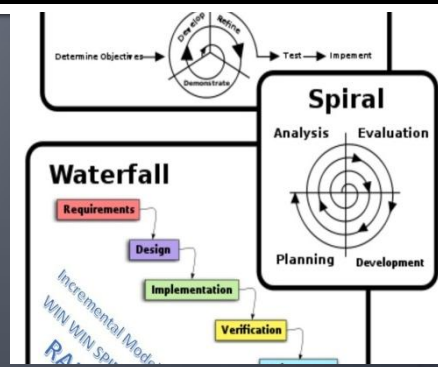


Chương 2:

Quy trình phần mềm



Mục tiêu

- Cung cấp các kiến thức về các mô hình quy trình phần mềm thông dụng
- SV phải nhận thức được tầm quan trọng, đánh giá, lựa chọn được mô hình quy trình phần mềm phù hợp cho dự án phát triển phần mềm cụ thể.

Nội dung

1. Quy trình phần mềm?
2. Mô hình quy trình phần mềm?
3. Tầm quan trọng của mô hình quy trình phần mềm
4. Một số mô hình quy trình phần mềm thông dụng.

1. Quy trình phần mềm?

- ~ Quy trình phát triển phần mềm, hoặc quy trình vòng đời phát triển phần mềm - SDLC
 - ~ tập các hoạt động được sắp xếp theo trình tự nhất định nhằm tạo ra/sửa đổi một sản phẩm phần mềm.

1. Quy trình phần mềm?

- Các hoạt động thường gặp trong các quy trình:
 - **Đặc tả:**
 - đặc tả những gì hệ thống phải làm và các ràng buộc trong quá trình xây dựng hệ thống.
 - **Phát triển:**
 - Thiết kế và xây dựng hệ thống phần mềm.
 - **Kiểm thử:**
 - kiểm tra xem liệu phần mềm đã thoả mãn yêu cầu của khách hàng?
 - **Mở rộng:**
 - điều chỉnh và thay đổi phần mềm tương ứng với sự thay đổi yêu cầu.

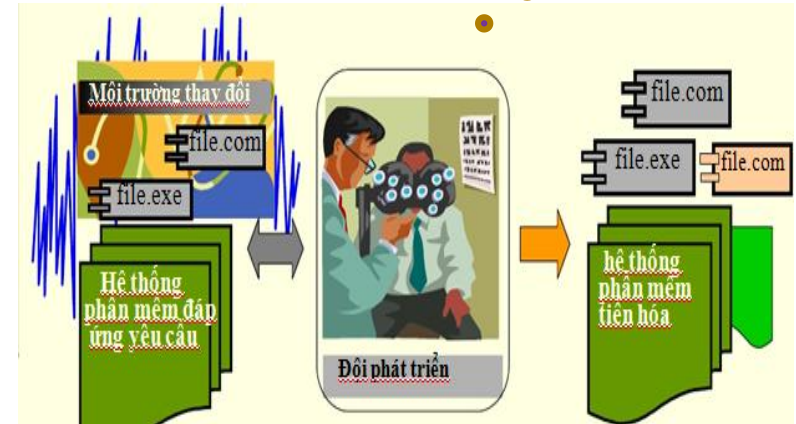
1. Đặc tả phần mềm



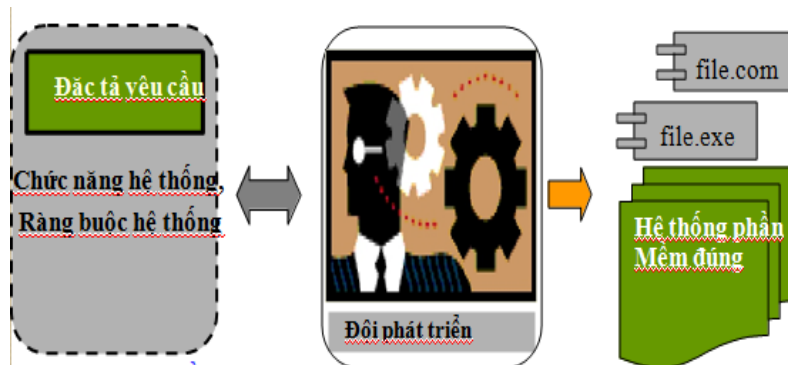
Các yêu cầu Phần mềm



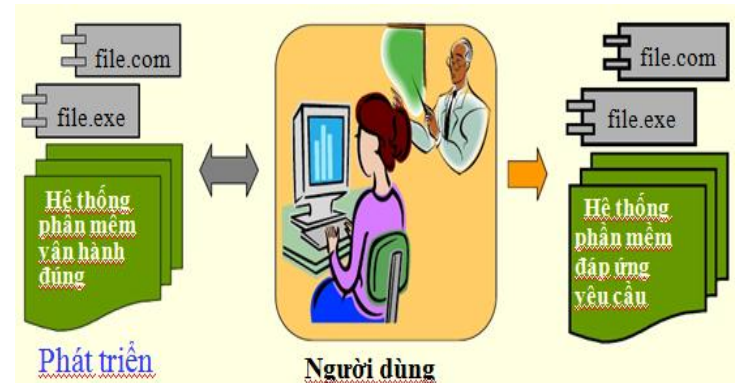
4. Mở rộng



2. Phát triển



3. Kiểm thử



1. Quy trình phần mềm?

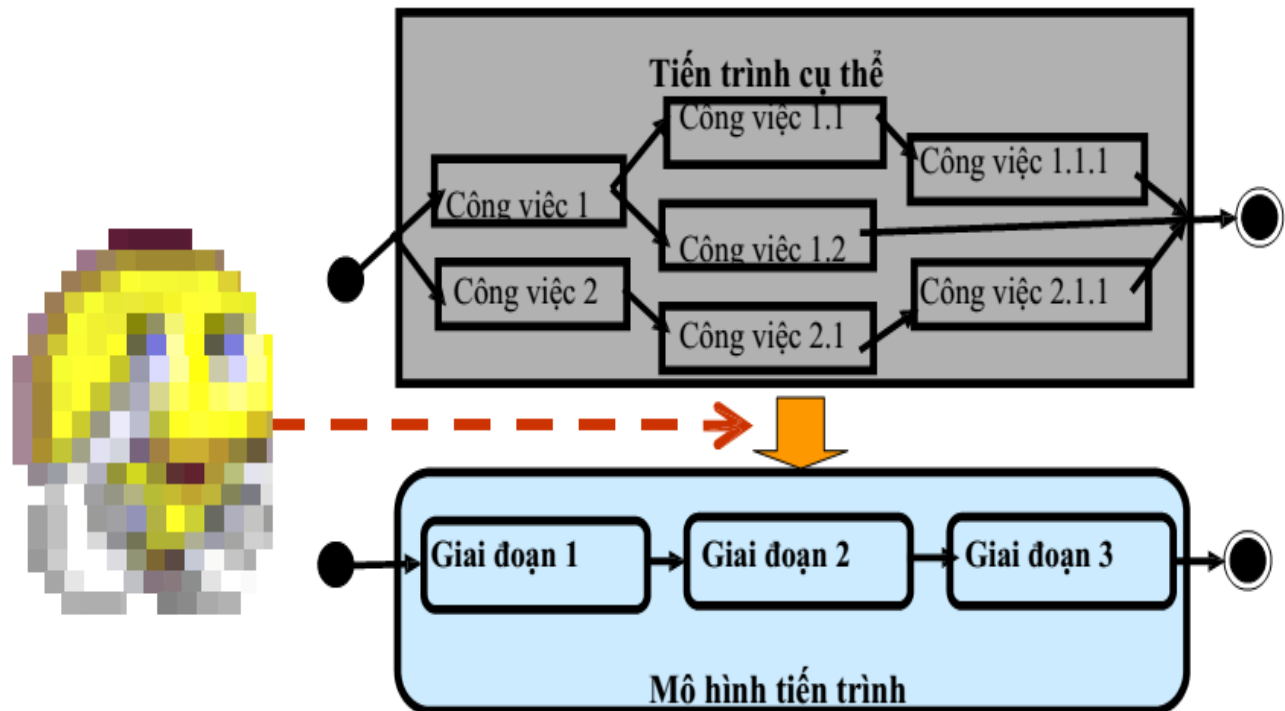
- Khác nhau
 - Phụ thuộc loại phần mềm cụ thể
 - Sử dụng quy trình không phù hợp
 - => giảm chất lượng, tăng chi phí phát triển, tăng rủi ro tiềm ẩn

Nội dung

1. Quy trình phần mềm?
2. **Mô hình quy trình phần mềm?**
3. Tầm quan trọng của mô hình quy trình phần mềm
4. Một số mô hình quy trình phần mềm thông dụng.

2. Mô hình quy trình phần mềm?

- ~ mô hình SDLC
 - Là một thể hiện đơn giản (trừu tượng) của một quy trình phần mềm cụ thể.



2. Mô hình quy trình phần mềm?

- Mỗi mô hình thường biểu diễn một góc nhìn cụ thể của tiến trình phần mềm.
 - Mô hình quy trình truyền thống
 - Ví dụ: Thác nước, xoắn ốc, ...
 - ~ Biểu diễn một góc nhìn cụ thể (thường là góc nhìn kiến trúc).
 - Mô hình quy trình hiện đại
 - Ví dụ: RUP
 - ~ Là hợp nhất của 3 góc nhìn: tĩnh, động, thực hành.

2. Mô hình quy trình phần mềm?

- Hệ thống lớn = {Hệ con}
 - Mỗi hệ con có thể sử dụng một mô hình quy trình p.m phù hợp riêng.
 - => Trong thực tế các mô hình thường được kết hợp với nhau để triển khai một dự án.

