Kỹ thuật phần mềm ứng dụng

Chương 2

Các pha trong phát triển phần mềm

(Phần 1)

Đặt vấn đề

9/5/2017

Nội dung

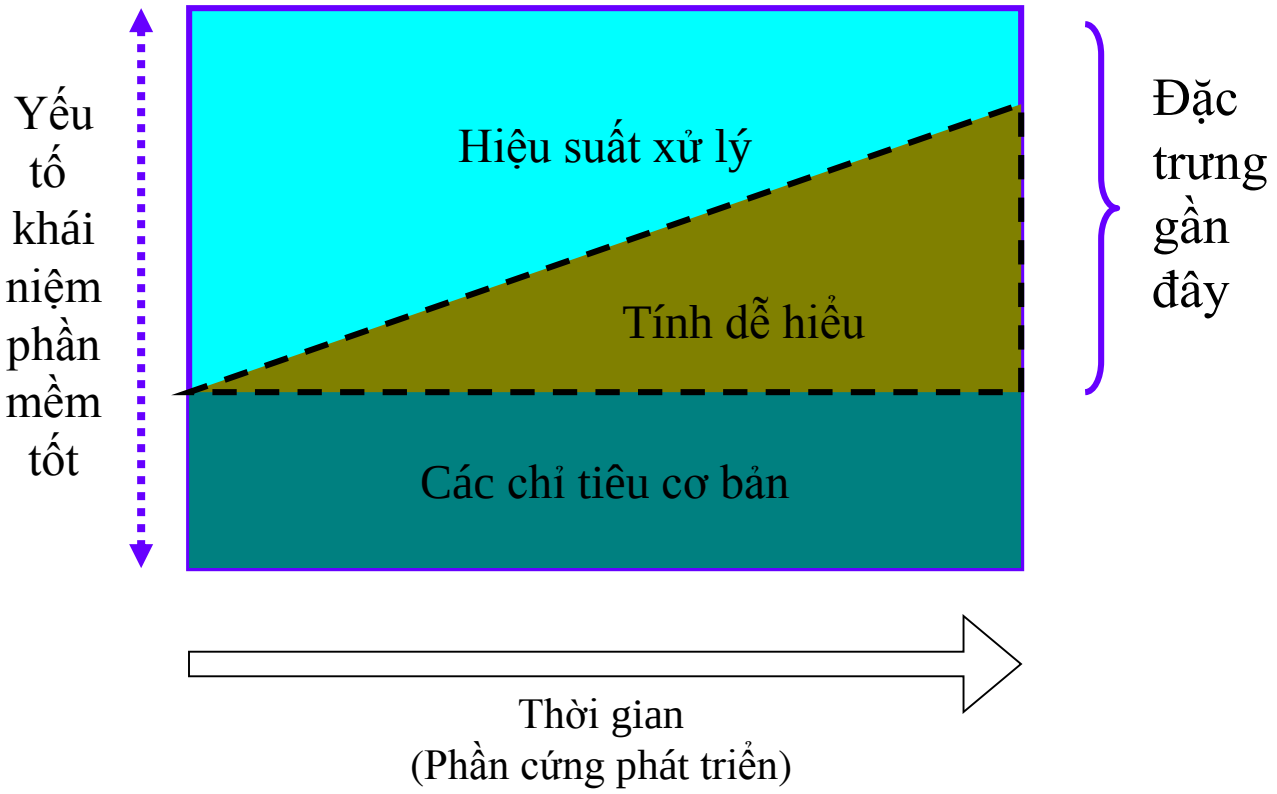
- *2.1 Đặt vấn đề*
  - *2.1.1Đặc điểm của phần mềm*
  - *2.1.2Các vấn đề của phát triển phần mềm*
  - *2.1.3Các mô hình phát triển phần mềm*
- *2.2 Các pha trong phát triển phần mềm*
  - *2.2.1Nghiên cứu yêu cầu (Requirements and Specifications)*
  - *2.2.2Phân tích và thiết kế (System Analysis and Design)*
  - *2.2.3Triển khai (Coding and Unit Test)*
  - *2.2.4Thử nghiệm (Test)*
  - *2.2.5Cài đặt và bảo trì (Deployment and Maintenance)*

2.1.1. Đặc điểm của phần mềm

- *Đặc tính chung của phần mềm:*
  - *Là hàng hóa vô hình*
  - *Chất lượng phần mềm không giảm đi theo thời gian*
  - *Phần mềm vốn chứa lỗi tiềm tàng*
  - *Lỗi phần mềm dễ được phát hiện bởi người ngoài*
  - *Chức năng của phần mềm thường biến hóa, thay đổi theo thời gian*
  - *Hiệu ứng làn sóng trong thay đổi phần mềm*
  - *Phần mềm vốn chứa ý tưởng và sáng tạo của tác giả/nhóm làm ra nó*
  - *Có thể sao chép rất đơn giản*

