



# Mô hình phát triển phần mềm(2)

---

Lê Thị Mỹ Hạnh

Khoa Công nghệ Thông tin

Trường Đại học Bách khoa

Đại học Đà Nẵng



# 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
- Sản phẩm: tài liệu đặc tả yêu cầu



# Các hoạt động phát triển phần mềm

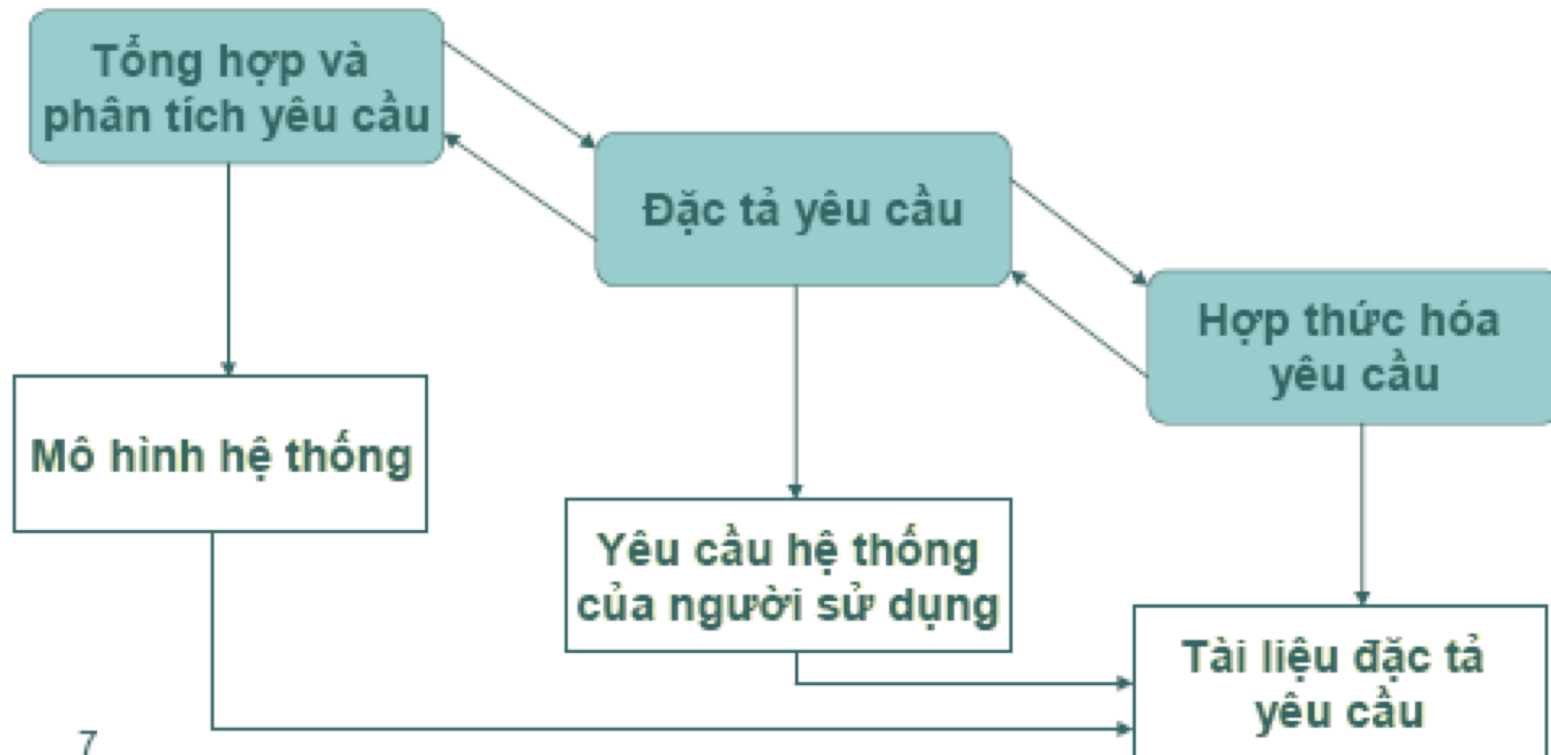
## ■ Phân tích và đặc tả yêu cầu (2)