Nội dung

1. Quy trình phần mềm?
2. Mô hình quy trình phần mềm?
3. **Tầm quan trọng của mô hình quy trình phần mềm**
4. Một số mô hình quy trình phần mềm thông dụng.

3. Lợi ích của mô hình quy trình phần mềm?

- Là 1 framework kế hoạch ↑ dự án
 - Thể hiện cách tiếp cận được sử dụng để phát triển p.m
 - Ảnh hưởng trực tiếp đến thành công/thất bại của dự án.
 - Hỗ trợ lập lịch biểu, phân công công việc và giám sát tiến trình tổng thể của hệ thống.
- Là từ điển chung cho mỗi giai đoạn phát triển
- Xác định kênh giao tiếp giữa các bên liên quan

Nội dung

1. Quy trình phần mềm?
2. Mô hình quy trình phần mềm?
3. Tầm quan trọng của mô hình quy trình phần mềm
4. **Một số mô hình quy trình phần mềm thông dụng.**

4. Mô hình quy trình phần mềm

- 5 loại mô hình đại diện cho các cách tiếp cận ↑:
 - i. **Mô hình thác nước (tuyến tính)**
 - ii. Các mô hình phát triển lặp
 - iii. Các mô hình phát triển lặp và tuyến tính (lai)
 - iv. Phát triển dựa trên sử dụng lại
 - v. Các mô hình phát triển hình thức

i. Mô hình thác nước (Water Fall Model)

- Là mô hình vòng đời phát triển phần mềm (SDLC) đầu tiên được giới thiệu bởi Winston W. Royce (1970).
- Còn gọi là mô hình SDLC tuyến tính (linear-sequential life cycle model), hoặc mô hình cổ điển.

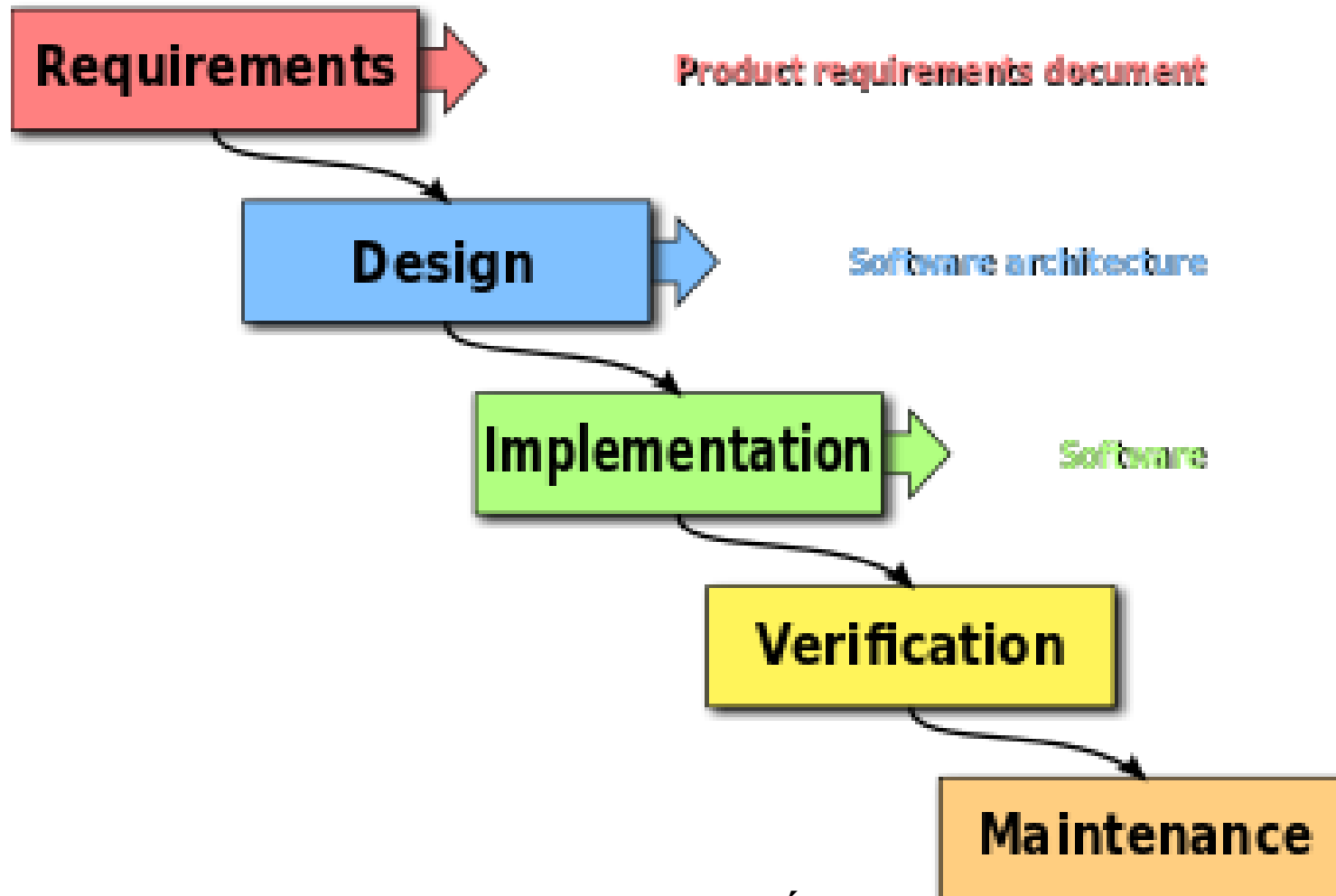


Winston Walker Royce was an American computer scientist, director at Lockheed Software Technology Center in Austin, Texas. He was a pioneer in the field of software development, known for his 1970 paper from which the Waterfall model for software development was mistakenly drawn.

i. Mô hình thác nước (Water Fall Model)

- Quy trình phát triển giống như một dòng chảy.
 - Chia quy trình phát triển thành nhiều giai đoạn (pha) tách biệt
 - 5 giai đoạn:
 - Phân tích, thiết kế, cài đặt, thẩm định và bảo trì
 - Các giai đoạn này được thực hiện theo 1 trật tự nghiêm ngặt, không có sự quay lui hay nhảy vượt pha.
 - Đầu ra của giai đoạn trước là đầu vào của giai đoạn sau.
 - Chỉ khi kết quả (tài liệu, sản phẩm) của giai đoạn trước được xét duyệt và phê chuẩn mới chuyển đến giai đoạn tiếp theo.

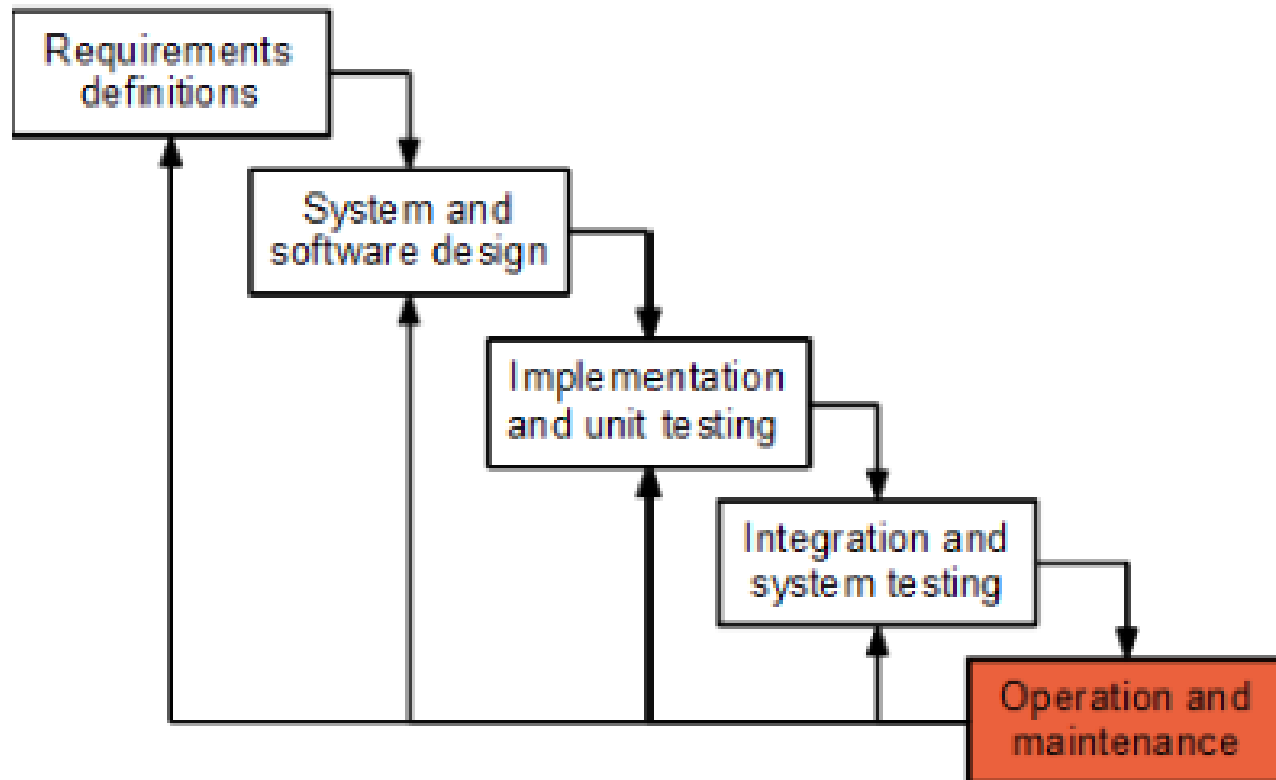
i. Mô hình thác nước (Water Fall Model)



Hình 2.1 Mô hình thác nước gốc

i. Mô hình thác nước (Water Fall Model)

- Mô hình thác nước được sửa đổi



Hình 2.2 Mô hình thác nước

i. Mô hình thác nước (Water Fall Model)

- **Ưu điểm?**

i. Mô hình thác nước (Water Fall Model)

■ Các ưu điểm:

1. Về việc áp dụng

- Mô hình đơn giản, dễ hiểu và dễ áp dụng
 - Mang tính tự nhiên, tư duy logic.

