CÔNG NGHỆ PHẦN MỀM

SOFTWARE ENGINEERING

TS. Võ Đức Hoàng

Khoa Công nghệ thông tin

Trường Đại học Bách khoa - Đại học Đà Nẵng

Nội dung

- Chương 1: Giới thiệu Công nghệ phần mềm
- Chương 2: Các mô hình phát triển phần mềm
- Chương 3: Phân tích và đặc tả yêu cầu
- Chương 4: Các kỹ thuật đặc tả
- Chương 5: Thiết kế
- Chương 6: Lập trình và ngôn ngữ lập trình
- Chương 7: Kiểm thử
- Chương 8: Quản trị dự án

Tài liệu học tập

- ***Giáo trình chính:***
  - Andrew Troelsen, Pro C# 5.0 and The .NET 4.5 Framework, Apress, 2012.
- ***Tài liệu tham khảo:***
  - Simon Kendal, *Object Oriented Programming using C#*, BookBoon, 2012.
  - Microsoft MSDN, *C# Programming Guide for Visual Studio 2013*.
  - Joe Mayo, *LINQ Programming*, McGraw-Hill Education, 2009.
  - Andrew Clymer, *Pro Asynchronous Programming with .NET*, Apress, 2013…

Mô hình phát triển phần mềm (2)

Nội dung

- Các hoạt động phát triển phần mềm
- Các mô hình phát triển phần mềm

Các hoạt động phát triển phần mềm

- Phân tích tính khả thi
- Phân tích và đặc tả yêu cầu
- Thiết kế
- Mã hóa
- Kiểm thử
- Bảo trì

Các hoạt động phát triển phần mềm

- Phân tích tính khả thi
  - Xác định những vấn đề cần giải quyết.
  - Xem xét các giải pháp và kỹ thuật khác nhau
    - Thuận lợi
    - Bất lợi
  - Đánh giá về thời gian, giá thành, nguồn tài nguyên cần thiết
  - Sản phẩm: **Tài liệu phát triển**

Các hoạt động phát triển phần mềm

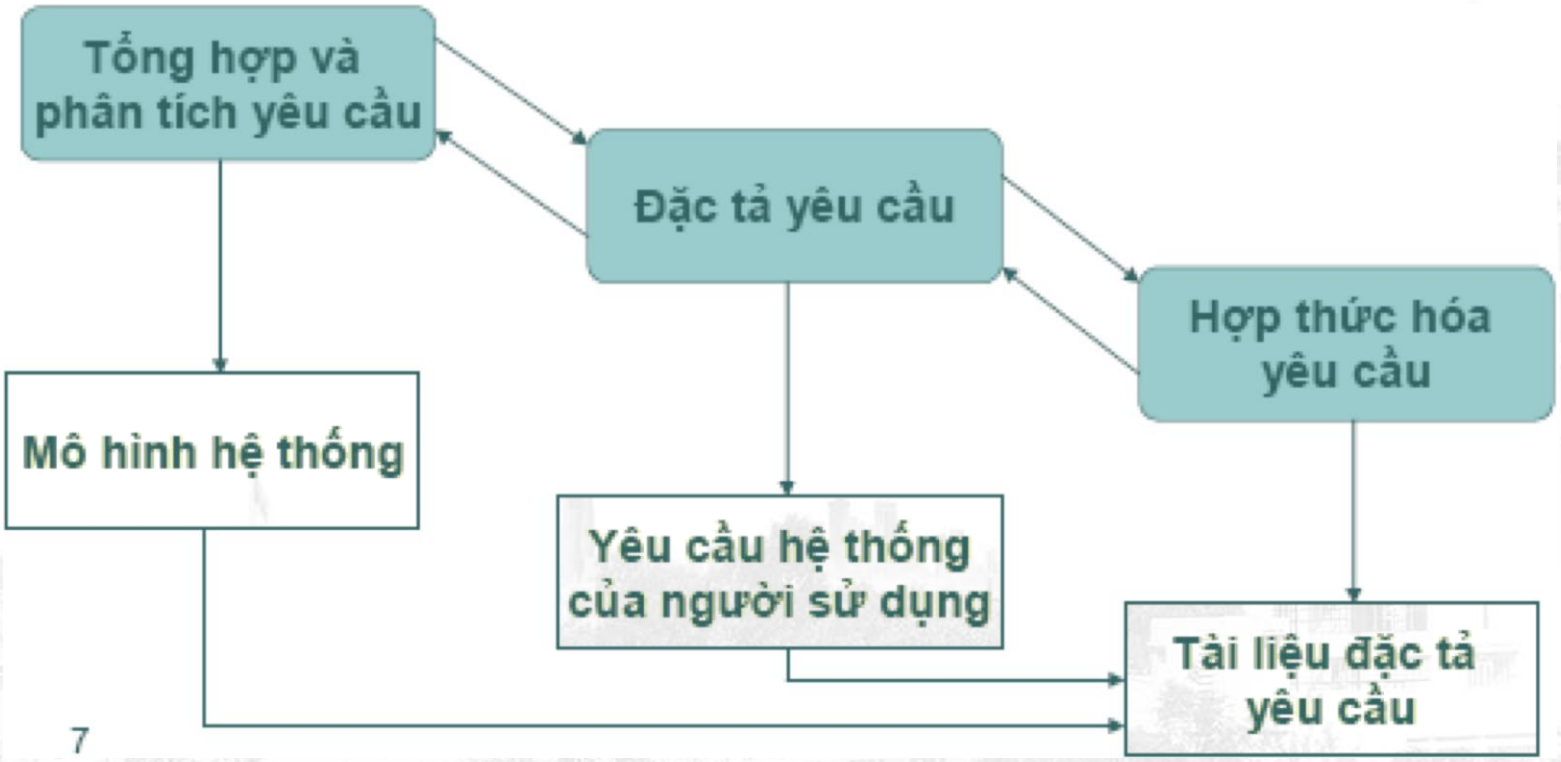
- Phân tích và đặc tả yêu cầu (1)
  - Xác định nhu cầu của khách hàng/người sử dụng
    - Xác định bài toán, chứ không phải là giải pháp
  - Khó khăn
    - Khách hàng không biết cái họ cần
    - Khách hàng không biết bày tỏ cái mình muốn
    - Các thay đổi khi phát triển sản phẩm
  - Sản phẩm: **Tài liệu đặc tả yêu cầu**

Các hoạt động phát triển phần mềm

- Phân tích và đặc tả yêu cầu (2)
  - Các bước
    - Khảo sát, tổng hợp yêu cầu
    - Phân tích yêu cầu
    - Đặc tả yêu cầu
    - Hợp thức hóa yêu cầu

Các hoạt động phát triển phần mềm

- Phân tích và đặc tả yêu cầu (2)

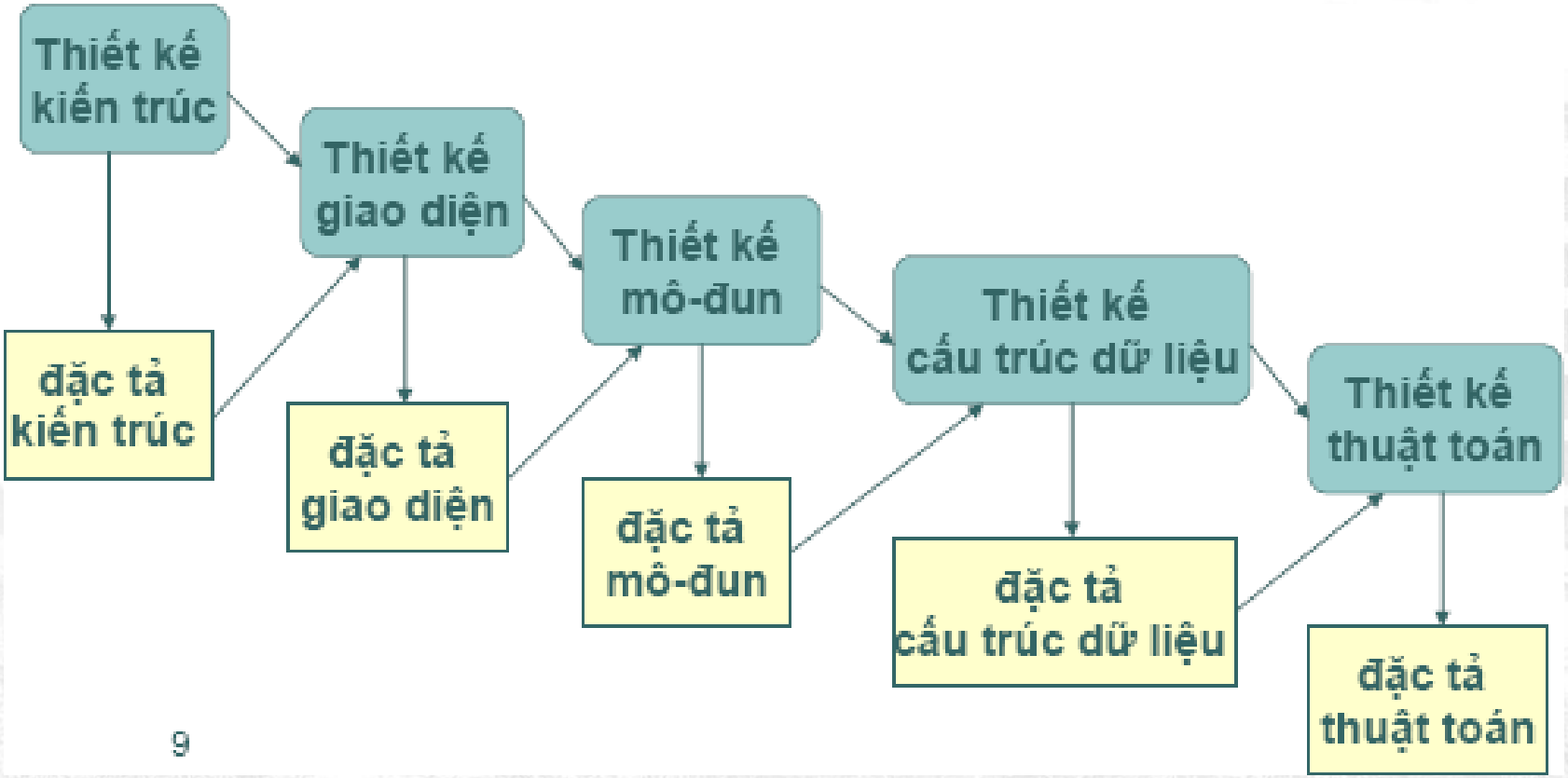


Các hoạt động phát triển phần mềm

- Thiết kế (1)
  - Chuyển từ tài liệu các yêu cầu thành cấu trúc logic có thể cài đặt được
  - Giải pháp cho vấn đề đã được đặt tả
  - Thiết kế kiến trúc
    - Các mô-đun và giao diện các mô đun
  - Thiết kế giao diện
  - Thiết kế các mô-đun
    - Cấu trúc dữ liêu
    - Thuật toán
  - Sản phẩm: **Tài liệu thiết kế**

