Chương 3

- **Phát triển Agile**

Slide được xây dựng theo cuốn

Software Engineering: A Practitioner's Approach, 7/e

của Roger S. Pressman

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, trong khi có các mục bên phải thì đánh giá những mục bên trái cao hơn."

Kent Beck et al

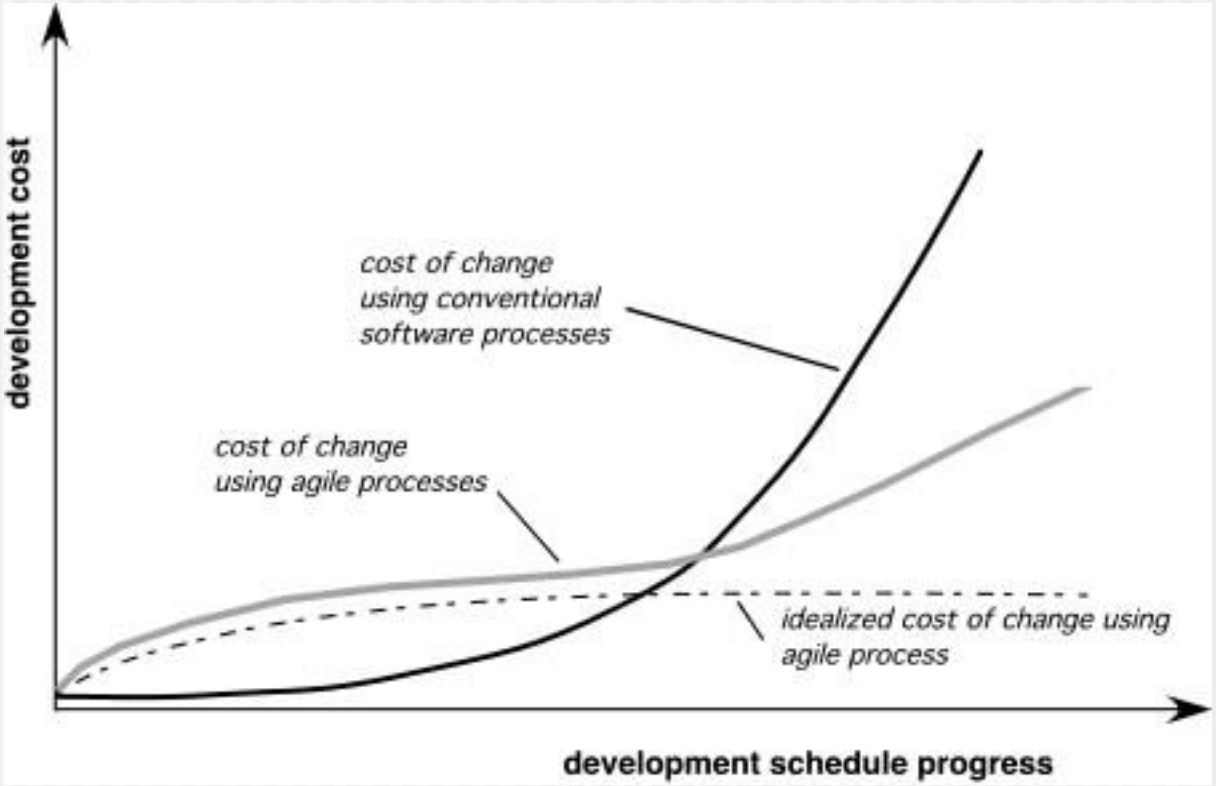
“Agility” là gì?

- 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



Một quy trình Agile

- Được điều khiển bởi những mô tả của khách hàng về những gì yêu cầu (kịch bản).
- Nhận ra rằng các kế hoạch là ngắn.
- Phát triển phần mềm lặp đi lặp lại với một nhấn mạnh vào hoạt động xây dựng.
- Cung cấp nhiều ‘số gia phần mềm’.
- Điều chỉnh như 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.