2. Về hoạt động quản lý

- Thuận lợi cho hoạt động quản lý
- Các giai đoạn xác định, tách biệt, minh bạch
 - Mỗi giai đoạn có các phát hành (các kết quả được ghi chép cẩn thận) cụ thể => Dễ thẩm tra và đánh giá kết quả từng giai đoạn.
 - Dễ lập lịch biểu, phân công công việc, phân bổ nguồn tài nguyên, đánh giá và giám sát lịch biểu.
- => *các lý do mà các nhà quản lý ưa thích và sử dụng mô hình.*

i. Mô hình thác nước (Water Fall Model)

■ Các ưu điểm:

3. Về chất lượng sản phẩm

- Phần mềm dễ bảo trì
 - Kiến trúc phần mềm tốt
 - Tài liệu chuyên trách, rõ ràng cho từng giai đoạn
- Độ tin cậy đảm bảo
 - Nếu tuân theo đúng lịch biểu, được giám sát và kiểm định chặt chẽ.

4. Công cụ hỗ trợ

- Hỗ trợ quản lý; Hỗ trợ truyền thông
 - => Phong phú.

i. Mô hình thác nước (Water Fall Model)

■ Hạn chế chính:

1. Về thời gian

- Chậm có phiên bản được phát hành
 - Đòi hỏi sự kiên nhẫn của khách hàng
- Thời gian dự án có thể kéo dài do các giai đoạn phải làm tuần tự.
 - Thời gian chờ đợi.

i. Mô hình thác nước (Water Fall Model)

■ Hạn chế chính:

2. Về chi phí

- Chi phí chuyên gia;
- Chi phí phát triển và bảo trì sản phẩm
- Lỗi phát hiện và bị trì hoãn qua các giai đoạn là ảnh hưởng nghiêm trọng đến nguồn chi phí.

i. Mô hình thác nước (Water Fall Model)

■ Hạn chế chính:

3. Về chất lượng

- Sản phẩm tiềm ẩn nhiều rủi ro và sự không chắc chắn
- Không đáp ứng các yêu cầu thay đổi.
- Không có sự tương tác của khách hàng trong quá trình phát triển
 - => Mang tính chủ quan, khó được người dùng chấp nhận.

4. Về khả năng áp dụng

- Mô hình quá đơn giản, khó tuân thủ một cách tuần tự:
 - Dự án lớn thường phải quay lui và sửa đổi.
 - Việc điều chỉnh phạm vi dự án có thể chấm dứt dự án

i. Mô hình thác nước (Water Fall Model)

■ Khả năng ứng dụng

- Mô hình thác nước phù hợp với các tình huống:

1. Các yêu cầu ổn định, được đặc tả rõ ràng
2. Các nguồn tài nguyên phong phú với sự trợ giúp các chuyên gia.
3. Dự án phát triển trong thời gian ngắn.

⇒ Có thể áp dụng cho các dự án lớn, phức tạp, thời gian sống lâu nếu lịch biểu tối ưu, giám sát và quản lý tốt.

2. Mô hình quy trình phần mềm

- 5 loại mô hình (SDLC) tiêu biểu:
 - i. Mô hình thác nước
 - ii. **Các mô hình phát triển lặp**
 - iii. Các mô hình phát triển lặp và tuyến tính (lai)
 - iv. Các mô hình phát triển hình thức
 - v. Phát triển dựa trên sử dụng lại

ii. Các mô hình phát triển lặp

- a. Mô hình mẫu thử (Prototyping)
- b. Phát triển ứng dụng nhanh (Application Development - RAD) (1991)

a. Mô hình mẫu thử (Software Prototype Model)

- Là một mô hình SDLC thông dụng hiện nay.
 - Được đề xuất bởi Frederick P. Brooks (1975)

Frederick Phillips "Fred" Brooks Jr. is an American computer architect, software engineer, and computer scientist, best known for managing the development of IBM's System/360 family of computers and the OS/360 software support package, then later writing

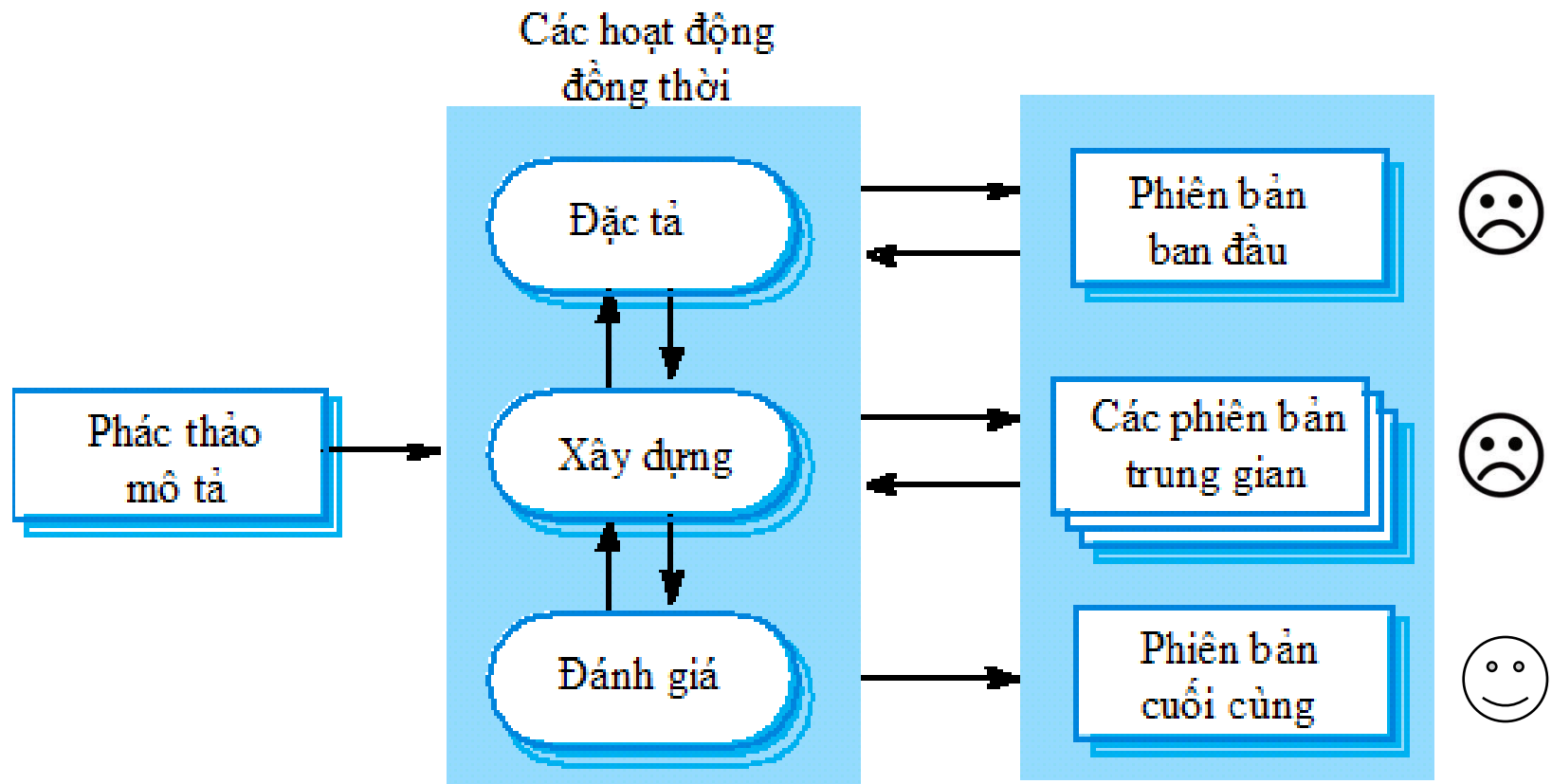


a. Mô hình mẫu thử (Software Prototype Model)

- Ý tưởng:
 - Xây dựng một **mẫu thử** ban đầu và đưa cho khách hàng dùng thử. Sau đó dựa trên các phản hồi của khách hàng để tinh chỉnh mẫu thử qua nhiều phiên bản cho đến khi mẫu thử hoàn thiện (đáp ứng mọi yêu cầu của khách hàng) thì dừng lại.

a. Mô hình mẫu thử (Software Prototype Model)

- Mô hình mẫu thử:



a. Mô hình mẫu thử (Software Prototype Model)

- Mục đích sử dụng:

1. **Phát triển thăm dò**

- Bàn giao mẫu thử và xem là sản phẩm cuối.