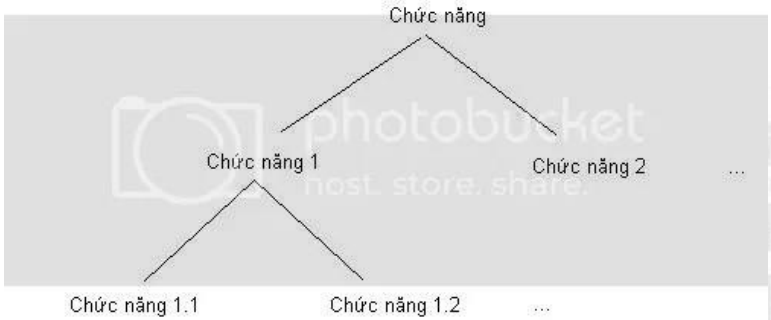
Các hoạt động phát triển phần mềm

- Thiết kế (2)

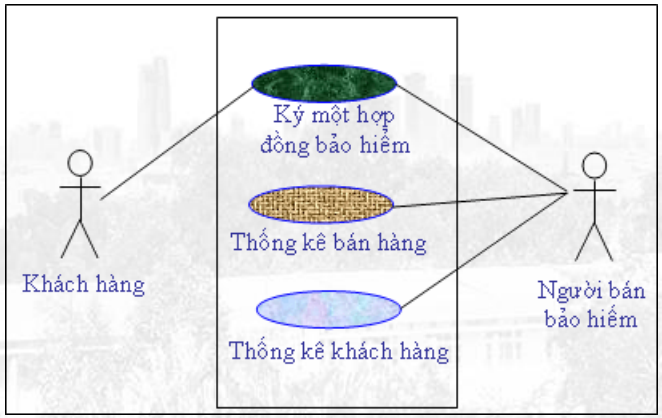


Các hoạt động phát triển phần mềm

- Thiết kế (3)
  - Các phương pháp thiết kế
    - Thiết kế hướng chức năng



    - Thiết kế hướng đối tượng



Các hoạt động phát triển phần mềm

- Thiết kế (3)
  - **Thiết kế hướng chức năng**
    - Đây là cách tiếp cận truyền thống.
    - Quan tâm chủ yếu tới những thông tin mà hệ thống sẽ giữ gìn
    - Thiết kế ngân hàng dữ liệu để chứa những thông tin mà người dung yêu cầu.
    - Cung cấp Forms để nhập thông tin và in báo cáo để trình bày các thông tin
    - ➢ Tập trung vào thông tin và không mấy để ý đến những gì có thể xảy ra với những hệ thống đó và cách hoạt động (ứng xử) của hệ thống là ra sao.

Các hoạt động phát triển phần mềm

- Thiết kế (3)
  - **Thiết kế hướng chức năng**

Ưu điểm

- Đơn giản, là phương pháp tốt cho việc thiết kế ngân hàng dữ liệu và nắm bắt thông tin

Nhược điểm

- Áp dụng cho việc thiết kế ứng dụng lại có thể khiến phát sinh nhiều khó khăn.
- Không phù hợp với hệ thống thường xuyên thay đổi.

Các hoạt động phát triển phần mềm

- Thiết kế (3)
  - **Thiết kế hướng đối tượng**
    - Lối tiếp cận hướng đối tượng là một lối tư duy về vấn đề theo lối ánh xạ các thành phần trong bài toán vào các đối tượng ngoài đời thực.
    - Chia ứng dụng thành các thành phần nhỏ, gọi là các đối tượng độc lập với nhau.
    - Xây dựng ứng dụng bằng cách chắp các đối tượng đó lại với nhau.
    - ➢ Chức năng của hệ thống được biểu diễn thông qua cộng tác của đối tượng, việc thay đổi chức năng, tiến hóa chức năng không làm thay đổi đến cấu trúc tĩnh của phần mềm..

Các hoạt động phát triển phần mềm

- Thiết kế (3)
  - **Thiết kế hướng đối tượng**

Ưu điểm

- Tập trung vào cả hai khía cạnh của hệ thống là dữ liệu và hành động.
- Hỗ trợ sử dụng lại mã nguồn.
- Phù hợp với các hệ thống lớn

➢ Giảm thiểu lỗi và các khó khăn trong việc bảo trì, giúp tăng tốc độ thiết kế và phát triển phần mềm.

Nhược điểm

- Tốc độ chậm
- Khó cho người mới bắt đầu.

Các hoạt động phát triển phần mềm

- Mã hóa và gỡ rối
  - Mã hóa
  - Cài đặt các thiết kế bằng ngôn ngữ lập trình
  - Không đơn thuần chỉ là lập trình
    - Viết tài liệu
    - Chuẩn lập trình (coding standards)
    - Lập trình theo cặp (pair programming)
    - Công cụ
    - Quản lý phiên bản
- Gỡ rối
  - Phát hiện các lỗi trong quá trình lập trình
- Sản phẩm: **chương trình**

Các hoạt động phát triển phần mềm

- Kiểm thử (1)
  - Phát hiện lỗi trong chương trình
  - Lập kế hoạch thực hiện kiểm thử
    - Tạo các trường hợp kiểm thử
    - Tiêu chuẩn kiểm thử
    - Nguồn tài nguyên kiểm thử
- Mã nguồn được kiểm thử theo tài liệu thiết kế
- Sản phẩm: **báo cáo kiểm thử**

Các hoạt động phát triển phần mềm

- Kiểm thử (2)
  - Các hoạt động kiểm thử
    - Kiểm thử đơn vị
    - Kiểm thử tích hợp
    - Kiểm thử hệ thống
    - Kiểm thử chấp nhận
  - Các phương pháp kiểm thử
    - Kiểm thử tĩnh
    - Kiểm thử động
      - Kiểm thử hộp trắng
      - Kiểm thử hộp đen

Các hoạt động phát triển phần mềm

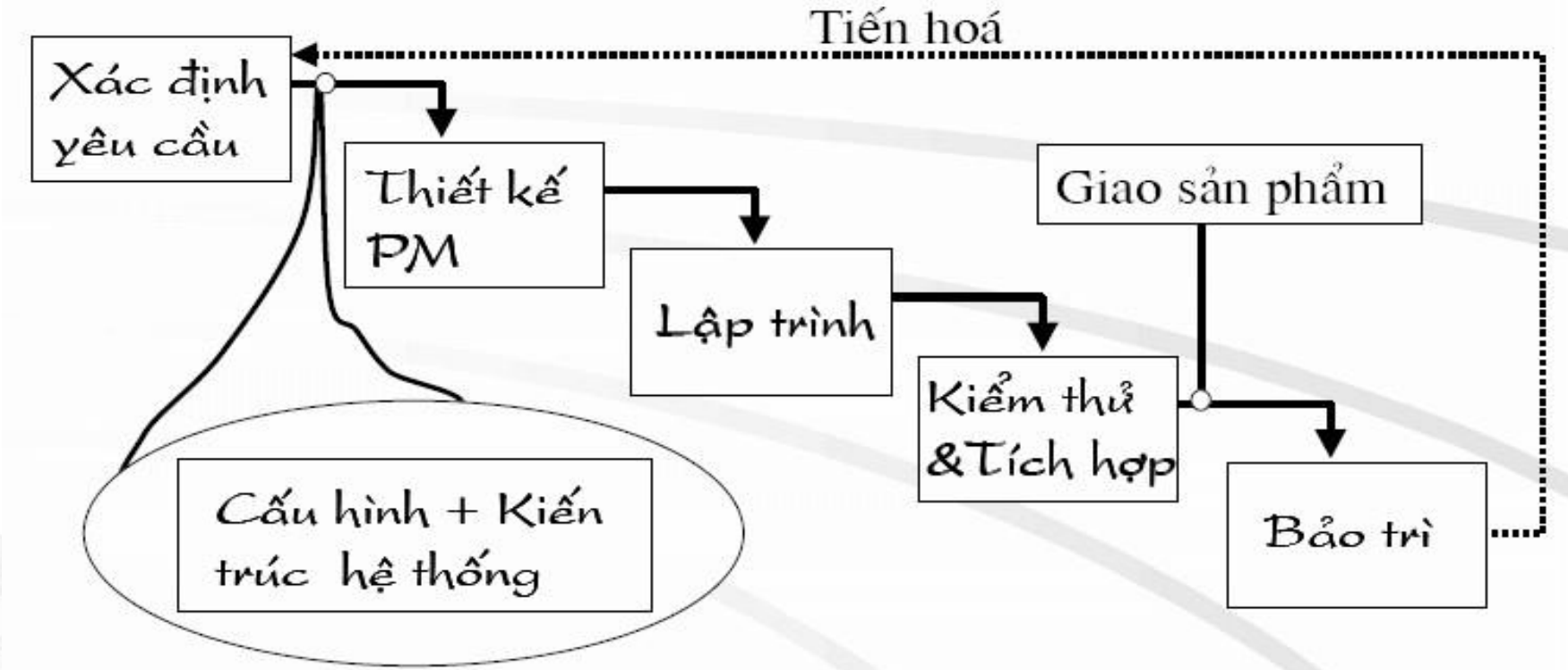
- Bảo trì
  - Bảo đảm các chương trình vận hành tốt
  - Cài đặt các thay đổi
  - Cài đặt các yêu cầu mới
  - Xử lý lỗi khi vận hành
  - Sản phẩm: **chương trình**

Các mô hình phát triển phần mềm

- Sự tổ chức các hoạt động phát triển phần mềm
- Mô hình phát triển phần mềm hay tiến trình phát triển phần mềm
- Có nhiều mô hình phát triển phần mềm
  - Mô hình thác nước
  - Mô hình nguyên mẫu
  - Mô hình chữ V
  - Mô hình tiến hóa
  - Mô hình xoắn ốc
  - Mô hình hợp nhất

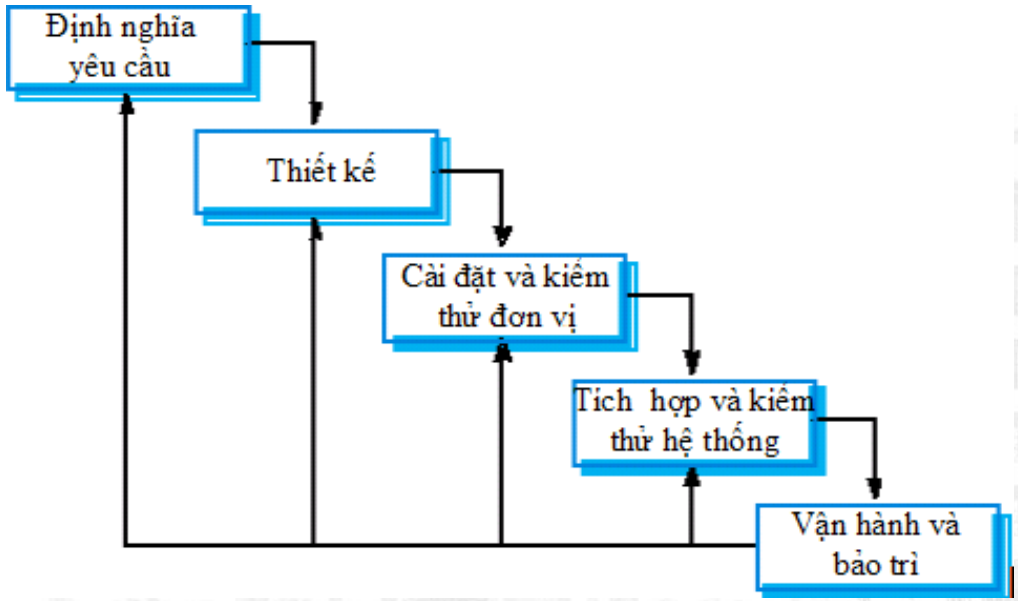
Các mô hình phát triển phần mềm

- Mô hình thác nước (Waterfall model)



Các mô hình phát triển phần mềm

- Mô hình thác nước



- Tách biệt giữa các pha, tiến hành tuần tự
- Chậm có phiên bản thực hiện được
- Đặc tả kỹ, phân công chuyên trách – hướng tài liệu

- Mô hình thác nước:
  - Có sớm nhất và được sử dụng rộng rãi
  - Thích hợp khi yêu cầu hiểu tốt
  - Bảo trì thuận lợi

Các mô hình phát triển phần mềm

- Mô hình thác nước
  - Ưu điểm
    - Đơn giản, dễ hiểu và sử dụng.
    - Đối với các dự án nhỏ hơn, mô hình thác nước hoạt động tốt và mang lại kết quả phù hợp.
    - Vì các giai đoạn của mô hình thác nước cứng nhắc và chính xác, một pha được thực hiện một lần, nó rất dễ dàng để maintain.
    - Các tiêu chí đầu vào và đầu ra được xác định rõ ràng, do đó nó dễ dàng và có hệ thống để tiến hành chất lượng.
    - Kết quả được ghi chép tốt.

