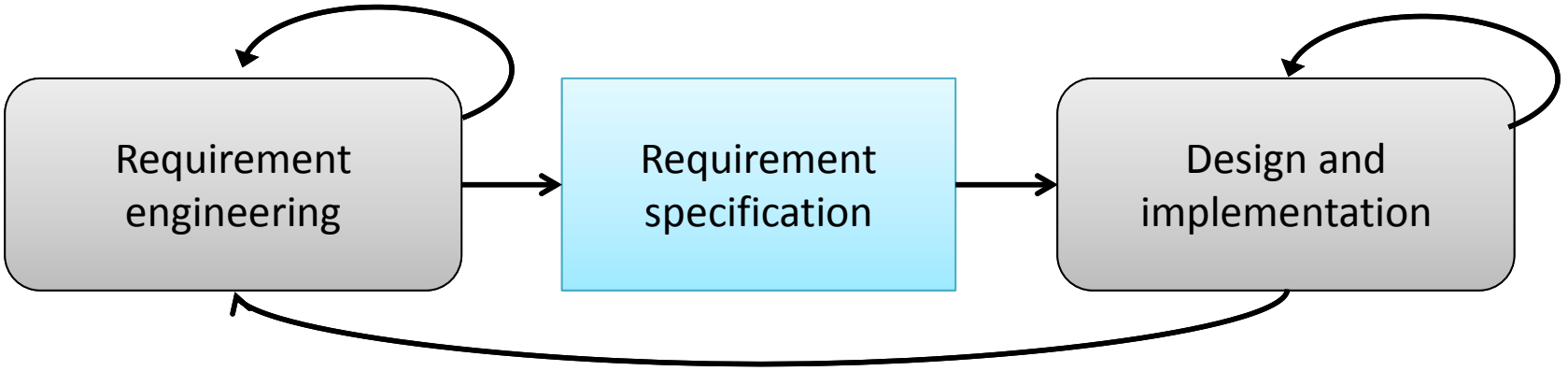
Đó là, mặc dù các mục bên phải có giá trị, chúng tôi đánh giá cao hơn những mục bên trái."

Kent Beck et al

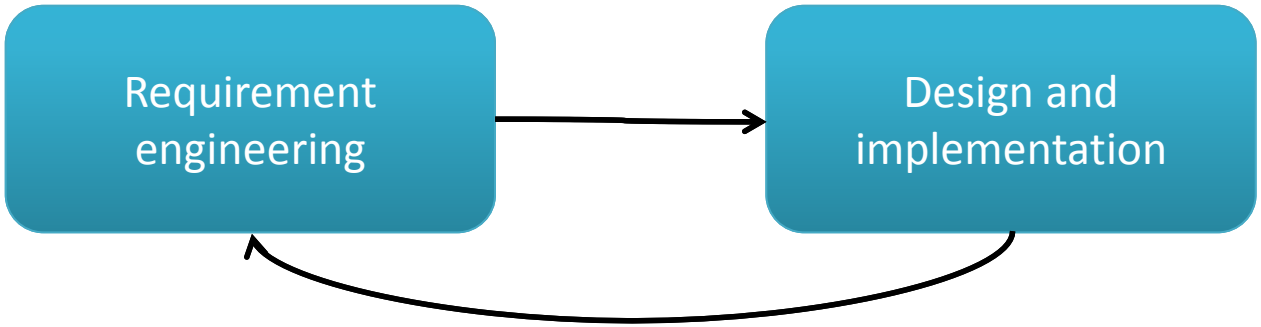
Phát triển theo kế hoạch và phát triển agile

(Agile Development)

Plan-based development



Agile development



Phát triển theo kế hoạch và phát triển agile

(Agile Development)

- **Phát triển theo kế hoạch**
  - **Các giai đoạn phát triển tách biệt**
    - Sản phẩm của mỗi giai đoạn được lập kế hoạch từ trước.
  - **Không nhất thiết chỉ dành cho mô hình thác nước, mô hình phát triển tăng dần cũng dùng được**
    - lặp xảy ra bên trong mỗi hoạt động.
- **Phát triển agile**
  - **Các hoạt động phát triển được thực hiện xen kẽ**
  - **Sản phẩm cần thực hiện được quyết định bởi thương thuyết trong quá trình phát triển.**

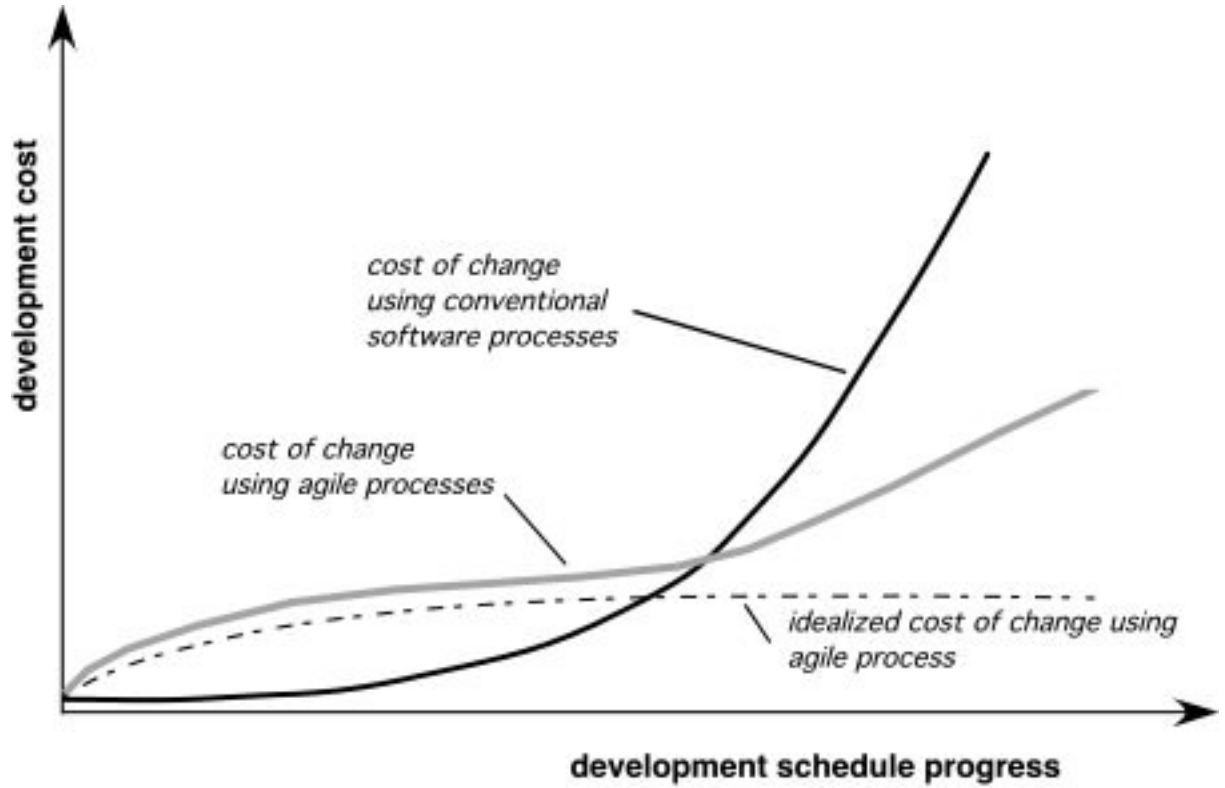
“Agility” là gì?

- Hiệu quả (nhanh chóng, thích ứng) đáp ứng được thay đổi.
- Giao tiếp hiệu quả giữa các bên liên quan.
- Đưa khách hàng vào trong team của mình.
- Tổ chức thành một team để có thể kiểm soát các công việc thực hiện.

Năng suất…

- Nhanh chóng, gia tăng cung cấp phần mềm

Agility và Chi phí của sự thay đổi



http://bit.ly/2ihOLB4 Slides copyright 2009 by Roger Pressman.

Một quy trình Agile

- Được điều khiển bởi những mô tả của khách hàng về những yêu cầu của mình (kịch bản).
- Nhận ra rằng các kế hoạch là ngắn hạn.
- Phát triển phần mềm lặp đi lặp lại, nhấn mạnh vào hoạt động xây dựng.
- Cung cấp nhiều bản ‘phần mềm tăng dần’.
- Thích nghi khi thay đổi xảy ra.

Quy tắc Agility - I

1. Ưu tiên cao nhất của dự án là thỏa mãn khách hàng bằng việc bàn giao sản phẩm sớm và liên tục.
2. Hoan nghênh các thay đổi từ phía khách hàng, kể cả các thay đổi vào giai đoạn cuối. Quy trình Agile khai thác những thay đổi cho lợi thế cạnh tranh của khách hàng.
3. Bàn giao sản phẩm theo chu kì từ vài tuần đến vài tháng. Chu kì ngắn tốt hơn chu kì dài.
4. Các nhân viên hiểu nghiệp vụ và các lập trình viên phải làm việc cùng nhau hàng ngày.
5. Tổ chức dự án xoay quanh những cá nhân tích cực. Cung cấp môi trường làm việc, hỗ trợ và tin tưởng họ để họ hoàn thành công việc.
6. Phương pháp giao tiếp tốt nhất trong đội dự án là gặp mặt trực tiếp.