2. **Loại bỏ mẫu thử**

- Thu thập các yêu cầu phần mềm từ mẫu thử và loại bỏ mẫu thử.

a. Mô hình mẫu thử (Software Prototype Model)

- **Ưu điểm?**

a. Mô hình mẫu thử (Software Prototype Model)

■ Ưu điểm

1. Thời gian

- Nhanh chóng có các phát hành đến người dùng
 - Người dùng có hiểu biết tốt về hệ thống
 - Kinh nghiệm vận hành, tiếp xúc trực tiếp
 - Tăng cường sự hợp tác với người dùng
 - Có được những phản hồi giá trị từ phía người dùng

a. Mô hình mẫu thử (Software Prototype Model)

■ Ưu điểm (t.t):

2. Chất lượng sản phẩm

- Đáp ứng các yêu cầu thay đổi
- Phần mềm mang tính khách quan
- Thường được người dùng chấp nhận thay vì từ chối
 - Thỏa mãn lập tức các yêu cầu của khách hàng, đáp ứng các yêu cầu thay đổi, các yêu cầu thực sự mong muốn.

a. Mô hình mẫu thử (Software Prototype Model)

- **Nhược điểm?**

a. Mô hình mẫu thử (Software Prototype Model)

■ Nhược điểm

1. Quản lý dự án

- Gây khó khăn, đòi hỏi khả năng thích nghi cao trong công tác quản lý
 - Quản lý, giám sát không đúng cách
 - => Hậu quả: quá nhiều nỗ lực chi cho phát triển mẫu thử
 - Khó lập lịch, giám sát và đảm bảo chất lượng sản phẩm
 - Thiếu tầm nhìn của cả quy trình
 - Phải điều chỉnh lịch biểu liên tục tùy theo phản hồi của khách hàng.
 - Đội phát triển phải có kỹ năng chuyên môn cao.

a. Mô hình mẫu thử (Software Prototype Model)

■ Nhược điểm

2. Chất lượng sản phẩm có thể ↓

- Phần mềm khó bảo trì:
 - Kiến trúc nghèo nàn, tài liệu thiếu
 - Hệ thống mang tính chắp vá, khó sửa đổi
- Áp lực về thời gian phát triển nhanh mẫu thử
 - => dẫn đến sử dụng giải thuật, ngôn ngữ lập trình, quy trình, công nghệ... không tối ưu.

a. Mô hình mẫu thử (Software Prototype Model)

■ Khả năng ứng dụng

- Phù hợp khi phát triển các ứng dụng có nhiều tương tác với người dùng

- Càng nhiều tương tác, mô hình càng mang lại lợi ích

- Ví dụ: các hệ thống on-line, các hệ thống cần người dùng điền form với nhiều loại màn hình khác nhau, ...

- => Phù hợp với các hệ tương tác cỡ vừa, nhỏ, có vòng đời ngắn hoặc một phần của hệ thống lớn.

- Hệ thống với yêu cầu thay đổi

=> *Không hiệu quả với các hệ thống ít tương tác người dùng (ví dụ batch processing) hoặc các hệ thống hầu hết là thực hiện các tính toán bên trong.*

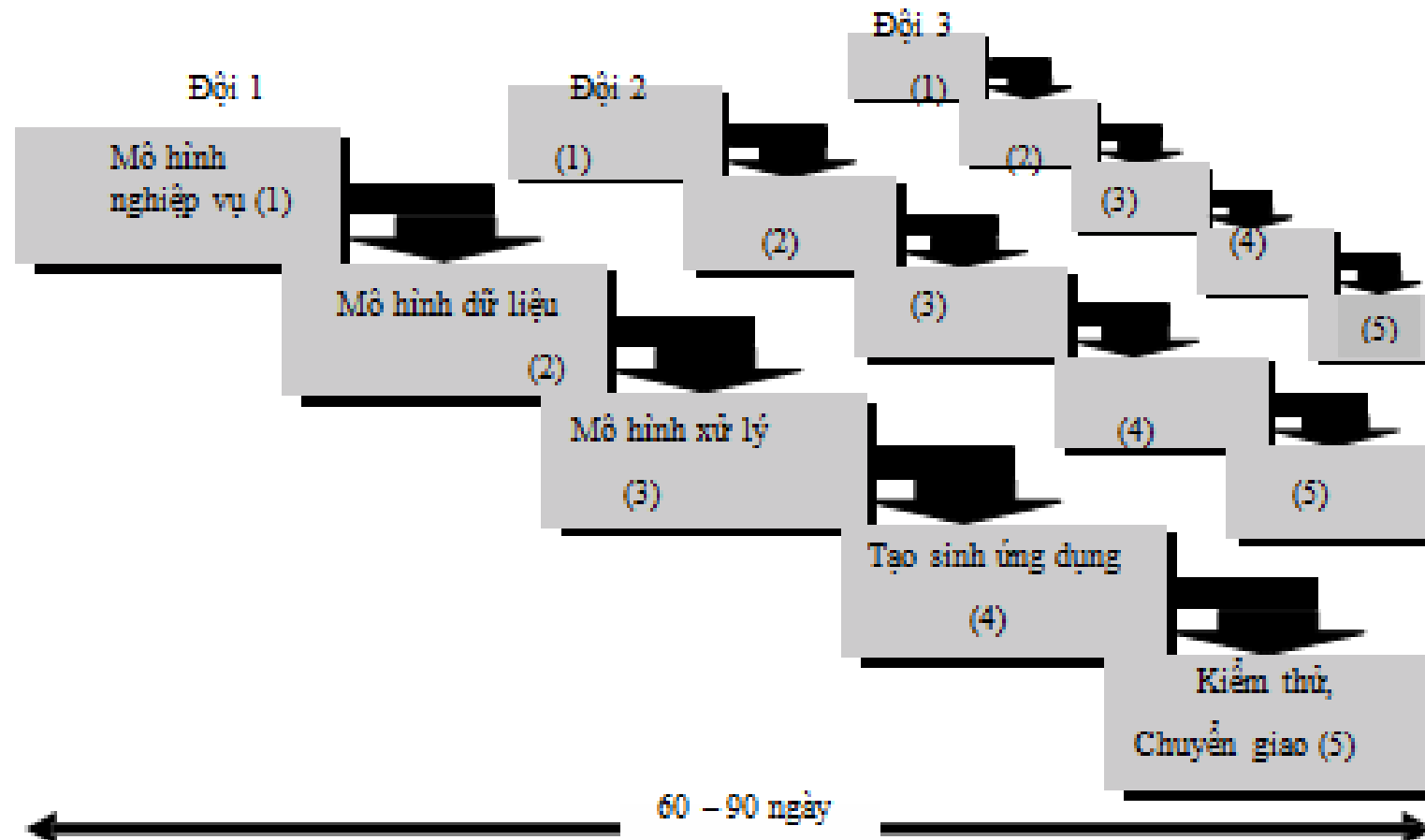
b. Phát triển ứng dụng nhanh (RAD)

- Mô hình RAD (**Rapid Application Development**)
 - Là sự kết hợp giữa phát triển lặp và mẫu thử.
 - Được phát triển bởi [James Martin](#) (1991)



James Martin was a British Information Technology consultant and author, known for his work on information engineering. Martin was nominated for a Pulitzer prize for his book, *The Wired Society: A Challenge for Tomorrow*.

b. Phát triển ứng dụng nhanh (RAD)



b. Phát triển ứng dụng nhanh (RAD)

- **Ưu điểm?**

b. Phát triển ứng dụng nhanh (RAD)

■ Ưu điểm:

1. Thời gian

- Cho ra các phát hành nhanh chóng
 - => Giảm thời gian phát triển tổng thể
- Tăng cường các khả năng:
 - Sử dụng lại các thành phần
 - Thực hiện // các hoạt động phát triển.
 - Khai thác tối đa các công cụ trợ giúp

b. Phát triển ứng dụng nhanh (RAD)

- **Ưu điểm:**
 2. Khuyến khích các phản hồi từ phía khách hàng
 - Các yêu cầu thay đổi có thể được kết hợp
 3. Tích hợp xảy ra sớm và giải quyết nhiều vấn đề tích hợp

b. Phát triển ứng dụng nhanh (RAD)

- **Nhược điểm?**

b. Phát triển ứng dụng nhanh (RAD)

■ Nhược điểm

- Đòi hỏi nhóm phát triển có trình độ chuyên môn cao, tăng độ phức tạp trong quản lý.
- Chỉ nên dùng RAD với hệ thống có thể mô đun hóa
 - Phụ thuộc nhiều vào các kỹ năng mô hình hóa
 - Không thể áp dụng cho các dự án kinh phí thấp vì cần khai thác tối đa các công cụ trợ giúp cho quy trình phát triển
 - Chi phí mua sắm, huấn luyện, thuê chuyên gia, ...

b. Phát triển ứng dụng nhanh (RAD)

■ Khả năng ứng dụng

- Hệ thống có thể mô đun hóa thành các phát hành theo cách thức tăng dần
- Có ngân quỹ cho phép khai thác tối đa các công cụ hiện có
- Các chuyên gia miền là sẵn sàng để tư vấn các kiến thức nghiệp vụ liên quan
- Hệ thống với các yêu cầu thay đổi
- Phù hợp với các dự án yêu cầu thời gian phát triển ngắn

2. Mô hình quy trình phần mềm

- 5 loại mô hình (SDLC) tiêu biểu:
 - i. Mô hình thác nước
 - ii. Các mô hình phát triển lặp
 - iii. Các mô hình phát triển kết hợp lặp và tuần tự (lai)**
 - iv. Phát triển dựa trên sử dụng lại
 - v. Các mô hình phát triển hình thức
(các mô hình quan trọng và phổ biến)

iii. Phát triển lặp & tuần tự

- a. Mô hình xoắn ốc
- b. Mô hình lặp và tăng trưởng

a. Mô hình xoắn ốc

- Được phát triển bởi Barry Boehm (1988)
 - Còn gọi là Boehm's model.