Các mô hình phát triển phần mềm

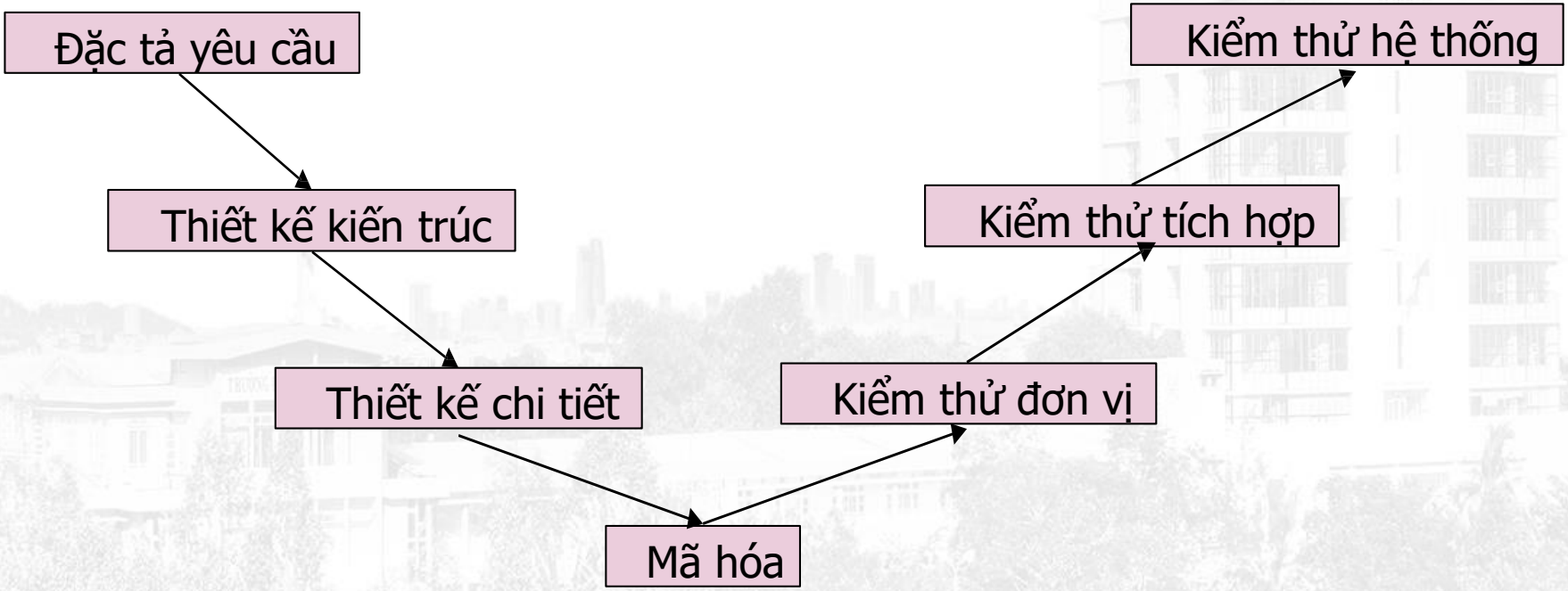
- Mô hình thác nước
  - Nhược điểm
    - Bản chất của phát triển phần mềm là quá trình lặp đi lặp lại chứ không phải tuần tự
    - Khách hàng đặc tả tất cả yêu cầu một cách chính xác và đầy đủ ngay từ ban đầu
    - Khách hàng thường phải chờ đợi rất lâu để thấy được phiên bản đầu tiên của sản phẩm
    - Tồn tại “delay”trong nhóm làm việc
    - Đối với các dự án lớn và phức tạp, mô hình này không tốt vì yếu tố rủi ro cao hơn.
    - Không thích hợp cho các dự án mà yêu cầu được thay đổi thường xuyên.
    - Không làm việc cho các dự án dài và đang diễn ra.

Các mô hình phát triển phần mềm

- Mô hình thác nước
  - Kết luận
    - Trong mô hình thác nước, điều rất quan trọng là đi theo dấu hiệu của các sản phẩm của từng giai đoạn. Tính đến ngày hôm nay hầu hết các dự án đang di chuyển với các mô hình Agile và Prototype, mô hình thác nước vẫn giữ tốt cho các dự án nhỏ hơn. Nếu yêu cầu là đơn giản và testable, mô hình thác nước sẽ mang lại kết quả tốt nhất..

Các mô hình phát triển phần mềm

- Mô hình chữ V (V model)
  - Khắc phục hạn chế của mô hình phát triển phần mềm thác nước là:
    - Các khiếm khuyết được tìm thấy rất chậm trong quá trình phát triển.
    - Fix bug càng chậm thì càng khó khăn và tốn kém
  - Nhấn mạnh vai trò kiểm thử



Các mô hình phát triển phần mềm

- Mô hình chữ V
  - Qui trình được chia thành hai nhóm giai đoạn tương ứng nhau:
    - Phát triển và kiểm thử.
    - Mỗi giai đoạn phát triển sẽ kết hợp với một giai đoạn kiểm thử tương ứng
  - Các hoạt động kiểm thử phải được tiến hành song song (theo khả năng có thể) ngay từ đầu chu trình cùng với các hoạt động phát triển.

Các mô hình phát triển phần mềm

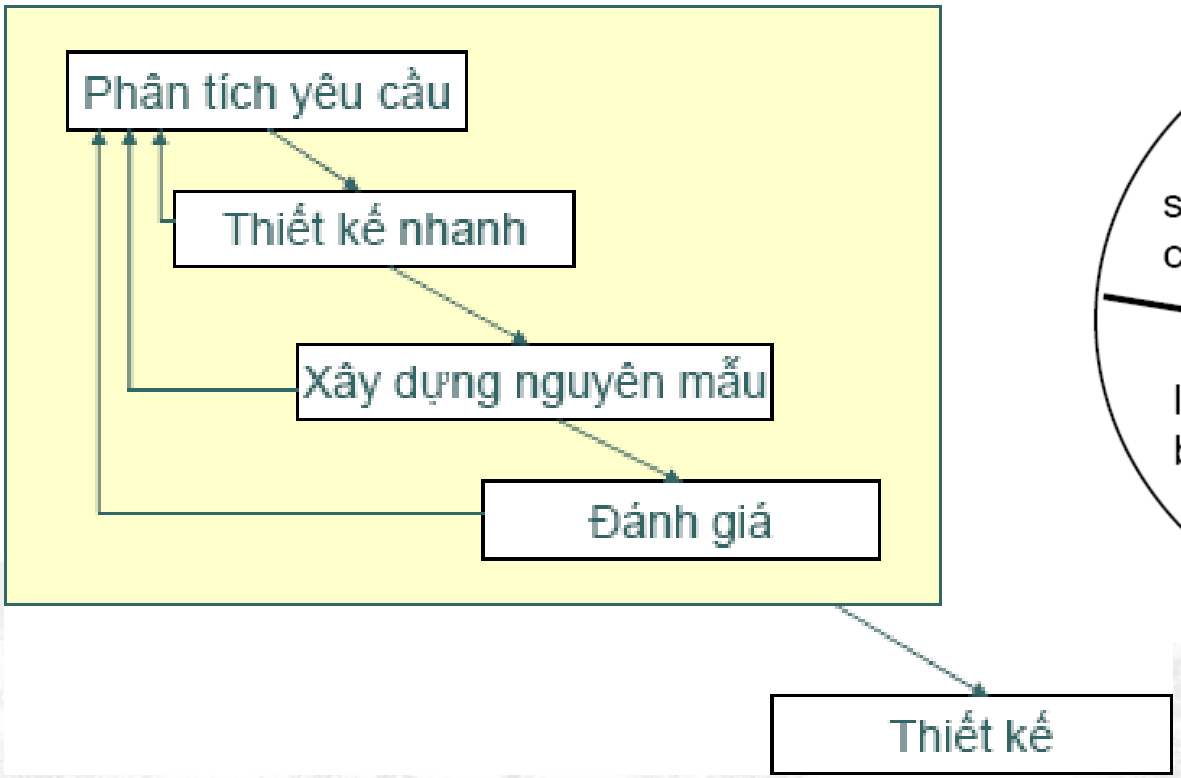
- Mô hình chữ V
  - Ưu điểm
    - Quá trình phát triển và quy trình quản lý có tính tổ chức và hệ thống
    - Hoạt động tốt cho các dự án có quy mô vừa và nhỏ.
    - Kiểm tra bắt đầu từ khi bắt đầu phát triển vì vậy sự mơ hồ được xác định ngay từ đầu.
    - Dễ dàng quản lý vì mỗi giai đoạn có các mục tiêu và mục tiêu được xác định rõ ràng.

Các mô hình phát triển phần mềm

- Mô hình chữ V
  - Nhược điểm
    - Không thích hợp cho các dự án lớn và phức tạp
    - Không phù hợp nếu các yêu cầu thường xuyên thay đổi.
    - Không có phần mềm làm việc được sản xuất ở giai đoạn trung gian.
    - Không có điều khoản cho việc phân tích rủi ro nên có sự không chắc chắn và có tính rủi ro.