2.1.1. Đặc điểm của phần mềm

- *Phần mềm tốt*

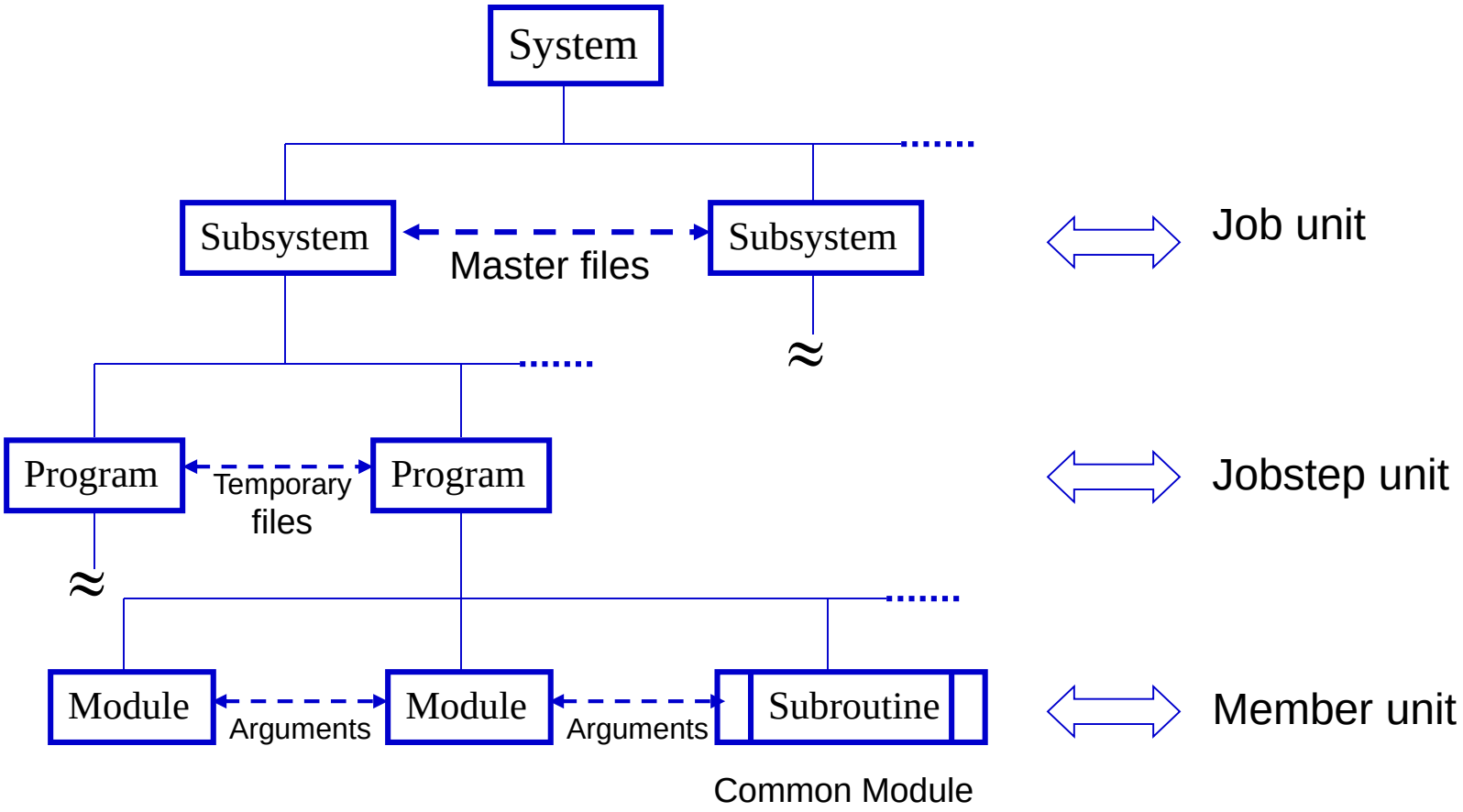


Các chỉ tiêu cơ bản để đánh giá phần mềm tốt

- *Phản ánh đúng yêu cầu người dùng (tính hiệu quả)*
- *Chứa ít lỗi tiềm tàng*
- *Giá thành không vượt quá giá ước lượng ban đầu*
- *Dễ vận hành, sử dụng*
- *Tính an toàn và độ tin cậy cao*

2.1.1. Đặc điểm của phần mềm

Cấu trúc phần mềm là cấu trúc phân cấp



Các khái niệm (Software concepts)

- *Khái niệm tính môđun (modularity concept)*
- *Khái niệm chi tiết hóa dần từng bước (stepwise refinement concept)*
- *Khái niệm trừu tượng hóa (abstraction concept): về thủ tục, điều khiển, dữ liệu*
- *Khái niệm che giấu thông tin (information hiding concept)*

Tính môđun (Modularity)

- *Là khả năng phân chia phần mềm thành các môđun ứng với các chức năng, đồng thời cho phép quản lý tổng thể*
- *Hai phương pháp phân chia môđun:*
  - *Theo chiều sâu (depth, thẳng đứng)?*
  - *Theo chiều rộng (width, nằm ngang)?*
- *Quan hệ giữa các môđun: qua các đối số (arguments)*

Khái niệm Che giấu thông tin

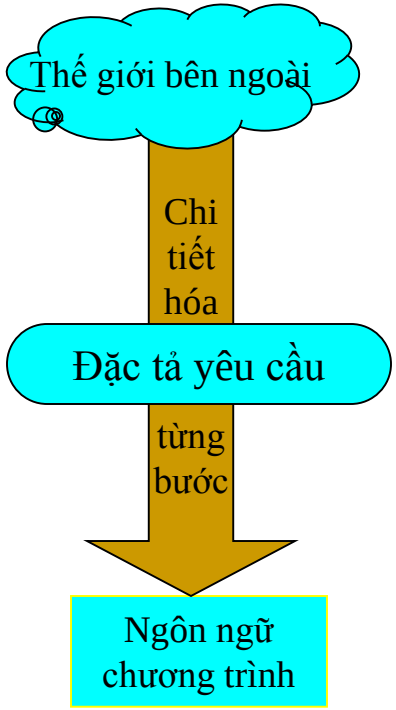
- *"các môđun nên được đặc trưng bởi những quyết định thiết kế sao cho mỗi môđun ẩn kín đối với các môđun khác" [Parnas1972]*

Khái niệm Trừu tượng hóa

- *Trừu tượng hóa cho phép tập trung vấn đề ở mức tổng quát, gạt đi những chi tiết mức thấp ít liên quan*
- *3 mức trừu tượng*
  - *Trừu tượng thủ tục*
  - *Trừu tượng dữ liệu*
  - *Trừu tượng điều khiển*

Chi tiết hóa dần từng bước

Cách tiếp cận từ trên xuống (top-down approach)