Nhân tố con người

- *Các quy trình mẫu dành cho nhu cầu của con người và team,* không phải là các cách xung quanh
- Những điểm quan trọng giữa những người trong 1 team Agile và nhóm thân:
  - **Năng lực.**
  - **Phổ biến chung.**
  - **Sự cộng tác.**
  - **Khả năng tự quyết định.**
  - **Khả năng giải quyết vấn đề mở.**
  - **Sự tin tưởng và tôn trọng lẫn nhau.**
  - **Tự tổ chức.**

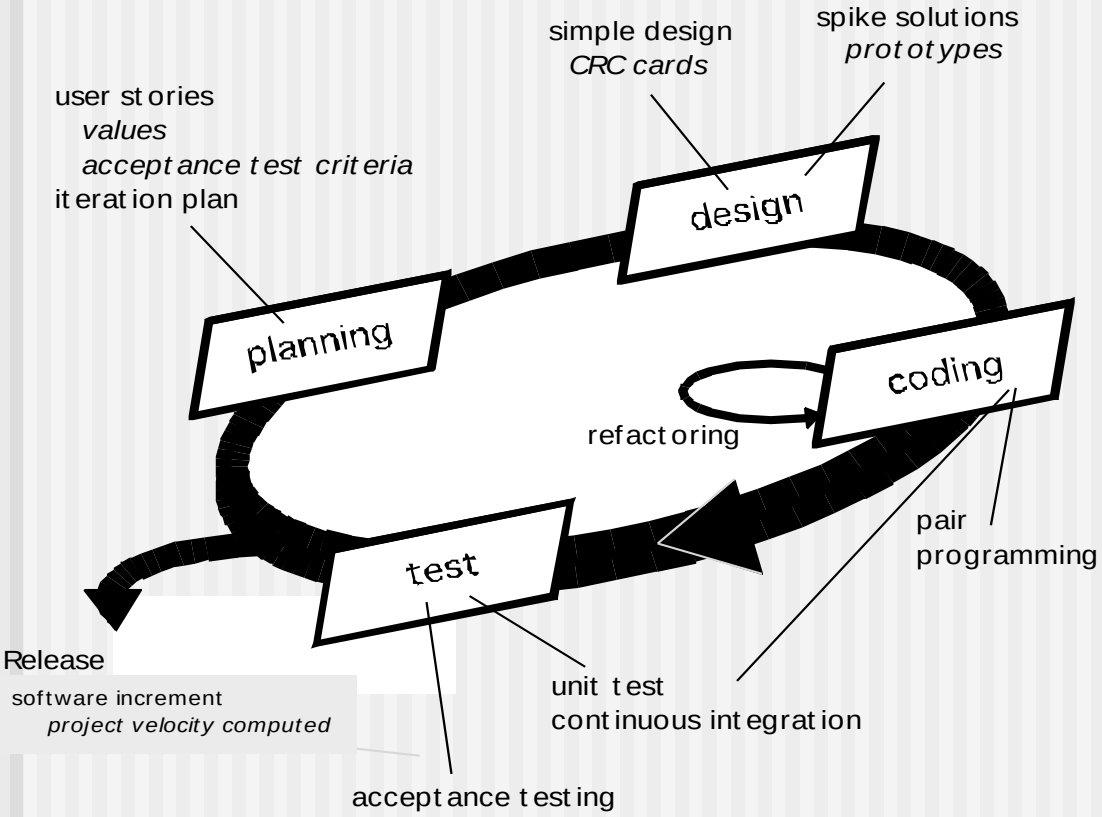
Extreme Programming (XP)

- 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"
  - Agile team đánh giá từng story và gán cho nó 1 giá (cost).
  - Các story được nhóm lại thành 1 một increment chuyển giao
  - Một cam kết được thực hiện vào ngày chuyển giao.
  - Sau increment đầu tiên "project velocity" được sử dụng để xác định số ngày cho những increment khác.

Extreme Programming (XP)

- XP Thiết kế
  - Tuân theo nguyên tắc KIS
  - Khuyến khích việc sử dụng CRC cards (see Chapter 8)
  - 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.