Các mô hình phát triển phần mềm

- Mô hình nguyên mẫu (prototype model)



Các mô hình phát triển phần mềm

- Mô hình bản mẫu dựa trên ý tưởng xây dựng một mẫu thử ban đầu (Prototype –nguyên mẫu) và đưa cho người sử dụng xem xét; sau đó, tinh chỉnh mẫu thử qua nhiều phiên bản cho đến khi thỏa mãn yêu cầu của người sử dụng thì dừng lại.
- Mẫu thử ban đầu như là một cơ chế để nhận diện chính xác yêu cầu của khách hàng
- Mẫu thử ban đầu có thể trở thành sản phẩm. Khi các yêu cầu của người sử dụng được thỏa mãn thì cũng là lúc chúng ta đã xây dựng xong hệ thống
- Mẫu thử ban đầu có thể loại bỏ, mẫu thử chỉ có tác dụng để làm sáng tỏ yêu cầu của người sử dụng.

Các mô hình phát triển phần mềm

- Áp dụng Mô hình bản mẫu
  - Những hệ thống tương tác ở mức độ nhỏ hoặc vừa
  - Trên một phần của những hệ thống lớn (giao diện)
  - Những hệ thống có thời gian chu kỳ tồn tại ngắn.
- Ưu điểm
  - Phát hiện yêu cầu
  - Hợp thức hóa yêu cầu
  - Thiết kế giao diện
    - Giao diện trên giấy
    - Giao diện “thật”

Các mô hình phát triển phần mềm

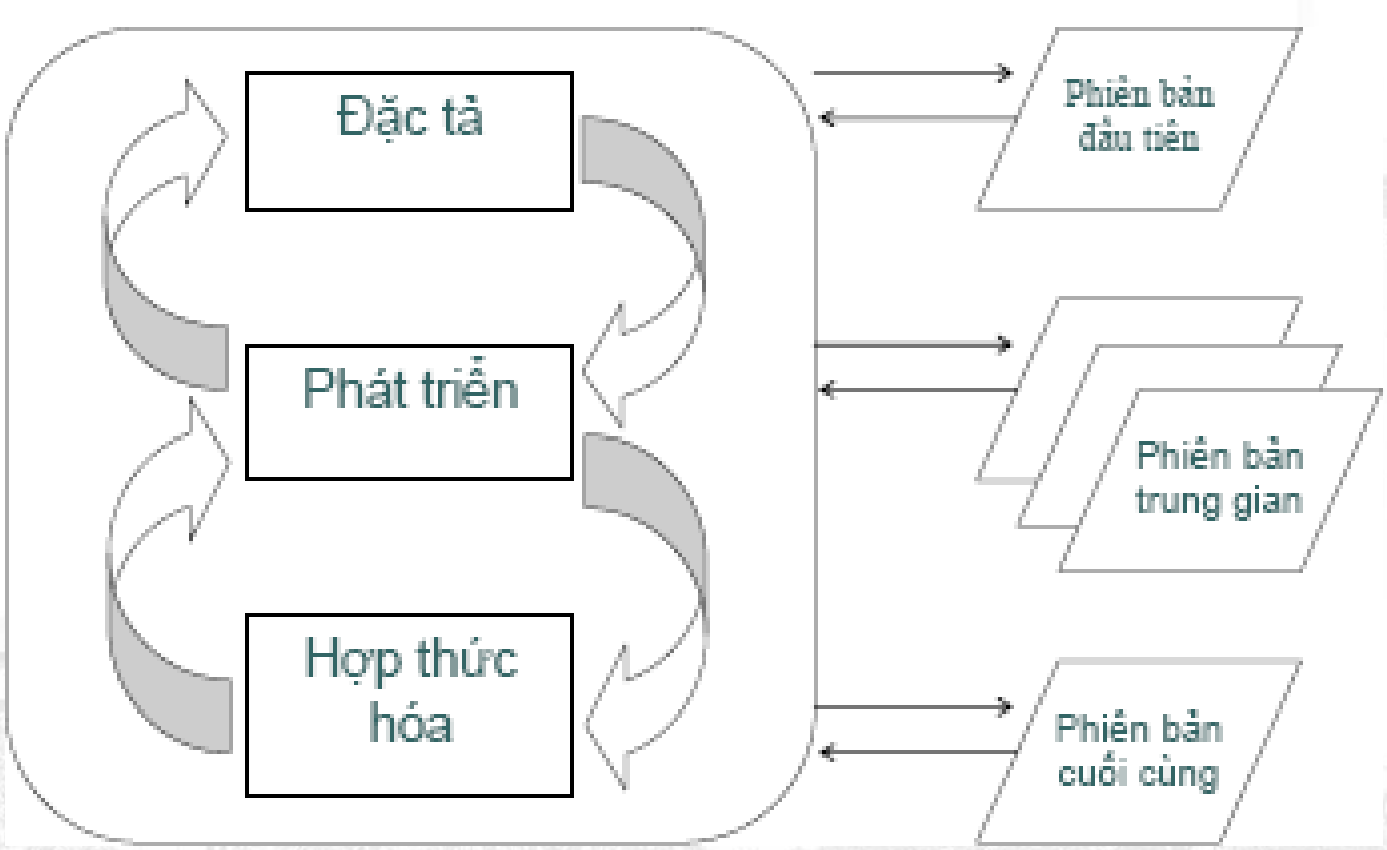
- Hạn chế
  - Thiếu tầm nhìn của cả quy trình, hệ thống rủi ro cao
    - Yêu câu không chắc chắn
    - Giao diện chưa rõ ràng
    - Chiến lược cài đặt chưa rõ ràng
  - Các hệ thống được cấu trúc một cách nghèo nàn. Những lựa không tốt có thể tích hợp trong hệ thống (Phù hợp cho nguyên mẫu nhưng không phù hợp cho hệ thống thực)
  - Yêu cầu kỹ năng đặc biệt (sử dụng công cụ CASE).
  - Khách hàng hối thúc nhà phát triển hoàn thành sản phẩm một khi thấy được các prototype đầu tiên
  - Nguyên mẫu không giống hoàn toàn hệ thống cuối cùng
  - Khách hàng sẽ có các phản ứng khác nhau

Các mô hình phát triển phần mềm

- **Mô hình tiến hóa (evolutionary model)**
  - Mô hình này thực sự cũng là một dạng dựa trên mô hình mẫu, tuy nhiên có sự khác biệt:
    - mô hình tiến hóa xây dựng nhiều phiên bản prototype liên tiếp nhau.
    - những phiên bản prototype trước sẽ được xây dựng với mục tiêu có thể tái sử dụng trong những phiên bản sau.
- Một số phần của hệ thống phần mềm có thể được xây dựng sớm ngay từ giai đoạn thực hiện phân tích yêu cầu và thiết kế.

Các mô hình phát triển phần mềm

- **Mô hình tiến hóa (evolutionary model)**



Các mô hình phát triển phần mềm

- **Mô hình tiến hóa (evolutionary model)**

  Có hai phương pháp để thực hiện mô hình này:

  - Phương pháp 1: Phát triển thăm dò
    - Mục đích là để làm việc với khách hàng và để đưa ra hệ thống cuối cùng từ những đặc tả sơ bộ ban đầu;
    - Phương pháp này thường bắt đầu thực hiện với những yêu cầu được tìm hiểu rõ ràng và sau đó, bổ sung những đặc điểm mới được đề xuất bởi khách hàng;
    - Cuối cùng, khi các yêu cầu của người sử dụng được thoả mãn thì cũng là lúc chúng ta đã xây dựng xong hệ thống.

Các mô hình phát triển phần mềm

- **Mô hình tiến hóa (evolutionary model)**

  Có hai phương pháp để thực hiện mô hình này:

  - Phương pháp 2: Loại bỏ mẫu thử
    - Mục đích là để tìm hiểu các yêu cầu của hệ thống;
    - Phương pháp này thường bắt đầu với những yêu cầu không rõ ràng và ít thông tin. Các mẫu thử sẽ được xây dựng và chuyển giao tới cho người sử dụng;
    - Từ đó, ta có thể phân loại những yêu cầu nào là thực sự cần thiết và lúc này mẫu thử không còn cần thiết nữa;
    - Như vậy, mẫu thử chỉ có tác dụng để làm sáng tỏ yêu cầu của người sử dụng..

Các mô hình phát triển phần mềm

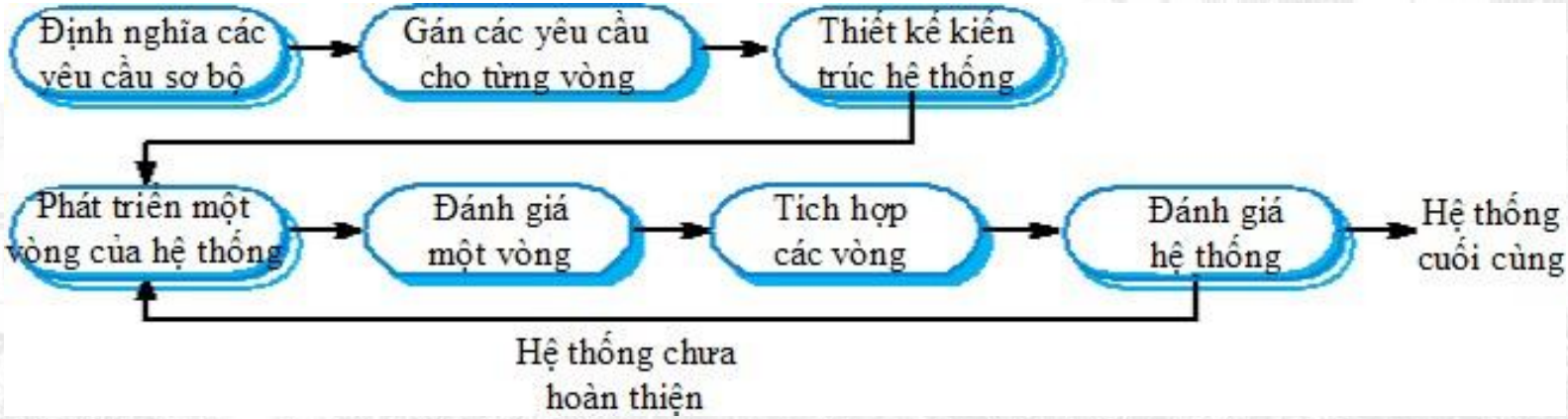
- **Mô hình tiến hóa (evolutionary model)**
  - Ưu điểm
    - Dự án vừa và nhỏ hoặc các phần của dự án phức tạp
    - Chú trọng việc tái sử dụng mẫu. Một phần của hệ thống có thể được phát triển ngay trong các giai đoạn phân tích phát triển yêu cầu và thiết kế;
    - Cho phép thay đổi yêu cầu và khuyến khích người sử dụng tham gia trong suốt chu kỳ của dự án.

Các mô hình phát triển phần mềm

- **Mô hình tiến hóa (evolutionary model)**
  - Hạn chế
    - Làm chậm quá trình phát triển yêu cầu và có thể ảnh hưởng sự chú ý đến các công việc trung gian như kiểm tra mã nguồn, thực hiện kiểm thử cấp thấp…;
    - Dễ dẫn đến kết cấu của hệ thống kém;
    - Thường thì với mô hình này, tính chặt chẽ, minh bạch của qui trình kém;
    - Chỉ nên áp dụng với những hệ thống có tương tác ở mức độ nhỏ hoặc vừa; trên một phần của những hệ thống lớn; hoặc những hệ thống có thời gian chu kỳ tồn tại ngắn.