Trừu tượng hóa mức cao:
Thế giới bên ngoài,
trạng thái chưa rõ ràng

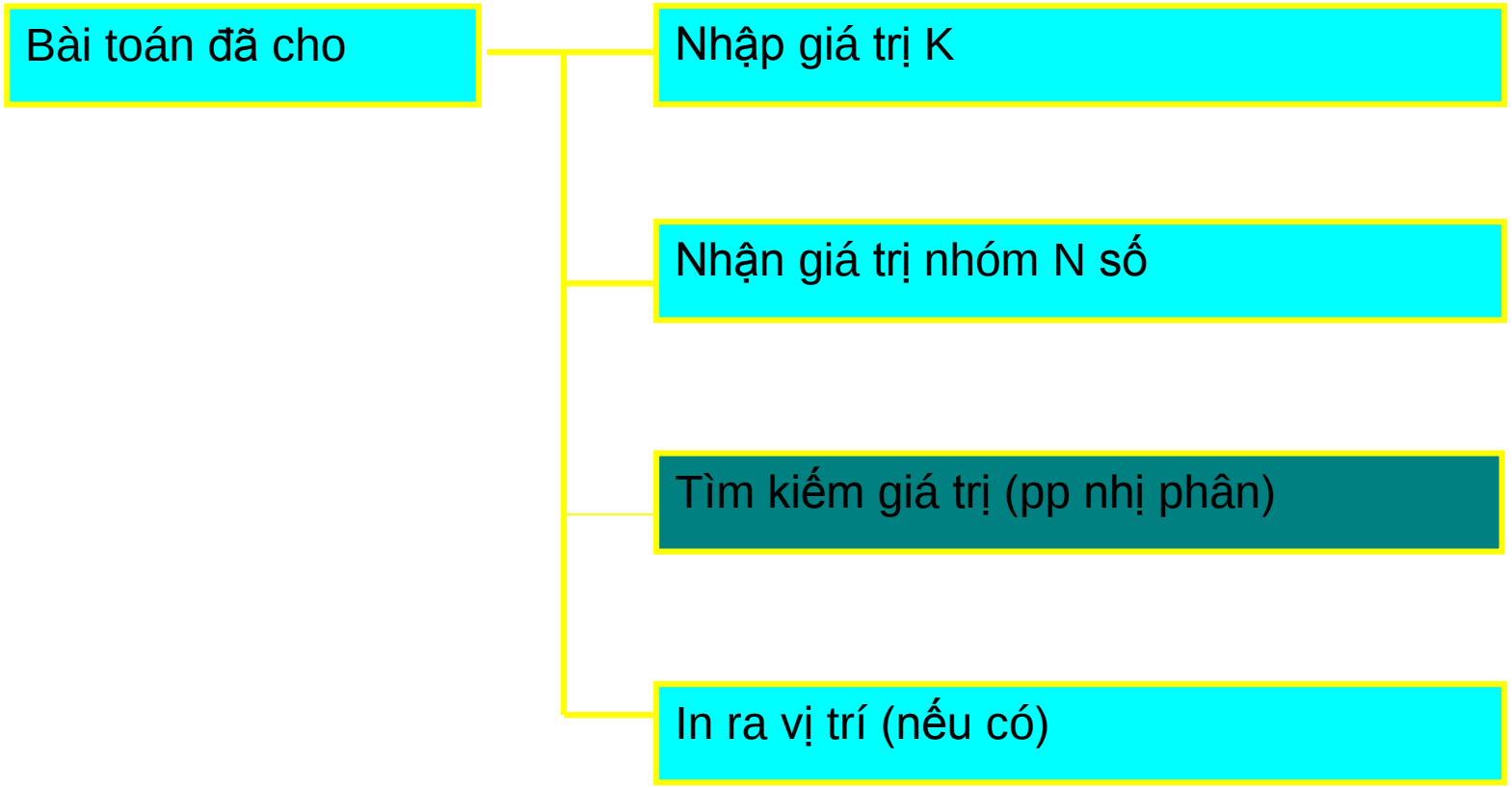
Trừu tượng hóa mức trung gian:
Xác định yêu cầu và đặc tả
những định nghĩa yêu cầu

Trừu tượng hóa mức thấp:
Từng lệnh của chương trình được
viết bởi ngôn ngữ thủ tục nào đó

Ví dụ: Trình tự giải quyết vấn đề từ mức thiết kế chương trình đến mức lập trình

- *Bài toán: từ một nhóm N số khác nhau tăng dần, hãy tìm số có giá trị bằng K (nhập từ ngoài vào) và in ra vị trí của nó*