### ■ 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)



7



# Các hoạt động phát triển phần mềm

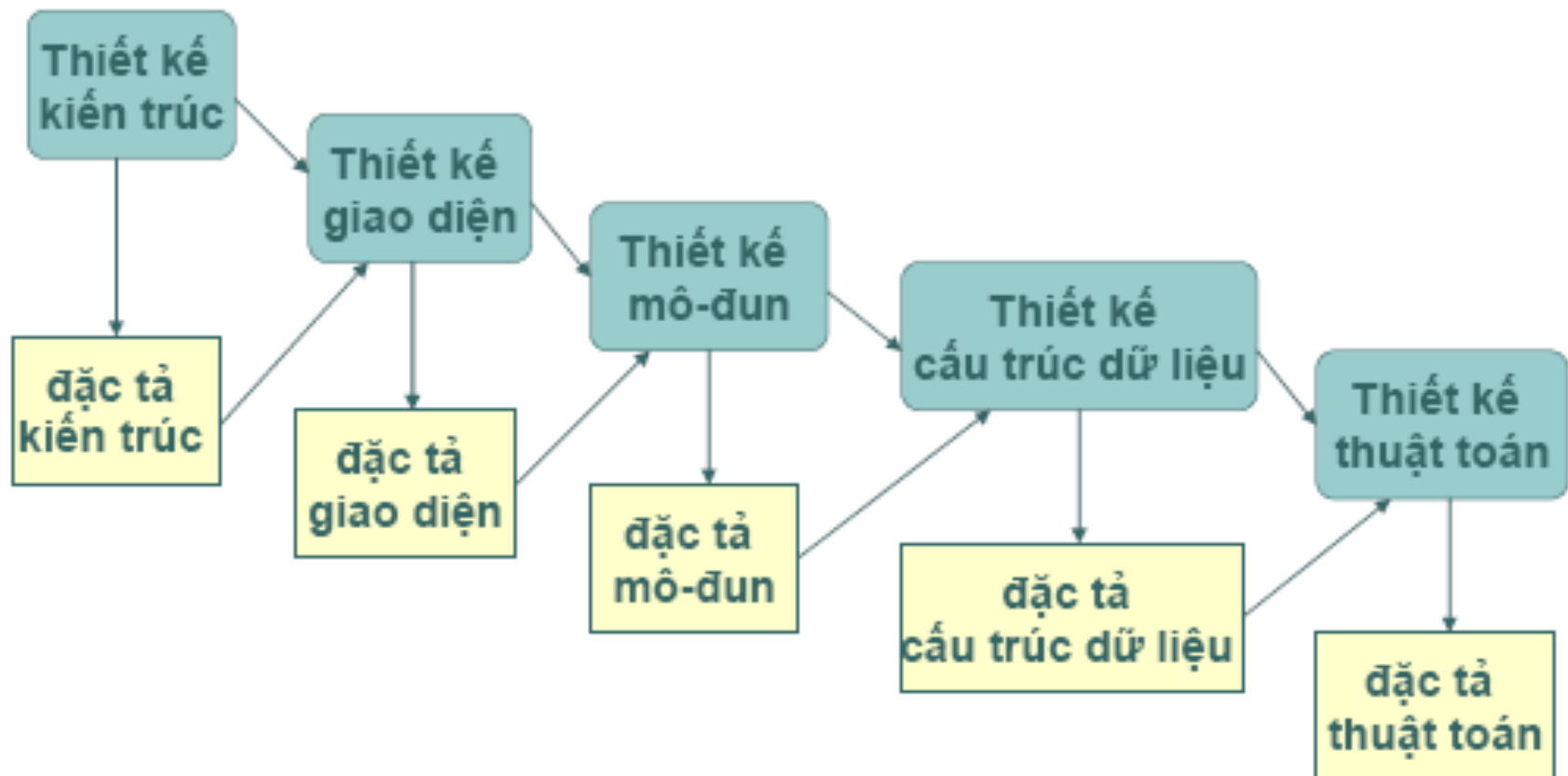
## ■ **Thiết kế (1)**

- Chuyển từ tài liệu các yêu cầu thành cấu trúc logic có thể cài đặt được
- Giải pháp cho vấn đề đã được đặt ra
- 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 liệu
  - 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

- **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
      - insertions/invariants
      - 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

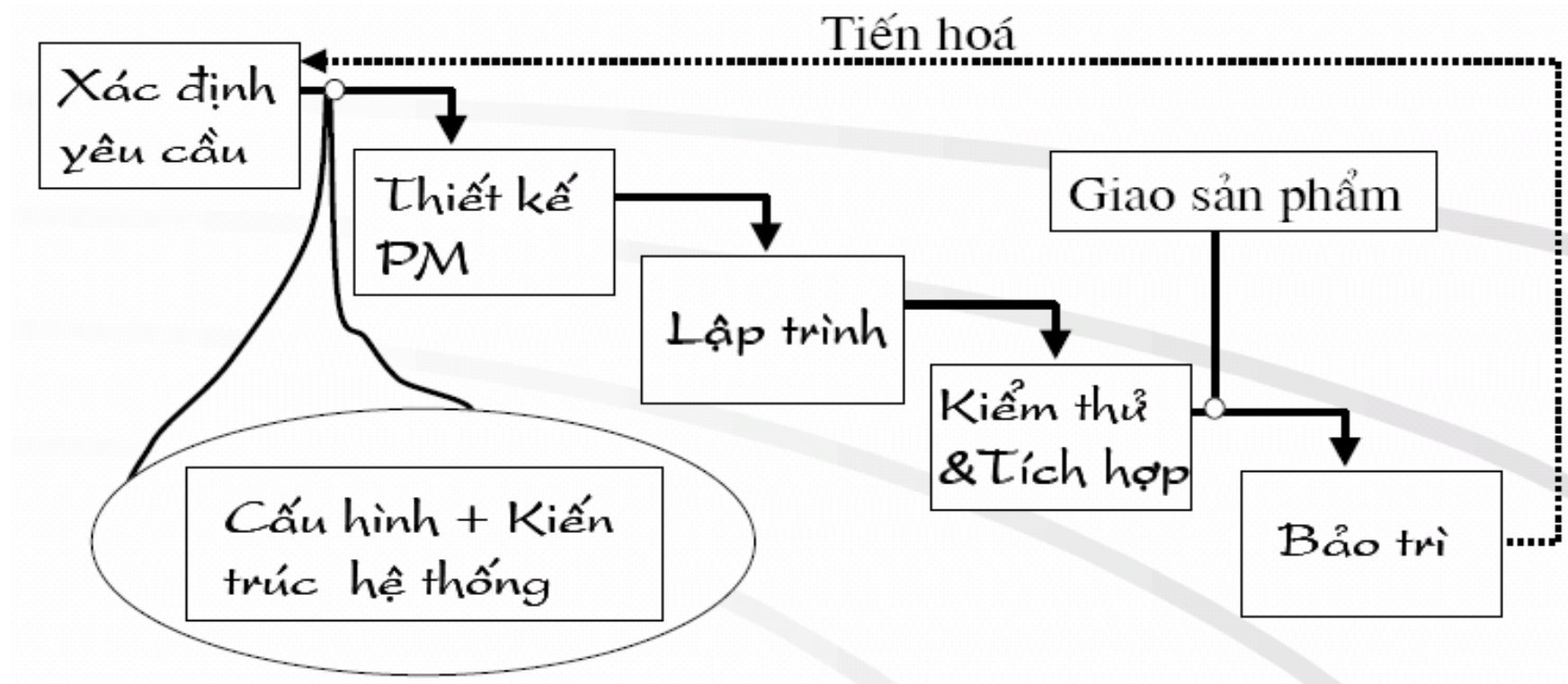


# Các mô hình phát triển phần mềm

- 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

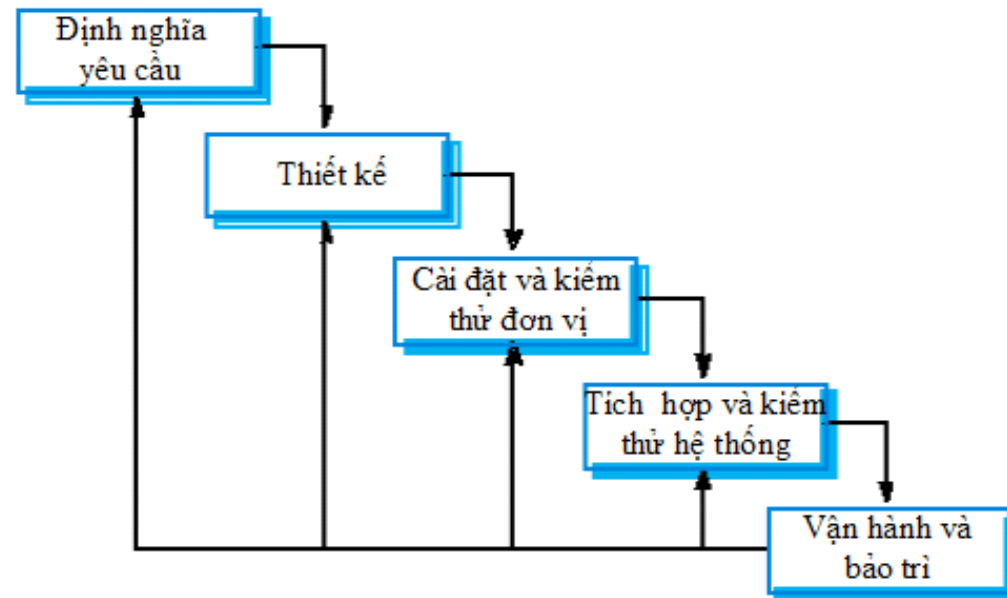
## ■ Mô hình thác nước





# Các mô hình phát triển phần mềm

## ■ Mô hình thác nước



- Tách biệt giữa các pha, tiến hành tuần tự
- Chưa 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