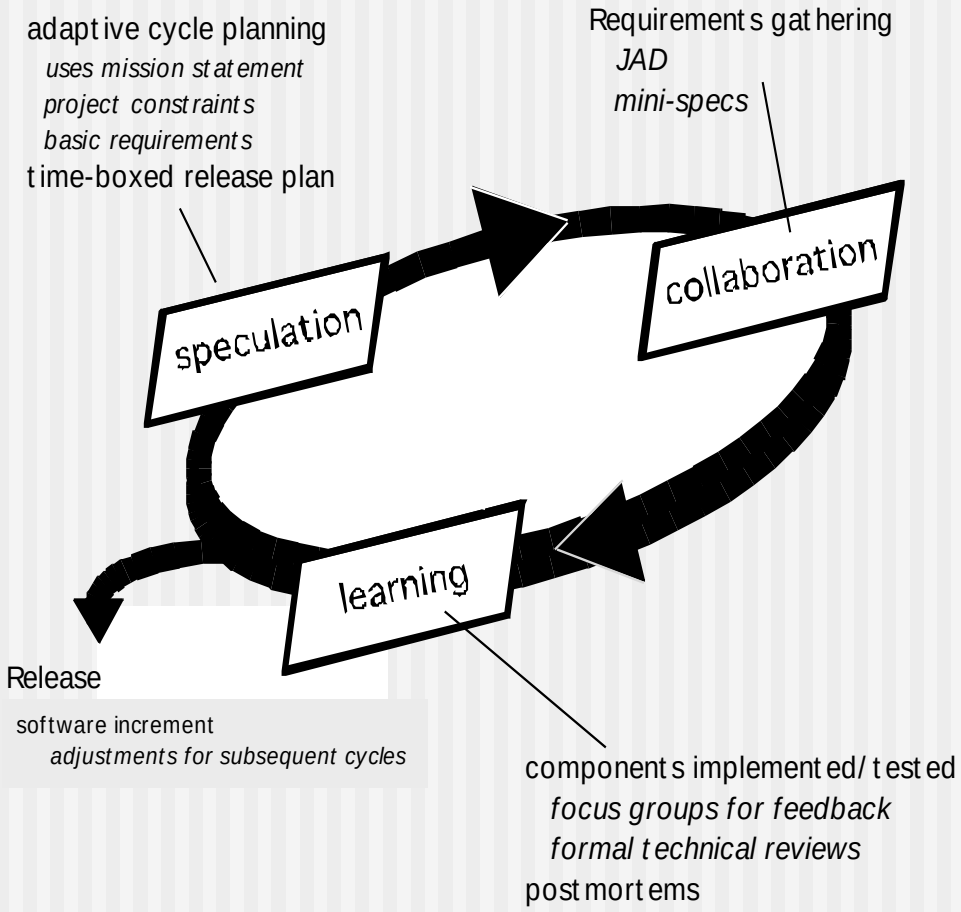
Extreme Programming (XP)



Adaptive Software Development

- Được đề xuất bởi Jim Highsmith
- ASD — Đặc điểm phân biệt
  - Nhiệm vụ định hướng kế hoạch
  - Tập trung dựa trên thành phần
  - Sử dụng "time-boxing" (xem Chapter 24)
  - Xem xét rõ ràng về rủi ro
  - Đề cao sự hợp tác để thu thập yêu cầu
  - Đề cao "learning" trong suốt quá trình

