



Chương 1: Tổng quan

Tài liệu tham khảo

- ❖ Bài giảng dựa chính trên bài giảng “Xây dựng phần mềm hướng đối tượng” - ThS Trần Minh Triết – ĐH KHTN
- ❖ Bài giảng “Phân tích và thiết kế hướng đối tượng” – TS Phạm Ngọc Nam – ĐH BK HN
- ❖ Bài giảng “Phân tích và thiết kế hướng đối tượng” – ThS Phạm Nguyễn Cường – ĐH KHTN
- ❖

Nội dung

- ❖ Khái niệm về Phần mềm
- ❖ Khái niệm về Chất lượng phần mềm
- ❖ Khái niệm về Công nghệ phần mềm
- ❖ Khái niệm về Quy trình phần mềm

Khái niệm về phần mềm

- ❖ Ví dụ: xét một số phần mềm sau
 - Phần mềm quản lý học sinh cấp 3.
 - Phần mềm quản lý thư viện.
 - Phần mềm quản lý nhà sách.
 - Phần mềm quản lý khách sạn.
 - Phần mềm quản lý phòng mạch tư.
 - Phần mềm quản lý sổ tiết kiệm.
 - Phần mềm quản lý giải vô địch bóng đá.
 - Phần mềm quản lý bán vé chuyến bay.
 - Phần mềm quản lý các đại lý.
 - Phần mềm xếp thời khóa biểu
 - ...

Khái niệm về phần mềm

❖ *Phần mềm dưới góc nhìn của người sử dụng:*

- **Chương trình** thực thi được trên **máy tính** hoặc **các thiết bị chuyên dụng khác**, nhằm **hỗ trợ** cho các **nhà chuyên môn** trong từng **lĩnh vực chuyên ngành** thực hiện **tốt hơn** các thao tác **nghịệp vụ** của mình

Khái niệm về phần mềm

- ❖ Môi trường triển khai phần mềm
 - Máy tính: Desktop, Laptop, Tablet PC...
 - Thiết bị chuyên dụng:
 - Thiết bị di động: PDA, Pocket PC, Điện thoại di động...
 - Các thiết bị chuyên dụng khác: set-top box, router, firewall (phần cứng...)
- ❖ Hỗ trợ làm tốt hơn các thao tác nghiệp vụ:
 - Tin học hóa nghiệp vụ hiện đang làm thủ công
 - Cải tiến chức năng nghiệp vụ hiện đang được thực hiện trên máy tính
 - Đề ra, xây dựng và triển khai chức năng nghiệp vụ mới

Khái niệm về phần mềm

❖ *Phần mềm dưới góc nhìn của chuyên viên Tin học:*

- Đây là một hệ thống bao gồm 3 thành phần cơ bản:

- Thành phần giao tiếp
- Thành phần xử lý
- Thành phần lưu trữ

cần được xây dựng để thực hiện theo yêu cầu của người sử dụng

Khái niệm về phần mềm

❖ Thành phần giao tiếp (Giao diện):

- Cho phép **tiếp nhận** các yêu cầu về việc sử dụng phần mềm từ người sử dụng (chọn công việc muốn thực hiện và cung cấp các dữ liệu nguồn liên quan đến công việc đó), từ các thiết bị thu thập dữ liệu (cân, nhiệt kế, tế bào quang điện...), hoặc từ các phần mềm khác.
- Cho phép **trình bày** các kết quả của việc thực hiện các yêu cầu cho người dùng (kết quả của công việc khi thực hiện trên máy tính) hoặc điều khiển hoạt động các thiết bị điều khiển (đóng/mở cửa, dùng hay cho chuyển động...)
- Một cách tổng quát, thành phần giao tiếp cho phép **nhập/xuất thông tin** cùng với hình thức trình bày/giao tiếp tương ứng.
- Mục tiêu chính của thành phần này là đưa thông tin từ thế giới bên ngoài phần mềm (người sử dụng, các thiết bị, phần mềm khác...) và bên trong, hoặc ngược lại.

Khái niệm về phần mềm

❖ Thành phần xử lý:

- **Kiểm tra tính hợp lệ** của thông tin nguồn được cung cấp từ người dùng theo các quy định ràng buộc trong thế giới thực
 - Ví dụ: chỉ cho mượn tối đa 3 quyển sách, mỗi lớp học không quá 50 học sinh...
- **Tiến hành xử lý** cho ra kết quả mong đợi theo **quy định** tính toán có sẵn **trong thế giới thực**
 - Ví dụ: quy tắc tính tiền phạt khi trả sách về, quy định tính tiền điện, trả góp khi mua nhà...
- Hoặc **tiến hành xử lý** theo **thuật giải tự đề xuất**
 - Ví dụ: xếp thời khóa biểu tự động, nén ảnh, phát sinh nước cờ của máy tính...

Khái niệm về phần mềm

❖ Thành phần xử lý:

- Việc xử lý dựa trên thông tin nguồn từ **người sử dụng cung cấp**
 - Ví dụ: tính nghiệm phương trình bậc 2 dựa trên các hệ số nhập vàohoặc **dữ liệu lưu trữ có sẵn**
 - Ví dụ: tính tồn kho dựa trên các phiếu nhập xuất đã lưu trữhoặc **cả hai**
 - Ví dụ: tính tiền phạt dựa trên ngày trả sách được nhập vào và thông tin về loại sách đã được lưu trữ
- Việc xử lý cho ra **kết quả** có thể dùng để **xuất cho người dùng xem qua thành phần giao diện**, hay **lưu trữ lại qua thành phần lưu trữ**, hoặc **cả hai**

Khái niệm về phần mềm


- ❖ Một cách tổng quát, thành phần xử lý là hệ thống chuyên xử lý tính toán, biến đổi dữ liệu,
 - dùng thông tin nguồn từ các thành phần giao diện (**chức năng nhập**) hay thành phần dữ liệu (**chức năng đọc**);
 - Kiểm tra tính hợp lệ (chức năng kiểm tra) và sau đó tiến hành xử lý (chức năng xử lý) – nếu cần thiết
 - Để cho ra kết quả sẽ được trình bày thông qua thành phần giao diện (**chức năng xuất**) hoặc lưu trữ lại trong thành phần dữ liệu (**chức năng ghi**)

Khái niệm về phần mềm

❖ Thành phần lưu trữ (thành phần dữ liệu)

- Cho phép lưu trữ lại (**chức năng ghi**) các kết quả đã xử lý trên bộ nhớ phụ với tổ chức lưu trữ được xác định trước
 - Ví dụ: Việc mượn sách đã được kiểm tra hợp lệ, bảng lương tháng đã được tính
 - Ví dụ: tập tin có cấu trúc, tập tin nhị phân, cơ sở dữ liệu
- Cho phép truy xuất lại (**chức năng đọc**) các dữ liệu đã lưu trữ phục vụ cho các hàm xử lý tương ứng
- Một cách tổng quát thành phần dữ liệu là hệ thống chuyên **đọc ghi** dữ liệu cùng với mô hình tổ chức lưu trữ dữ liệu tương ứng. Mục tiêu chính của thành phần này là **chuyển đổi dữ liệu** giữa bộ nhớ chính và bộ nhớ phụ

Khái niệm về phần mềm



Các loại
phần mềm
khác?

Khái niệm lớp phần mềm

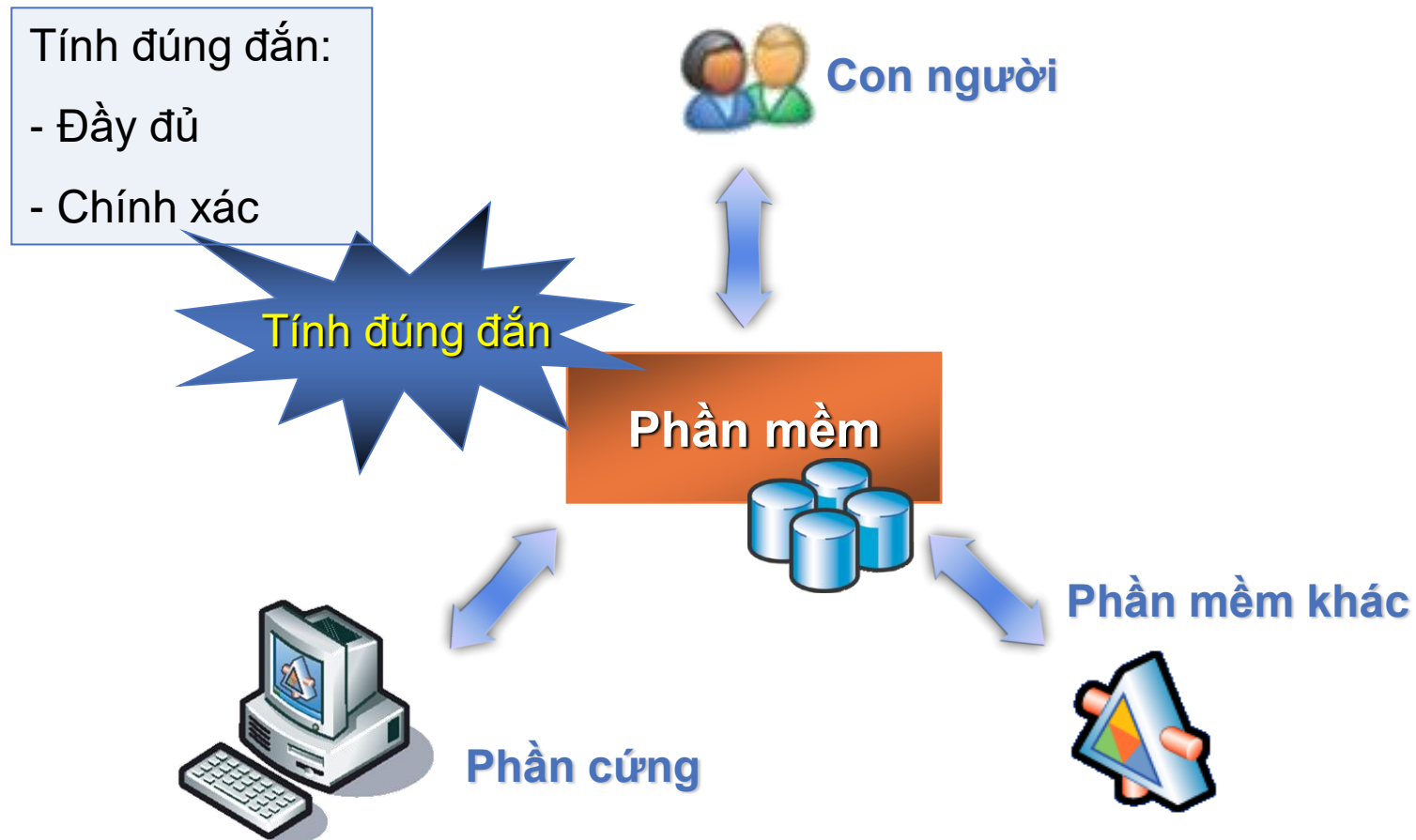
❖ Lớp phần mềm:

- Lớp phần mềm là hệ thống các phần mềm trên **cùng một lĩnh vực hoạt động** nào đó
- Do cùng lĩnh vực hoạt động nên các phần mềm cùng lớp thường có **cấu trúc** và **chức năng** tương tự nhau

❖ Mục tiêu của ngành Công nghệ Phần mềm

- Xây dựng được phần mềm **có chất lượng**
- Dễ dàng **xây dựng phần mềm mới** từ các phần mềm **có sẵn cùng lớp**

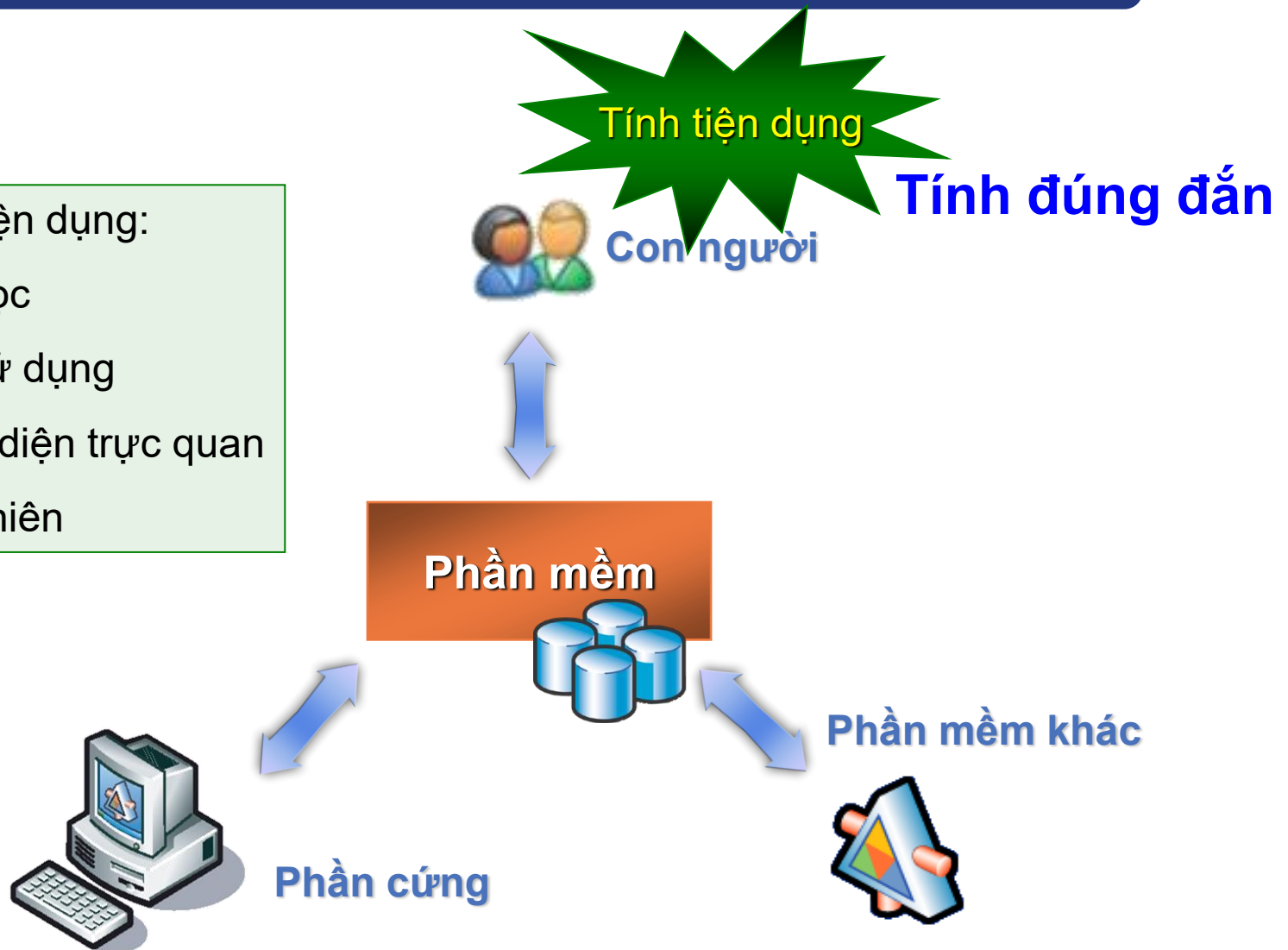
Chất lượng phần mềm



Chất lượng phần mềm

Tính tiện dụng:

- Dễ học
- Dễ sử dụng
- Giao diện trực quan
- Tự nhiên



Chất lượng phần mềm

Tính hiệu quả:

- Tối ưu sử dụng CPU*
- Tối ưu sử dụng bộ nhớ*
- Tối ưu sử dụng thiết bị



Con người

Tính đúng đắn
Tính tiện dụng



Phần mềm



Tính hiệu quả



Phần cứng



Phần mềm khác



Chất lượng phần mềm

Tính tương thích:

- Import/Export dữ liệu
- Tương tác



Con người

Tính đúng đắn
Tính tiện dụng
Tính hiệu quả



Phần mềm



Phần mềm khác

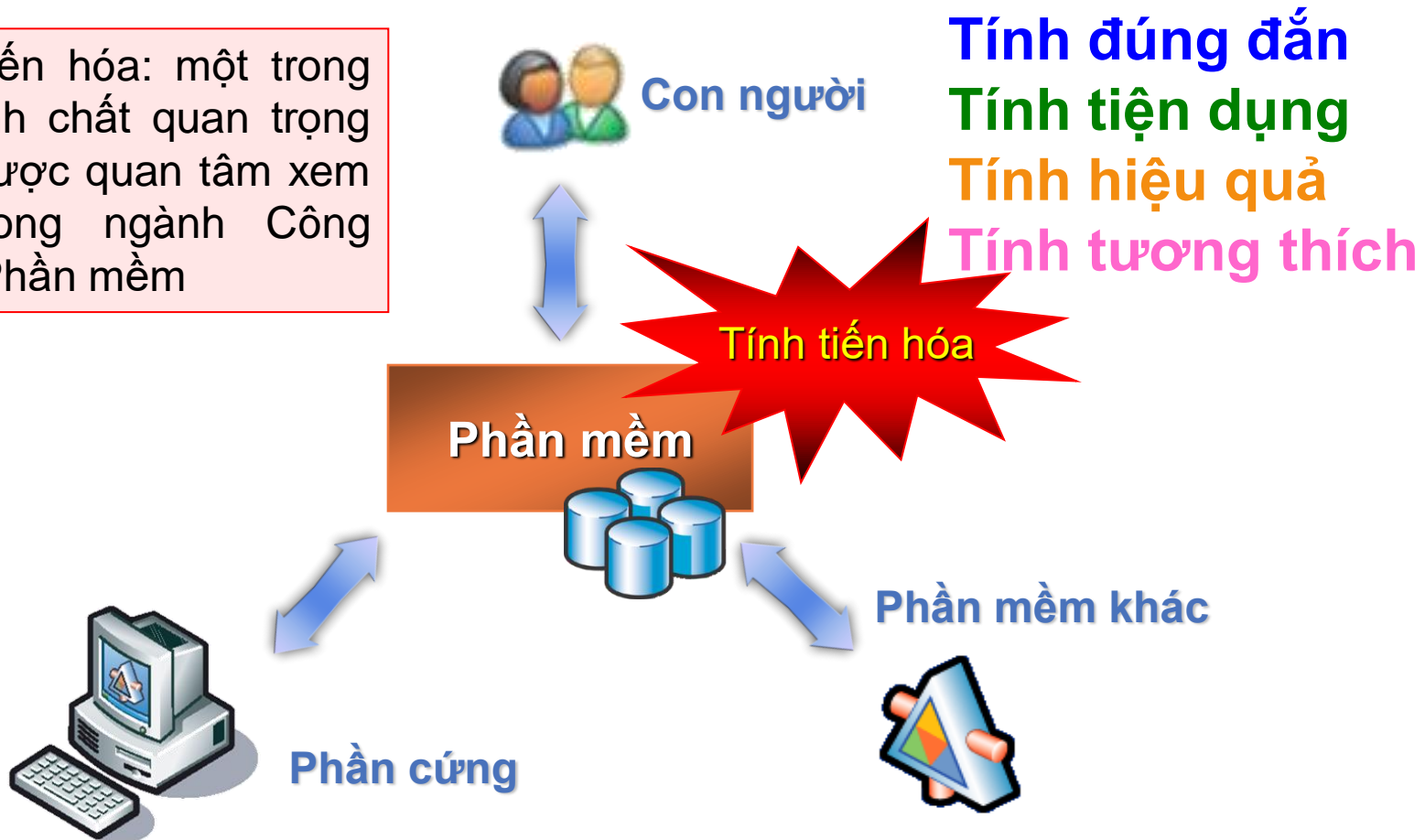
Phần cứng



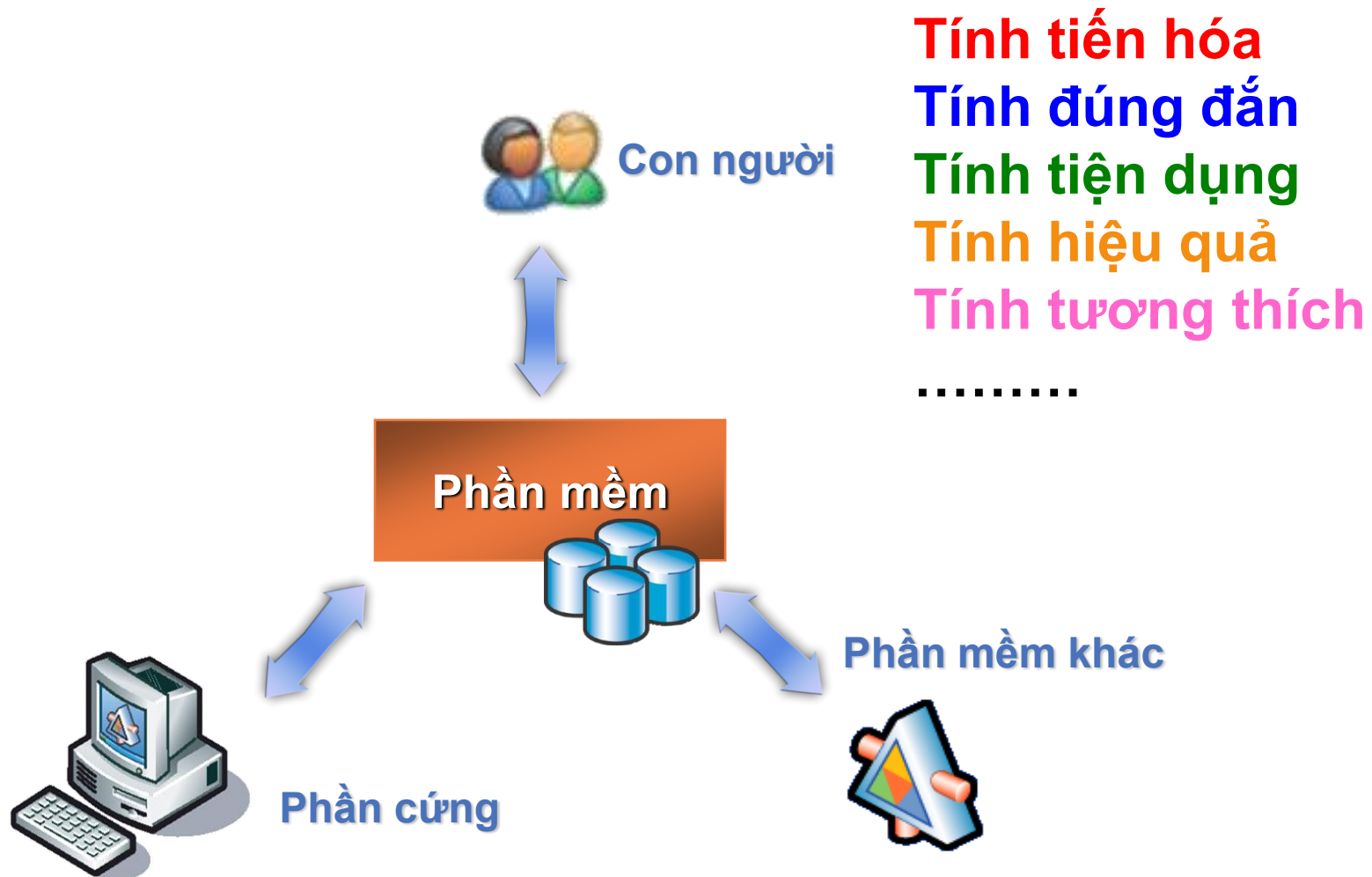
Tính tương thích

Chất lượng phần mềm

Tính tiến hóa: một trong các tính chất quan trọng nhất được quan tâm xem xét trong ngành Công nghệ Phần mềm



Chất lượng phần mềm



Chất lượng phần mềm

❖ Tính dễ kiểm tra: việc kiểm tra các thành phần phù hợp với yêu cầu phần mềm là dễ dàng nhất có thể được

❖ Tính dễ sửa lỗi: khi có sự không phù hợp (so với yêu cầu) trong quá trình kiểm tra một thành phần, việc phát hiện chính xác “vị trí lỗi” và sửa lỗi là nhanh nhất có thể được.

❖ Tính dễ bảo trì: khi cần nâng cấp, cải tiến một thành phần (theo yêu cầu mới), việc cập nhật phần mềm là nhanh, chính xác nhất có thể được và đặc biệt là cố gắng hạn chế ảnh hưởng đến các thành phần khác

❖ Tính tái sử dụng: các thành phần đã thực hiện có thể dùng lại trong các phần mềm cùng lớp (hoặc cùng lĩnh vực) với thời gian và công sức ít nhất có thể được

.....

Phát triển của Công nghệ phần mềm

- ❖ Giai đoạn 1 (1950 – giữa 1960)
 - Xử lý theo lô, xử lý tập trung, ít xử lý phân tán, ít sửa đổi phần mềm
- ❖ Giai đoạn 2 (từ giữa 1960 đến giữa 1970)
 - Hệ thống đa chương trình và đa người dùng
 - Bắt đầu cuộc “khủng hoảng” phần mềm
- ❖ Giai đoạn 3 (từ giữa 1970 đến giữa 1980)
 - Sự phát triển và sử dụng rộng rãi máy tính cá nhân
 - Sự phát triển của các công ty phần mềm
- ❖ Giai đoạn 4 (từ giữa 1980 đến nay)
 - Phần cứng ngày càng phát triển
 - Hệ thống phần mềm ngày càng đa dạng, phong phú, xử lý ngày càng phức tạp, công nghệ ngày càng phát triển...

Cuộc khủng hoảng phần mềm

- ❖ Số lượng các phần mềm tăng vọt (do sự phát triển của phần cứng: tăng khả năng, giá thành hạ)
- ❖ Có quá nhiều khuyết điểm trong các phần mềm được dùng trong xã hội:
 - Thực hiện không đúng yêu cầu (tính toán sai, không ổn định...)
 - Thời gian bảo trì nâng cấp quá lâu, chi phí cao, hiệu quả thấp
 - Khó sử dụng
 - Thực hiện chậm
 - Không chuyển đổi dữ liệu giữa các phần mềm
 - ...

Cuộc khủng hoảng phần mềm

❖ Một số kết luận:

- Việc tăng vọt số lượng phần mềm là điều hợp lý và sẽ còn tiếp diễn
- Các khuyết điểm của phần mềm có nguồn gốc chính từ phương pháp, cách thức và quy trình tiến hành xây dựng phần mềm:
 - **Cảm tính:** mỗi người theo một phương pháp riêng
 - **Thô sơ, đơn giản:** chỉ tập trung vào việc lập trình mà ít quan tâm đến các công việc cần làm khác (khảo sát hiện trạng, phân tích yêu cầu, thiết kế...)
 - **Thủ công:** còn thiếu các công cụ hỗ trợ quy trình phát triển

Hiện trạng cuộc sống

- ❖ Thế giới phụ thuộc vào sự phát triển công nghệ
- ❖ Yêu cầu sự phát triển của công nghệ phần mềm
- ❖ Các ứng dụng của công nghệ phần mềm: càng ngày càng được mở rộng và phức tạp hơn
- ❖ Nhu cầu thị trường tăng: đòi hỏi tăng năng suất, nâng cao chất lượng nhưng lại giảm thiểu thời gian.
- ❖ Tuy nhiên lại thiếu nguồn nhân lực thực sự có trình độ.

Thống kê về các dự án phần mềm

❖ The Robbins-Gioia Survey (2001)

- Khảo sát trên 232 người làm việc tại các công ty đang triển khai hệ thống ERP. 51%: không thành công. 49% còn lại (46% không thể nâng cấp)

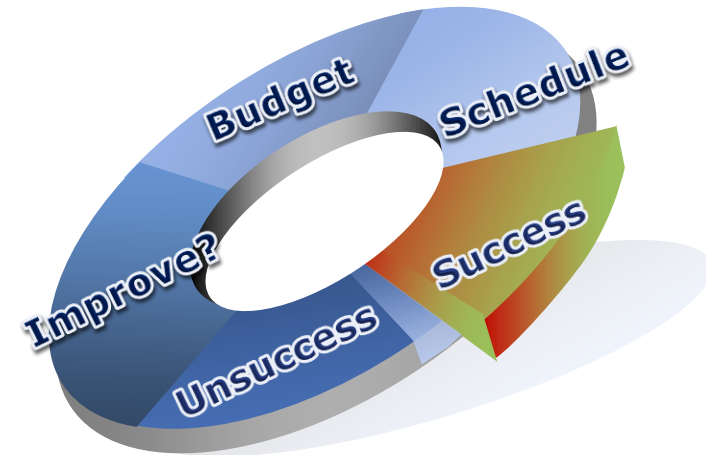
❖ The KPMG Canada Survey (1997)

- Khảo sát trên 1450 phiếu khảo sát, phân tích 176 trường hợp.
- 61% các dự án được xem là thất bại
- Hơn 1/3 các dự án vượt quá ngân sách ước tính ban đầu.
- Thất thoát ngân sách cho các dự án công nghệ thông tin hàng năm lên đến hàng tỷ đôla. Canada (\$25 billion per year)

Thống kê về các dự án phần mềm

❖ The Chaos Report (1995)

- 365 respondents representing 8,380 applications.
- 31.1% cancelled.
- 52.7% cost over 189% original budget.
- 16.2% on-time and on-budget



❖ The OASIG Study (1995)

- 7 in 10 IT projects “fail” in some respect

Theo nguồn thống kê của Ó IT Cortex

Những thách thức



Thách thức

- Công nghệ thay đổi nhanh
- Công việc phát triển phần mềm là công việc tập thể
- Sự chuyên môn hoá và cách thức làm việc phân tán



Kết luận:

- 1 IT Project: Tỷ lệ thất bại nhiều hơn thành công
- Chỉ 1 trong 5 dự án thì thực sự đáp ứng được nhu cầu
- Dự án càng lớn khả năng thất bại càng cao.