- Có sớm nhất và được sử dụng rộng rãi
- Trong mô hình thác nước, năm pha trên phải được thực hiện tách biệt và tuần tự
- Mô hình này chỉ thích hợp khi các yêu cầu rõ ràng và những thay đổi được giới hạn
- Bảo trì thuận lợi



# Các mô hình phát triển phần mềm

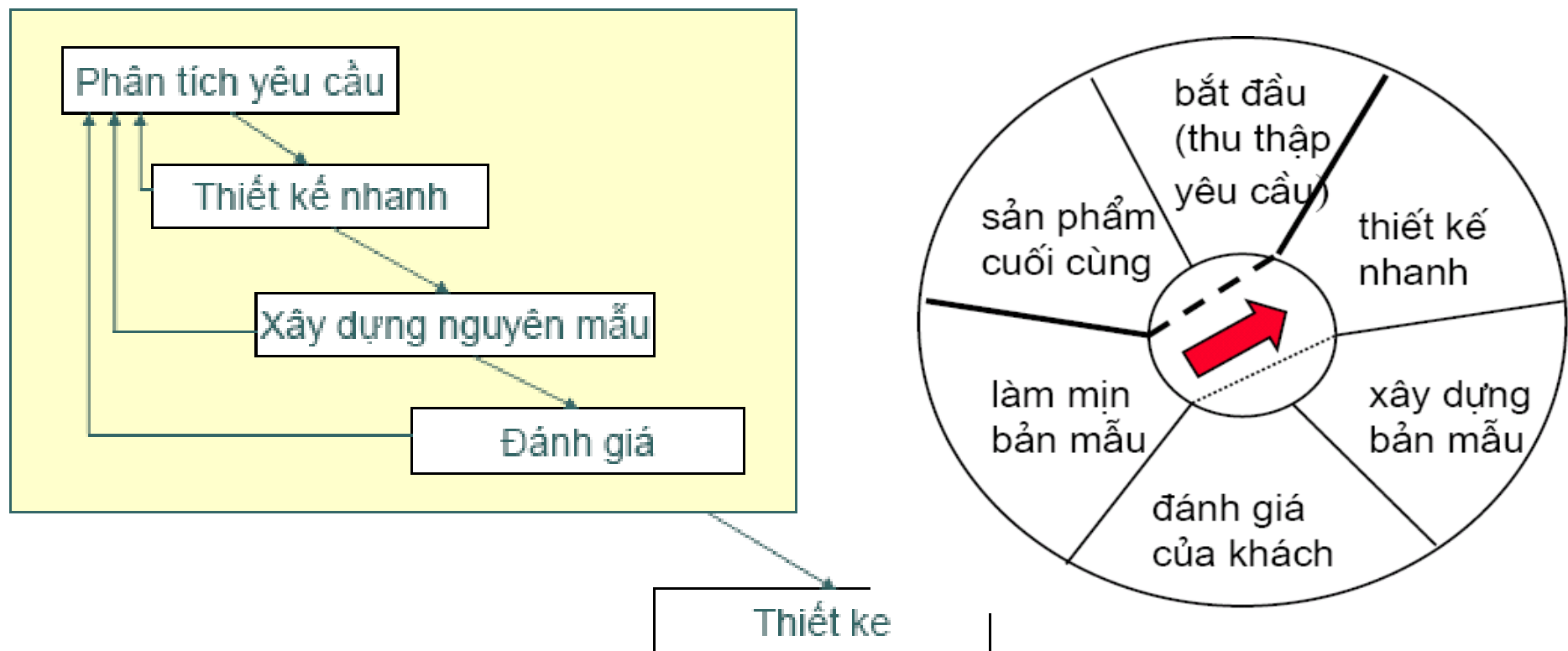
## ■ Mô hình thác nước

### ■ 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

# Các mô hình phát triển phần mềm

## ■ Mô hình nguyên mẫu





# 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

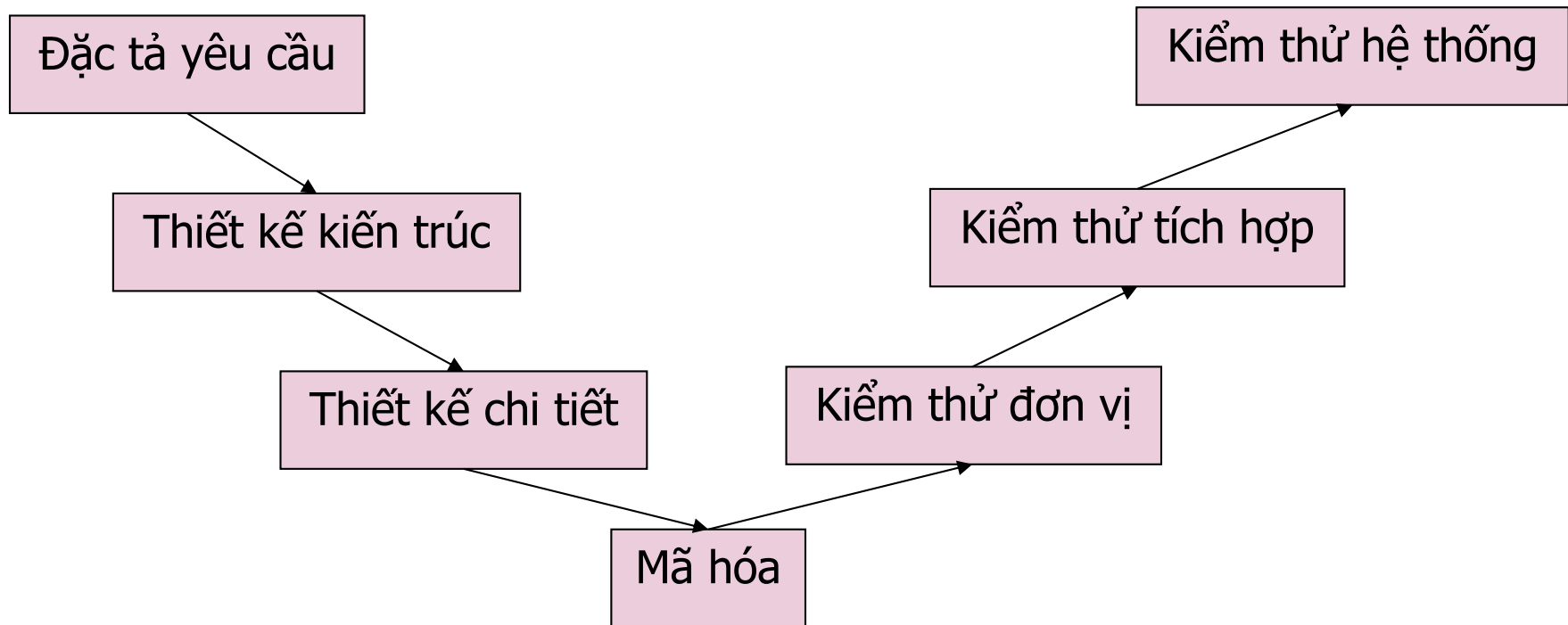