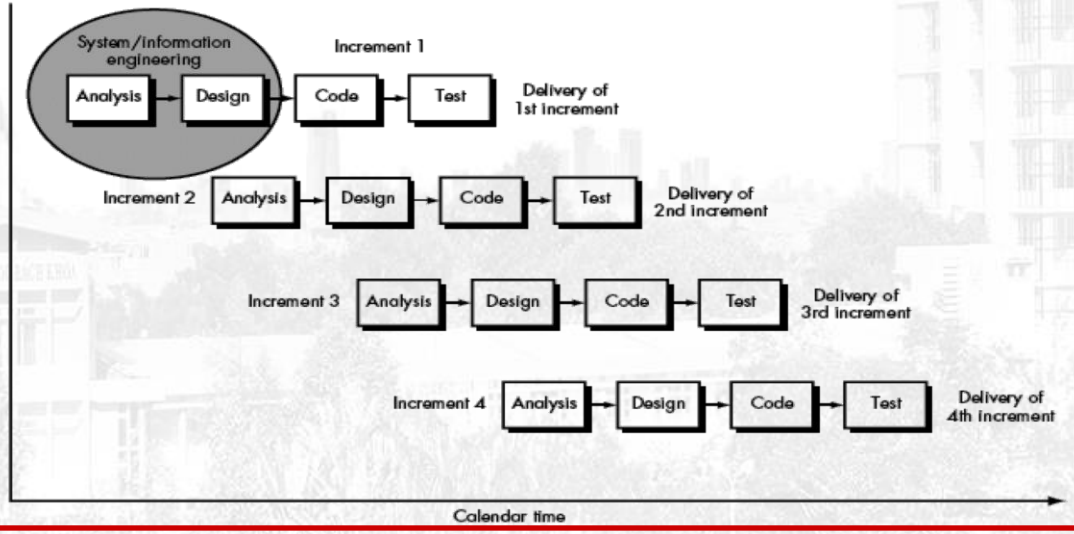
Các mô hình phát triển phần mềm

- **Mô hình phát triển lặp lại, tăng thêm**
  - Mô hình này được đề xuất dựa trên ý tưởng thay vì phải xây dựng và chuyển giao hệ thống một lần thì sẽ được chia thành nhiều vòng, tăng dần. Mỗi vòng là một phần kết quả của một chức năng được yêu cầu.
  - Các yêu cầu của người sử dụng được đánh thứ tự ưu tiên. Yêu cầu nào có thứ tự ưu tiên càng cao thì càng ở trong những vòng phát triển sớm hơn.



Các mô hình phát triển phần mềm

- **Mô hình phát triển lặp lại, tăng thêm**
  - Các bước lặp (iteration) đầu tập trung vào yêu cầu của phần mềm và thiết lập một kiến trúc ổn định cho hệ thống
  - Các bước sau tập trung vào việc xây dựng sản phẩm để cuối cùng chuyển sang giai đoạn kiểm tra hệ thống
  - Mỗi bước hiện thực một phần cụ thể trong toàn bộ yêu cầu của hệ thống



Các mô hình phát triển phần mềm

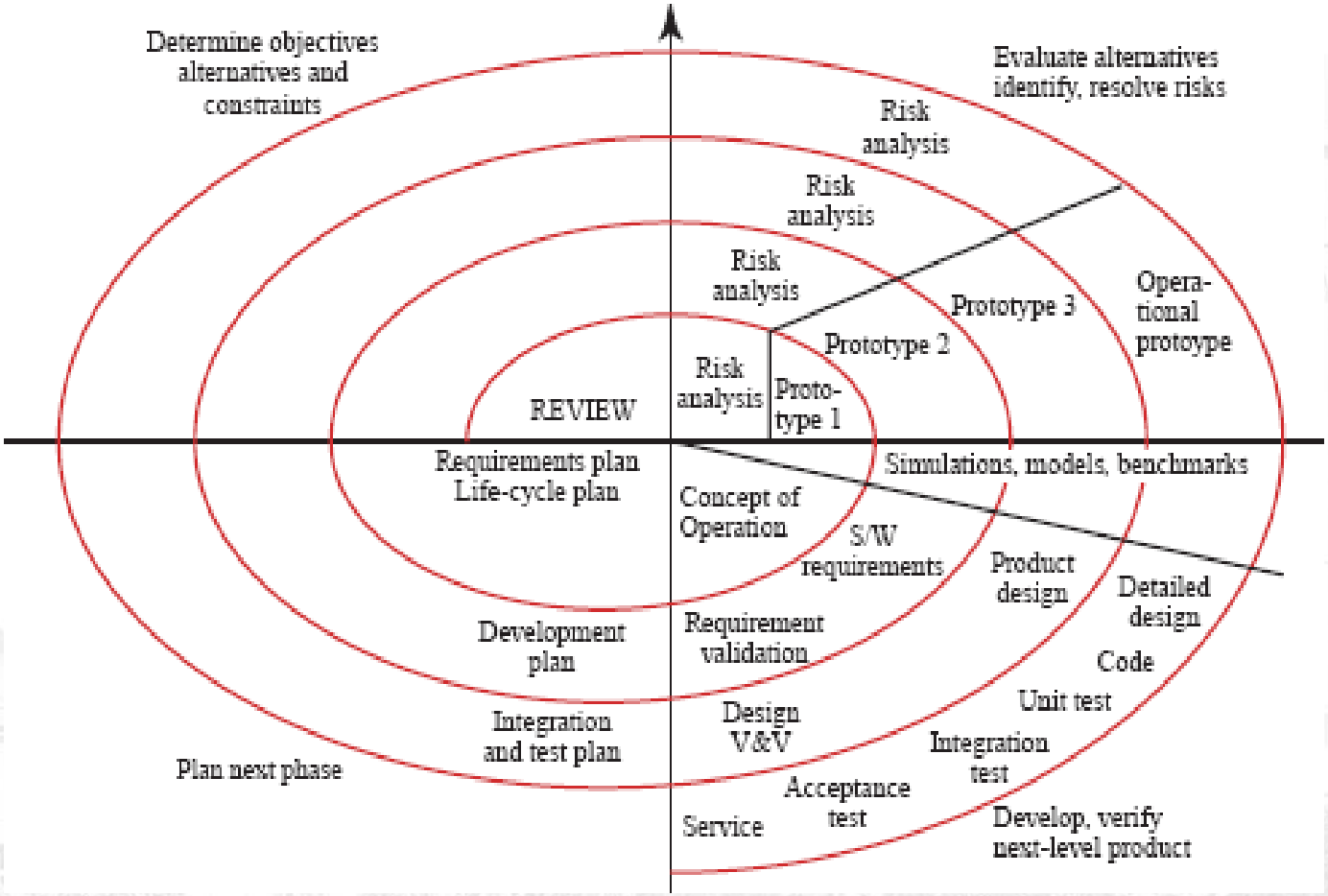
- **Mô hình phát triển lặp lại, tăng thêm**
  - Ưu điểm
    - Giảm rủi ro sớm trong chu kỳ phát triển phần mềm
      - Sau mỗi lần tăng vòng thì có thể chuyển giao kết quả thực hiện được cho khách hàng nên các chức năng của hệ thống có thể nhìn thấy sớm hơn.
      - Phản hồi của người sử dụng về những vấn đề phát sinh trong phiên bản trước được dùng để cải tiến và ngăn ngừa những vấn đề tương tự xảy ra trong những phiên bản tiếp theo.
      - Các vòng trước đóng vai trò là mẫu thử để giúp tìm hiểu thêm các yêu cầu ở những vòng tiếp theo.
      - Những chức năng của hệ thống có thứ tự ưu tiên càng cao thì sẽ được kiểm thử càng kỹ.
    - Có thể thực hiện nhiều bước đồng thời

Các mô hình phát triển phần mềm

- **Mô hình phát triển lặp lại, tăng thêm**
  - Nhược điểm:
    - Tổng chi phí lập kế hoạch phát triển cho toàn hệ thống có thể cao hơn.
      - Lưu ý, ở đây chỉ đề cập chi phí lập kế hoạch ban đầu, không bao gồm tất cả chi phí phát sinh.
      - Trong thực tế, nếu ứng dụng hợp lý, toàn bộ chi phí và thời gian cho đến khi sản phẩm được nghiệm thu có thể thấp hơn so với mô hình khác.
    - Các yêu cầu về kế hoạch và hoạt động trong qui trình cụ thể sẽ phức tạp hơn.

Các mô hình phát triển phần mềm

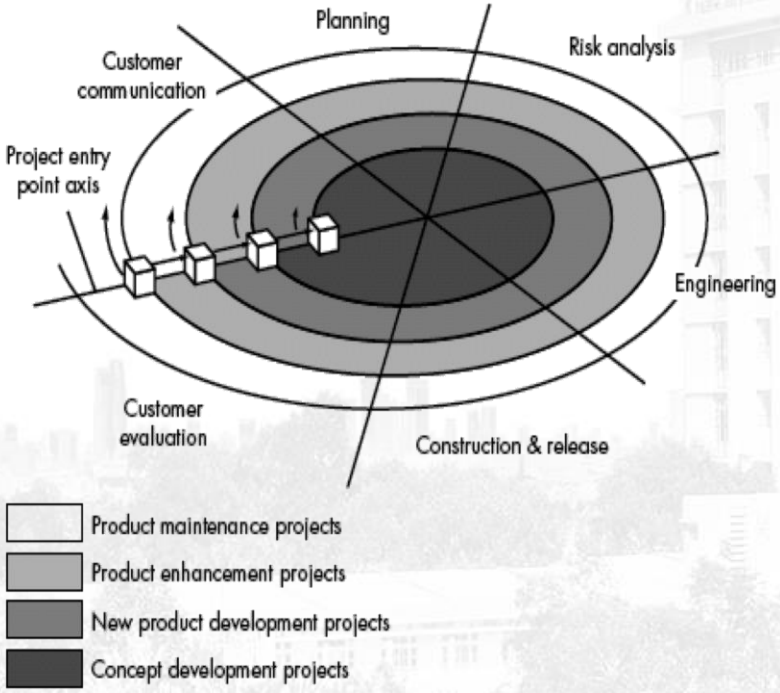
- **Mô hình xoắn ốc (Spiral Model)**



Các mô hình phát triển phần mềm

- **Mô hình xoắn ốc (Spiral Model)**
  - **Nhấn mạnh việc đánh giá rủi ro**
  - **Phần mềm được xây dựng theo nhiều chu kì**

FIGURE 2.8
A typical spiral model



Các mô hình phát triển phần mềm

- **Mô hình xoắn ốc (Spiral Model)**
  - Mỗi chu kì tương ứng với một sản phẩm của một giai đoạn phát triển phần mềm

    1. Xác định mục tiêu chất lượng cho sản phẩm được thực hiện, đồng thời xác định sự lựa chọn mua, tái sử dụng hay tự thiết kế và hiện thực các thành phần của hệ thống.

    2. Đánh giá các giải pháp, xác định các nguy cơ và tìm cách giải quyết chúng. Việc này được thực hiện bởi nhiều hoạt động khác nhau thông qua làm mẫu hay mô phỏng.