Quy tắc Agility - II

7. Các chức năng đã hoạt động là thước đo chính cho tiến độ dự án.
8. Khuyến khích phát triển bền vững: Lập trình viên, người dùng, nhà quản lý phải có khả năng tham gia dự án một cách liên tục.
9. Liên tục cải tiến chất lượng thiết kế và mã nguồn.
10. Tính đơn giản giữ vai trò cốt yếu. Làm càng ít càng tốt.
11. Những yêu cầu và thiết kế tốt nhất được nảy nở từ những nhóm làm việc tự chủ.
12. Sau những khoảng thời gian nhất định, đội dự án xem xét cách thức cải tiến hiệu quả công việc.

Quy tắc Agile các nét chính

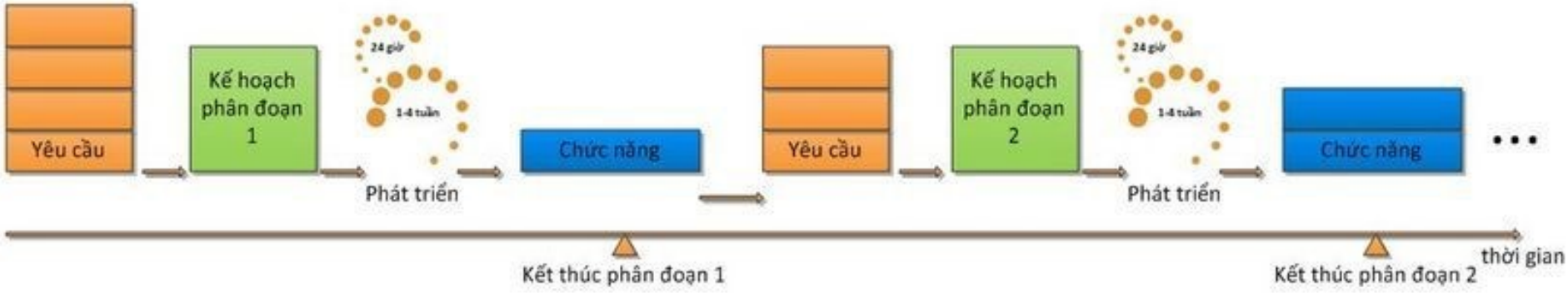
Nguyên tắc	Miêu tả
Khách hàng tham gia	Khách hàng nên tham gia chặt chẽ trong suốt quá trình phát triển. Vai trò của họ là cung cấp và quy định mức độ ưu tiên các yêu cầu mới về hệ thống, và đánh giá hệ thống tại các lần lặp (iterations of the system).
Chuyển giao tăng dần	Phần mềm được phát triển một cách tăng dần từng đợt (increment), trong đó khách hàng chỉ ra các yêu cầu cần được đưa vào mỗi đợt.
Con người thay vì quy trình	Kĩ năng của đội phát triển cần được ghi nhận và khai thác. Các thành viên của đội cần được tự do phát triển cách làm việc của riêng mình mà không cần đến các quy trình quy phạm định trước.
Chấp nhận thay đổi	Hiểu rằng yêu cầu hệ thống sẽ thay đổi nên thiết kế hệ thống sao cho nó có thể chấp nhận được các thay đổi đó.
Gìn giữ tính giản dị dễ hiểu	Chú trọng vào tính giản dị dễ hiểu của phần mềm đang được phát triển cũng như của quy trình phát triển. Chủ động nỗ lực loại bỏ sự phức tạp ra khỏi hệ thống bất cứ khi nào có thể.

Nhân tố con người

- *Các quy trình mẫu dành cho nhu cầu của con người và team,* không phải là dành cho các cách thức bên ngoài
- Những điểm quan trọng phải có giữa những người trong 1 team Agile & bản thân cả nhóm:
  - Năng lực.
  - Tập trung vào cái chung.
  - Hợp tác.
  - Năng lực quyết định.
  - Năng lực giải quyết vấn đề mở.
  - Sự tin tưởng và tôn trọng lẫn nhau.
  - Tự giác.

Đặc trưng của Agile

- *Tính lặp (Iterative):* Dự án sẽ được thực hiện trong các phân đoạn lặp đi lặp lại.



- *Tính tăng trưởng* (Incremental) và *tiến hóa* (Evolutionary)
- *Tính thích nghi* (adaptive)
- *Nhóm tự tổ chức và liên chức năng*
- *Quản lý tiến trình thực nghiệm* (Empirical Process Control)
- *Giao tiếp trực diện* (face-to-face communication)
- *Phát triển dựa trên giá trị* (value-based development)

Agile methods

- **Khả năng ứng dụng phương pháp agile**
  - **Phát triển phần mềm dùng chung, trong đó một công ty phần mềm đang phát triển một sản phẩm cỡ nhỏ-trung bình để bán.**
  - **Phát triển phần mềm đặt hàng (custom system development) trong phạm vi một tổ chức, trong đó**
    - có một cam kết rõ ràng từ khách hàng về việc tham gia quy trình phát triển
    - có không nhiều các quy tắc và quy định từ bên ngoài có ảnh hưởng tới phần mềm.
  - **Do sự chú trọng vào các nhóm nhỏ làm việc một cách gắn kết, có vấn đề trong việc mở rộng quy mô của các phương pháp agile cho các hệ thống lớn.**

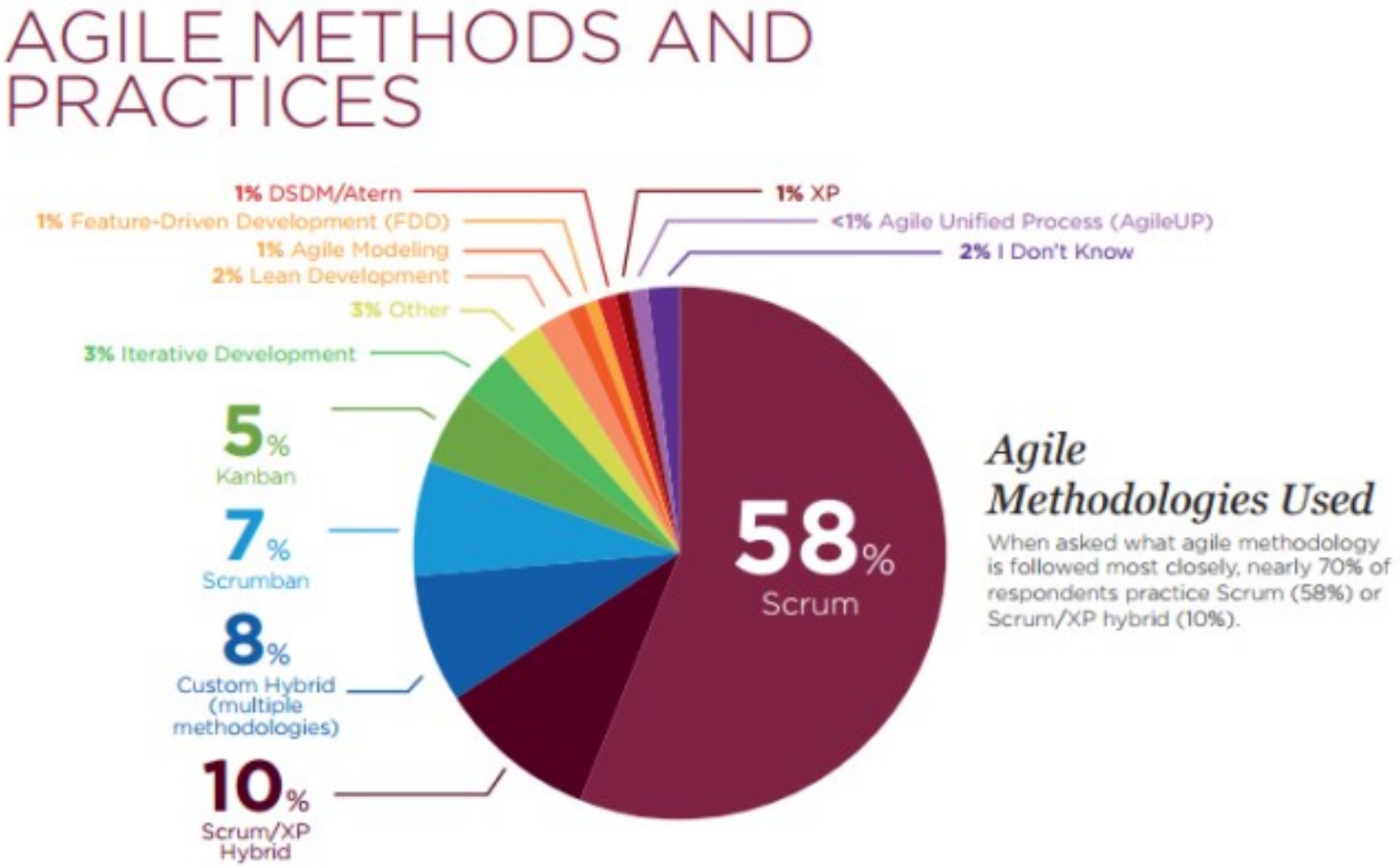
Agile methods

- **Vấn đề với các phương pháp agile**
  - Có thể khó giữ sự quan tâm của khách hàng tham gia quy trình.
  - Các thành viên trong đội có thể không phù hợp với cường độ làm việc đặc thù của các phương pháp agile.
  - Khi có nhiều hơn một người có quyền xác định mức độ ưu tiên của các yêu cầu, có thể khó thay đổi mức độ ưu tiên đó.
  - Việc gìn giữ tính giản dị dễ hiểu cũng đòi hỏi công sức.
  - Cũng như các phương pháp phát triển lặp khác, hợp đồng có thể là vấn đề.