## ■ Hạn chế

- Thiếu tầm nhìn của cả quy trình, Hệ thống rủi ro cao
  - Yêu cầu 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 chọn 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 chữ V

- 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.



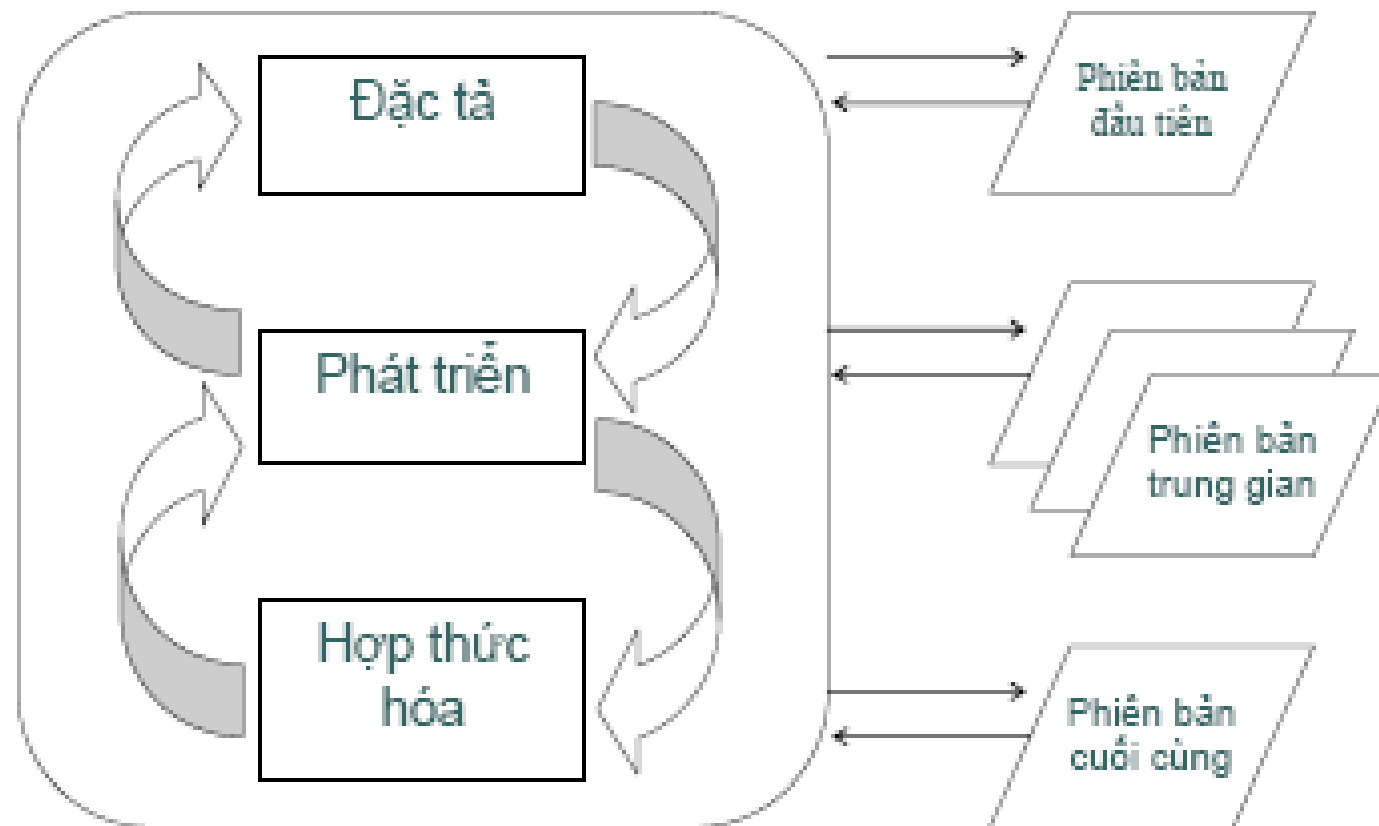
# Các mô hình phát triển phần mềm

## ■ **Mô hình 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

### ■ Ưu điểm

- Dự án vừa và nhỏ
- Các phần của dự án phức tạp
- Các hệ thống có thời gian sống ngắn