- *Giải từng bước từ khái niệm đến chi tiết hóa từng câu lệnh bởi ngôn ngữ lập trình nào đó*
- *Chọn giải thuật tìm kiếm nhị phân (pp nhị phân)*

Cụ thể hóa thủ tục qua các chức năng



Cụ thể hóa bước tiếp theo

Tìm kiếm giá trị
(pp nhị phân)

- Xác lập phạm vi mảng số
- Lặp lại xử lý tìm kiếm giá trị K trong phạm vi tìm kiếm

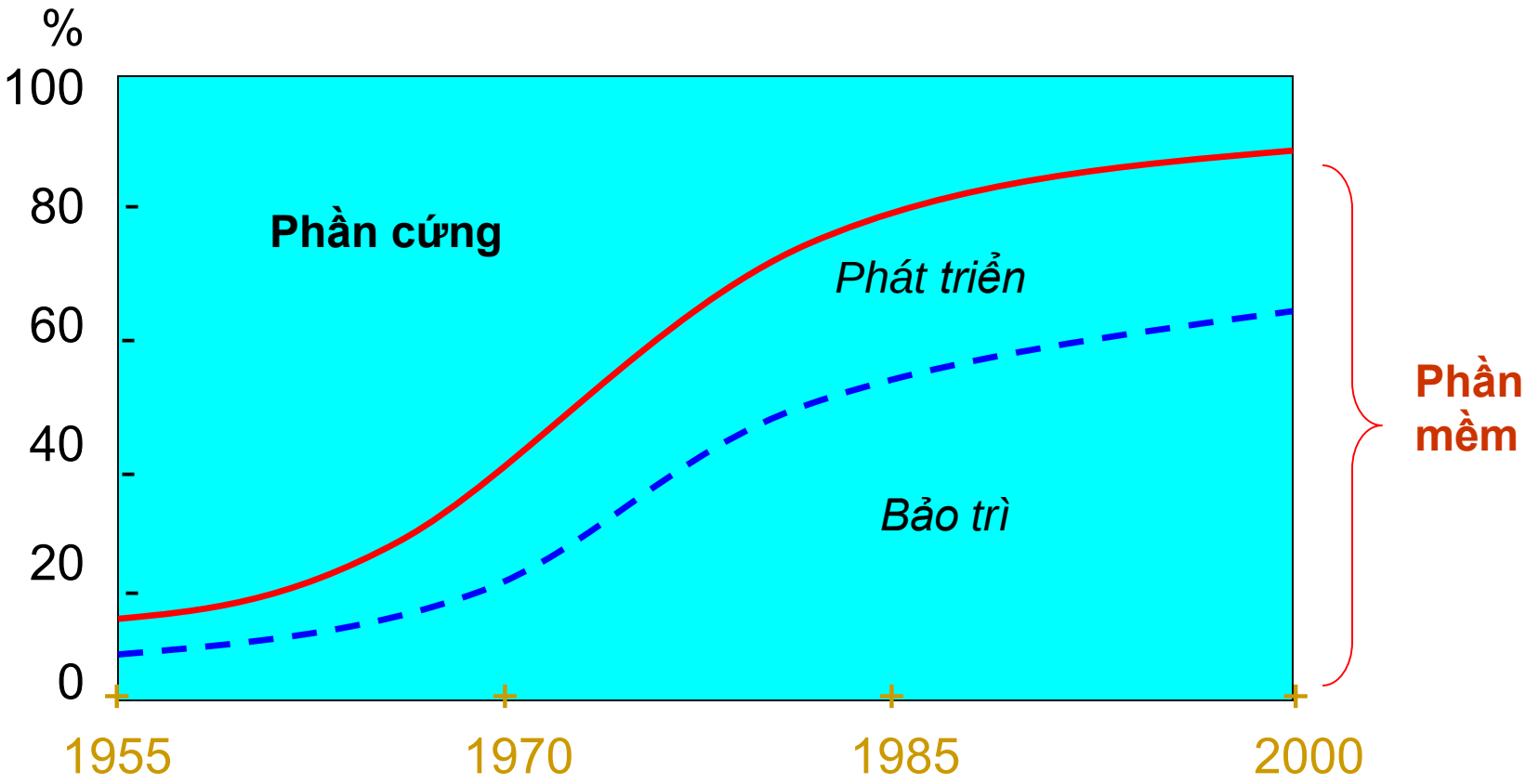
Lặp lại tìm kiếm K
trong phạm vi tìmkiếm