Agile methods

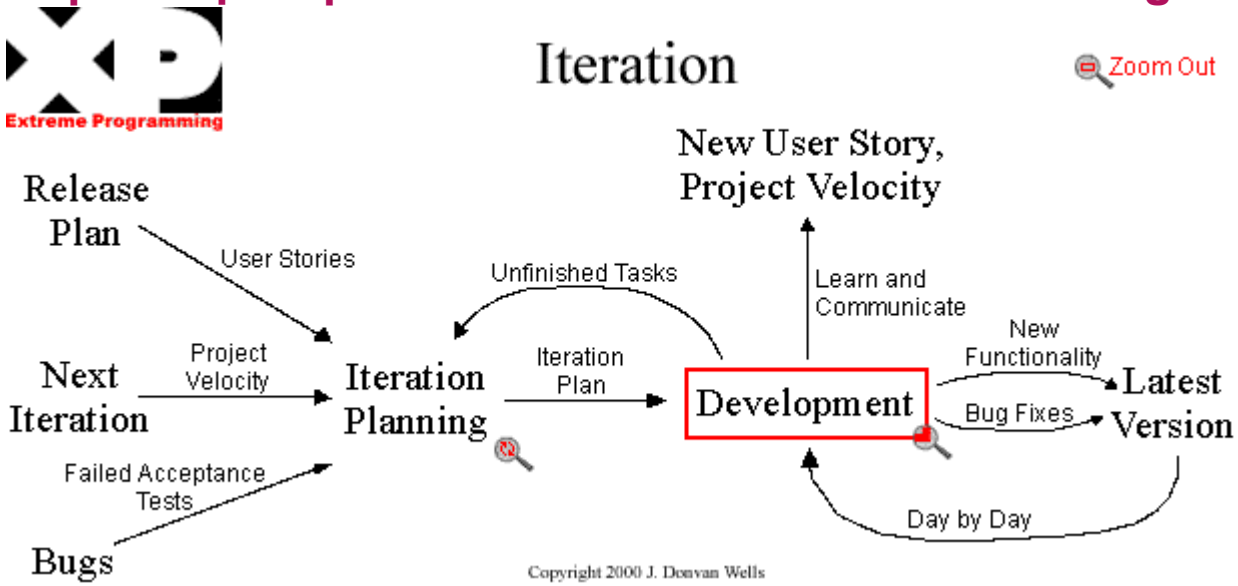
- Agile Modeling
- Agile Unified Process (AUP)
- Dynamic Systems Development Method (DSDM)
- Essential Unified Process (EssUP)
- Extreme Programming (XP)
- Feature Driven Development (FDD)
- Open Unified Process (OpenUP)
- Scrum
- Velocity tracking

Agile methods



Extreme programming

- **Phương pháp agile nổi tiếng nhất và được sử dụng rộng rãi nhất.**
- **Extreme Programming (XP) có một cách tiếp cận 'cực đoan' đối với hoạt động phát triển vòng lặp.**
  - **Các phiên bản mới có thể được xây dựng vài lần mỗi ngày;**
  - **Hai tuần một lần chuyển giao sản phẩm đợt mới (increment) cho khách hàng;**
  - **Phải chạy tất cả các test cho từng phiên bản (build) và một phiên bản chỉ được chấp nhận nếu tất cả các test đều thành công.**



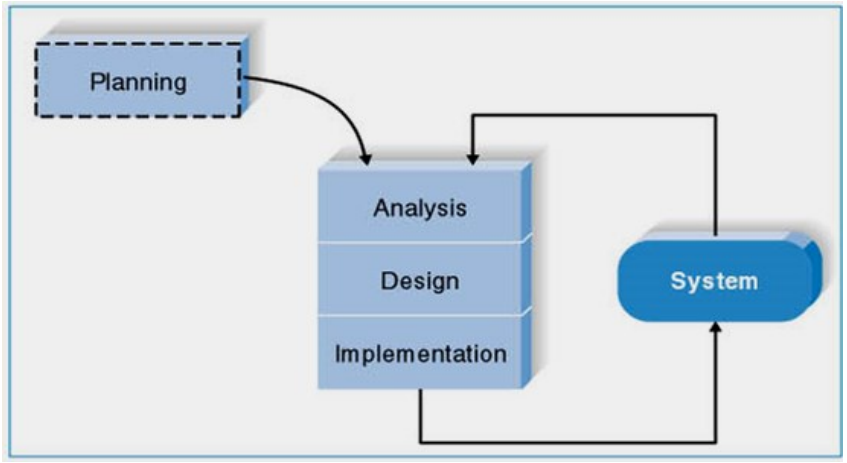
Extreme Programming (XP)

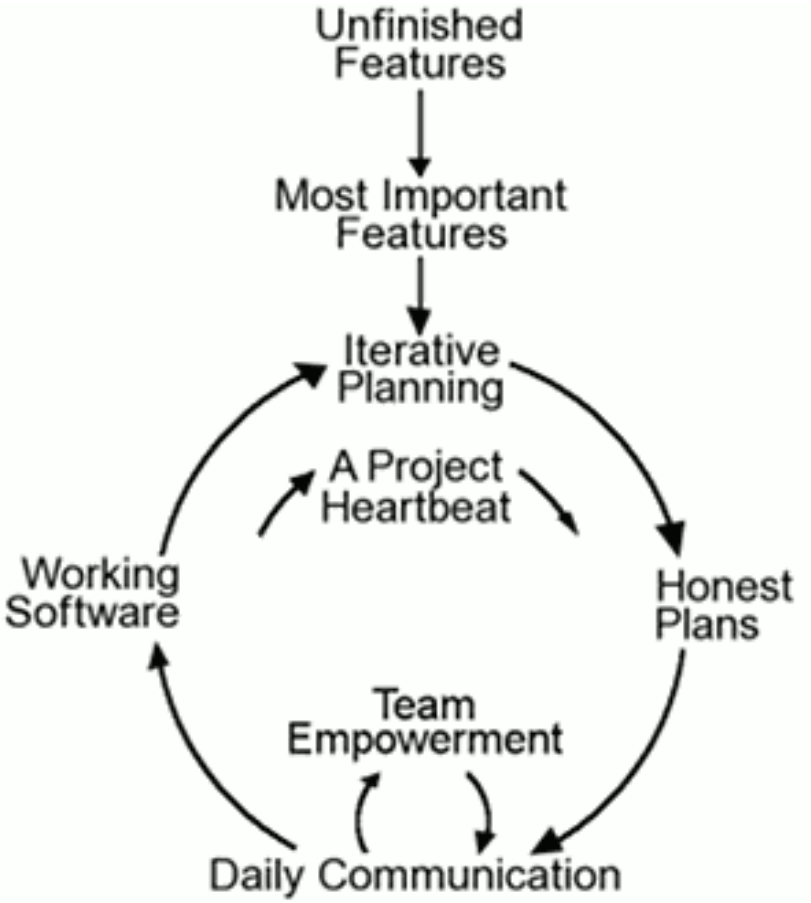
- Là quy trình Agile được sử dụng rộng rãi nhất, ban đầu được đề xuất bởi Kent Beck.
- Kế hoạch XP
  - Bắt đầu với việc tạo ra "user stories" (giống như use case)
  - Agile team đánh giá từng story và gán cho nó 1 chi phí (cost).
  - Các story được nhóm lại thành 1 một Chuyển giao tiếp theo
  - Một cam kết được thực hiện vào ngày chuyển giao kết quả.
  - Sau Chuyển giao đầu tiên "project velocity" được sử dụng để xác định số ngày cho những Chuyển giao kết quả tiếp theo.

eXtreme-Programming -based

(Agile Development)

- **Trao đổi thông tin (communication)**
- **Đơn giản (simplicity)**
- **Phản hồi (feedback)**
- **Thế mạnh (courage)**





Đặc điểm

(Agile Development)

- **Tương tác liên tục**
- **Thiết kế đơn giản, sử dụng các nguyên lý và dạng thức thiết kế chung**
- **Nhóm làm việc: người lập trình, khách hàng, người quản trị - khách hàng trực diện**

Extreme Programming (XP)

- XP Thiết kế
  - Tuân theo nguyên tắc KIS
  - Khuyến khích việc sử dụng CRC cards (Class Responsibility Colaboration)
  - Đối với vấn đề khó khăn trong thiết kế, cho thấy việc tạo ra các "spike solutions"(gia tăng đột biến)—1 nguyên mẫu thiết kế
  - Khuyến khích "refactoring"—1 tính lặp trong những thiết kế
- XP Lập trình
  - Đề nghị xây dựng các unit test trước khi bắt đầu lập trình
  - Khuyến khích "pair programming"
- XP Kiểm thử
  - Tất cả các unit tests được thực hiện hàng ngày
  - "Acceptance tests" được định nghĩa bởi khách hàng và được dùng để đánh giá chức năng mà khác hàng nhìn thấy.