➔ Có thành công nhưng quá nhiều thất bại

Vấn đề phát sinh trong phát triển HT

- ❖ Hiểu không đúng những gì người dùng cần
- ❖ Không thể thích ứng với các thay đổi về yêu cầu của hệ thống
- ❖ Các Module không khớp với nhau
- ❖ Phần mềm khó bảo trì và nâng cấp, mở rộng
- ❖ Phát hiện trễ các lỗi hổng của dự án
- ❖ Chất lượng phần mềm kém
- ❖ Hiệu năng của phần mềm thấp
- ❖ Các thành viên trong nhóm không biết được ai đã thay đổi cái gì, khi nào, ở đâu, tại sao phải thay đổi
- ❖ Quá trình build-and-release không đáng tin cậy

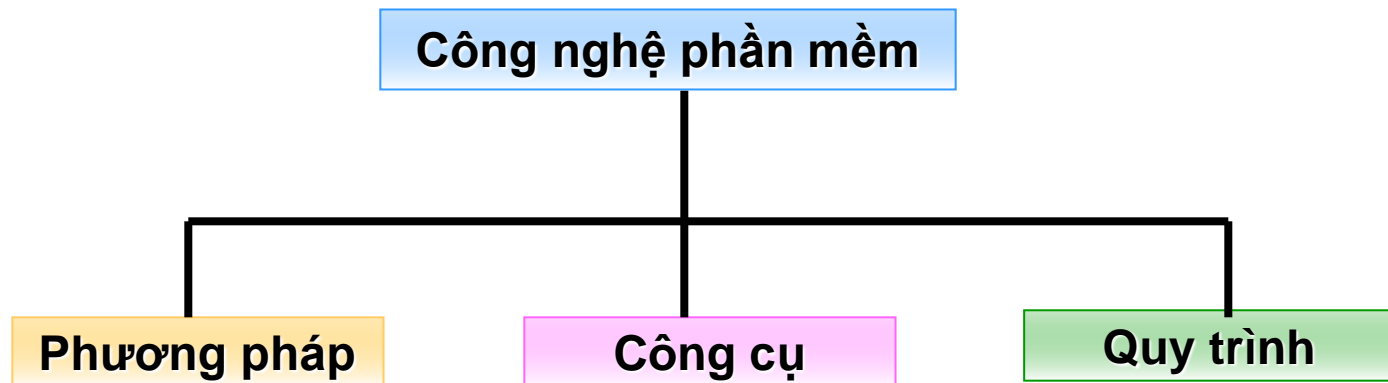
Các nguyên nhân cốt lõi

- ❖ Sự quản lý yêu cầu người dùng không đầy đủ
- ❖ Trao đổi thông tin mơ hồ và không đầy đủ
- ❖ Kiến trúc không vững chắc
- ❖ Độ phức tạp vượt quá tầm kiểm soát
- ❖ Có những mâu thuẫn không phát hiện được giữa yêu cầu, thiết kế, và cài đặt.
- ❖ Kiểm chứng không đầy đủ
- ❖ Sự lượng giá chủ quan về tình trạng của dự án
- ❖ Sự chậm trễ trong việc giảm rủi ro do mô hình thác nước
- ❖ Sự lan truyền không thể kiểm soát của các thay đổi
- ❖ Thiếu các công cụ tự động hóa

Công nghệ phần mềm

❖ Khái niệm:

- Công nghệ phần mềm là ngành khoa học nghiên cứu về việc xây dựng các phần mềm có chất lượng trong thời gian và chi phí hợp lý



Công nghệ phần mềm

- ❖ Các đối tượng nghiên cứu của Công nghệ phần mềm :
 - Quy trình phần mềm:
 - Hệ thống các giai đoạn mà quá trình phát triển phần mềm phải trải qua,
 - với mỗi giai đoạn cần xác định rõ:
 - Mục tiêu, kết quả nhận từ giai đoạn trước đó,
 - Kết quả chuyển giao cho giai đoạn kế tiếp
 - Phương pháp phát triển phần mềm:
 - Hệ thống các hướng dẫn cho phép từng bước thực hiện một giai đoạn nào đó trong quy trình phần mềm
 - Công cụ và Môi trường phát triển phần mềm:
 - Hệ thống các phần mềm trợ giúp trong lĩnh vực xây dựng phần mềm
 - Hỗ trợ các chuyên viên tin học trong các bước xây dựng phần mềm theo một phương pháp nào đó với một quy trình được chọn trước

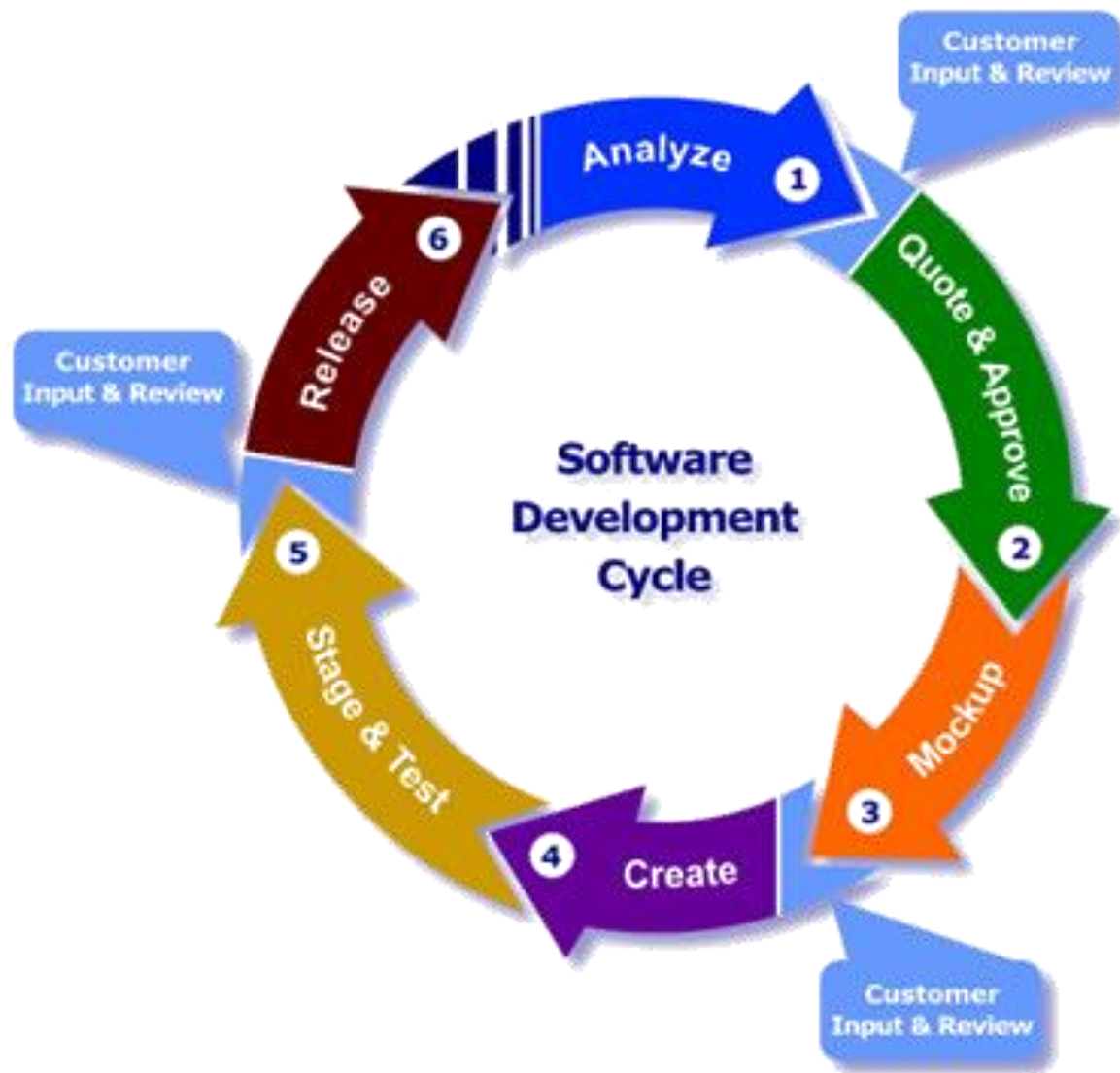
Để phát triển phần mềm cần nắm

- ❖ Tiến trình làm phần mềm
- ❖ Cách quản lý tiến trình
- ❖ Các thể hiện, ghi chép các kết quả phát sinh trong quá trình thực hiện tiến trình.
- ❖ Các công nghệ cụ thể thực hiện trong tiến trình làm phần mềm.