### ■ Hạn chế

- Cấu trúc của hệ thống tồi, Tiến trình không rõ ràng

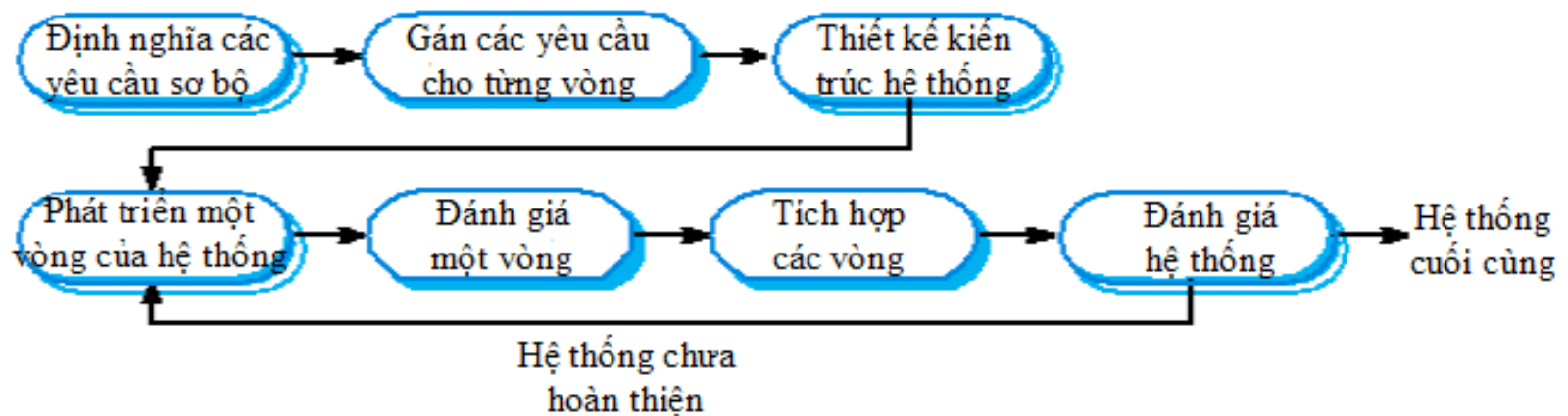
### ■ Ứng dụng

- Hệ thống tương tác nhỏ và vừa; phần GUI của những hệ thống lớn; những hệ thống cần chu kỳ phát triển ngắn.
- Đội ngũ phát triển không quen thuộc với lĩnh vực của dự án.

# Các mô hình phát triển phần mềm

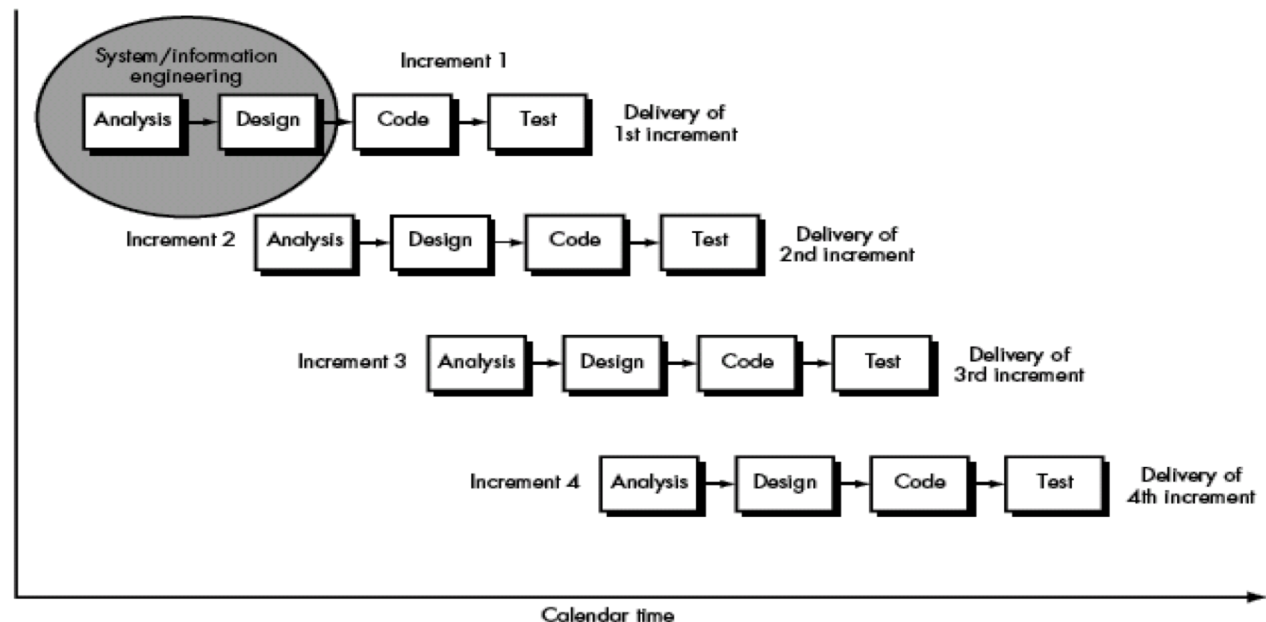
## ■ Mô hình phát triển lặp lại, tăng thêm

- 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.