Kịch bản yêu cầu

- **Với XP, một khách hàng hoặc người dùng là thành viên của đội XP và có trách nhiệm đưa ra các quyết định về các yêu cầu.**
- **Yêu cầu của người dùng được diễn đạt dưới dạng các kịch bản hoặc user story.**
  - **được viết trên các story card**
  - **đội phát triển chia story thành các tác vụ cài đặt.**
    - Các tác vụ này là cơ sở cho việc lập kế hoạch và ước lượng chi phí.
- **Khách hàng lựa chọn các story cần được đáp ứng trong bản release tiếp theo**
  - **dựa theo đánh giá ưu tiên và ước lượng về kế hoạch của họ.**

User story

- **Mẫu**

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

- **Ví dụ**

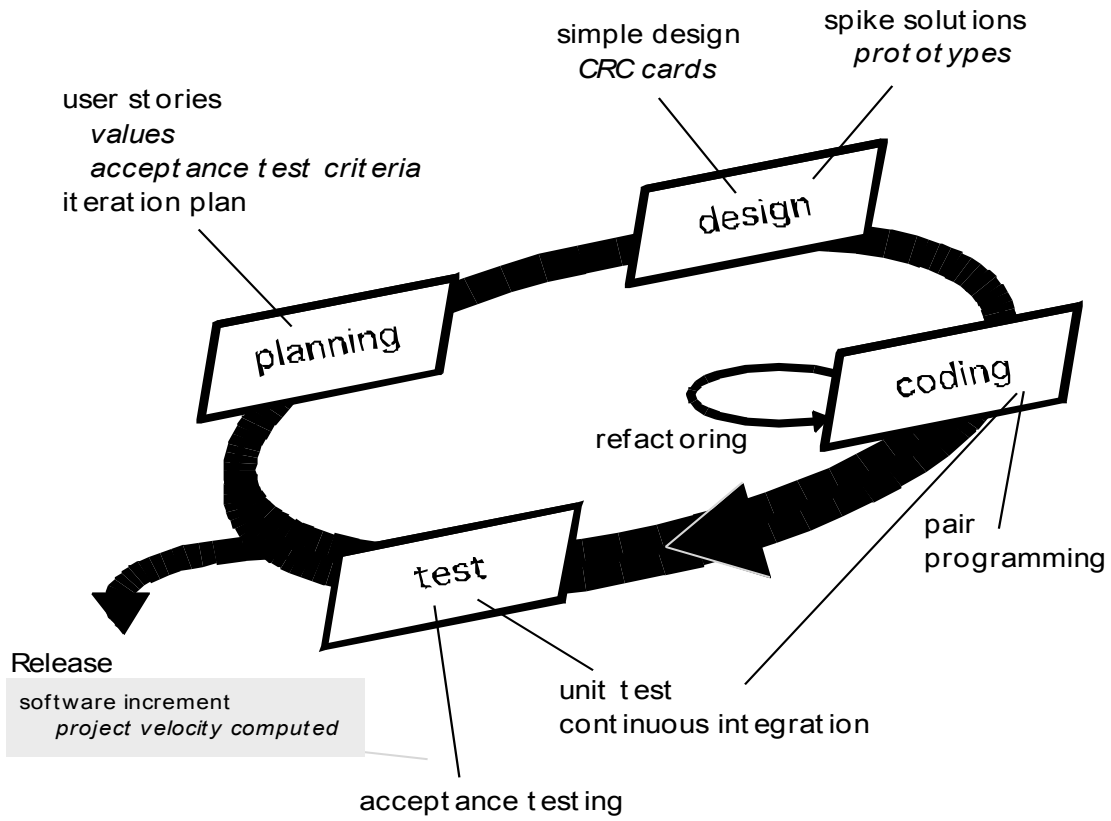
Với vai trò một đại diện khách hàng, tôi muốn tra cứu các khách hàng của tôi theo tên và họ.

Khởi động ứng dụng
Ứng dụng bắt đầu bằng việc mở tài liệu cuối mà user đã dùng lần trước.

Nhà tư vấn sẽ nhập chi phí vào một form chi phí. nhà tư vấn sẽ nhập các mục trong form như dạng chi phí, miêu tả, số lượng, và các ghi chú về chi phí đó. Khi nào muốn, nhà tư vấn có thể thực hiện một công việc tùy chọn trong số dưới đây.

(1)Sau khi điền xong form, nhà tư vấn sẽ "Submit". Nếu chi phí nhỏ hơn 50, chi phí đó sẽ được gửi thẳng cho hệ thống để xử lý.
(2)Nếu nhà tư vấn chưa điền xong form chi phí, nhà tư vấn có thể muốn "Save for later". Form đó sau đó sẽ được hiện trong một danh sách (hàng đợi) dành cho nhà tư vấn với ghi chú trạng thái là "Incomplete".
(3)Nếu nhà tư vấn quyết định xóa toàn bộ dữ liệu và đóng forrm, nhà tư vấn sẽ "Cancel and exit". Dữ liệu soạn dở sẽ không được lưu lại ở bất cứ đâu.

Extreme Programming (XP)

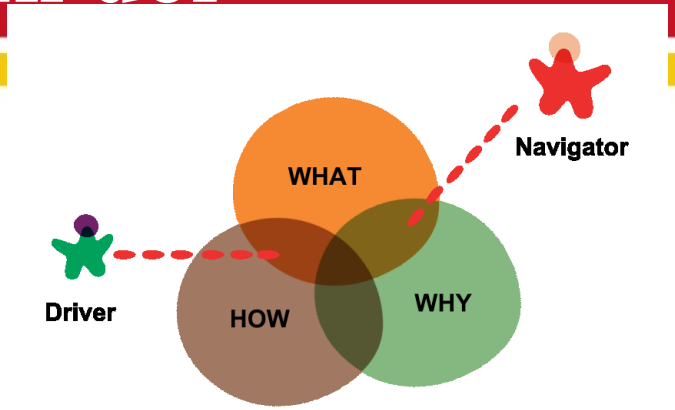


http://bit.ly/2ihOLB4 Slides copyright 2009 by Roger Pressman.

XP practices

- **Các nguyên lý và thực hành (practices) XP**
  - Incremental planning (lập kế hoạch tăng dần)
  - Small releases (các bản phát hành nhỏ)
  - Simple design (thiết kế đơn giản)
  - Test-first development
  - Refactoring (cải tiến)
  - Pair programming (lập trình cặp đôi)
  - Collective ownership (sở hữu tập thể)
  - Continuous integration (tích hợp liên tục)
  - Sustainable pace (tiến độ bền vững)
  - On-site customer (khách hàng tại chỗ)

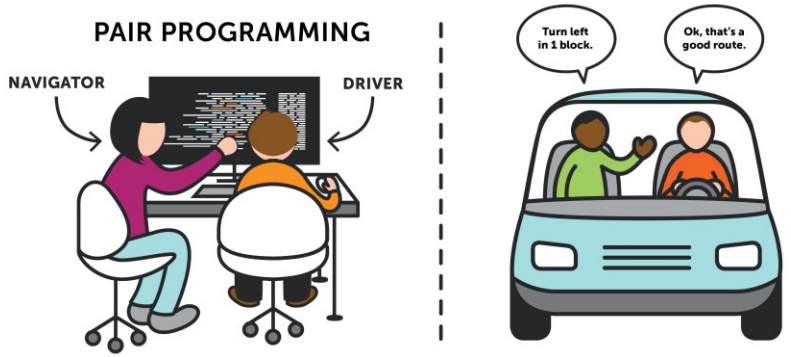
Lập trình đôi



- Trong XP, việc theo từng cặp,
  - Cùng ngồi viết mã.
  - Nhờ vậy cùng làm chủ phần mã và lan truyền kiến thức cho cả đội.
  - Khuyến khích refactoring.
- Các số liệu đo đạc cho thấy năng xuất của lập trình đôi tương đương với năng xuất của hai người làm việc độc lập.