    3. Phát triển và kiểm định sản phẩm ở mức tiếp theo dựa trên kết quả định hướng được chỉ ra trong giai đoạn con số 2 (phân tích rủi ro)

    4. Kiểm duyệt tất cả các kết quả của các giai đoạn con xảy ra trước đó và lập kế hoạch cho chu kỳ lặp tiếp theo.

Các mô hình phát triển phần mềm

- **Mô hình xoắn ốc (Spiral Model)**
  - Rủi ro và giải pháp cho rủi ro
    - Thất bại về nhân sự
      - Tuyển dụng nhân sự cao cấp, đào tạo lẫn nhau, có đầy đủ nhân sự với các chức năng khác nhau.
    - Thời gian biểu và ngân sách không thực tế
      - Đánh giá thật chi tiết và phát triển dần dần, tái sử dụng, loại bỏ bớt các yêu cầu thật không cần thiết,…
    - Phát triển các chức năng không phù hợp
      - Trao đổi thường xuyên với người sử dụng có tài liệu hướng dẫn sử dụng sớm,..
    - Phát triển giao diện người dùng không thích hợp
      - Cần phân tích các công việc, xây dựng các hình mẫu trước,..
    - Thiếu yêu cầu đặt ra
      - Phát triển các phần ổn định trước
    - Vấn đề về hiệu quả
      - Cần phải mô phỏng, đo lường và thử nghiệm,..
    - Đòi hỏi vượt quá sự đáp ứng của công nghệ hiện hành

Các mô hình phát triển phần mềm

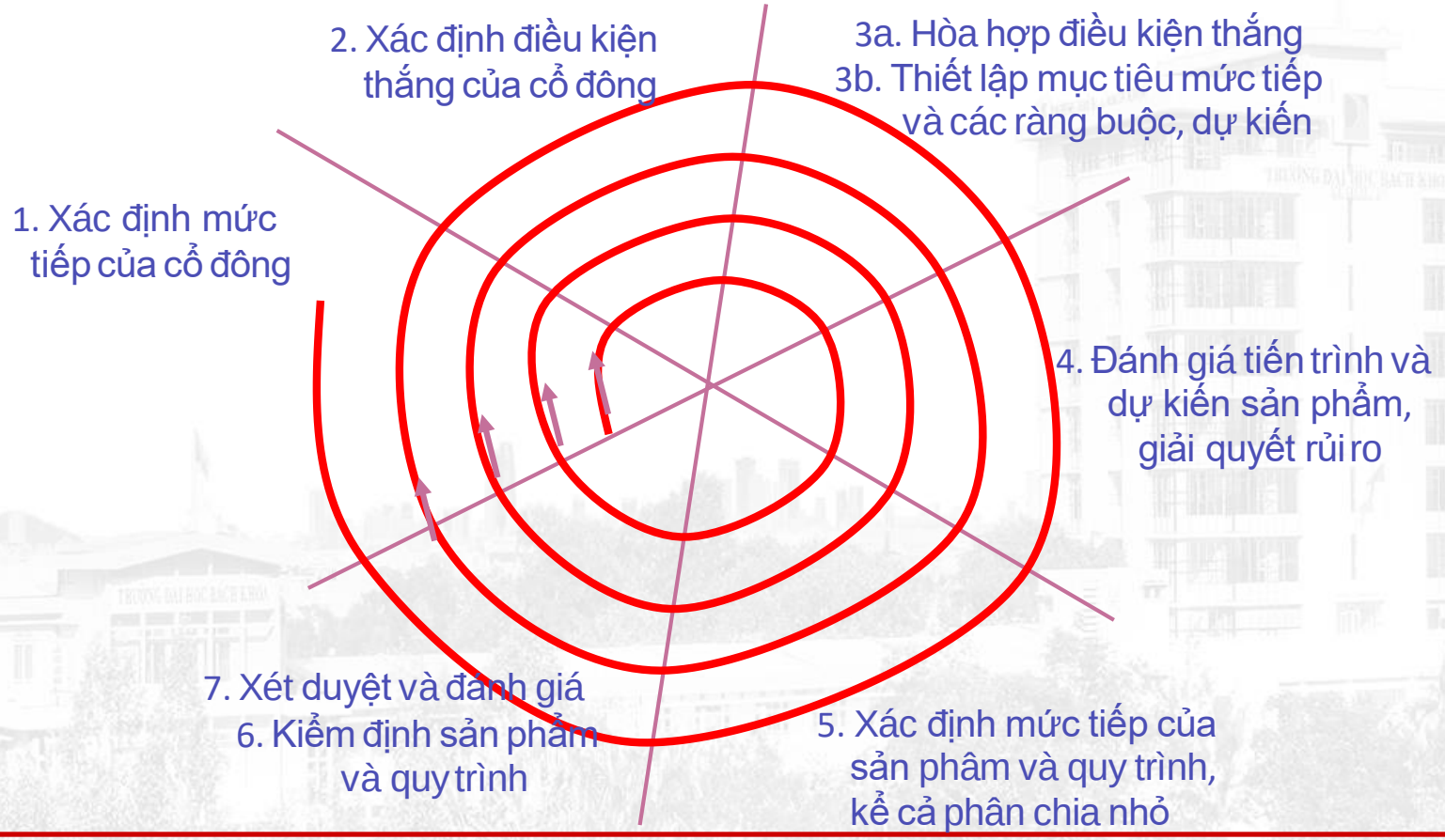
- **Mô hình xoắn ốc (Spiral Model)**
  - Ưu điểm
    - Hạn chế rủi ro sớm, Nhận được feedbacks từ khách hàng sớm
    - Dự án lớn, phức tạp, Hệ thống cần phát triển nhiều phiên bản
  - Nhược điểm
    - Yêu cầu chưa xác định rỏ ràng.
    - Khó thuyết phục khách hàng về việc kiểm soát tiến trình
    - Dựa vào những chuyên gia đánh giárủi ro, các rủi ro phải được phát hiện và quản lý
  - Ứng dụng
    - Dự án lớn có nhiều rủi ro hay sự thành công của dự án không có được sự đảm bảo nhất định; những dự án đòi hỏi nhiều tính toán, xử lý như hệ thống hỗ trợ quyết định.
    - Đội ngũ thực hiện dự án có khả năng phân tích rủi ro.

Các mô hình phát triển phần mềm

- **Mô hình xoắn ốc (Spiral Model)**
  - Mô hình xoắn ốc WINWIN
    - Nhằm thỏa hiệp giữa người phát triển và khách hàng, cả hai cùng "Thắng" (win-win)
    - Khách thì có phần mềm thỏa mãn yêu cầu chính
    - Người phát triển thì có kinh phí thỏa đáng và thời gian hợp lý
    - Các hoạt động chính trong xác định hệ thống:
    - Xác định cổ đông (stakeholders)
    - Xác định điều kiện thắng của cổ đông
    - Thỏa hiệp điều kiện thắng của các bên liên quan.

Các mô hình phát triển phần mềm

- **Mô hình xoắn ốc (Spiral Model)**
  - **Mô hình xoắn ốc WINWIN**



Các mô hình phát triển phần mềm

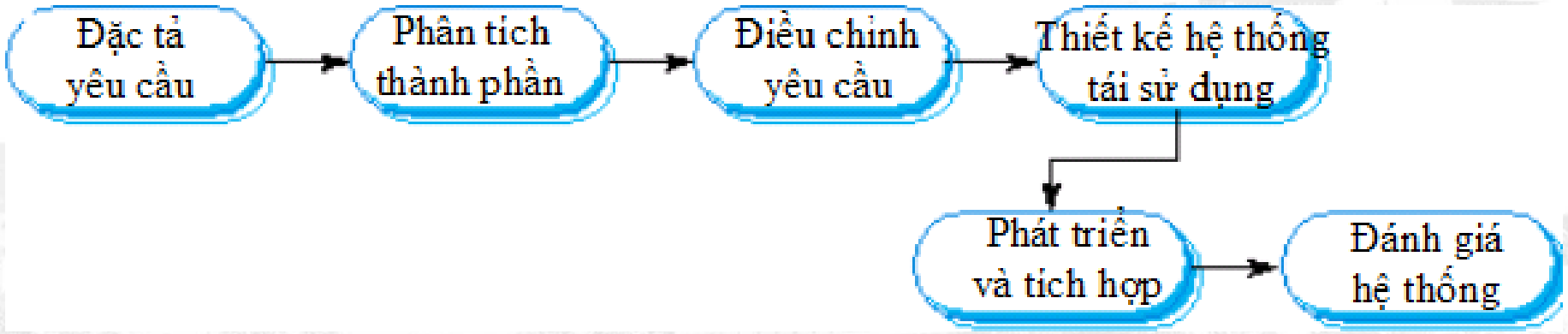
- **Mô hình phát triển đồng thời (The concurrent development model)**
  - Xác định mạng lưới những hoạt động đồng thời
  - (Network of concurrent activities)
  - Các sự kiện (events) xuất hiện theo điều kiện vận động trạng thái trong từng hoạt động
  - Dùng cho mọi loại ứng dụng và cho hình ảnh khá chính xác về trạng thái hiện trạng của dự án
  - Thường dùng trong phát triển các ứng dụng khách/chủ (client/server applications): system and componets are developed concurrently

Các mô hình phát triển phần mềm

- **Công nghệ phần mềm dựa thành phần**
  - Mô hình này dựa trên kỹ thuật tái sử dụng một cách có hệ thống
  - Hệ thống được tích hợp từ nhiều thành phần đang tồn tại hoặc các thành phần thương mại COTS (Commercial-Off-The-Shelf).

Các mô hình phát triển phần mềm

- **Công nghệ phần mềm dựa thành phần**
  - Các trạng thái chính của quy trình bao gồm:
    - Phân tích thành phần sẵn có
    - Điều chỉnh yêu cầu
    - Thiết kế hệ thống với kỹ thuật tái sử dụng
    - Xây dựng và tích hợp hệ thống