- Tìm vị trí giữa phân đôi mảng
- So sánh K với giá trị giữa
- Đặt lại phạm vi tìm kiếm

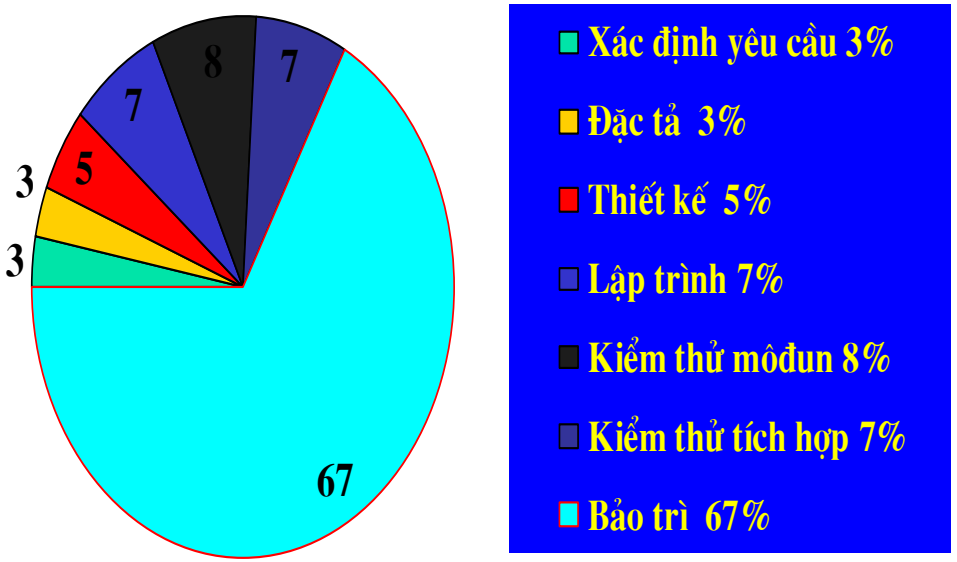
2.1.1. Đặc điểm của phần mềm

- Ngày càng phức tạp.
- Yêu cầu triển khai nhanh.
- Yêu cầu chất lượng cao
- Nhưng có nhiều phần mềm không được hoàn thành. Tại sao?

So sánh chi phí cho Phần cứng và Phần mềm



So sánh chi phí cho các pha



2.1.2. Các vấn đề của phát triển phần mềm

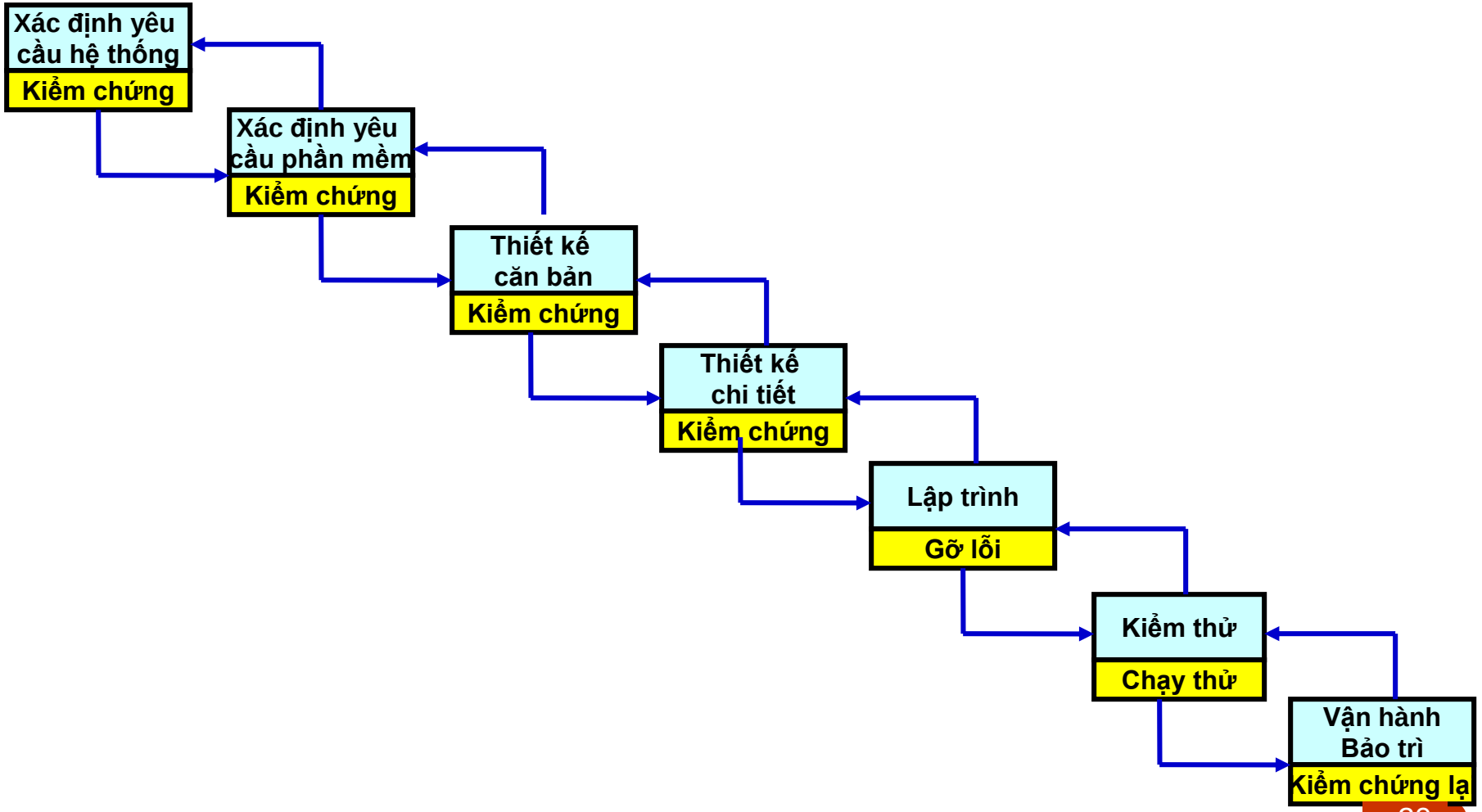
- *Hiểu sai yêu cầu của người dùng*
- *Không có khả năng đáp ứng tốc độ thay đổi nhanh chóng của yêu cầu*
- *Các modules không ghép được với nhau*
- *Phần mềm làm ra khó bảo trì và nâng cấp*
- *Phát hiện muộn các sai lầm trong dự án*
- *Chất lượng phần mềm thấp*
- *Các thành viên trong nhóm không cùng nỗ lực*
- *Quá trình xây dựng và phát hành không tin cậy*