Lập trình đôi

- Khi lập trình theo từng cặp, các lập trình viên ngồi cùng nhau trước một máy tính để phát triển phần mềm.
- Các cặp được sắp xếp một cách linh động để tất cả các thành viên đều có thời gian làm việc cùng nhau trong suốt quá trình phát triển.
- Sự chia sẻ kiến thức xảy ra trong quá trình lập trình theo cặp rất quan trọng
  - Nó giảm rủi ro cho dự án khi một số thành viên rời khỏi nhóm.



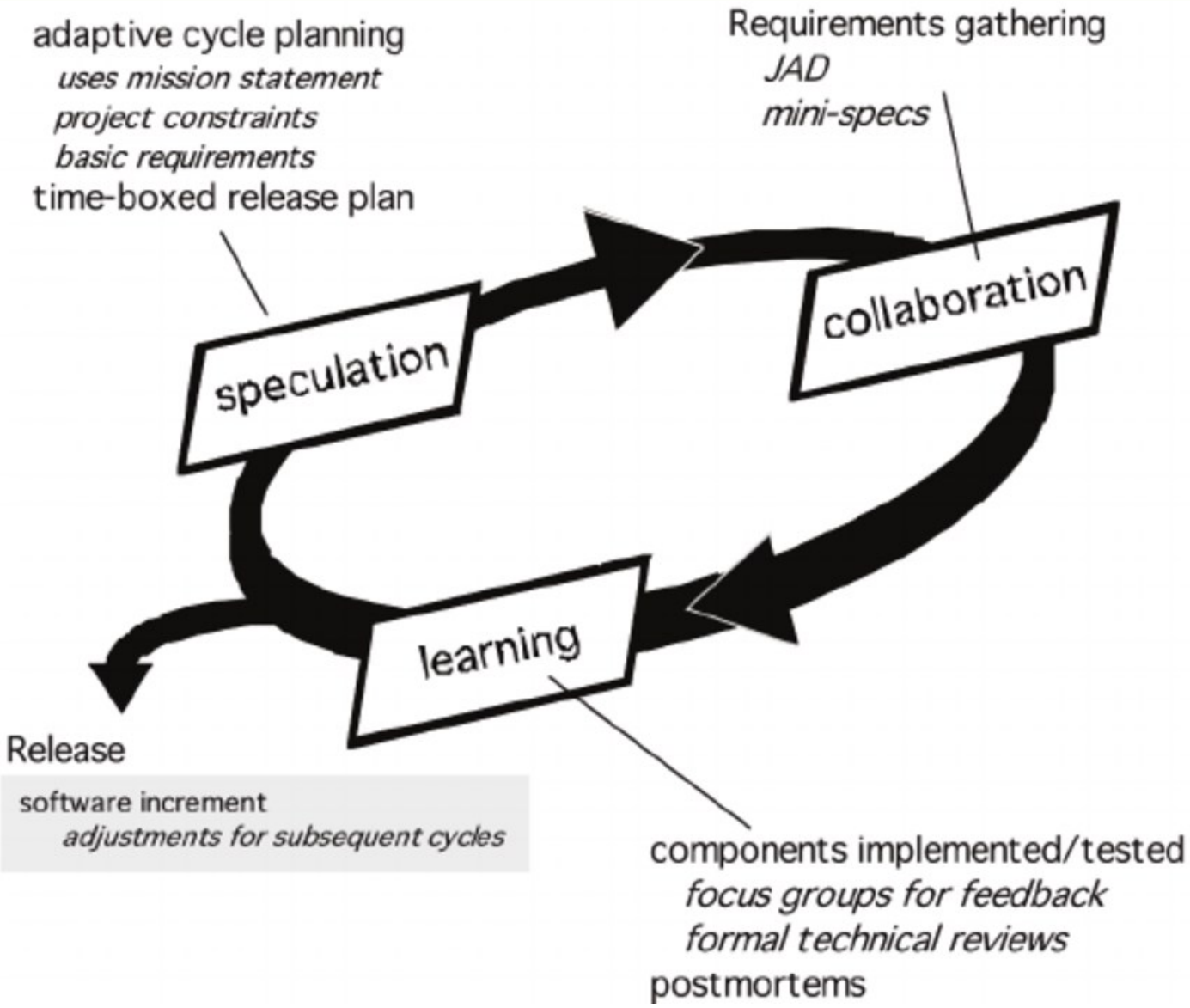
Ưu điểm lập trình đôi

- **Nó hỗ trợ quan niệm sở hữu tập thể và trách nhiệm tập thể đối với hệ thống.**
  - **Các cá nhân không chịu trách nhiệm đối với các vấn đề của mã chương trình.**
    - Thay vào đó, cả đội cùng có trách nhiệm giải quyết các vấn đề đó.
- **Nó hoạt động như là một quy trình review không chính thức vì mỗi dòng lệnh đều có ít nhất hai người xem.**
- **Tạo điều kiện cho refactoring – một quá trình trong việc nâng cấp phần mềm.**
  - **Khi áp dụng lập trình đôi và sở hữu tập thể, những người khác có lợi trực tiếp từ việc refactoring nên họ sẽ dễ hỗ trợ quá trình này.**

Adaptive Software Development

- Được đề xuất bởi Jim Highsmith
- ASD — Đặc điểm phân biệt:
  - Lập kế hoạch hướng Nhiệm vụ
  - Tập trung vào Dựa trên thành phần
  - Sử dụng "time-boxing"
  - Xem xét rõ ràng về rủi ro
  - Đề cao sự hợp tác để thu thập yêu cầu
  - Đề cao "học tập" trong suốt quá trình