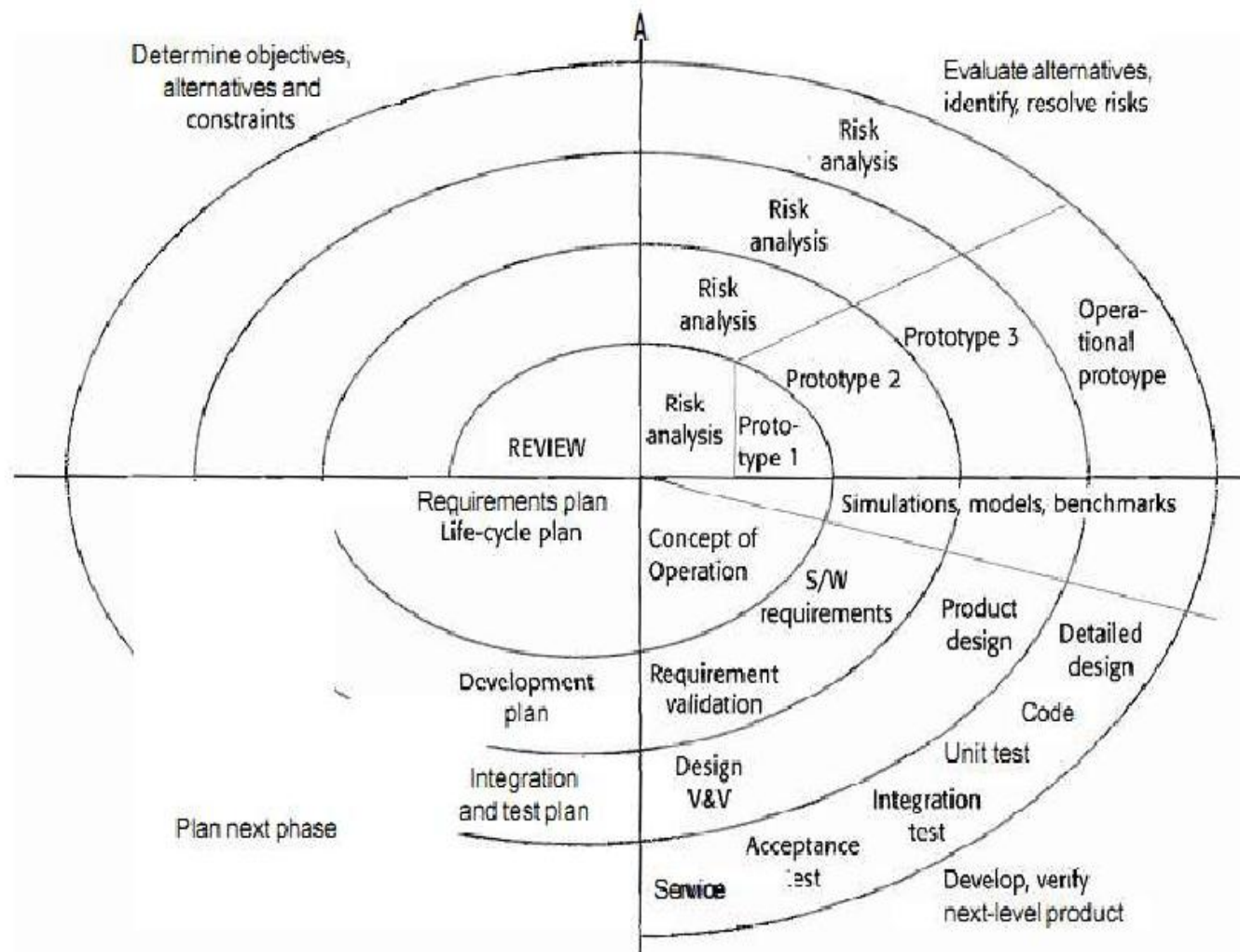


Barry W. Boehm is an American software engineer, distinguished professor of computer science, industrial and systems engineering; the TRW Professor of Software Engineering; and founding director of the Center for Systems and Software Engineering at the University of Southern California.

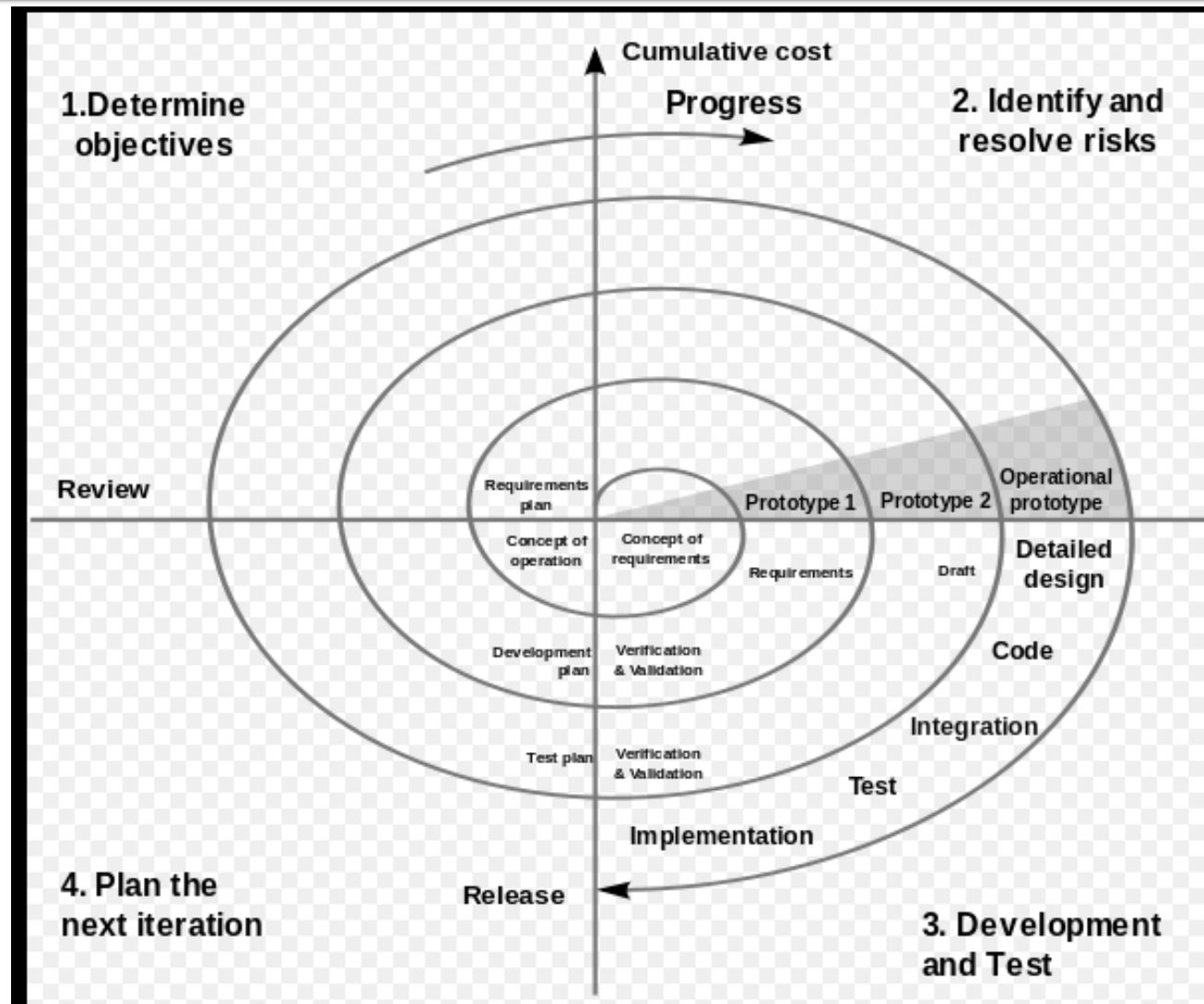
a. Mô hình xoắn ốc

- Là sự kết hợp giữa mô hình tuần tự và mô hình thác nước, đồng thời bổ sung thêm hoạt động phân tích rủi ro
 - Tiến trình phát triển phần mềm được biểu diễn bởi các vòng xoắn ốc.
 - Mỗi vòng xoắn ốc đại diện một giai đoạn phát triển phần mềm.
 - Vòng lặp trong nhất liên quan đến nghiên cứu tính khả thi, vòng lặp tiếp theo định nghĩa yêu cầu, vòng lặp tiếp theo là thiết kế hệ thống, ...

a. Mô hình xoắn ốc



a. Mô hình xoắn ốc



a. Mô hình xoắn ốc

- Là quá trình lặp: hướng mở rộng, hoàn thiện dần các pha trước đó.
 - Mỗi lần tăng vòng là một lần tiến triển dự án sang giai đoạn tiếp theo và xét duyệt lại các giai đoạn phát triển trước đó để khắc phục lỗi và hoàn thiện dần
 - => Còn gọi là cách tiếp cận phát triển lặp và tăng trưởng.

a. Mô hình xoắn ốc

- Ưu điểm?

a. Mô hình xoắn ốc

■ Ưu điểm

- Đạt được các ưu điểm của cả 2 mô hình thác nước và mẫu thử
- Khắc phục được một số hạn chế của 2 mô hình trên
- Kiểm soát được các rủi ro trong từng giai đoạn.

a. Mô hình xoắn ốc

■ Nhược điểm

- Việc quản lý là phức tạp hơn
 - Tiến trình phức tạp
- Không phù hợp với các dự án nhỏ, độ rủi ro thấp và có thể đắt đỏ với các dự án nhỏ
 - Chi phí chuyên gia lớn.
 - Một số lớn các giai đoạn trung gian yêu cầu tài liệu quá nhiều.

a. Mô hình xoắn ốc

■ Khả năng ứng dụng

- Mô hình xoắn ốc được sử dụng rộng rãi trong công nghiệp phần mềm vì nó đồng bộ với tiến trình phát triển tự nhiên của bất cứ sản phẩm nào.