Tiến trình phát triển hệ thống

- ❖ Tổng quát Tiến trình (Process) xác định:
 - Who - ai.
 - What - làm gì
 - Where - ở đâu
 - When - làm khi nào
 - How - làm như thế nào để đạt tới mục đích mong muốn.
- ❖ Software Development Process
- ❖ Rational Unified Process - RUP

Software Development Life Cycle - SDLC



Tiến trình phát triển hệ thống

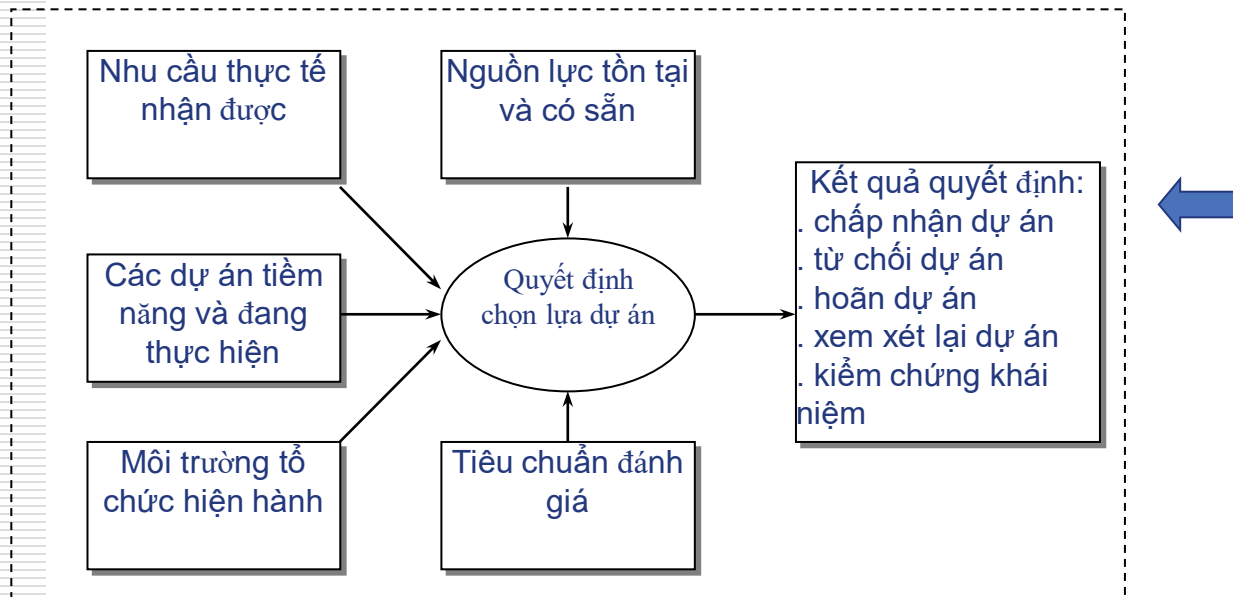
- ❖ Lập kế hoạch (Planning)
- ❖ Phân tích yêu cầu (Requirements Analysis)
- ❖ Thiết kế (Design)
- ❖ Phát triển và kiểm tra (Development & Testing)
- ❖ Triển khai và thực hiện (Deployment & Implement)
- ❖ Bảo trì (Maintenance)
- ❖ Nâng cấp (Improvement)

Lập kế hoạch (Planning)

- ❖ Xác định giá trị kinh doanh của hệ thống
- ❖ Phân tích tính khả thi
- ❖ Xây dựng kế hoạch công việc
- ❖ Xác định nguồn nhân lực cho dự án
- ❖ Ước lượng chi phí
- ❖ Điều khiển và quản lý dự án

Lập kế hoạch (Planning)

❖ Xác định và chọn lựa dự án



Xác định dự án
tiềm năng

Phân loại và xếp
hạng dự án

Chọn lựa dự án

Lập kế hoạch (Planning)

- ❖ Lập kế hoạch và khởi tạo dự án
 - Thành lập đội ngũ ban đầu
 - Khảo sát tổng thể hệ thống
 - Lập kế hoạch
 - Xác định phạm vi, nguồn lực, các nguyên tắc làm việc
 - Đánh giá khả thi
 - Xây dựng tài liệu mô tả hệ thống

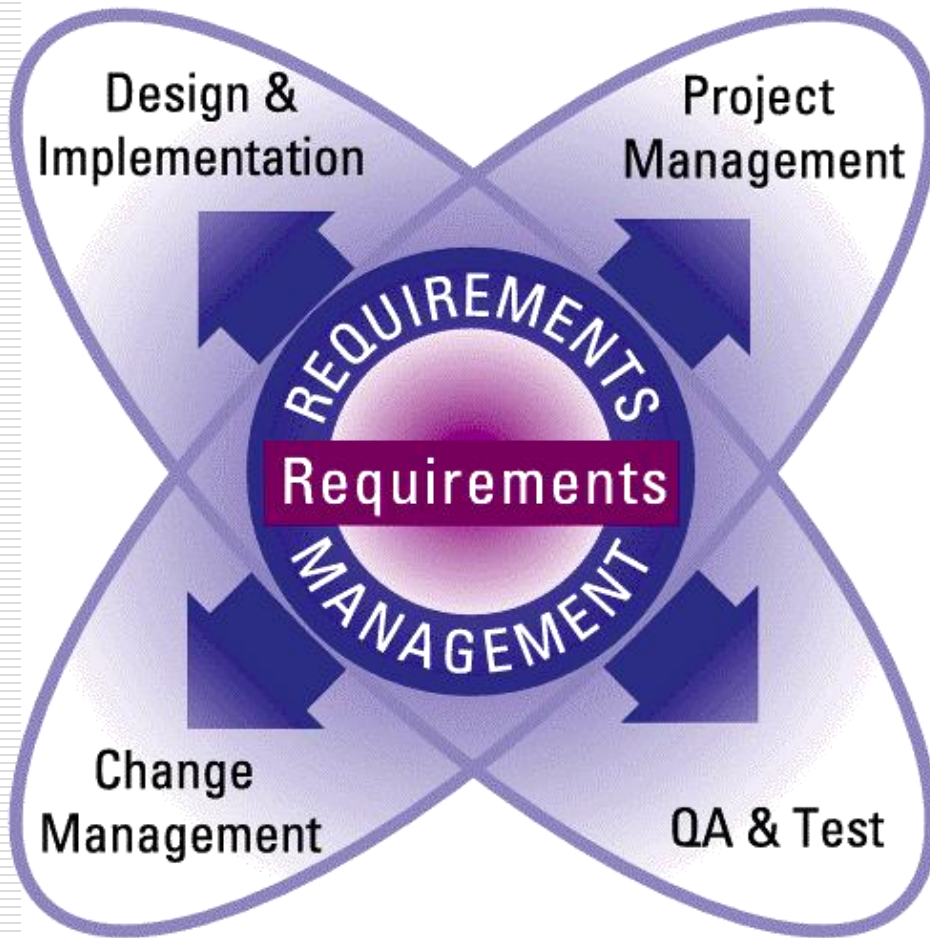
- ❖ Yêu cầu người sử dụng là mục tiêu chính phát triển hệ thống
- ❖ Một thiết kế ban đầu có thể không hoàn chỉnh so với các yêu cầu chính
- ❖ Việc phát hiện và bổ sung các thiếu sót trong thiết kế sẽ làm tăng giá thành, tổn thời gian hoặc hỏng dự án.
- ❖ Thời gian và chi phí để phát triển một thiết kế sai là không thể bù đắp

Phân tích hệ thống

- ❖ Xác định yêu cầu hệ thống
- ❖ Cấu trúc hóa yêu cầu: mô hình hoá và phân tích yêu cầu có thể dùng
- ❖ Phát sinh các phương án hệ thống và chọn lựa phương án khả thi nhất
- ❖ Phương pháp hướng đối tượng:
 - Xác định các lớp và đối tượng quan trọng
 - Mô tả yêu cầu hệ thống qua việc xác định tác nhân và use case
 - Mô tả kịch bản chức năng qua các sơ đồ động

- ❖ Yêu cầu chức năng mô tả hệ thống phải làm gì (what) nhưng không mô tả hệ thống phải làm thế nào (how)
- ❖ Yêu cầu phi chức năng: ràng buộc về thời gian, các hỗ trợ ngôn ngữ, ...
- ❖ Thu thập và phân tích yêu cầu là công việc rất khó khăn
 - Các yêu cầu không hoàn chỉnh
 - Các thuật ngữ trong nhiều lĩnh vực gây khó hiểu
 - Các yêu cầu khách hàng thường thiếu chính xác, thừa, không có cấu trúc và thiếu nhất quán
 - Có cả các yêu cầu thiếu tính khả thi.

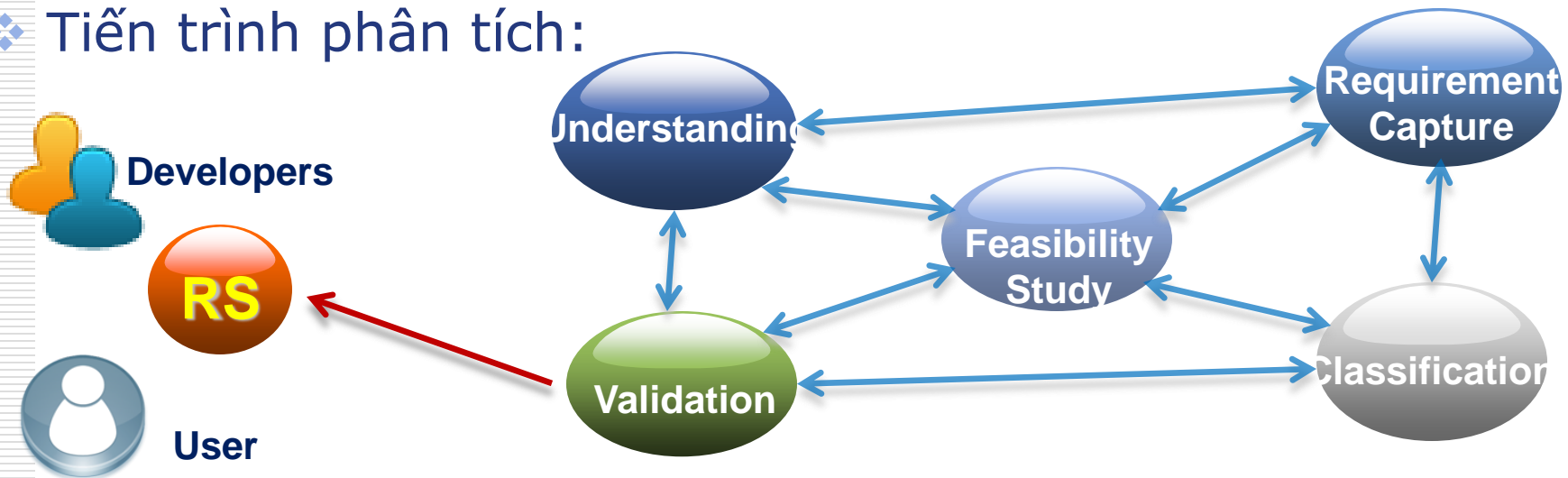
Requirement Management



Quản lý yêu cầu là một công việc

- Quan trọng
- Không dễ dàng
- Ảnh hưởng đến các giai đoạn khác
- Quyết định việc thành bại của PM