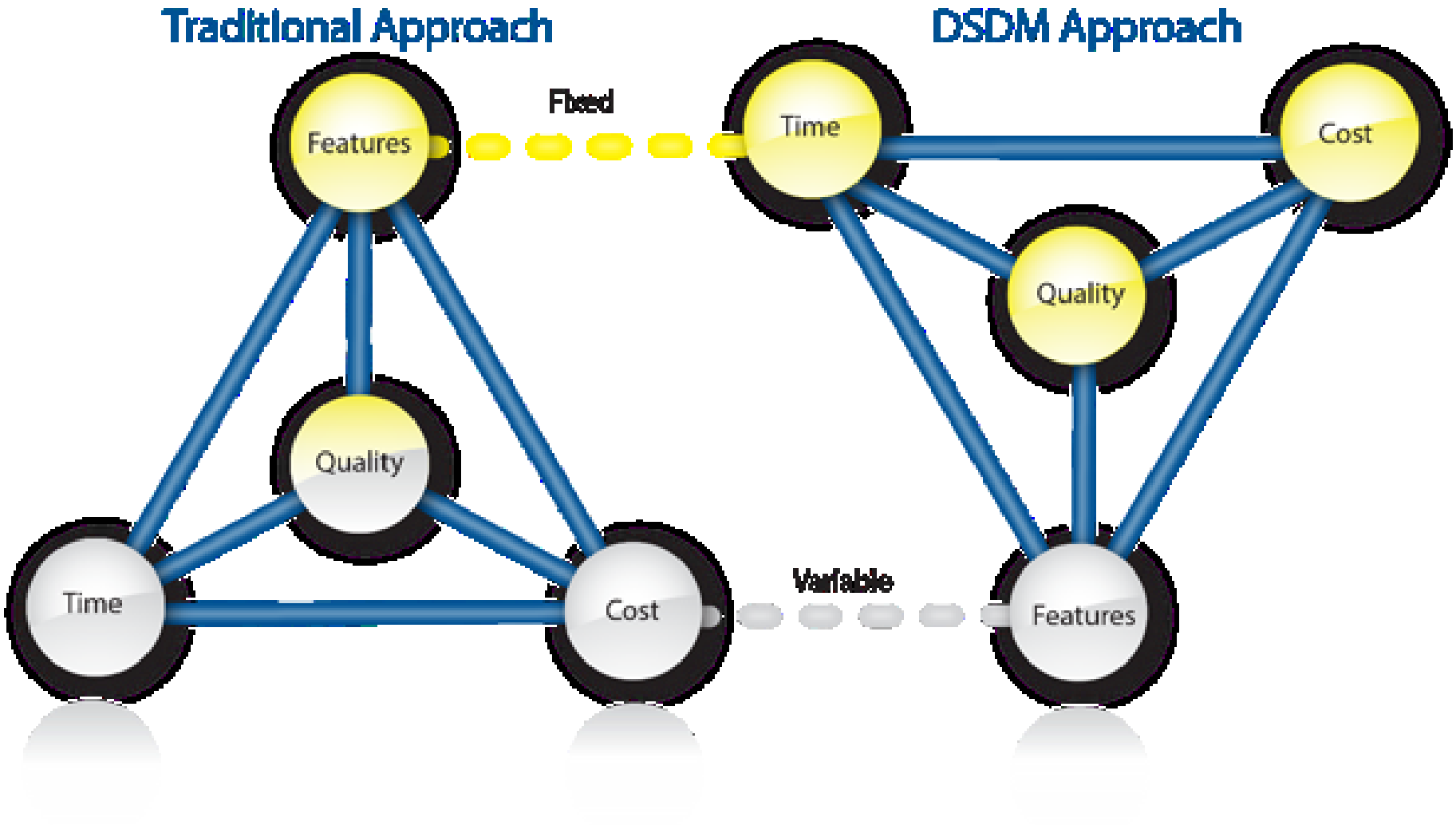
Adaptive Software Development



Phương thức phát triển hệ thống động

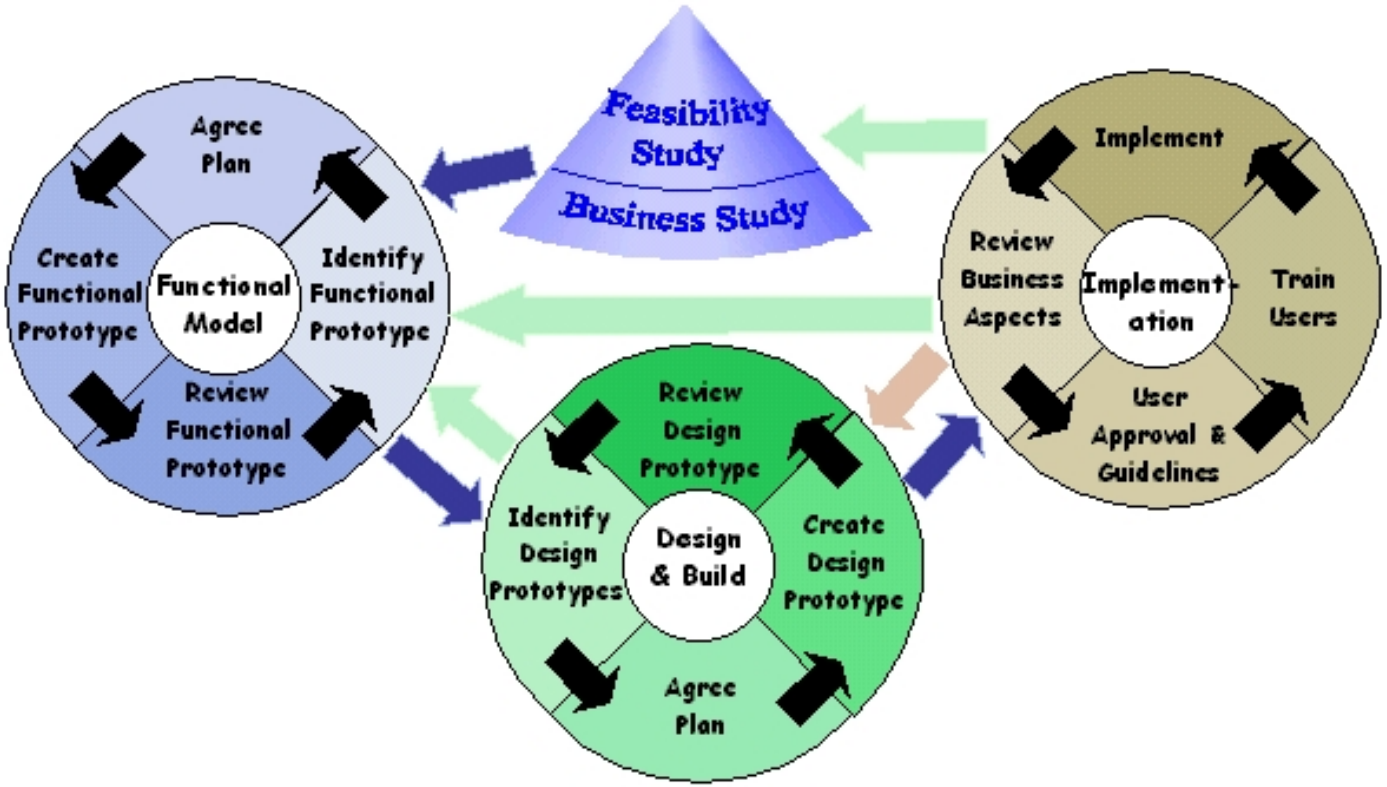
- Được thúc đẩy bởi DSDM Consortium (www.dsdm.org)
- DSDM— Đặc điểm phân biệt (**Dynamic systems development method**)
  - Tương tự như hầu hết các khía cạnh với XP và/ hoặc ASD
  - Chín nguyên tắc hướng dẫn
    - Người dùng tham gia là bắt buộc.
    - DSDM teams được trao quyền quyết định.
    - Trọng tâm là thường xuyên giao hàng sản phẩm.
    - Sự tương ứng cho mục đích kinh doanh là tiêu chuẩn cần thiết cho sự chấp nhận phân phối.
    - Phát triển lặp và tăng dần là cần thiết cho hội tụ vào một giải pháp kinh doanh đúng đắn.
    - Tất cả những thay đổi trong quá trình phát triển có thể đảo ngược.
    - Yêu cầu được baselined ở mức cao.
    - Kiểm thử được tích hợp trong suốt life-cycle.

DSDM - Phương thức phát triển hệ thống động



DSDM - Phương thức phát triển hệ thống động

The DSDM Development Process

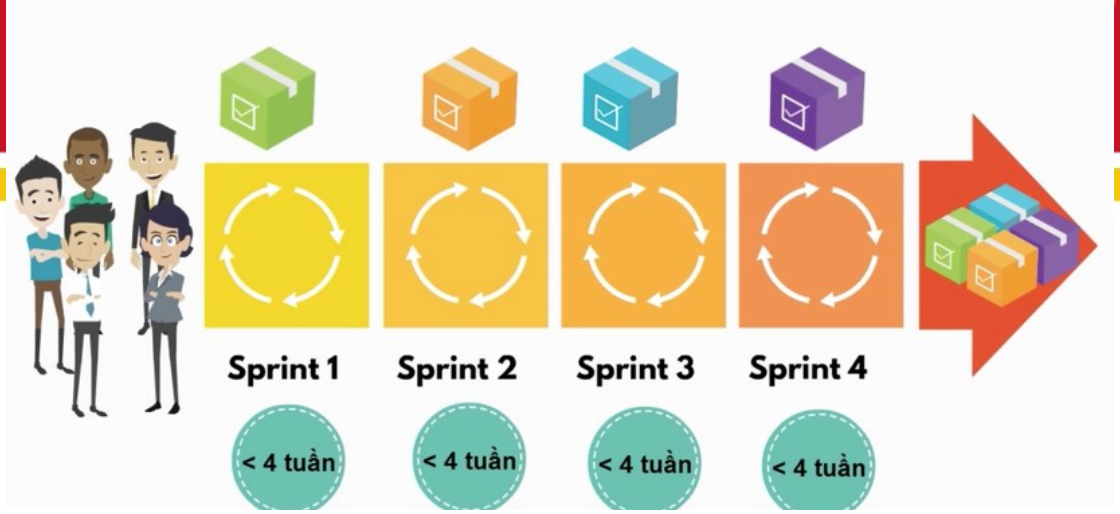


DSDM Life Cycle (with permission of the DSDM consortium)

http://bit.ly/2ihOLB4 Slides copyright 2009 by Roger Pressman.

Scrum

- Được đề xuất bởi Schwaber and Beedle
- Scrum—đặc điểm phân biệt
  - Công việc phát triển được phân chia thành các "gói"
  - Kiểm thử và tài liệu được thực hiện trong quá trình phát triển sản phẩm
  - Công việc trong "sprints" and được bắt nguồn từ 1 "backlog" của các yêu cầu hiện tại
  - Thường xuyên có các cuộc họp ngắn và đôi khi tiến hành không chính quy
  - "Demos"(trình diễn) được giao cho khách hàng với time-box xảy ra