# 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

- 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

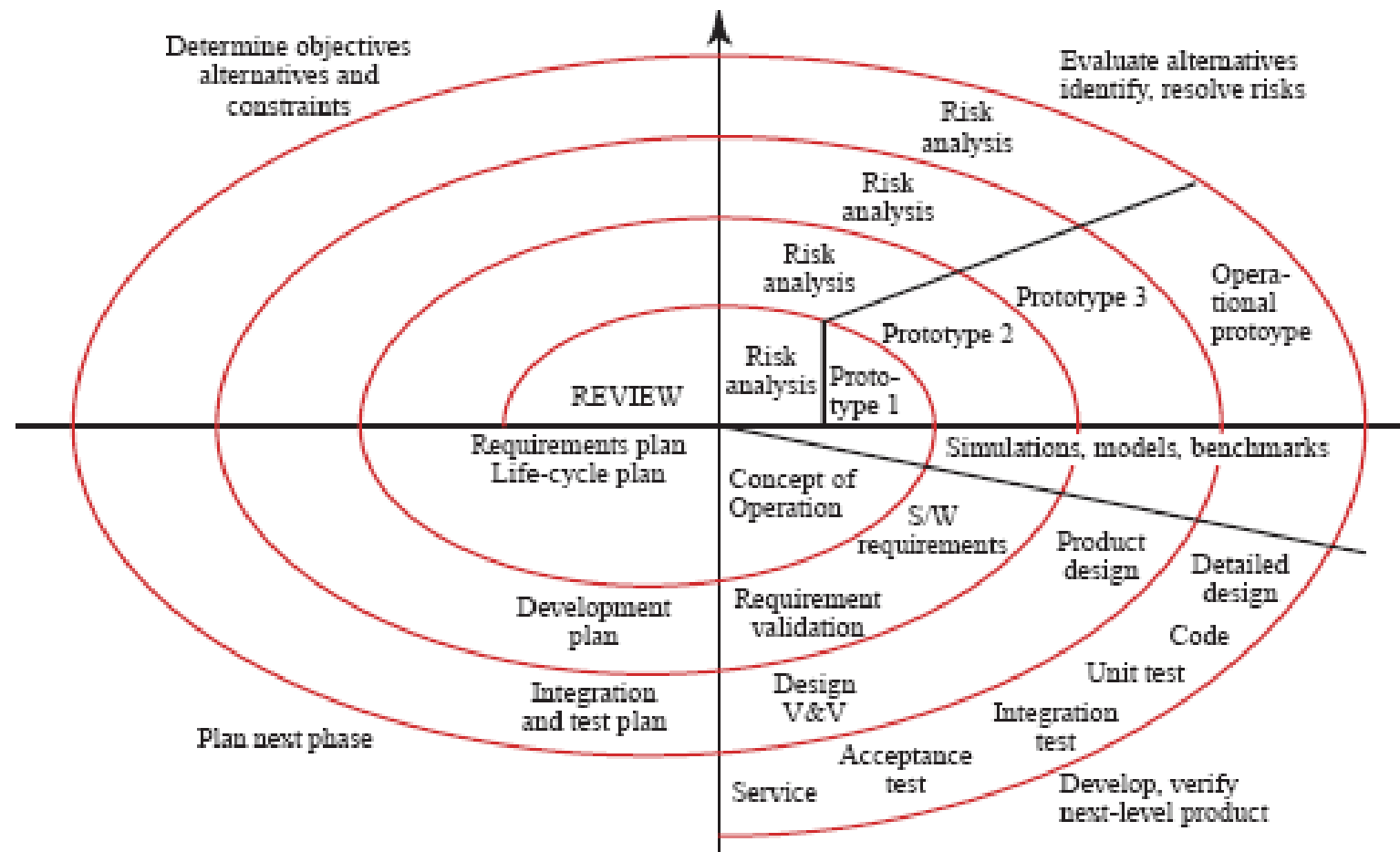
## ■ **Nhược điểm:**

- 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

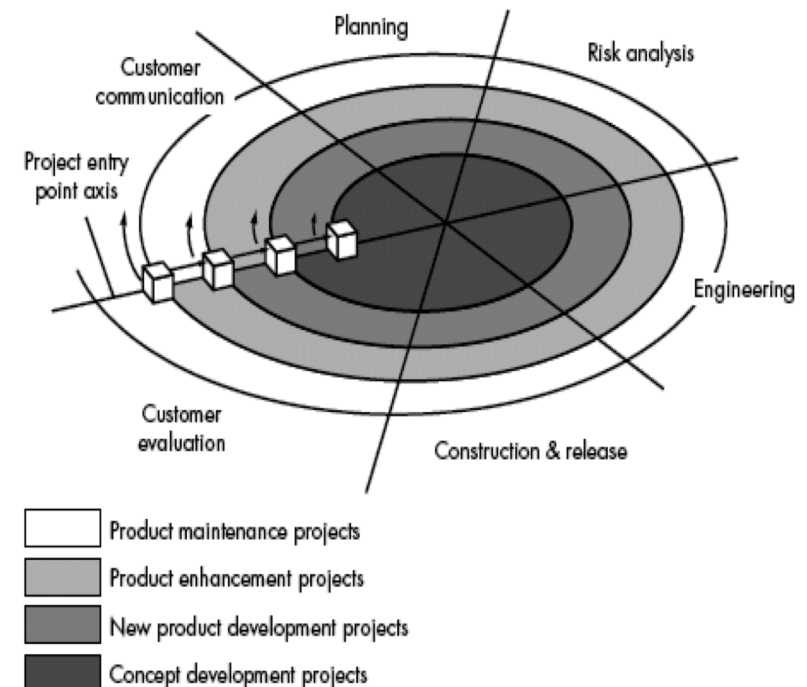
## ■ Mô hình xoắn ốc (Spiral Model)



# 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 kỳ

**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
  - 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

- 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 rõ ràng.
  - Khó thuyết phục khách hàng về việc kiểm soát tiến trình
  - Dựa vào những chuyên gia đánh giá rủi 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.

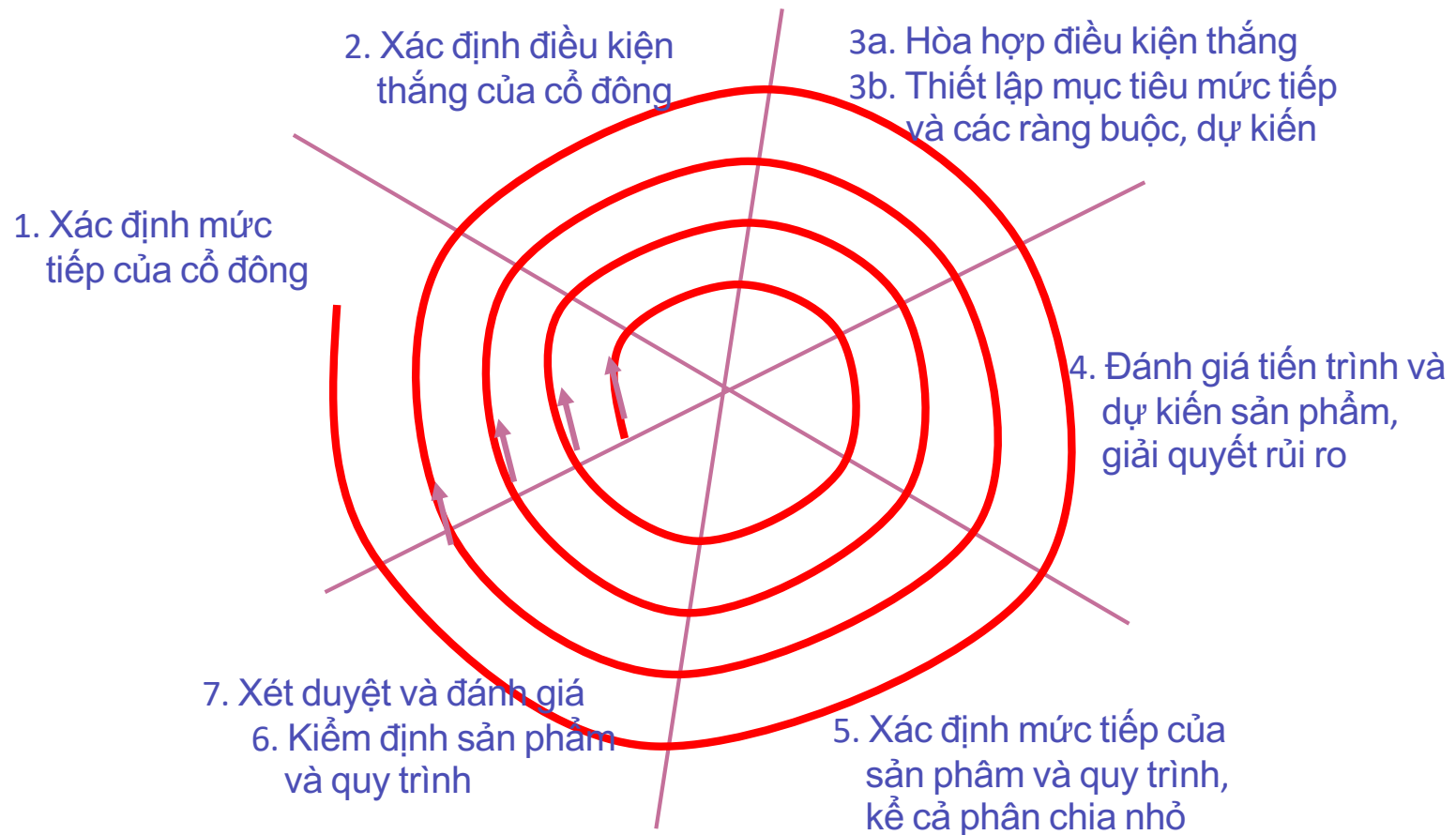


# Các mô hình phát triển phần mềm

- Mô hình xoắn ốc 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 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