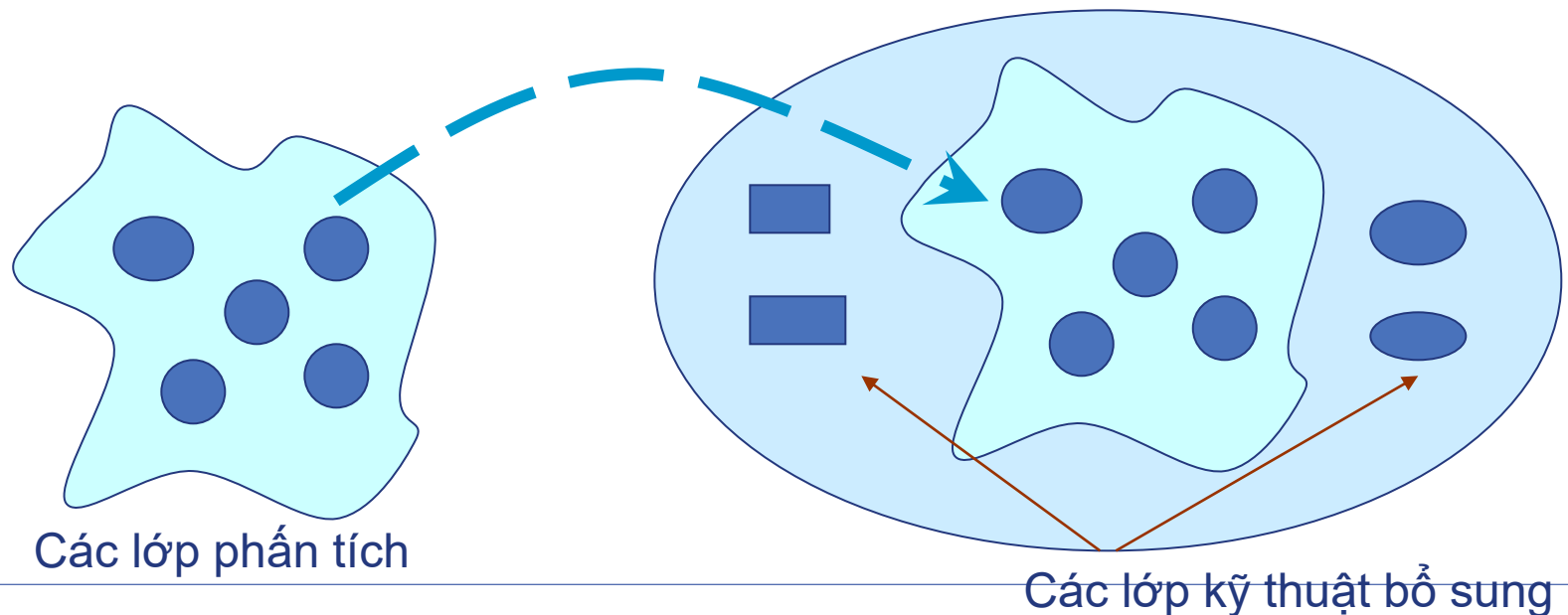
- ❖ Mục tiêu: Requirement Specification (RS)
- ❖ Tài liệu đặc tả:
 - Cam kết giữa khách hàng và tổ chức phát triển hệ thống
 - Nguồn nhân lực để phát triển hệ thống
 - Mô hình tương đối hoàn chỉnh về những yêu cầu hệ thống
- ❖ Tiến trình phân tích:



Thiết kế (Design)

- ❖ Xác định chiến lược thiết kế
 - ❖ Xây dựng cấu trúc thành phần
 - ❖ Thiết kế giao diện
 - ❖ Thiết kế chức năng
 - ❖ Thiết kế cơ sở dữ liệu
- ➔ Hồ sơ thiết kế cho cả hệ thống

- ❖ Chi tiết hóa các lớp gđ phân tích thành một giải pháp kỹ thuật thực hiện → bổ sung các lớp kỹ thuật tạo ra một kiến trúc kỹ thuật thực hiện



Thiết kế

❖ Thiết kế luận lý

- Đặc tả hệ thống ở mức độ trừu tượng hóa dựa trên giải pháp được chọn
- Độc lập với phần cứng, phần mềm và môi trường kỹ thuật
- Tập trung vào biểu diễn hành vi và tính năng đối tượng hệ thống

❖ Thiết kế vật lý: chuyển đổi thiết kế luận lý sang các đặc tả phần cứng, phần mềm, kỹ thuật được chọn để cài đặt hệ thống

Thiết kế (Design)

- ❖ Tài liệu thiết kế: mô hình kiến trúc
 - Đặt tả thành phần, mô tả các thành phần phải làm gì thông qua giao diện. Chủ yếu mô tả “what”
- ❖ Thiết kế chi tiết (thực hiện nhiều bước)
- ❖ Mô hình thiết kế chi tiết:
 - Thiết kế chức năng cho mỗi thành phần
 - Thiết kế giao diện cho mỗi thành phần
- ❖ Mô hình hệ thống cốt lõi
 - Cụ thể - phụ thuộc cài đặt – xác định làm thế nào “how”

- ❖ Mỗi thành phần trong giai đoạn thiết kế được thực hiện thành một module của hệ thống
- ❖ Kiểm chứng và thử nghiệm mỗi module theo đặt tả từ thiết kế
- ❖ Lập lại qui trình trên cho đến khi từng module hoàn thiện
- ❖ Tổ hợp các module hệ thống
- ❖ Kiểm tra toàn hệ thống để đảm bảo đáp ứng đúng và đủ yêu cầu đồng thời không phát sinh lỗi.
- ❖ Yêu cầu khách hàng thử nghiệm hệ thống
- ❖ Khách hàng chấp nhận sản phẩm → Tiến trình này đã hoàn tất

- ❖ Lên kế hoạch triển khai hệ thống
- ❖ Các báo cáo và tài liệu đính kèm
- ❖ Có kế hoạch tập huấn sử dụng cho khách hàng hay không
- ❖ Kế hoạch và thời gian bảo hành hệ thống
- ❖ Hỗ trợ kèm theo hệ thống?
- ❖ Nhận phản hồi của đối tác ra sao
- ❖ Cài đặt hệ thống

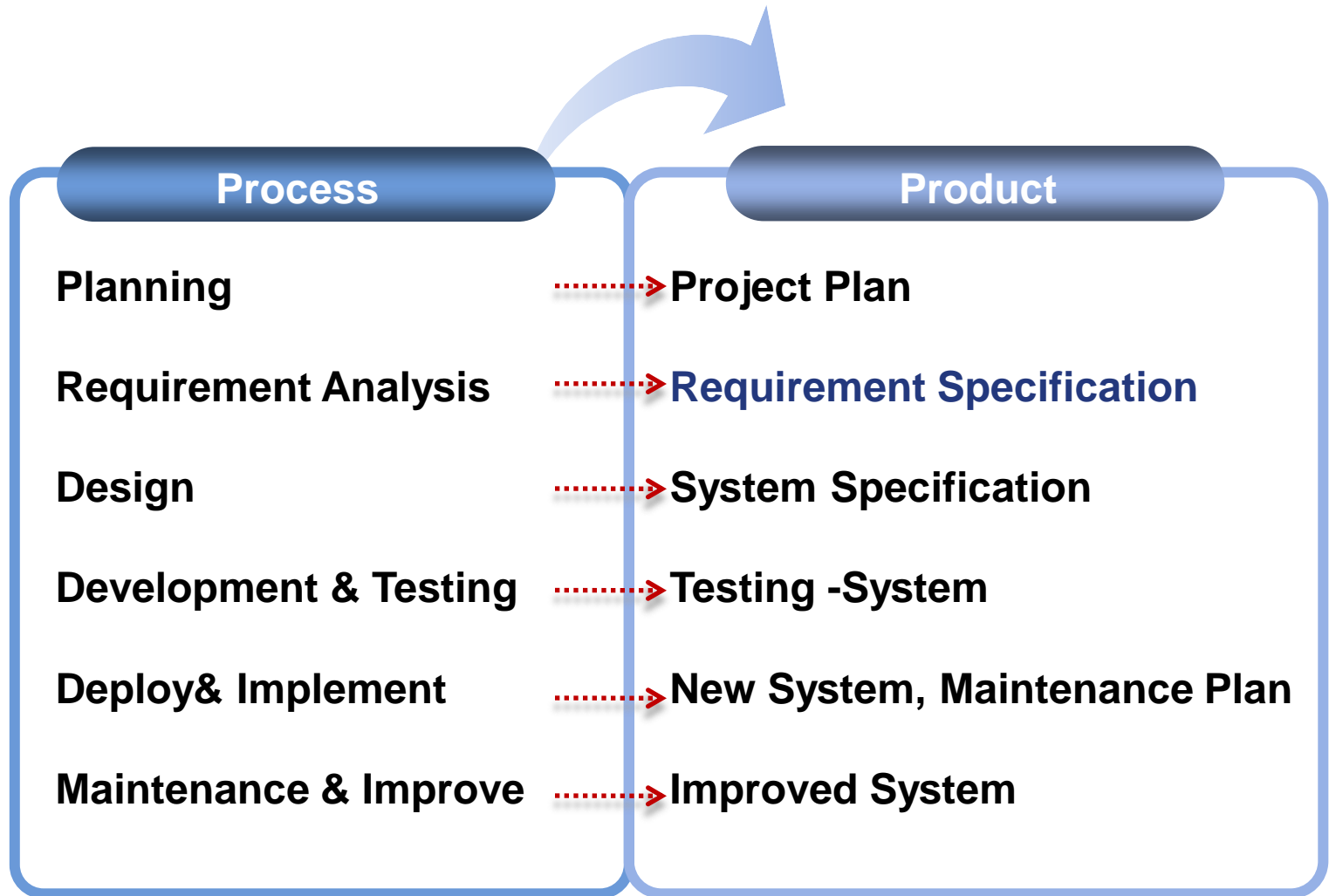
❖ Bảo trì:

- Sau khi hệ thống được cài đặt và sử dụng thực tế
- Bảo vệ và duy trì cho hệ thống hoạt động
- Sửa các lỗi phát sinh (lỗi hệ thống hay lỗi do người sử dụng)
- Tính chi phí bảo trì bảo dưỡng hệ thống

❖ Nâng cấp: gồm

- Bổ sung chức năng
- Thay đổi mới do nâng cấp phần cứng hay các phần mềm khác có liên quan

Các giai đoạn và kết quả



Software development Methodologies

- ❖ 1970s [Structured programming](#)
- ❖ 1980s [Structured Systems Analysis and Design Methodology \(SSADM\)](#)
- ❖ 1990s [Object-oriented programming \(OOP\)](#) phát triển từ những năm đầu 1960s, và phát triển như phương pháp lập trình suốt giữa thập niên 1990s.
- ❖ [Rapid application development \(RAD\)](#) từ 1991.
- ❖ [Scrum](#) (development), từ cuối 1990s
- ❖ [Team software process](#) bởi Watts Humphrey tại [SEI](#)
- ❖ 2000s [Extreme Programming](#)
- ❖ [Rational Unified Process \(RUP\)](#) từ 1998.
- ❖ [Agile Unified Process \(AUP\)](#) từ 2005 bởi [Scott Ambler](#)
- ❖ [Integrated Methodology](#) từ 2007

Software development approaches

- ❖ Thiết kế cấu trúc (Structured design)
 - Phương pháp thác nước (**waterfall method**)
 - Phương pháp phát triển song song (**Parallel development**)
- ❖ Phương pháp dạng Spiral
- ❖ 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)
- ❖ Phương pháp phát triển linh động (Agile development)
 - XP (**extreme programming**)

Software development approaches

- ❖ Object-Oriented Analysis and Design - OOAD
 - Use-case Diagram
 - Class Diagram
 - Object Diagram
 - State Diagram
 - Transition Diagram
 - Interaction Diagram
 - Module Diagram
 - Process Diagram
- ❖ Integrated Methodology Software Development