Sprint 1
Sprint 2
Sprint 3
Sprint 4
< 4 tuần
< 4 tuần
< 4 tuần
< 4 tuần

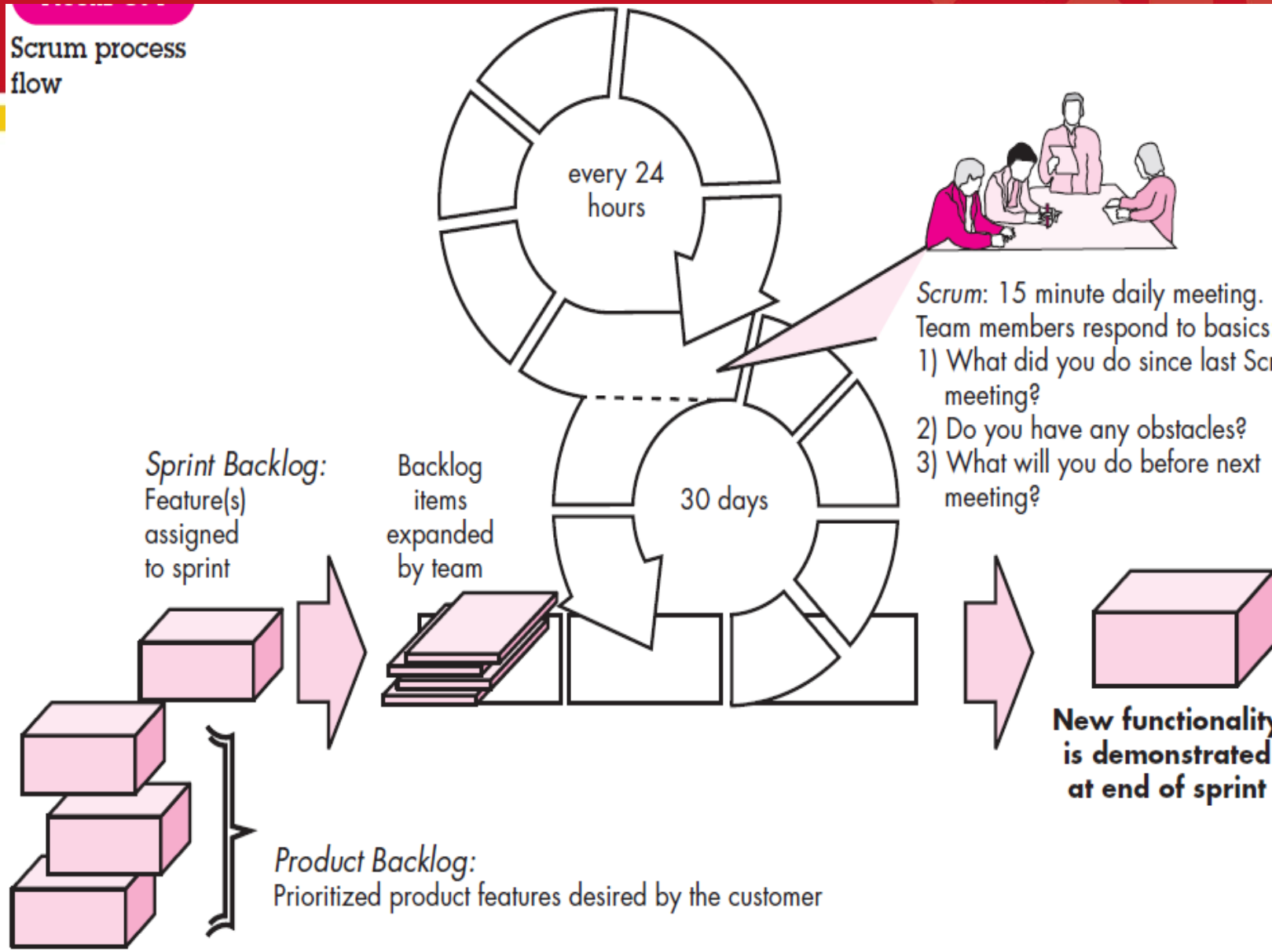
SCRUM


SCRUM MASTER
PRODUCT OWNER
NHÓM PHÁT TRIỂN
Làm mịn Product Backlog
Sơ kết Sprint
SCRUM HÀNG NGÀY
TỪ 1 ĐẾN 4 TUẦN
SPRINT
KHÔNG THAY ĐỔI MỤC TIÊU
Product Backlog
Lập kế hoạch Sprint Phần 1 và 2
Sprint Backlog
Phần tăng trưởng chuyển giao được của sản phẩm
Cải tiến Sprint

Agile

Scrum

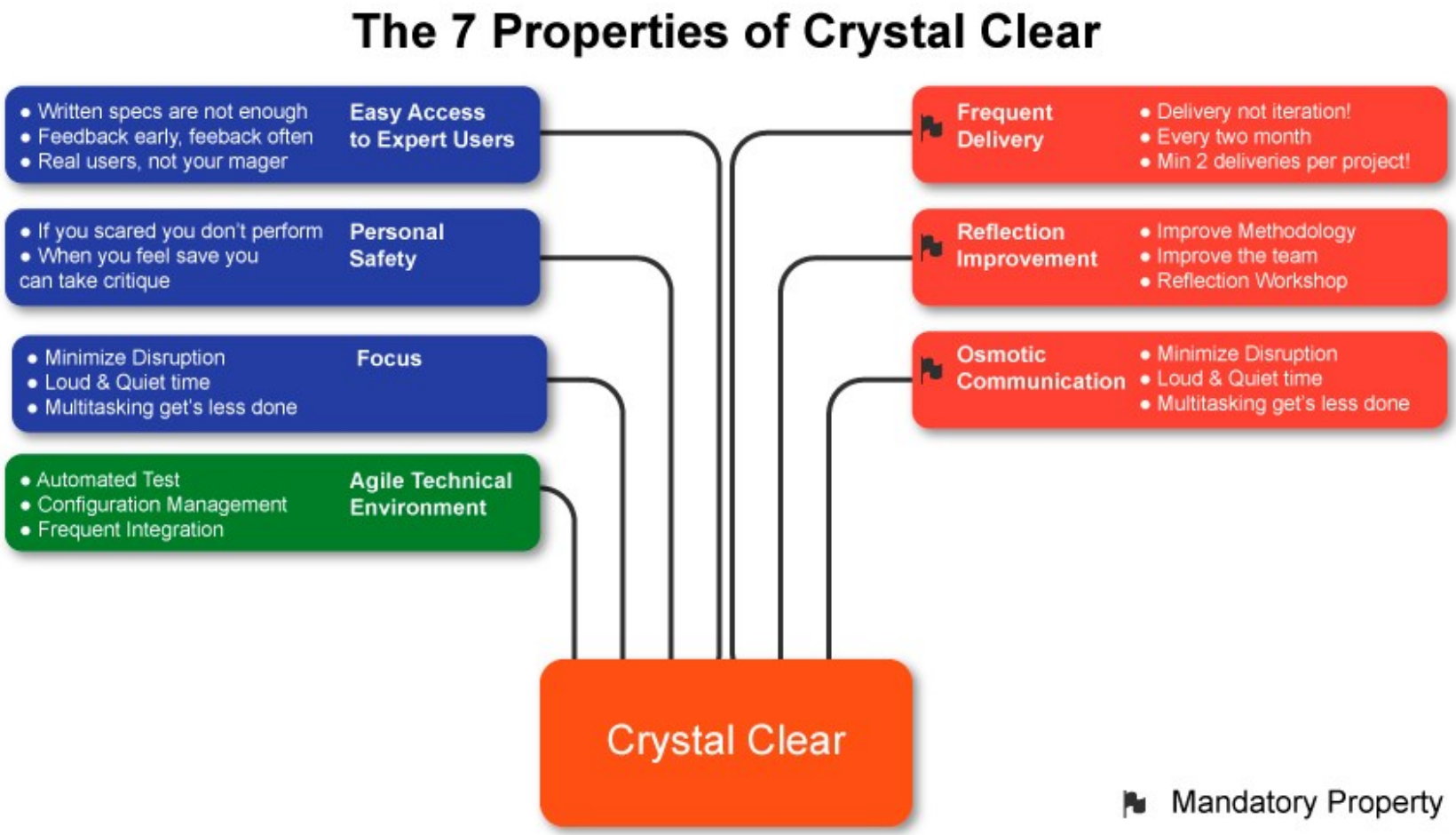
Scrum process flow



Crystal

- Được đề xuất bởi Cockburn and Highsmith
- Crystal— Đặc điểm phân biệt
  - Thực tế là một họ các mô hình cho phép "cơ động" dựa trên các đặc chưng của bài toán
  - Trao đổi thông tin face-to-face được nhấn mạnh
  - Gợi ý việc sử dụng các "hội thảo phản hồi" để xem xét thói quen làm việc của các team.