Vòng đời phần mềm (*Software life-cycle*)

- *Vòng đời phần mềm là thời kỳ tính từ khi phần mềm được sinh (tạo) ra cho đến khi chết đi (từ lúc hình thành đáp ứng yêu cầu, vận hành, bảo dưỡng cho đến khi loại bỏ không đâu dùng)*
- *Quy trình phần mềm (vòng đời phần mềm) được phân chia thành các pha chính: phân tích, thiết kế, triển khai , kiểm thử, bảo trì. Biểu diễn các pha có khác nhau theo từng người*

Vòng đời phần mềm (*Software life-cycle*)

- *Mô hình vòng đời phần mềm của Boehm*



Vòng đời phần mềm (*Software life-cycle*)

(1) Pha xác định yêu cầu và thiết kế có vai trò quyết định đến chất lượng phần mềm, chiếm phần lớn công sức so với lập trình, kiểm thử và chuyển giao phần mềm

(2) Pha cụ thể hóa cấu trúc phần mềm phụ thuộc nhiều vào suy nghĩ trên xuống (top-down) và trừu tượng hóa, cũng như chi tiết hóa

(3) Pha thiết kế, chế tạo thì theo trên xuống, pha kiểm thử thì dưới lên (bottom-up)

Vòng đời phần mềm (*Software life-cycle*)

(4) Trước khi chuyển sang pha kế tiếp phải đảm bảo pha hiện nay đã được kiểm thử không còn lỗi

(5) Cần có cơ chế kiểm tra chất lượng, xét duyệt giữa các pha nhằm đảm bảo không gây lỗi cho pha sau

(6) Tư liệu của mỗi pha không chỉ dùng cho pha sau, mà chính là đối tượng quan trọng cho kiểm tra và đảm bảo chất lượng của từng quy trình và của chính phần mềm

Vòng đời phần mềm (*Software life-cycle*)

(7) Cần chuẩn hóa mẫu biểu, cách ghi chép tạo tư liệu cho từng pha, nhằm đảm bảo chất lượng phần mềm

(8) Thao tác bảo trì phần mềm là việc xử lý quay vòng trở lại các pha trong vòng đời phần mềm nhằm biến đổi, sửa chữa, nâng cấp phần mềm

2.3.1 Các PP phát triển hệ thống

A. Thiết kế cấu trúc (Structured design)

- ***Phương pháp thác nước (waterfall method)***
- ***Phương pháp phát triển song song (Parallel development)***

B. Phương pháp phát triển nhanh ứng dụng (RAD)

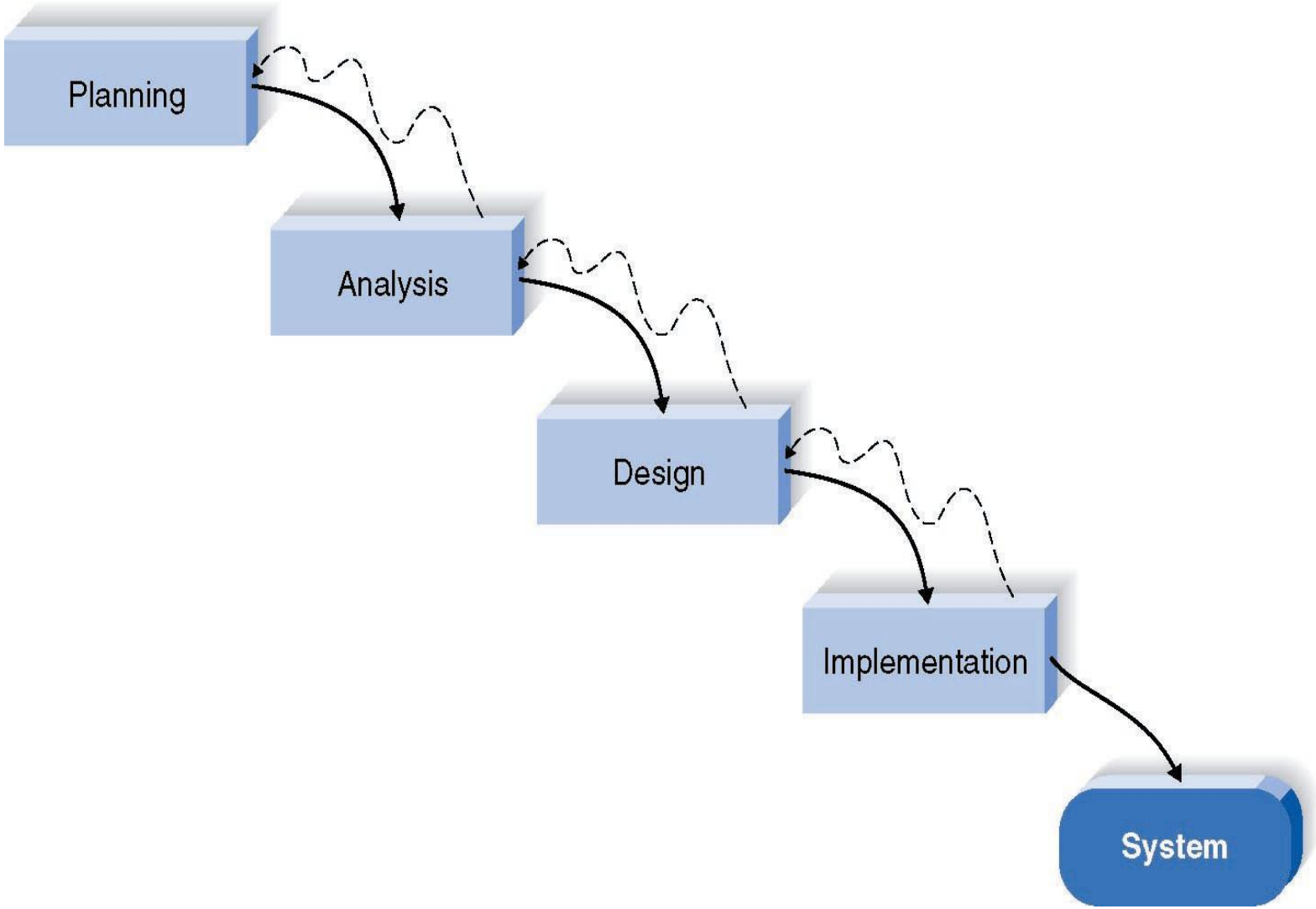
- ***Phương pháp phát triển theo các pha***
- ***Phương pháp xây dựng nguyên mẫu (prototyping)***
  - ***Thông thường (regular)***
  - ***Loại bỏ (throwaway)***