a. Mô hình xoắn ốc

■ Khả năng ứng dụng

- Một số ngữ cảnh ứng dụng phù hợp:
 - Hệ lớn có thể phân chia phần cốt lõi và phần thứ yếu
 - Các dự án có độ rủi ro mức trung bình và cao
 - Người dùng không chắc chắn về các yêu cầu của họ
 - Các yêu cầu là phức tạp và có thể thay đổi

b. Mô hình lặp & tăng trưởng

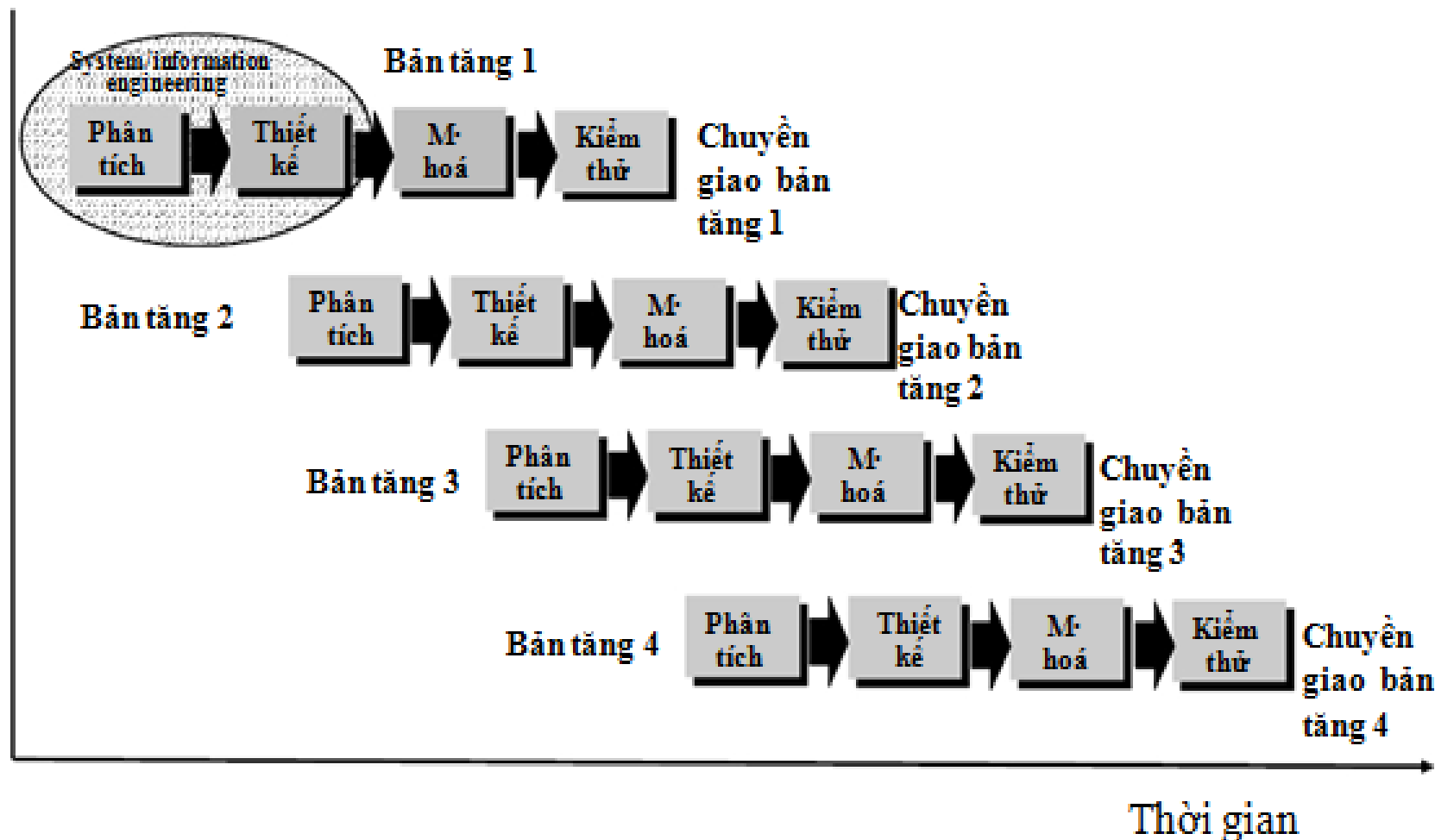
- Iterative and Incremental development
 - Còn gọi là mô hình thác nước thu nhỏ
 - Được đề cập trong bộ chuẩn quân sự: DOD-STD-2167 (1985) – phát triển phần mềm hệ thống phòng thủ.

b. Mô hình lặp & tăng trưởng

- **Ý tưởng:**

- Phát triển sản phẩm qua nhiều lần lặp. Mỗi lần lặp là một lần tăng trưởng hướng hệ thống đến ngày càng hoàn thiện hơn.
 - Ví dụ: Phát triển ứng dụng RUP, Agile,...

b. Mô hình lặp & tăng trưởng



b. Mô hình lặp & tăng trưởng

■ Khả năng ứng dụng

- Các yêu cầu được xác định rõ ràng và đầy đủ/lần lặp
 - Các đòi hỏi nâng cấp có thể tiến hóa theo thời gian
- Công nghệ mới được trang bị và sử dụng bởi nhóm phát triển
- Có một số đặc trưng rủi ro mức cao và các mục tiêu có thể thay đổi trong tương lai
- Dự án lớn, phức tạp

b. Mô hình lặp & tăng trưởng

■ Ưu điểm:

1. Một số chức năng chính có thể được phát triển sớm và nhanh chóng trong vòng đời
 - Phát hành nhanh sản phẩm đến người dùng
 - Mỗi phát hành ~ 1 lần tăng
 - Tạo cơ hội thu thập phản hồi từ người dùng về sản phẩm
2. Các kết quả thu được sớm và theo định kỳ
 - Tạo cơ hội chiếm lĩnh thị trường
 - Đáp ứng nhanh các yêu cầu người dùng
3. Phát triển // có thể được lập kế hoạch
4. Sự tiến triển dự án có thể đo

b. Mô hình lặp & tăng trưởng

■ Ưu điểm (t.t):

5. Hỗ trợ các yêu cầu thay đổi

- Ít tốn kém cho việc thay đổi phạm vi và yêu cầu dự án

6. Kiểm thử và gỡ lỗi cho các lần lặp nhỏ hơn là dễ dàng

7. Các rủi ro có thể được xác định và giải quyết trong mỗi lần lặp

- Dễ quản lý rủi ro – Phần rủi ro mức cao được làm đầu tiên

b. Mô hình lập & tăng trưởng

- **Nhược điểm:**

- Cần nhiều nguồn tài nguyên hơn
- Quản lý phức tạp và khó khăn hơn
 - Có thể nảy sinh các vấn đề thiết kế và kiến trúc hệ thống
 - Sửa đổi kiến trúc, đáp ứng nhu cầu thay đổi.
- Không phù hợp với các dự án nhỏ

b. Mô hình lặp & tăng trưởng

- RUP và Agile là các mô hình SDLC mở rộng từ cách tiếp cận lặp và tăng trưởng.
 - SV tự tìm hiểu.

2. Mô hình quy trình phần mềm

- 5 loại mô hình (SDLC) tiêu biểu:
 - i. Mô hình thác nước
 - ii. Các mô hình phát triển lặp
 - iii. Các mô hình phát triển kết hợp lặp và tuần tự (lai)
 - iv. Phát triển hướng sử dụng lại**
 - v. Các mô hình phát triển hình thức
(các mô hình quan trọng và phổ biến)