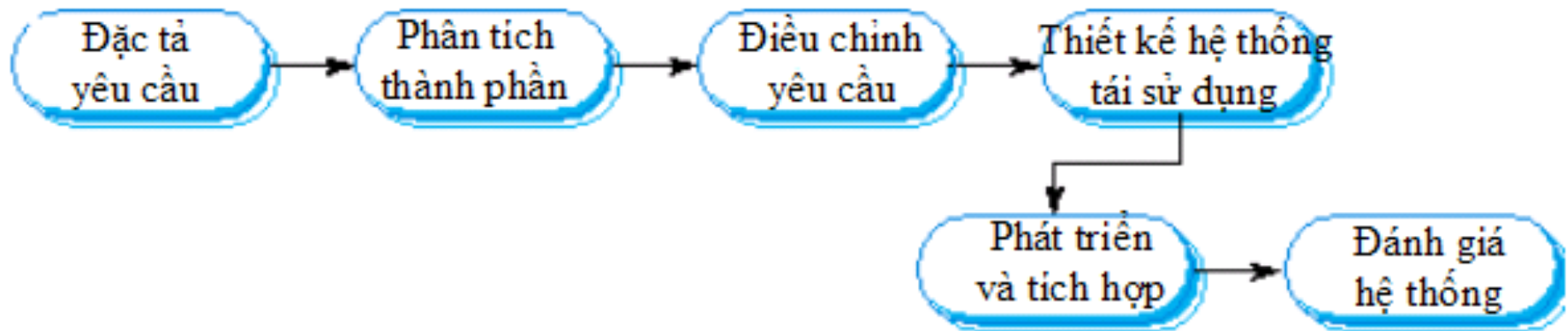
# Các mô hình phát triển phần mềm

- **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



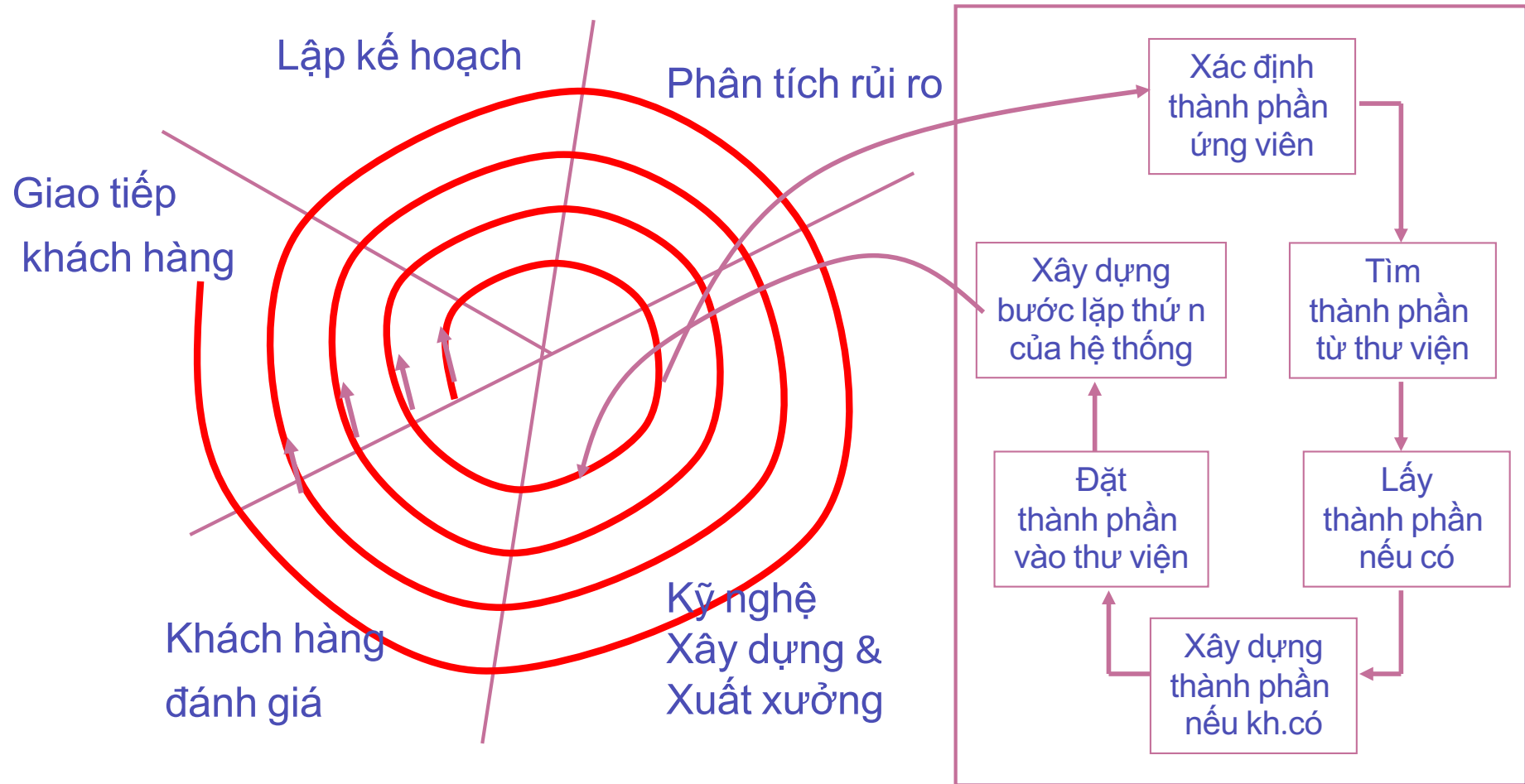


# Các mô hình phát triển phần mềm

- 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

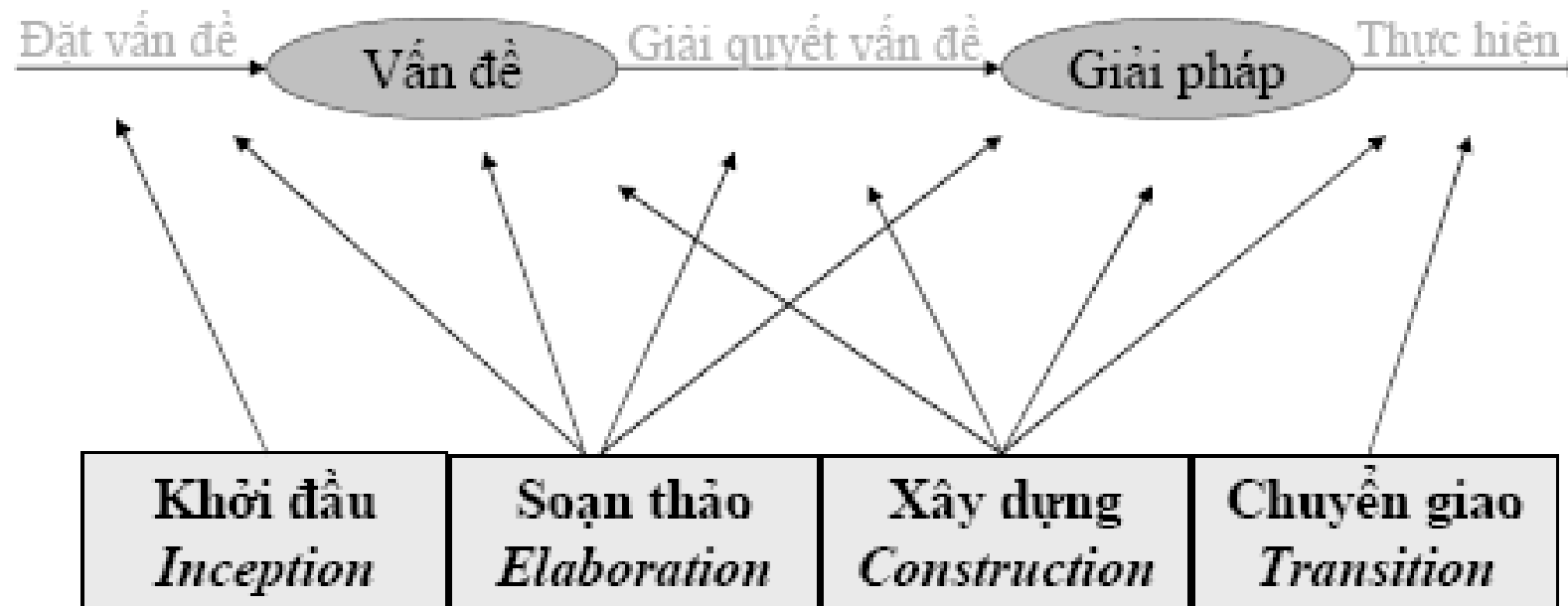


# Các mô hình phát triển phần mềm

- 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