A. Thiết kế cấu trúc

- *Sử dụng phổ biến vào thập niên 1980*
- *Dự án sẽ tiến triển từ bước này sang bước tiếp theo một cách có hệ thống*
- *Thông thường, một bước phải được hoàn thành trước khi bắt đầu bước tiếp theo*

A. Thiết kế cấu trúc

Phương pháp thác nước



Phương pháp thác nước

- **Ưu điểm:**
  - Trước khi lập trình thì các yêu cầu về hệ thống được xác định rất chi tiết và đầy đủ => giảm thiểu được sự thay đổi về yêu cầu trong quá trình phát triển hệ thống
- **Nhược điểm:**
  - Việc thiết kế phải hoàn thành hoàn toàn trước khi bắt đầu viết chương trình.
  - Thời gian từ khi đề xuất dự án đến khi có sản phẩm cuối cùng thường rất dài (vài tháng -> vài năm)

A. Thiết kế cấu trúc

Phương pháp phát triển song song

- Thiết kế chung cho toàn bộ hệ thống
- Chia dự án thành một chuỗi các dự án con

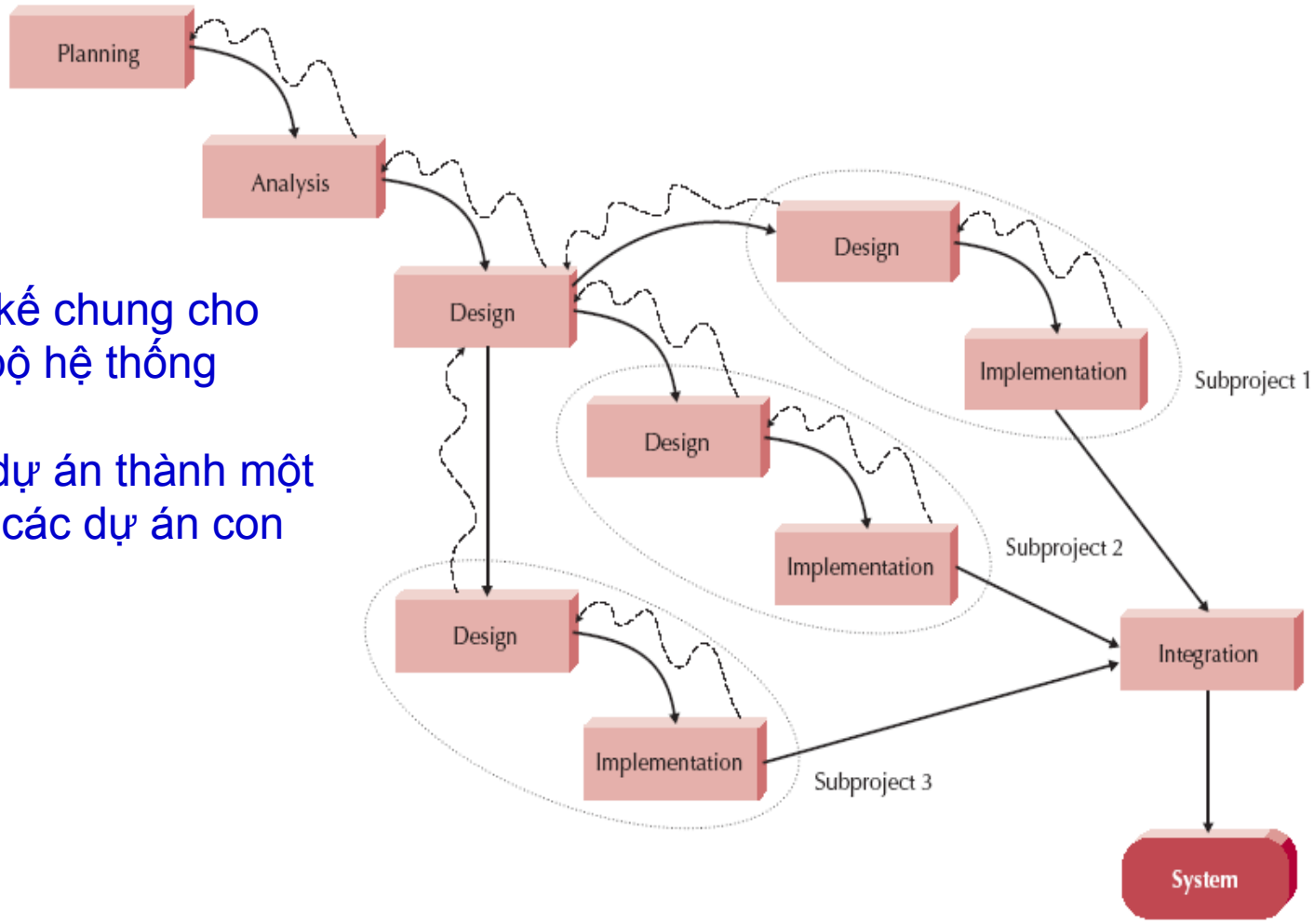


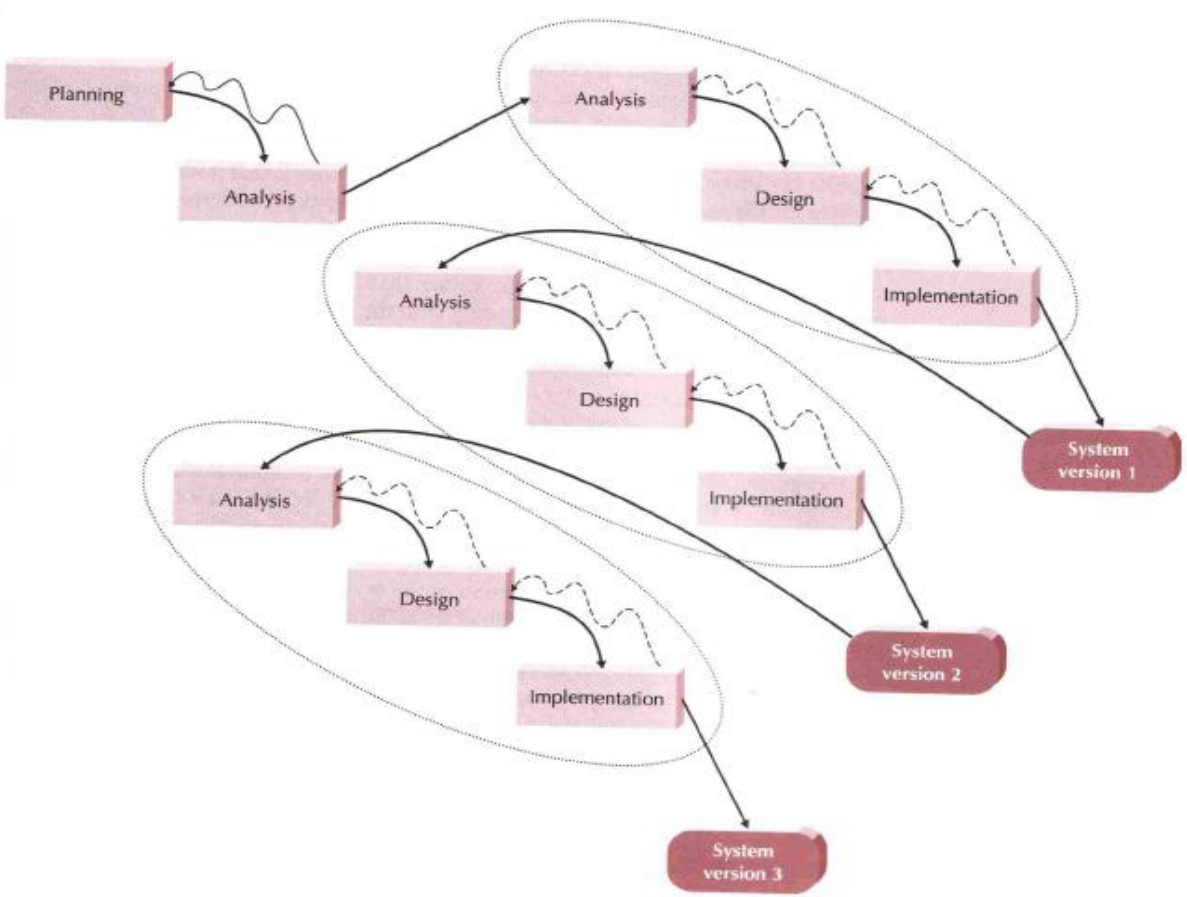
FIGURE 1-3 A Parallel Development-based Methodology

B. RAD

- *RAD (Rapid Application Development)*
  - *Tập trung giải quyết những điểm yếu của PP thiết kế cấu trúc.*
- *Các nhân tố quan trọng:*
  - *Công cụ CASE*
  - *JAD (joint application design)*
  - *Ngôn ngữ lập trình thế hệ thứ tư/visual*
  - *Công cụ tạo mã*