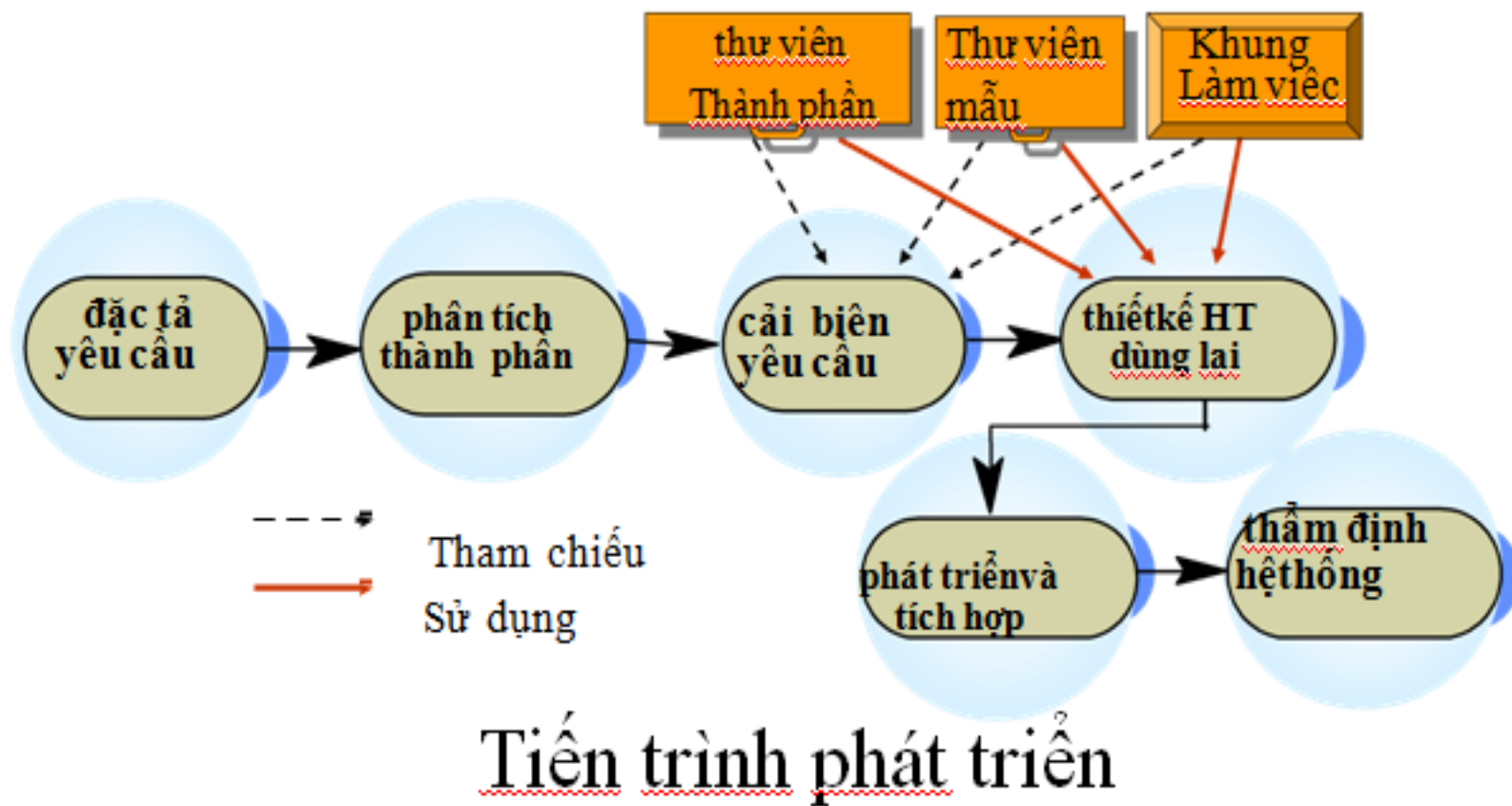
iv. Mô hình hướng sử dụng lại

- Ý tưởng
 - Sử dụng lại một cách nhiều nhất các nguồn tài nguyên có sẵn.
- Các hướng sử dụng lại:
 - Phát triển phần mềm hướng đối tượng
 - Phát triển phần mềm mã nguồn mở
 - **Phát triển phần mềm hướng cấu phần**
 - Phát triển phần mềm hướng mẫu, hướng khung.

f. Mô hình hướng sử dụng lại

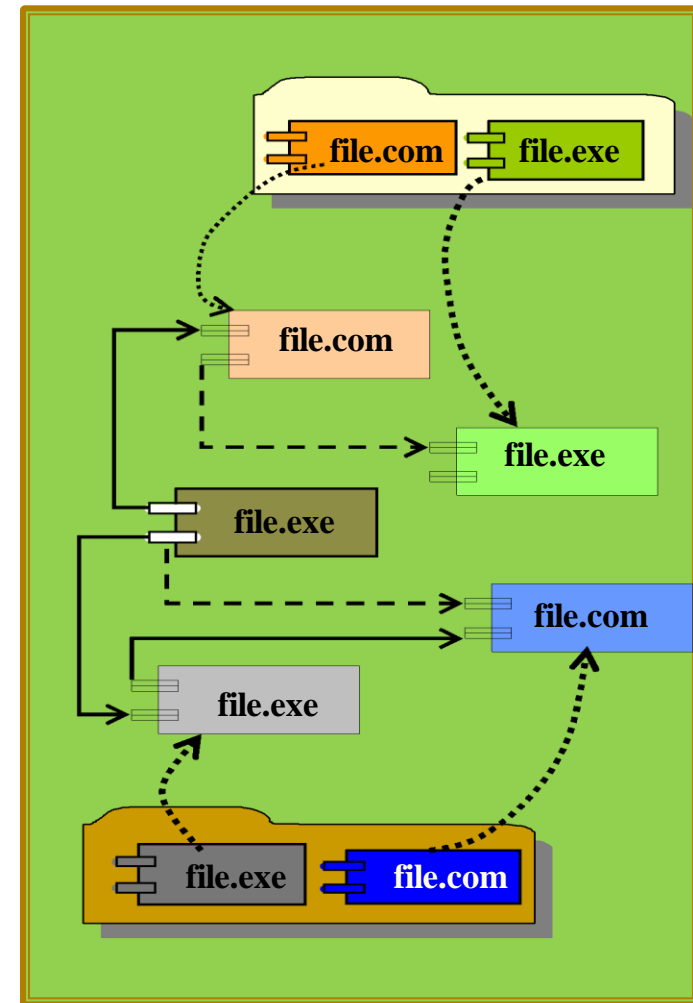
- Các giai đoạn của tiến trình sử dụng lại
 - Phân tích hệ thống thành các yêu cầu tương ứng với từng thành phần
 - Cải biên các yêu cầu để phù hợp với thành phần tái sử dụng (thành phần, mẫu, khung)
 - Thiết kế hệ thống hướng tới tái sản sử dụng lại
 - Phát triển và tích hợp.

f. Mô hình hướng sử dụng lại

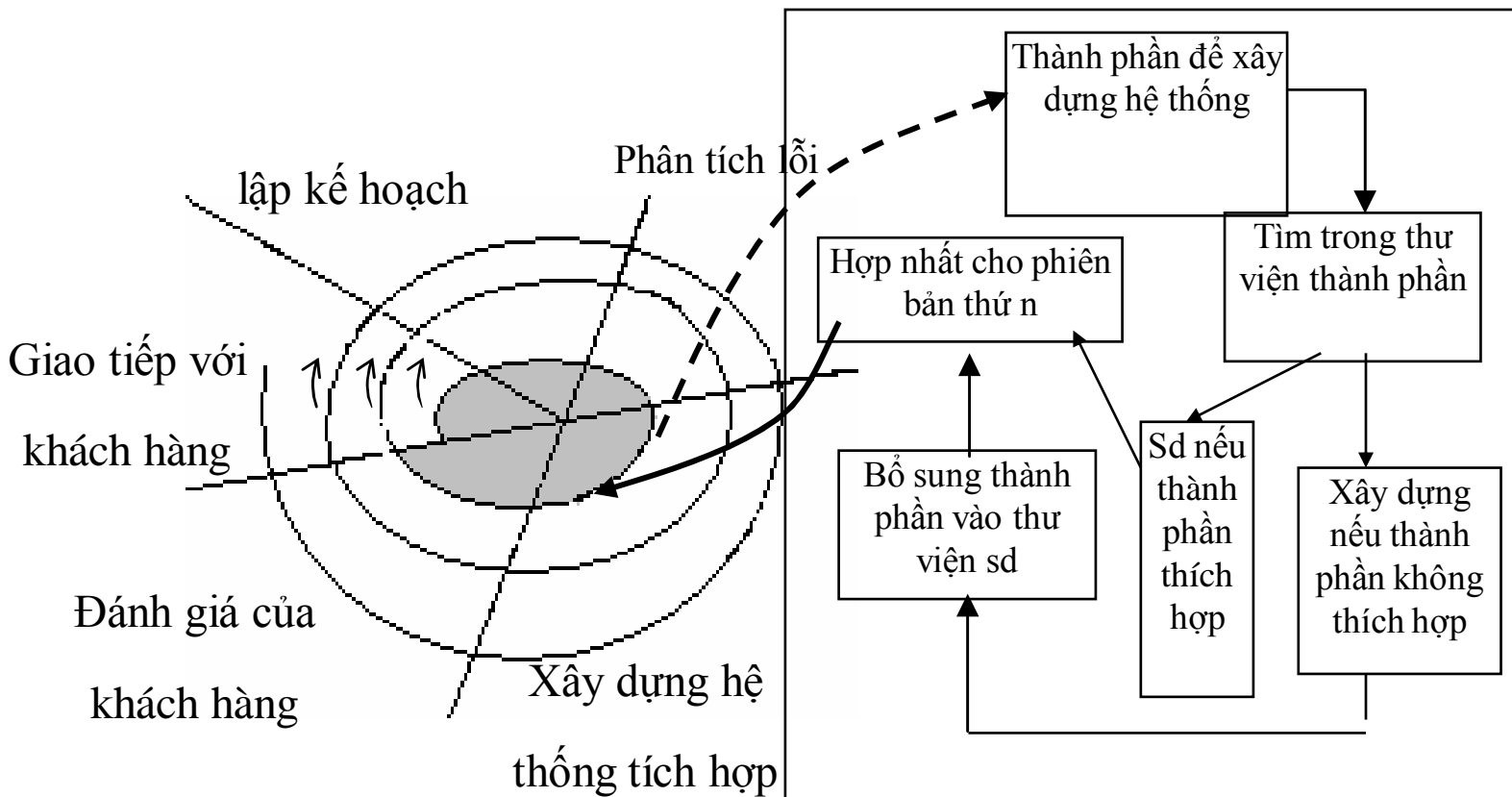


Mô hình phát triển dựa trên cấu phần

- Thành phần
 - Độc lập với ngôn ngữ lập trình
 - Có kích cỡ khác nhau
 - Hệ thống, hệ con, chương trình, mô đun.
 - Mức độ trừu tượng khác nhau
 - Hộp đen, hộp xám, hộp trắng
- Thay thế và tích hợp các thành phần là động



Mô hình phát triển dựa trên cấu phần



Mô hình phát triển dựa trên cấu phần

- Đặc điểm:
 - Giảm số lượng phần mềm được phát triển
 - Giảm chi phí và rủi ro
 - Phát hành phần mềm nhanh hơn
 - Hệ thống thường không thỏa mãn các yêu cầu thực sự (gốc) của người dùng.