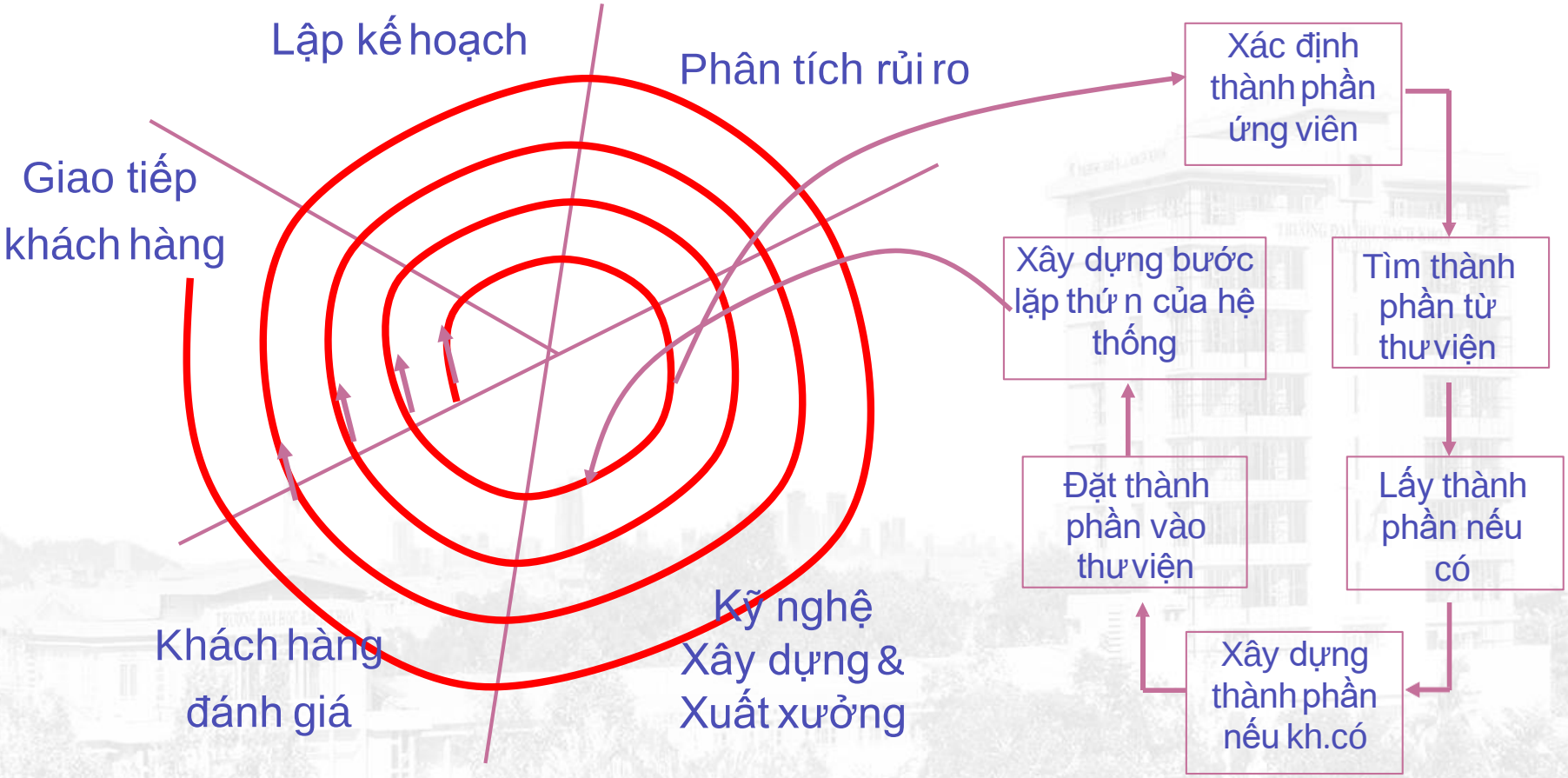


Các mô hình phát triển phần mềm

- **Mô hình theo thành phần (Component-based model)**
  - Gắn với những công nghệ hướng đối tượng (Object- oriented technologies) qua việc tạo các lớp (classes) có chứa cả dữ liệu và giải thuật xử lý dữ liệu
  - Có nhiều tương đồng với mô hình xoắn ốc
  - Với ưu điểm tái sử dụng các thành phần qua Thư viện / kho các lớp: tiết kiệm 70% thời gian, 80% giá thành, chỉ số sản xuất 26.2/16.9
  - Với UML như chuẩn công nghiệp đang triển khai

Các mô hình phát triển phần mềm

- **Mô hình theo thành phần**



Các mô hình phát triển phần mềm

- **Mô hình hình thức (Formal model)**
  - Còn gọi là CNPM phòng sạch (Cleanroom SE)
  - Tập hợp các công cụ nhằm đặc tả toán học phần mềm máy tính từ khâu định nghĩa, phát triển đến kiểm chứng
  - Giúp kỹ sư phần mềm phát hiện và sửa các lỗi khó
  - Thường dùng trong phát triển SW cần độ an toàn rất cao (y tế, hàng không, . . .)

Các mô hình phát triển phần mềm

- **Mô hình hình thức: Điểm yếu ?**
  - Cần nhiều thời gian và công sức để phát triển
  - Phí đào tạo cao vì ít người có nền căn bản cho áp dụng mô hình hình thức
  - Khó sử dụng rộng rãi vì cần kiến thức toán và kỹ năng của khách hàng

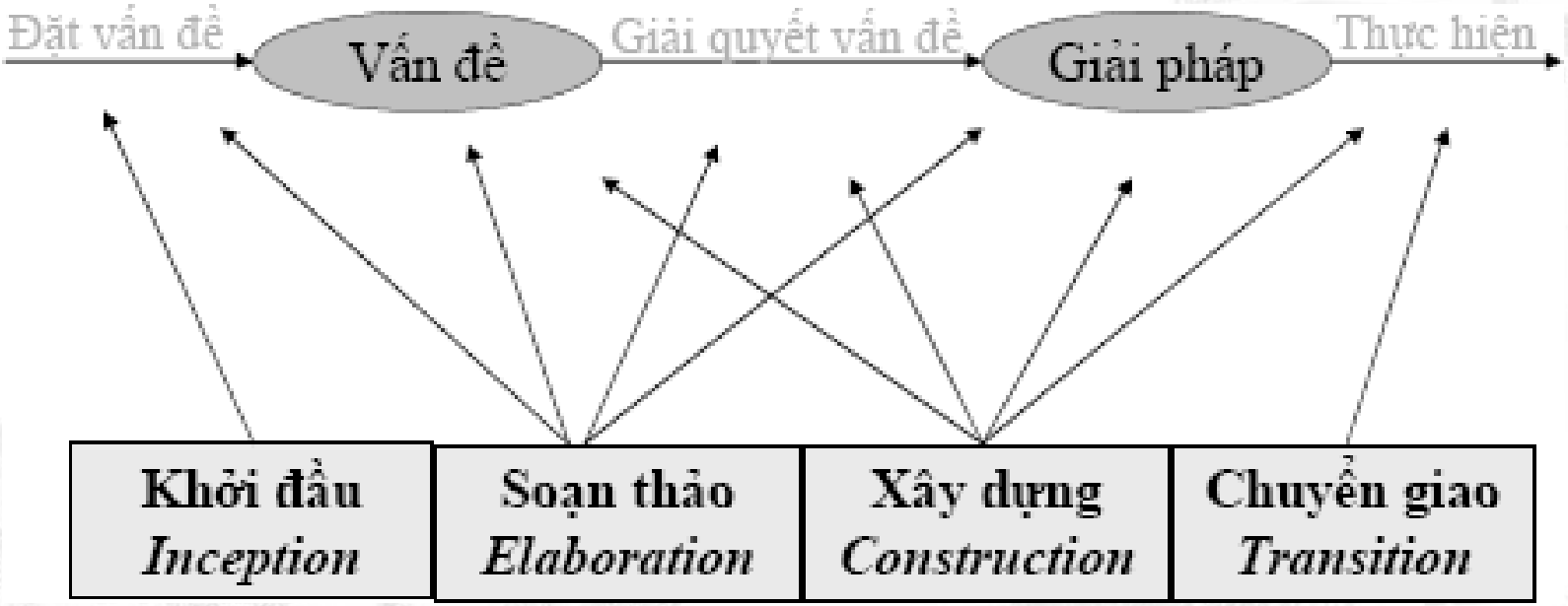
Các mô hình phát triển phần mềm

Mô hình hợp nhất

- Tiến trình hợp nhất có thể được nhìn dưới hai góc nhìn khác nhau
  - Góc nhìn quản lý:quan tâm đến lĩnh vực kinh tế, chiến thuật, con người
    - Tiền trình gồm 4 giai đoạn
  - Góc nhìn kỹ thuật: quan tâm đến công nghệ, kiểm tra chất lượng, phương pháp
    - Tiến trình gồm nhiều bước lặp

Các mô hình phát triển phần mềm

- **Mô hình hợp nhất**
  - **Góc nhìn quản lý**



Các mô hình phát triển phần mềm

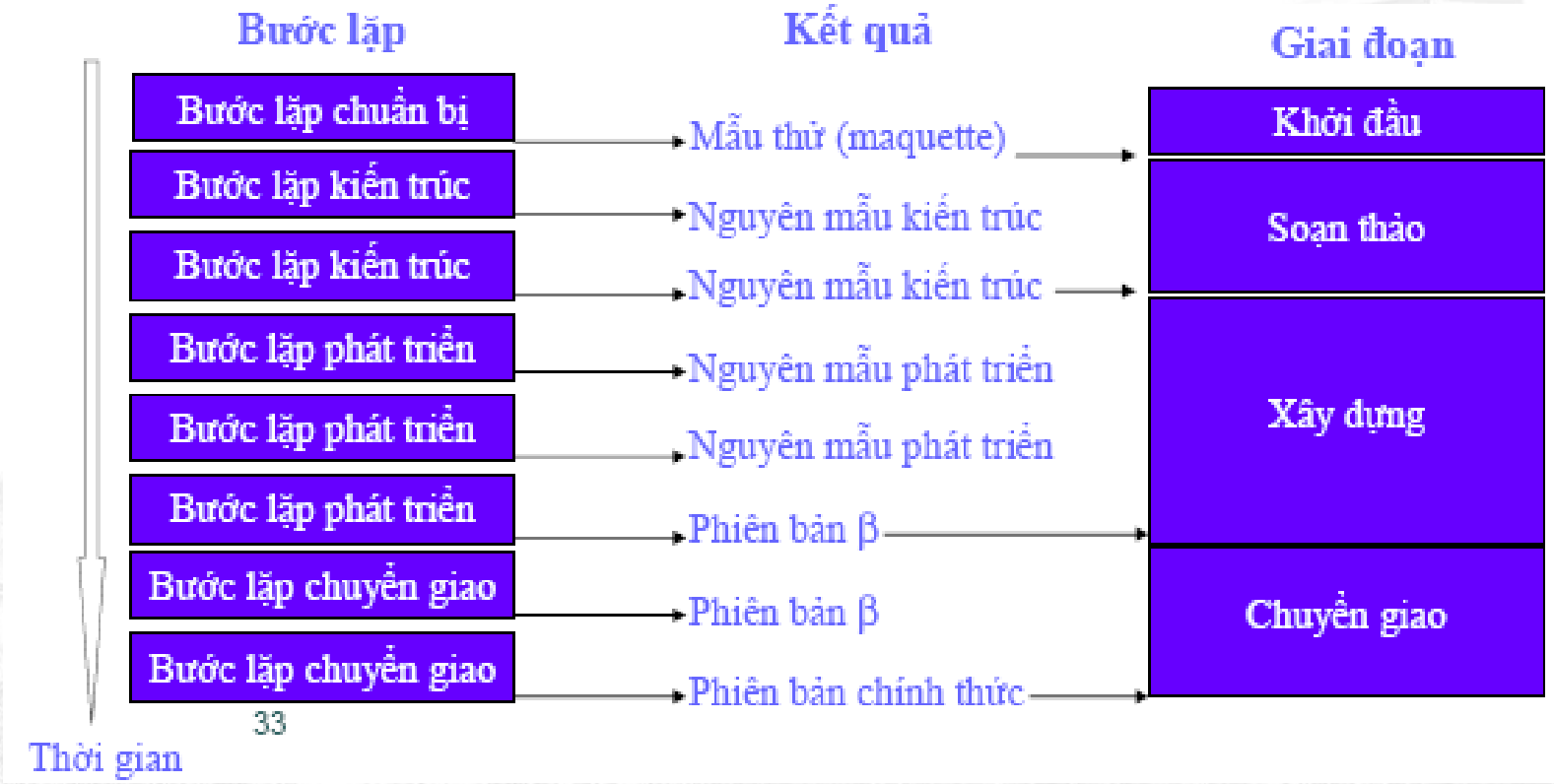
- **Mô hình hợp nhất**
  - **Góc nhìn kỹ thuật: các bước lặp**
    - **Đặc tả**
    - **Phân tích**
    - **Thiết kế**
    - **Mã hóa**
    - **Kiểm thử**
    - **Cài đặt**
  - **Mỗi bước lặp là một tiến trình thác đổ**