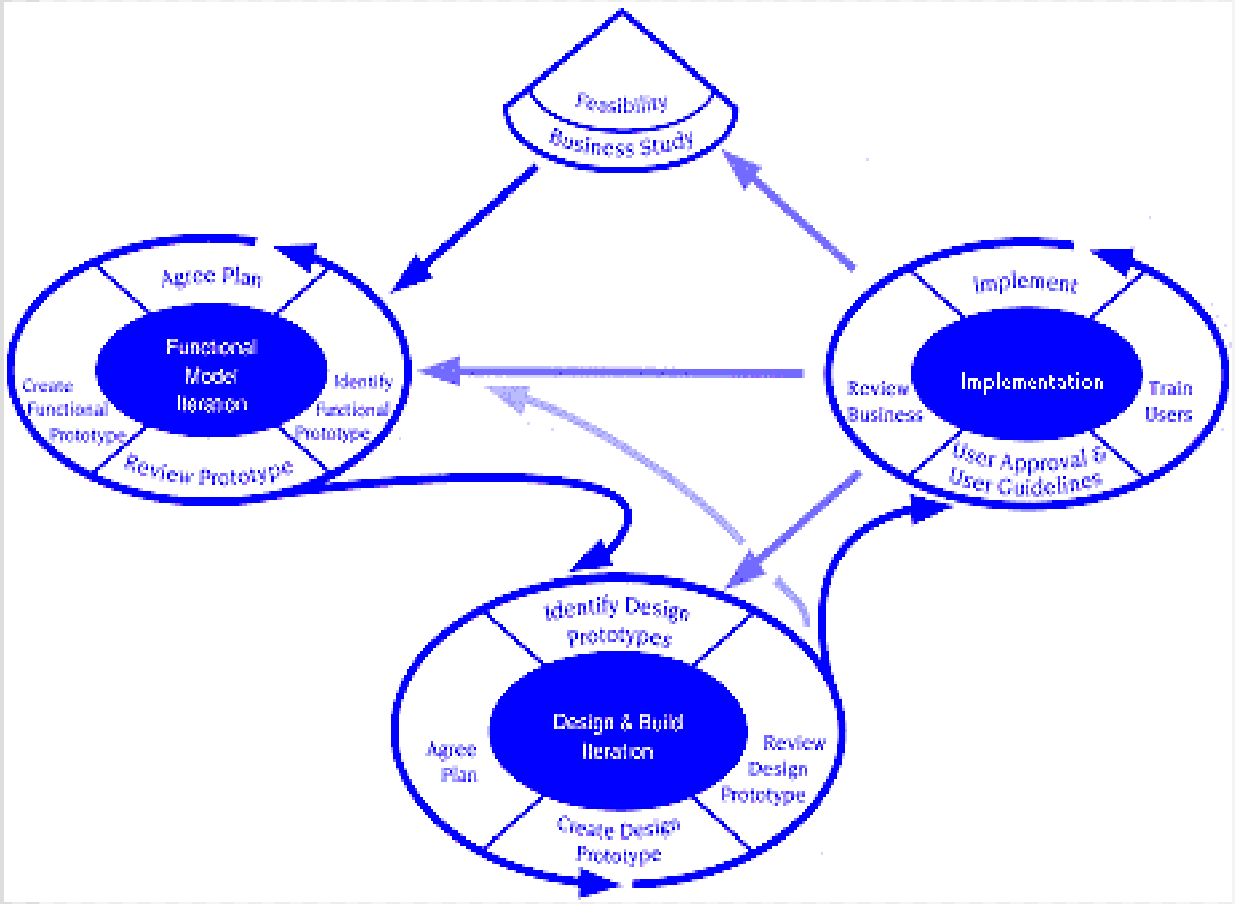
Adaptive Software Development



Phương thức phát triển hệ thống động

- Được thúc đẩy bởi DSDM Consortium (www.dsdm.org)
- DSDM— Đặc điểm phân biệt
  - 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.

Phương thức phát triển hệ thống động

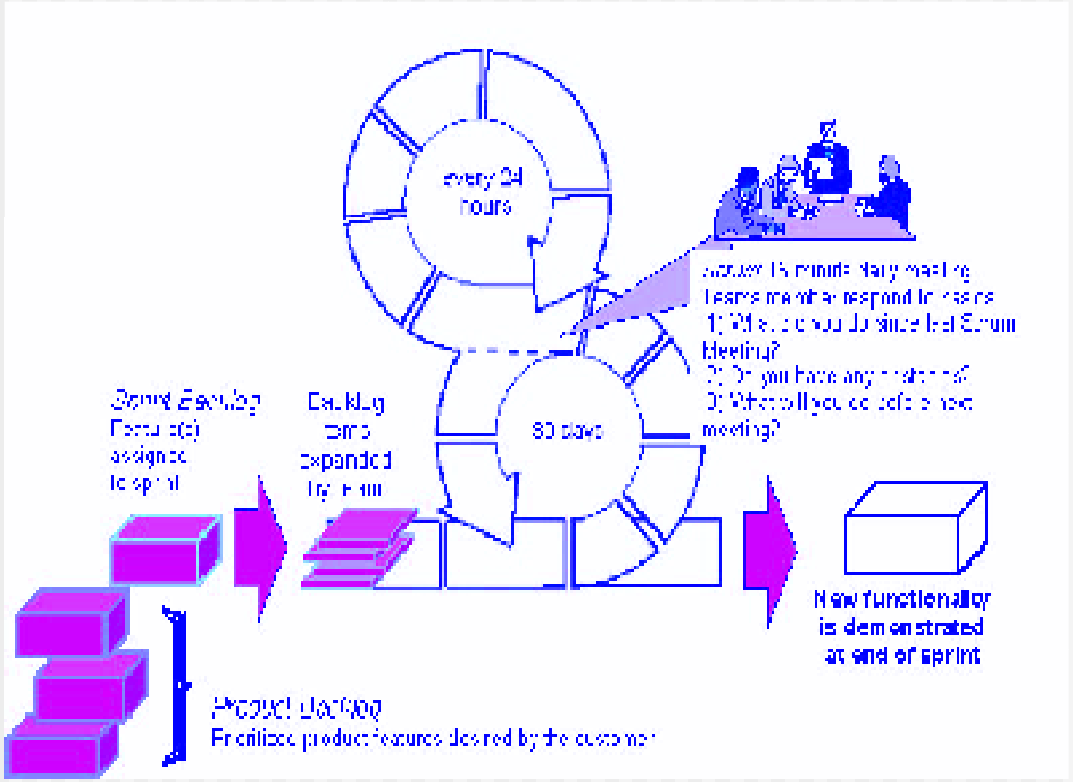


DSDM Life Cycle (with permission of the DSDM consortium)