2. Mô hình quy trình phần mềm

- 5 loại mô hình (SDLC) tiêu biểu:
 - i. Mô hình thác nước
 - ii. Các mô hình phát triển lặp
 - iii. Các mô hình phát triển kết hợp lặp và tuần tự (lai)
 - iv. Phát triển hướng sử dụng lại
 - v. **Mô hình phát triển hình thức**
(các mô hình quan trọng và phổ biến)

v. Phát triển hệ thống hình thức

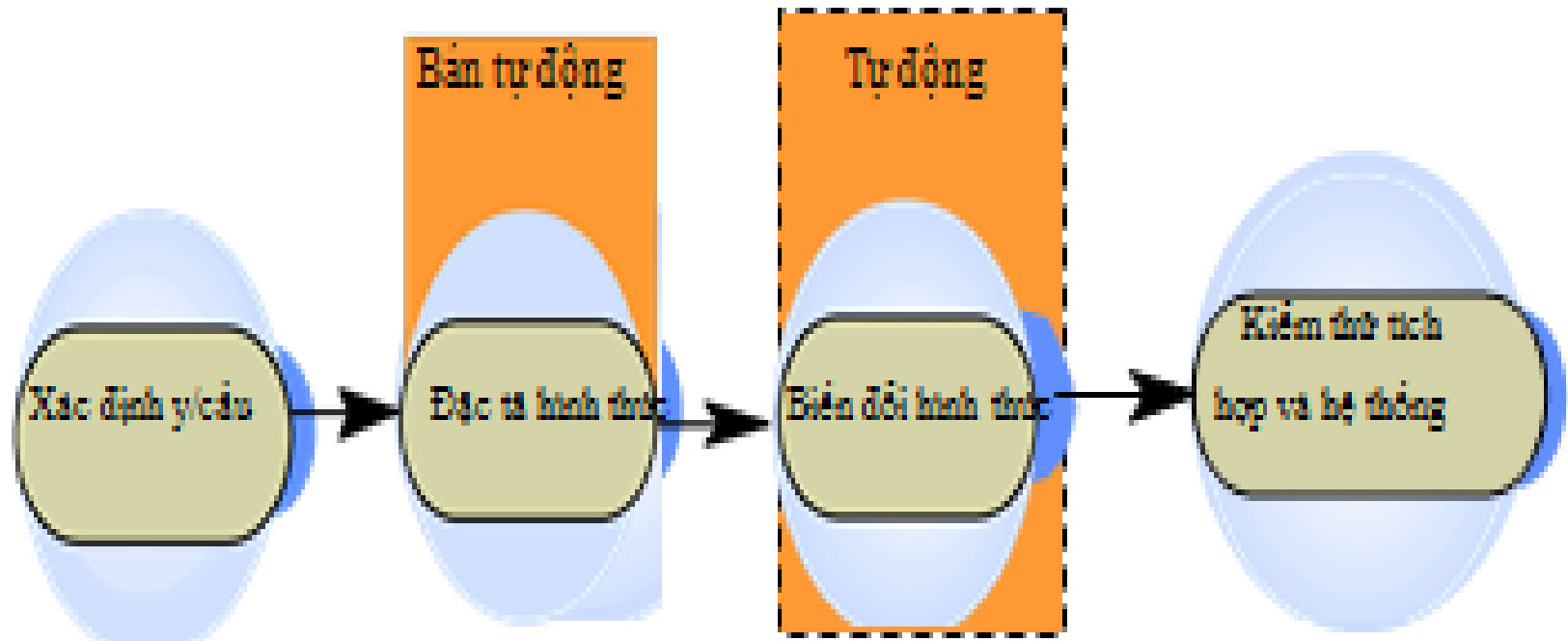
(Formal System Development)

- ~ sử dụng *các phương pháp hình thức* để phát triển & thẩm định phần mềm.
 - ~ *các kỹ thuật dựa trên toán học*;
 - Áp dụng tại các điểm khác nhau trong tiến trình phát triển:
 - Đặc tả (mô tả hệ thống được phát triển ~ các hướng dẫn phát triển và thẩm định sản phẩm);
 - Phát triển (biến đổi hình thức: thiết kế, sinh mã hệ thống dựa trên các đặc tả);
 - Thẩm định (chứng minh tính đúng đắn – c.m mệnh đề tự động, kiểm chứng mô hình, ...)

v. Phát triển hệ thống hình thức

(*Formal System Development*)

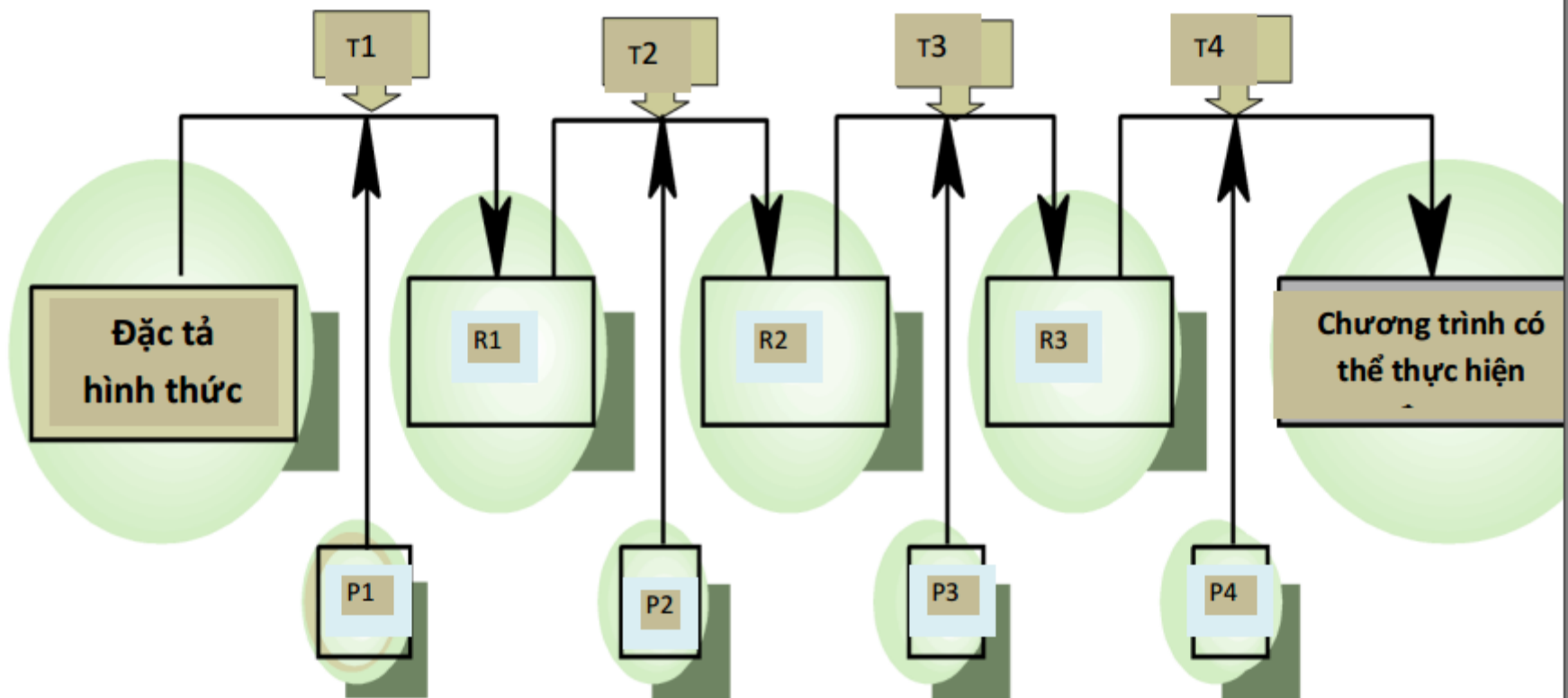
- Mô hình



v. Phát triển hệ thống hình thức

(Formal System Development)

Các phép biến đổi hình thức



Các chứng minh tính đúng đắn của phép biến đổi

v. Phát triển hệ thống hình thức

(Formal System Development)

■ Khả năng ứng dụng

■ Ví dụ:

- Các bộ định tuyến, chuyển mạch Ethernet, giao thức định tuyến, các bộ vi xử lý (Intel Core i7 processor, IBM Power7 microprocessor) và các ứng dụng bảo mật
- Một số dự án của NASA:
 - Hệ thống vận tải hàng không (Next Generation Air Transportation System);
 - Tích hợp hệ thống máy bay không người lái trong hệ thống không gian quốc gia (Unmanned Aircraft System integration in National Airspace System)
 - Phát hiện và giải quyết các xung đột trong không trung (Airborne Coordinated Conflict Resolution and Detection – ACCoRD).

v. Phát triển hệ thống hình thức

(Formal System Development)

■ Hạn chế

- Cần có các kỹ năng đặc tả và sử dụng kỹ thuật tiên tiến => Kinh phí lớn.
- Khó đặc tả được mọi khía cạnh của hệ thống
 - Ví dụ: Các giao diện phần mềm
 - => Áp dụng trong các hệ thống có khối lượng công việc xử lý nhiều, tương tác ít.

Tổng kết

1. Quy trình phần mềm?
2. Mô hình quy trình phần mềm?
3. Tầm quan trọng của mô hình quy trình phần mềm
4. Một số mô hình quy trình phần mềm thông dụng.

Câu hỏi & Bài tập

1. Mô hình thác nước được khuyến cáo sử dụng cho các dự án phát triển phần mềm gì?
2. Mô hình thác nước được khuyến cáo là không nên sử dụng cho các dự án phát triển phần mềm gì?
3. Liệt kê các vấn đề gặp phải khi phát triển ứng dụng sử dụng cách tiếp cận thác nước?
4. Ý nghĩa của việc thẩm định các yêu cầu là gì?
5. Các hoạt động chính của giai đoạn thiết kế phần mềm là gì?
6. Sự khác biệt giữa kiểm thử chấp thuận và kiểm thử hệ thống là gì?
7. Giải thích vai trò của mẫu thử trong phát triển tiến hóa?
8. Giải thích lý do phần mềm được phát triển bởi tiến hóa?
9. Các luồng công việc được hỗ trợ bởi RUP là gì?
10. RUP hỗ trợ tiến trình phần mềm lặp như thế nào?
11. Nghiên cứu và tìm hiểu các phương pháp phát triển phần mềm Agile.

Tài liệu tham khảo

- https://en.wikipedia.org/wiki/Waterfall_model
- https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm
- <http://tryqa.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/>
- https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm
- SELECTING A DEVELOPMENT APPROACH (2008)
- https://www.tutorialspoint.com/sdlc/sdlc_software_prototyping.htm
- https://en.wikipedia.org/wiki/Software_prototyping