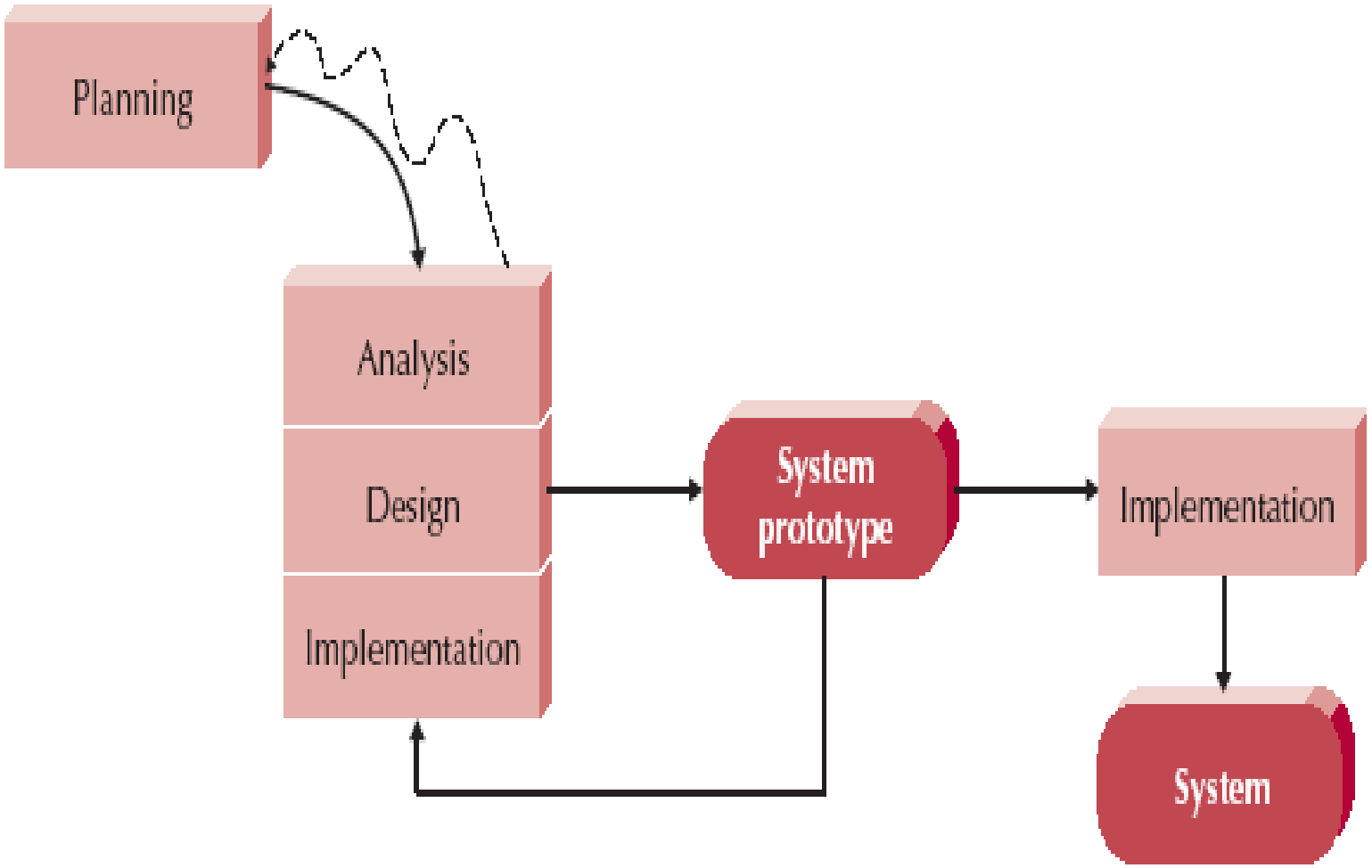
Phương pháp phát triển theo pha

- *Phân rã hệ thống tổng quát thành các chuỗi các phiên bản được phát triển một cách tuần tự*



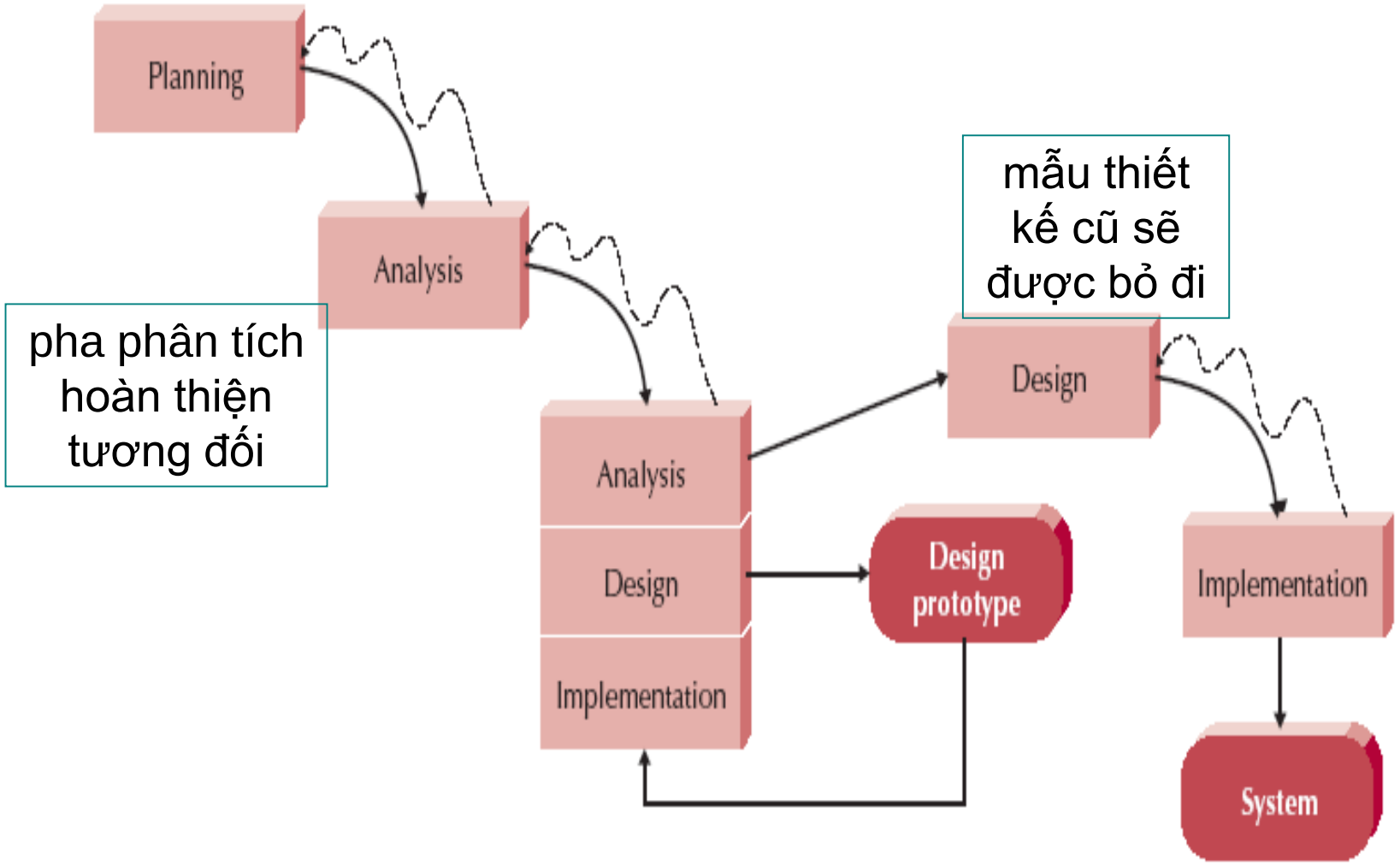
B. RAD

Pp xây dựng nguyên mẫu thông thường



B. RAD

Phương pháp xây dựng nguyên mẫu loại bỏ



Lựa chọn phương pháp phù hợp

- *Tiêu chí:*
  - *Độ rõ ràng, đầy đủ của các yêu cầu của người sử dụng*
  - *Khả năng, mức độ thành thạo về công nghệ*
  - *Độ phức tạp của hệ thống*
  - *Độ tin cậy của hệ thống*
  - *Quỹ thời gian*

Lựa chọn phương pháp phù hợp

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

Tên thư:

KTPMUD_Nhom1_baocaoso1_20150920

Tên báo cáo:

KTPMUD_Nhom1_baocaoso1_20150920.docx

Nộp báo cáo trước 12h trưa ngày thứ 3.

Tuần tới: tên đề tài + tên nhóm trưởng (số điện thoại)