Các mô hình phát triển phần mềm

- **Mô hình hợp nhất**
  - **Góc nhìn kỹ thuật**

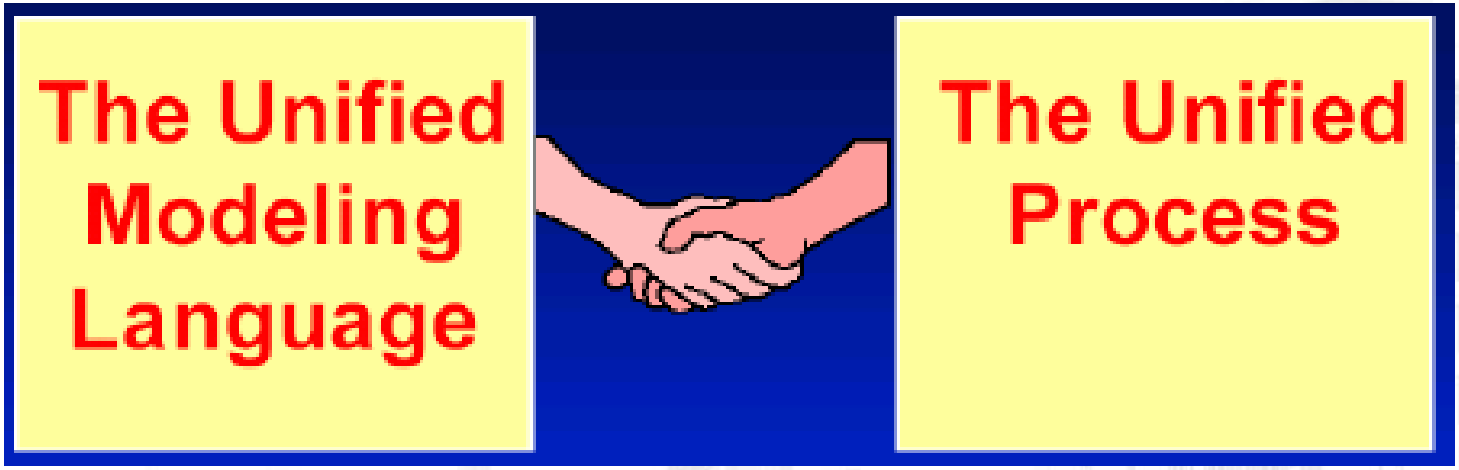
Các mô hình phát triển phần mềm

- **Mô hình hợp nhất**
  - **Kết hợp hai góc nhìn**



Các mô hình phát triển phần mềm

- **Mô hình hợp nhất và UML**

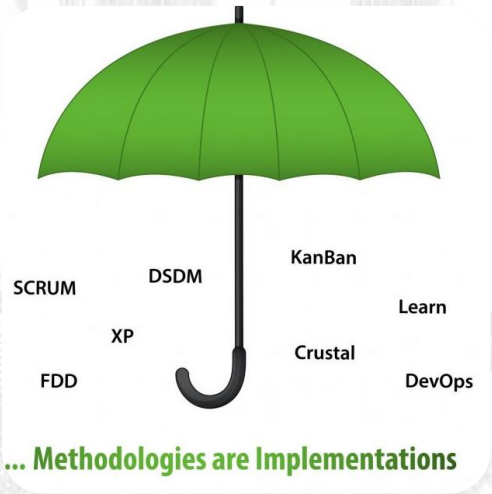


Các mô hình phát triển phần mềm

- **Mô hình Agile**
  - Gọi tắt là Agile đã trở nên rất phổ biến trong ngành phát triển phần mềm.
  - Agile thực chất là một triết lý hay một khung tư duy để nhanh chóng thích ứng và phản hồi với thay đổi, từ đó đạt được thành công trong một môi trường liên tục biến động và không chắc chắn.
  - Được mô tả bằng **4 giá trị** và **12 nguyên lý cốt lõi** trong Tuyên ngôn phát triển phần mềm linh hoạt hay Tuyên ngôn Agile

Các mô hình phát triển phần mềm

- **Agile Software Development là gì?**
  - là một thuật ngữ chung chỉ tất cả các kỹ thuật và phương pháp phát triển phần mềm theo triết lý Agile.
  - Các phương pháp Agile sẽ làm nhiệm vụ định nghĩa rõ hơn để các cá nhân và tập thể dễ dàng vận dụng vào bối cảnh công việc của mình.
  - khuyến khích việc lập kế hoạch thích ứng, phát triển tăng dần, chuyển giao sớm và cải tiến liên tục nhằm thích ứng nhanh với sự thay đổi – *một điểm yếu cố hữu của các phương pháp phát triển phần mềm truyền thống (waterfall).*



Các mô hình phát triển phần mềm

- **Lợi ích của Quản lý dự án theo mô hình Agile**
  - Lợi ích lớn nhất và dễ nhận thấy mà mô hình Agile mang lại là *rút ngắn thời gian cho ra mắt sản phẩm*, thường từ 1-2 tuần sẽ phát hành ra một tính năng cơ bản, mang đến giá trị sử dụng cho khách hàng.
  - Điểm vượt trội khác là *khả năng thích ứng nhanh* với sự thay đổi từ khách hàng.
  - Quy mô team dự án Agile hoàn hảo sẽ nằm trong giới hạn 7±2, tối đa 9 member, tối thiểu là 5

Các mô hình phát triển phần mềm

- **Agile Manifesto - Tuyên ngôn Agile gồm 4 giá trị cốt lõi**

1. Cá nhân và tương tác hơn là quy trình và công cụ
2. Phần mềm chạy tốt hơn là tài liệu đầy đủ
3. Cộng tác với khách hàng hơn là đàm phán hợp đồng
4. Phản hồi với thay đổi hơn là bám sát kế hoạch

Các mô hình phát triển phần mềm

- **12 nguyên tắc Agile trong Tuyên ngôn Agile**

1. Thỏa mãn yêu cầu của khách hàng - là ưu tiên hàng đầu thông qua việc chuyển giao những sản phẩm giá trị trong thời gian sớm và liên tục.
2. Sẵn sàng cho những thay đổi - thậm chí những thay đổi này xuất hiện muộn. Quy trình Agile linh hoạt trong việc ứng phó với sự thay đổi từ khách hàng, gia tăng tính cạnh tranh cho khách hàng.
3. Cung cấp phần mềm hoạt động được trong thời gian ngắn từ 1 vài tuần đến 1 vài tháng, với sự ưu tiên thời gian ngắn hơn.

Các mô hình phát triển phần mềm

- **12 nguyên tắc Agile trong Tuyên ngôn Agile**

4. Người kinh doanh và người lập trình phải làm việc cùng nhau mỗi ngày trong suốt dự án.
5. Xây dựng các dự án xung quanh cá nhân có động lực. Cho họ môi trường làm việc thuận lợi và sự hỗ trợ cần thiết. Hãy có niềm tin rằng họ sẽ làm tốt công việc của mình.
6. Đối thoại trực tiếp mặt đối mặt là phương pháp hữu hiệu nhất trong việc truyền đạt thông tin.
7. Phần mềm chạy được là thước đo chính của tiến độ dự án.

Các mô hình phát triển phần mềm

- **12 nguyên tắc Agile trong Tuyên ngôn Agile**

8. Phát triển bền vững và duy trì việc phát triển liên tục. Các nhà tài trợ, người phát triển và người dùng nên có thể duy trì tốc độ không đổi vô thời hạn.
9. Liên tục quan tâm đến kỹ thuật và thiết kế để tăng cường tính linh hoạt
10. Đơn giản - nghệ thuật tối đa hóa số lượng công việc không làm - là điều cần thiết
11. Nhóm tự tổ chức. Các kiến trúc, yêu cầu và thiết kế tốt nhất xuất hiện từ các nhóm tự tổ chức.
12. Tự phản ánh thường xuyên. Trong khoảng thời gian đều đặn, nhóm phản ánh về cách trở nên hiệu quả hơn, sau đó điều chỉnh cho phù hợp.

Các mô hình phát triển phần mềm

- **Các phương pháp Agile phổ biến**
  - Scrum
  - Extreme Programming (XP)
  - Adaptive Software Development (ASD)
  - Dynamic System Development Method (DSDM)
  - Feature Driven Development (FDD)
  - Lean Software Development
  - …
  - Scrum là phương pháp phổ biến nhất được áp dụng hiện nay. Một số doanh nghiệp sử dụng mô hình Scrum:

Các mô hình phát triển phần mềm

- **Những lợi ích trong cách tiếp cận theo mô hình Agile**
  - Khách hàng thường xuyên có cơ hội thấy và trải nghiệm thực tế sản phẩm được chuyển giao từng giai đoạn, giúp họ có những quyết định và thay đổi trong quá trình phát triển sản phẩm
  - Khách hàng có nhận thức mạnh mẽ về quyền sở hữu trong quá trình làm việc trực tiếp với nhóm dự án
  - Với phương pháp quản lý Agile, sản phẩm có thể chuyển giao nhanh với những tính năng hoàn thiện cơ bản
  - Sự phát triển tập trung vào người dùng cuối cùng hơn, vì sự tương tác thường xuyên và trực tiếp với khách hàng trong quá trình thực hiện

Các mô hình phát triển phần mềm

- **Một vài bất cập trong phương pháp Agile**
  - Mức độ tham gia của khách hàng rất cao đôi khi là vấn đề cho một số khách hàng – những người không thật sự hứng thú với cách tiếp cận này
  - Mô hình Agile thật sự hiệu quả khi các team member hoàn toàn tập trung vào dự án
  - Giao hàng đúng tiến độ và việc thường xuyên thay đổi mức độ ưu tiên, có khả năng dẫn đến một số tính năng không được chuyển giao đúng thời hạn
  - Phát sinh thêm một số sprint nếu cần thiết và ảnh hưởng đến chi phí dự án

Các mô hình phát triển phần mềm

- **Kết luận**
  - Có nhiều mô hình phát triển phần mềm
  - Kết hợp nhiều mô hình cho một dự án
    - Hệ thống phức tạp, chia dự án thành các hệ thống con
    - Mô hình xoắn ốc hay mô hình hợp nhất cho toàn bộ dự án
    - Mỗi hệ thống con có thể áp dụng một mô hình khác nhau
      - Mô hình nguyên mẫu cho các hệ thống con phức tạp
      - Mô hình thác nước cho các hệ thống con khác

Bài tập

- Đánh giá các mô hình phát triển phần mềm
  - Ưu, nhược điểm
  - Ứng dụng