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 gọi ngắn và đôi khi tiến hành không chính quy
  - “Demos”(trình diễn) được giao cho khách hàng với time-box xảy ra

Scrum



Scrum Process Flow (used with permission)

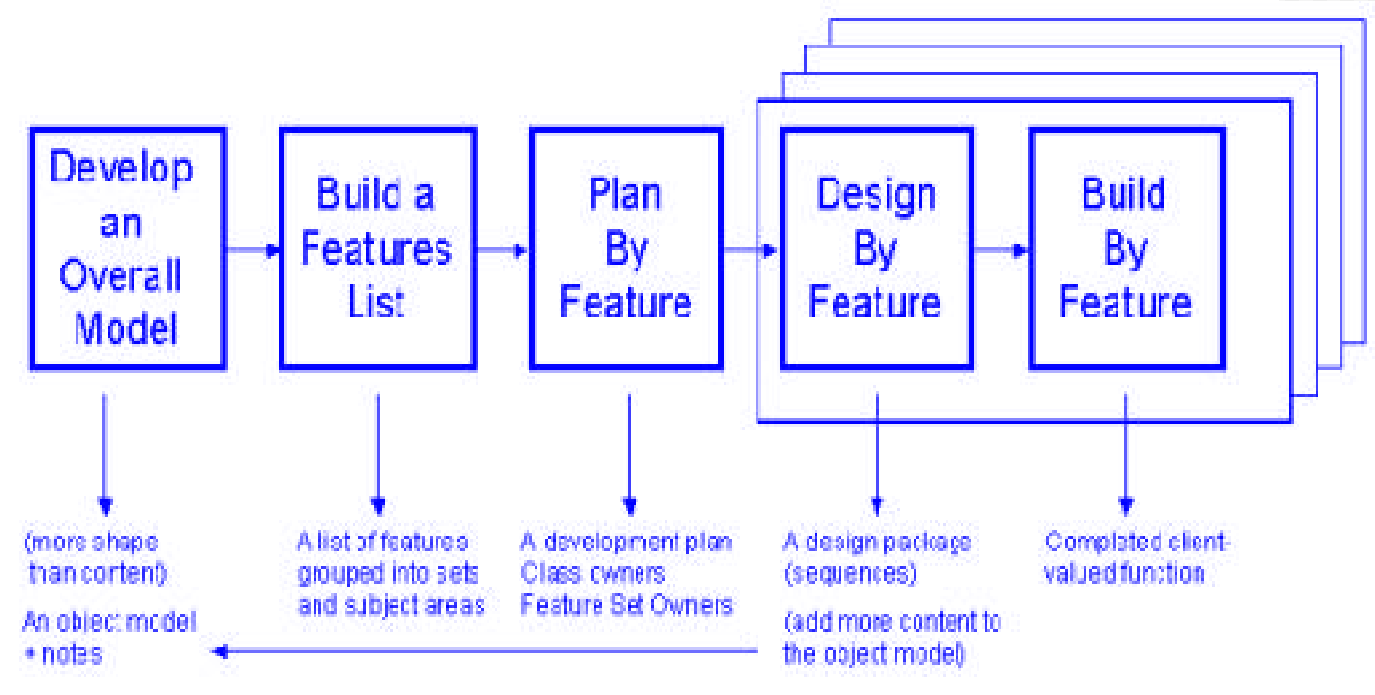
Crystal

- Được đề xuất bởi Cockburn and Highsmith
- Crystal— Đặc điểm phân biệt
  - Thực sự là một gia đình trong của các mô hình cho phép “cơ động” dựa trên các đặc điểm vấn đề
  - Thông tin trực tiếp được nhấn mạnh
  - Gợi ý việc sử dụng các “hội thảo phản ánh ” để xem xét thói quen làm việc của các team.

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