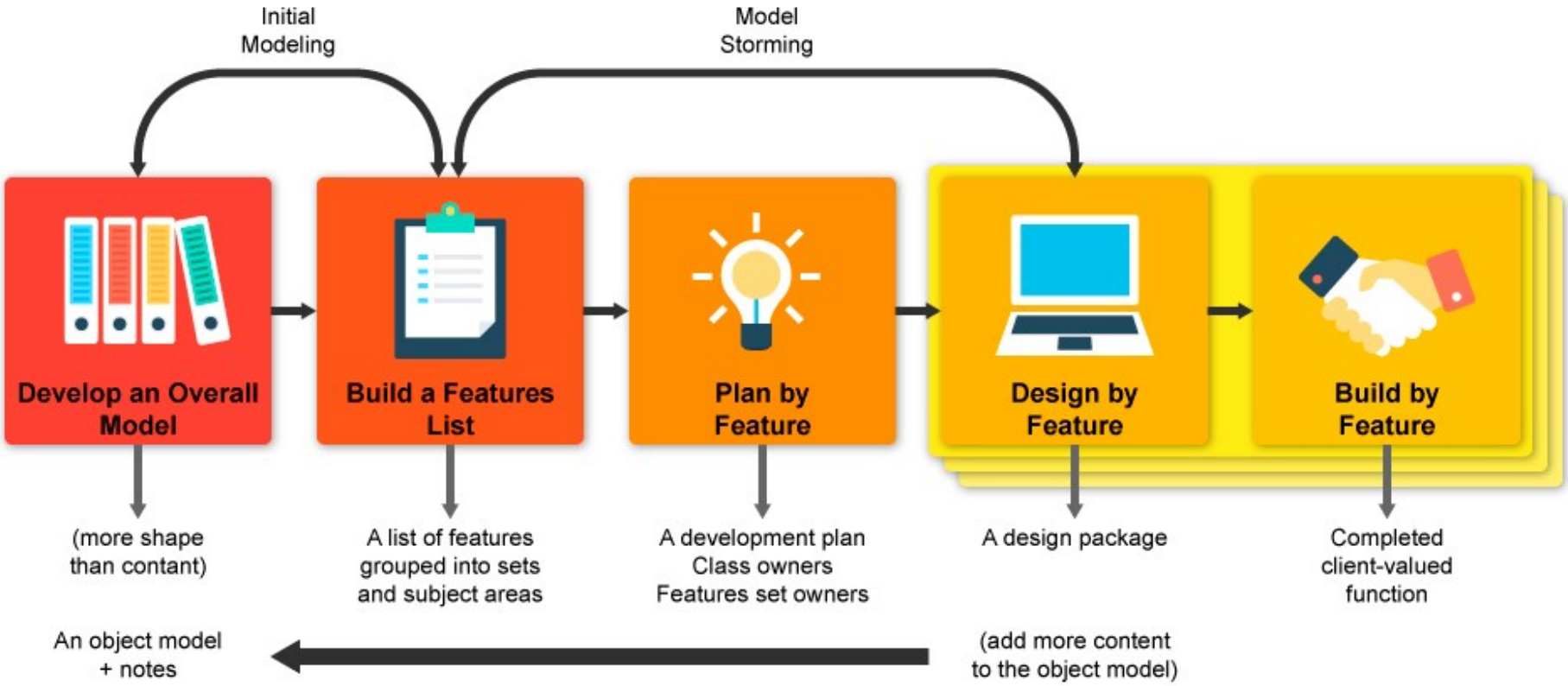
Crystal


The 7 Properties of Crystal Clear
Written specs are not enough
Feedback early, feeback often
Real users, not your mager
Easy Access to Expert Users
If you scared you don't perform
When you feel save you can take critique
Personal Safety
Minimize Disruption
Loud & Quiet time
Multitasking get's less done
Focus
Automated Test
Configuration Management
Frequent Integration
Agile Technical Environment
Frequent Delivery
Delivery not iteration!
Every two month
Min 2 deliveries per project!
Reflection Improvement
Improve Methodology
Improve the team
Reflection Workshop
Osmotic Communication
Minimize Disruption
Loud & Quiet time
Multitasking get's less done
Crystal Clear
Mandatory Property

Feature Driven Development

- Được đề xuất bởi Peter Coad et al
- FDD— đặc điểm phân biệt
  - Nhấn mạnh vào việc xác định "feature"(tính năng)
    - một *feature* "là một chức năng của khách hàng có giá trị được thực hiện trong 2 tuần hoặc ít hơn."
  - Sử dụng feature template
    - <action> the <result> <by | for | of | to> a(n) <object>
  - Một danh sách features được tạo và "plan by feature" được tiến hành
  - Thiết kế và xây dựng được hợp nhất trong FDD

Feature Driven Development



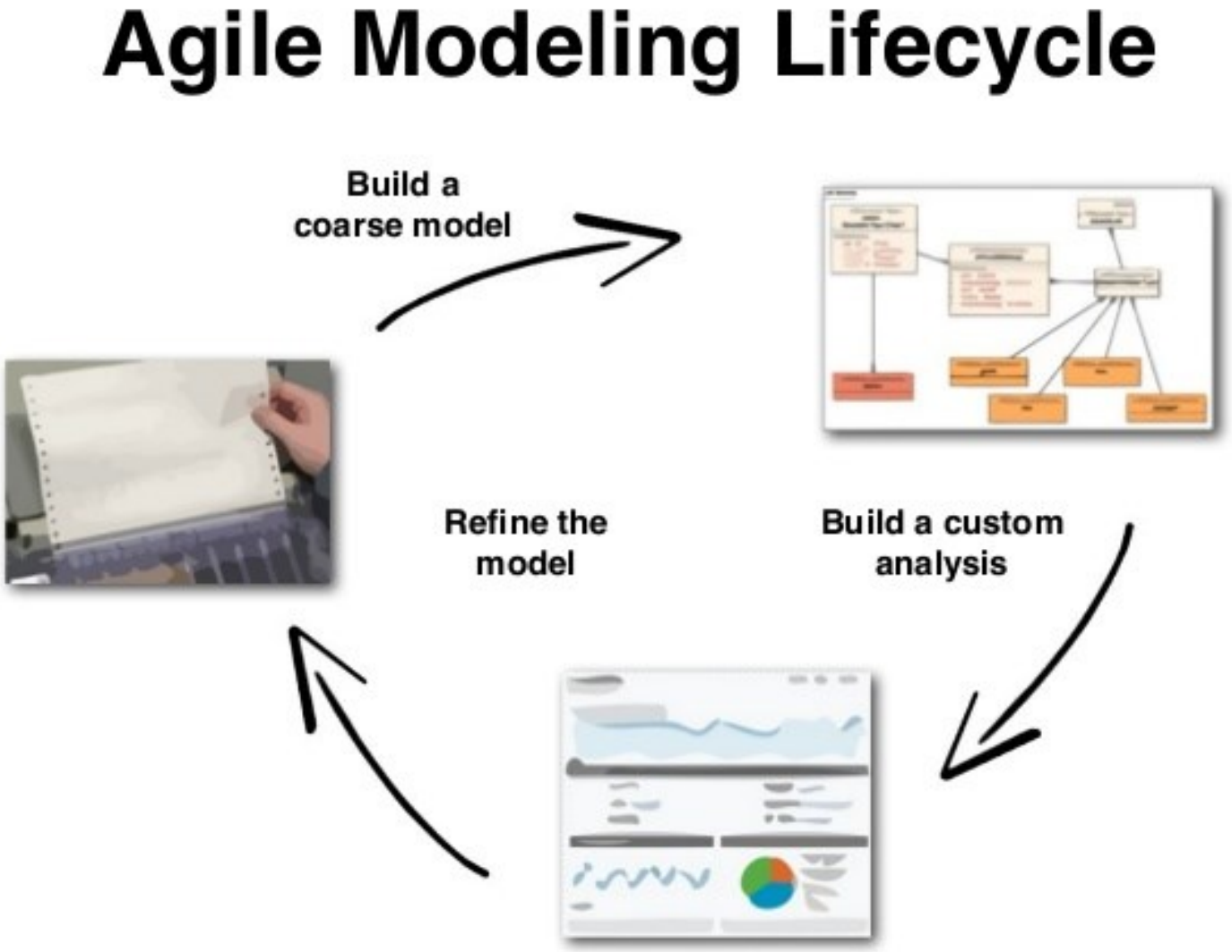
Reprinted with permission of Peter Coad

Agile Modeling

- Được đề xuất bởi Scott Ambler
- Gợi ý một tập hợp các nguyên tắc Agile
  - Mô hình với một mục đích
  - Sử dụng nhiều mô hình
  - Travel light (cần đủ mô hình và tài liệu hướng dẫn)
  - Nội dung quan trọng hơn sự diễn tả
  - Biết các mô hình và công cụ mà bạn có thể tạo ra chúng
  - Thích nghi địa phương

Agile Modeling

http://agilemodeling.com/



So sánh tổng hợp

Ability to Develop Systems	Structured Methodologies		RAD Methodologies			Agile Methodologies
	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP
with Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent
with Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Poor
that are Complex	Good	Good	Good	Poor	Excellent	Poor
that are Reliable	Good	Good	Good	Poor	Excellent	Good
with a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent
with Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Good

Tài liệu tham khảo

- Software & Software Engineering
- Slide được xây dựng theo cuốn Software Engineering: A Practitioner's Approach, 7/e by Roger S. Pressman
- Slides copyright © 1996, 2001, 2005, 2009 by Roger S. Pressman
- http://bit.ly/2ihOLB4

Câu hỏi và bài tập

1. Gợi ý loại mô hình tiến trình tổng quát phù hợp với các dự án sau và giải thích:
   - Hệ thống kế toán mới ở trường đại học thay thế cho hệ thống đang tồn tại
   - Hệ thống hỗ trợ hành khách tra cứu lịch trình tàu trên terminal đặt ở ga tàu.
2. Giải thích tại sao phần mềm được phát triển theo mô hình tăng dần lại khó khăn cho bảo trì?
3. Khảo sát và mô tả ngắn một vài công cụ hỗ trợ các phương pháp Agile.

Câu hỏi và bài tập

4. Nếu áp dụng XP vào dự án nhóm thì những hoạt động cần thực hiện là gì, mô tả ngắn gọn?
5. Nếu áp dụng XP vào dự án nhóm thì có những vai trò gì trong dự án và trách nhiệm/nhiệm vụ của từng vai trò?