      - Tiến 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

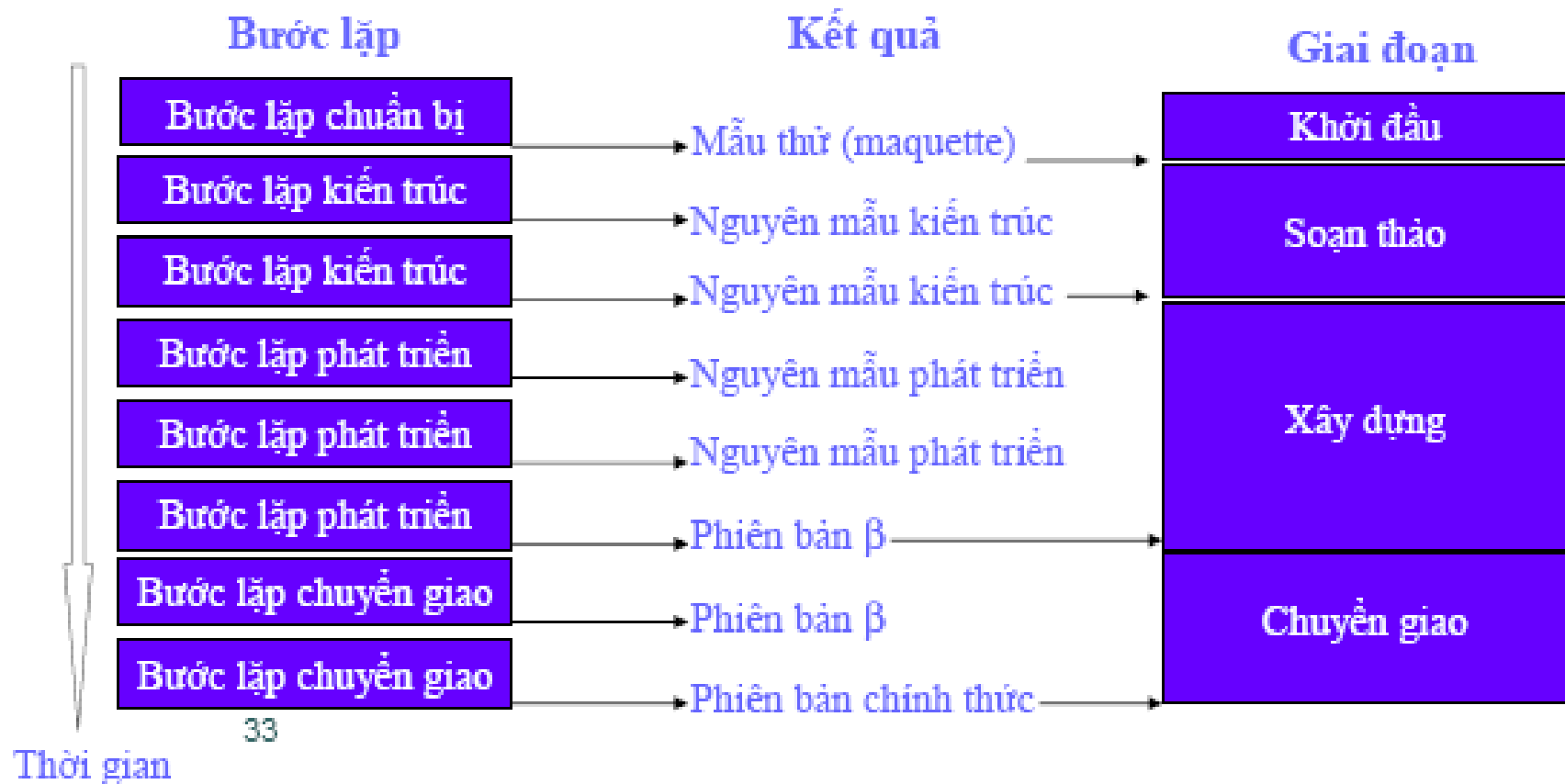
# Các mô hình phát triển phần mềm

- Mô hình hợp nhất
  - Góc nhìn kỹ thuật



# Các mô hình phát triển phần mềm

- Mô hình hợp nhất
  - Kết hợp hai góc nhìn



33

# Các mô hình phát triển phần mềm

- Mô hình hợp nhất và UML





# 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