Structured Design

- ❖ 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

Waterfall method

Waterfall method

Planning

Analysis

Design

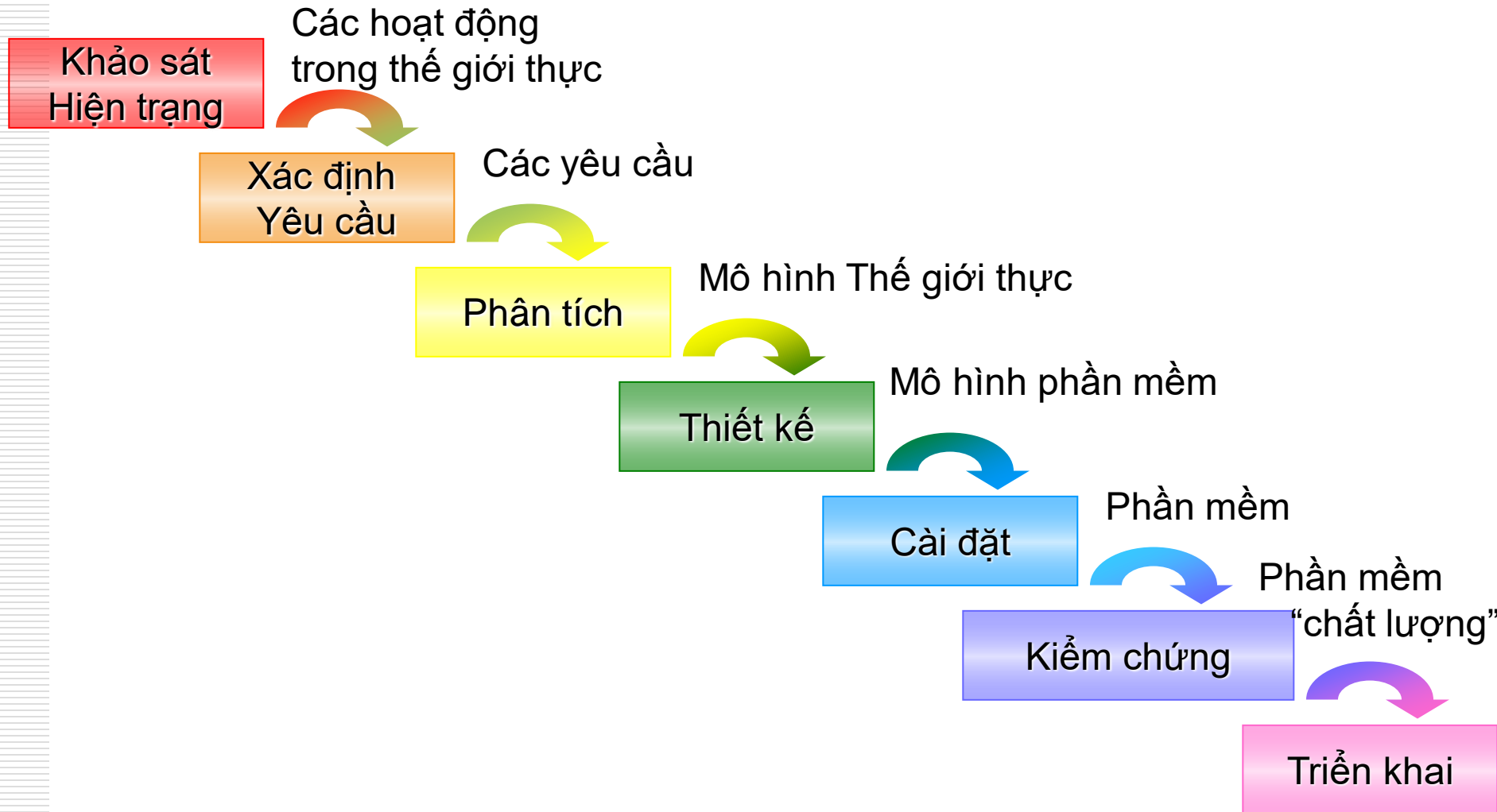
Implementation

System

Quy trình thác nước nhiều rủi ro

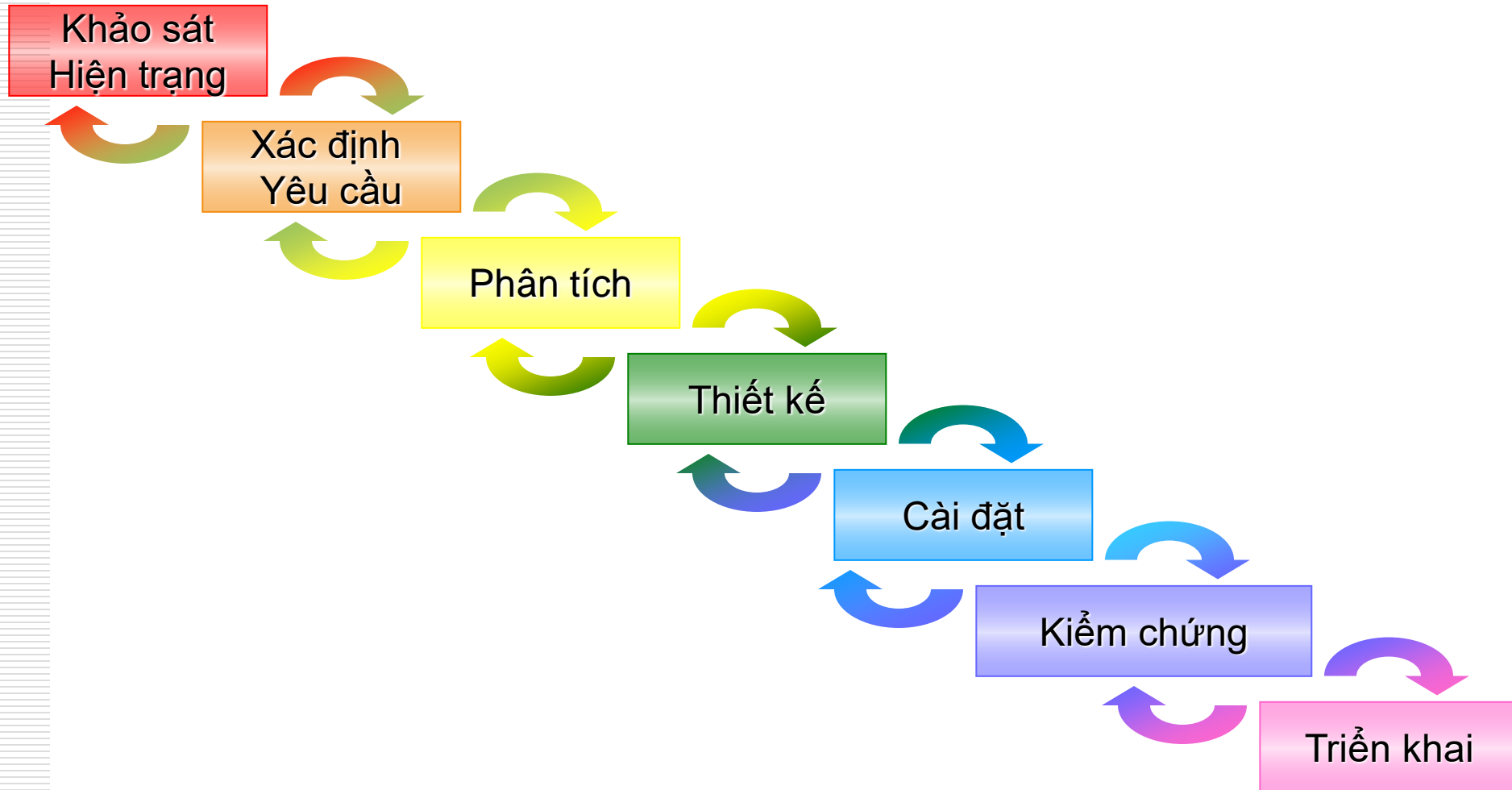
Time

Quy trình thác nước

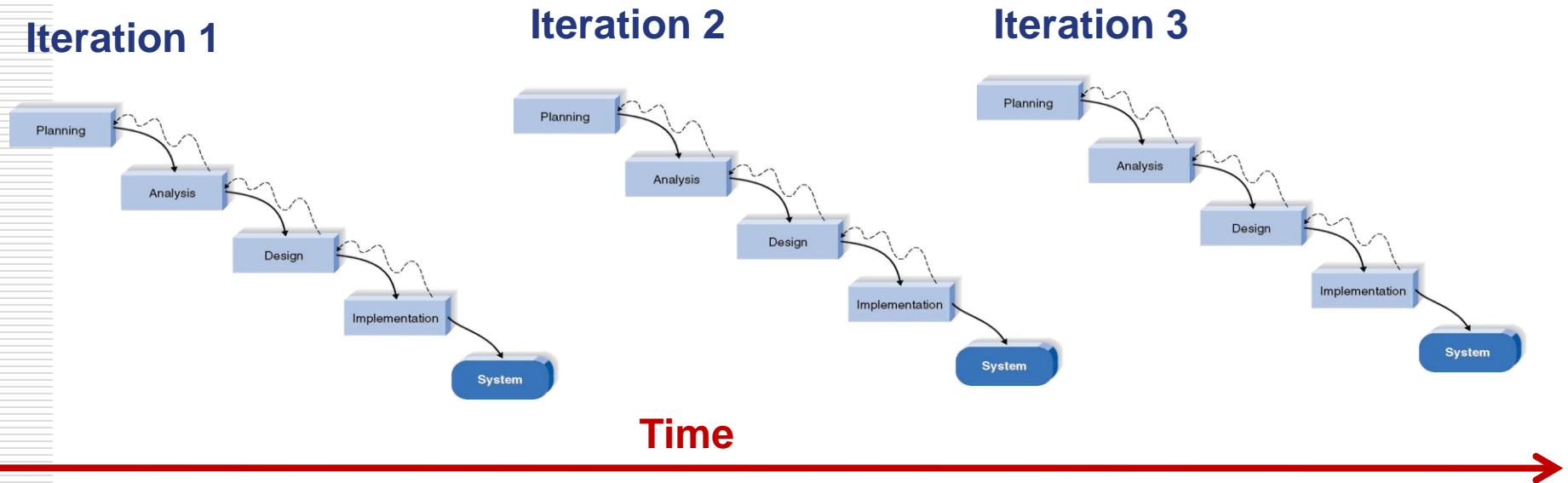


Waterfall

Quy trình thác nước cải tiến

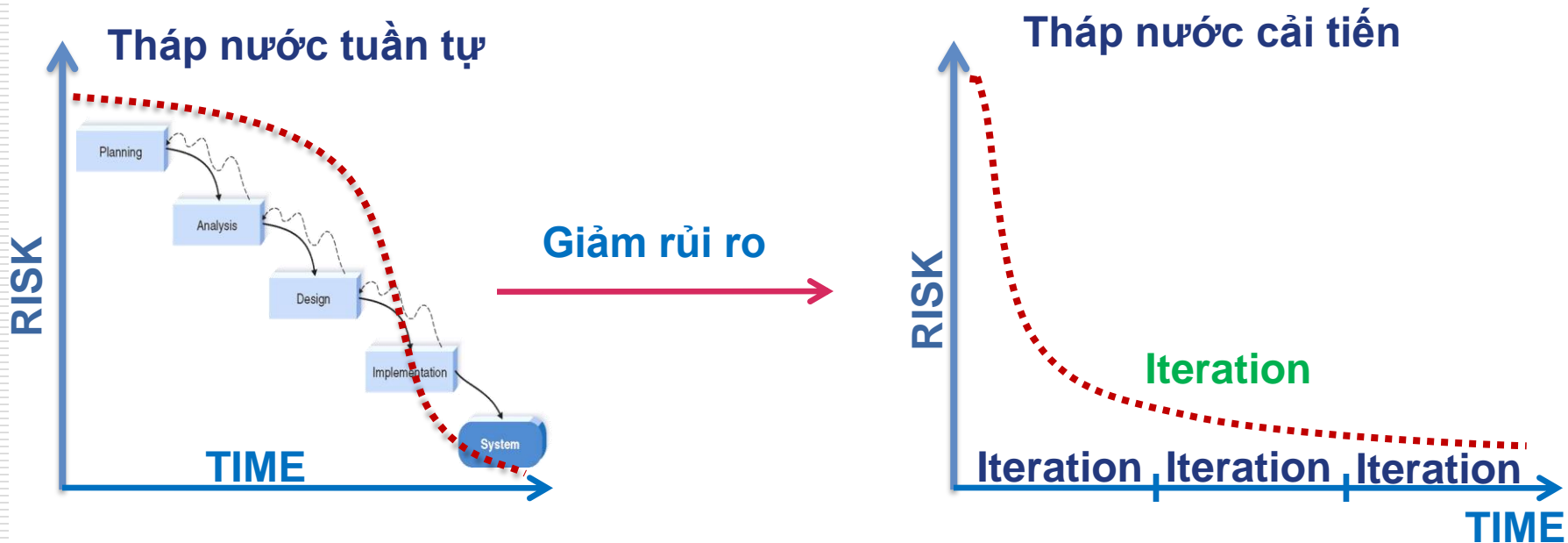


Ứng dụng thác nước theo vòng lặp



- ❖ Các vòng lặp đầu dành cho các vấn đề nhiều rủi ro
- ❖ Mỗi vòng lặp sinh ra một phiên bản với một sự bổ sung cho hệ thống
- ❖ Mỗi vòng lặp bao gồm cả việc tích hợp và kiểm chứng

Qui trình lặp



- ❖ Các rủi ro chính được giải quyết trước khi có các phát triển lớn
- ❖ Các vòng lặp đầu tiên cho phép nhận feedback
- ❖ Việc kiểm chứng và tích hợp diễn ra liên tục
- ❖ Các cột mốc cục bộ sẽ định ra các tiêu điểm ngắn hạn
- ❖ Sự tiến triển được đo bằng bản cài đặt
- ❖ Các cài đặt bộ phận có thể triển khai riêng

Phương pháp thác nước

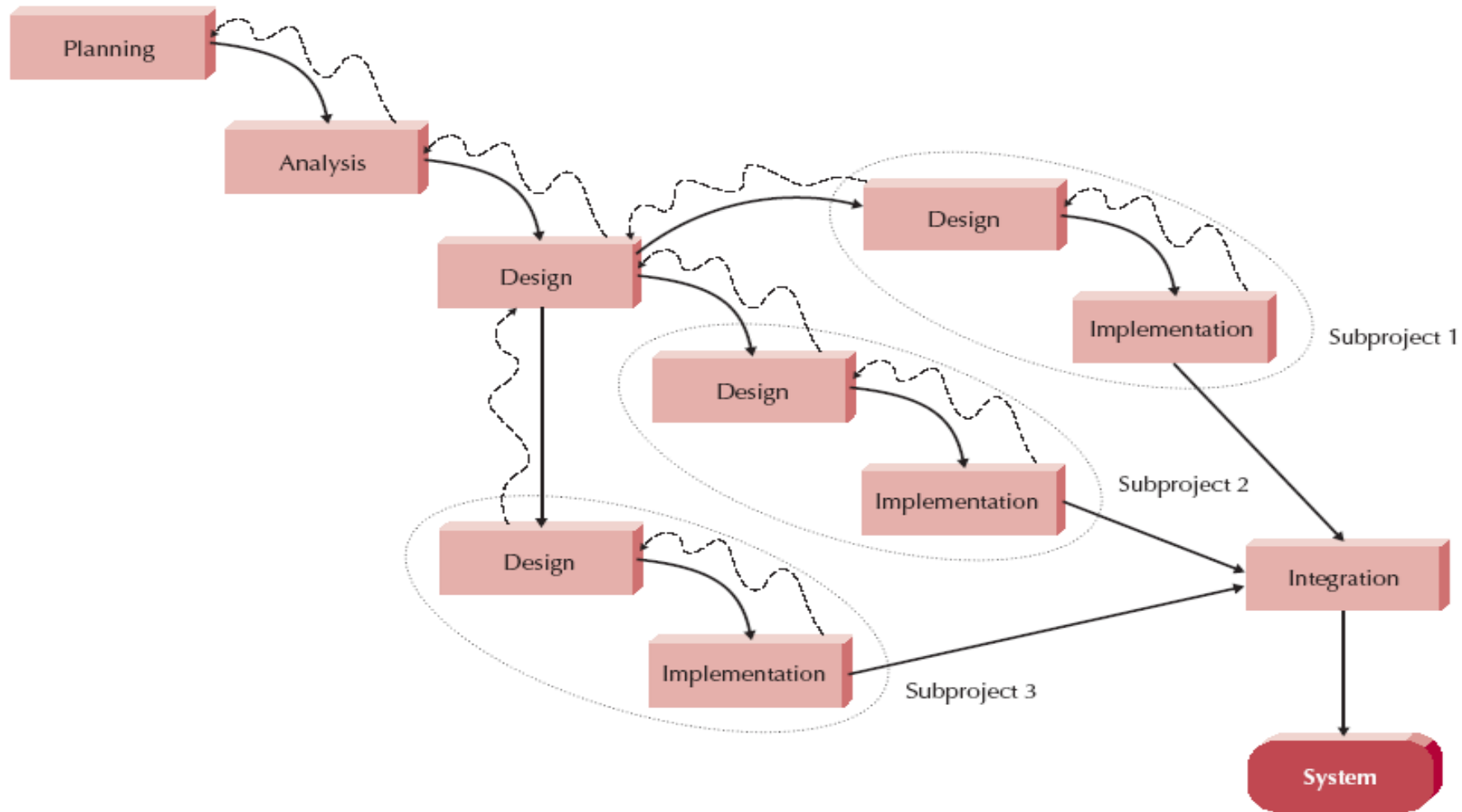
❖ Ưu điểm:

- 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:

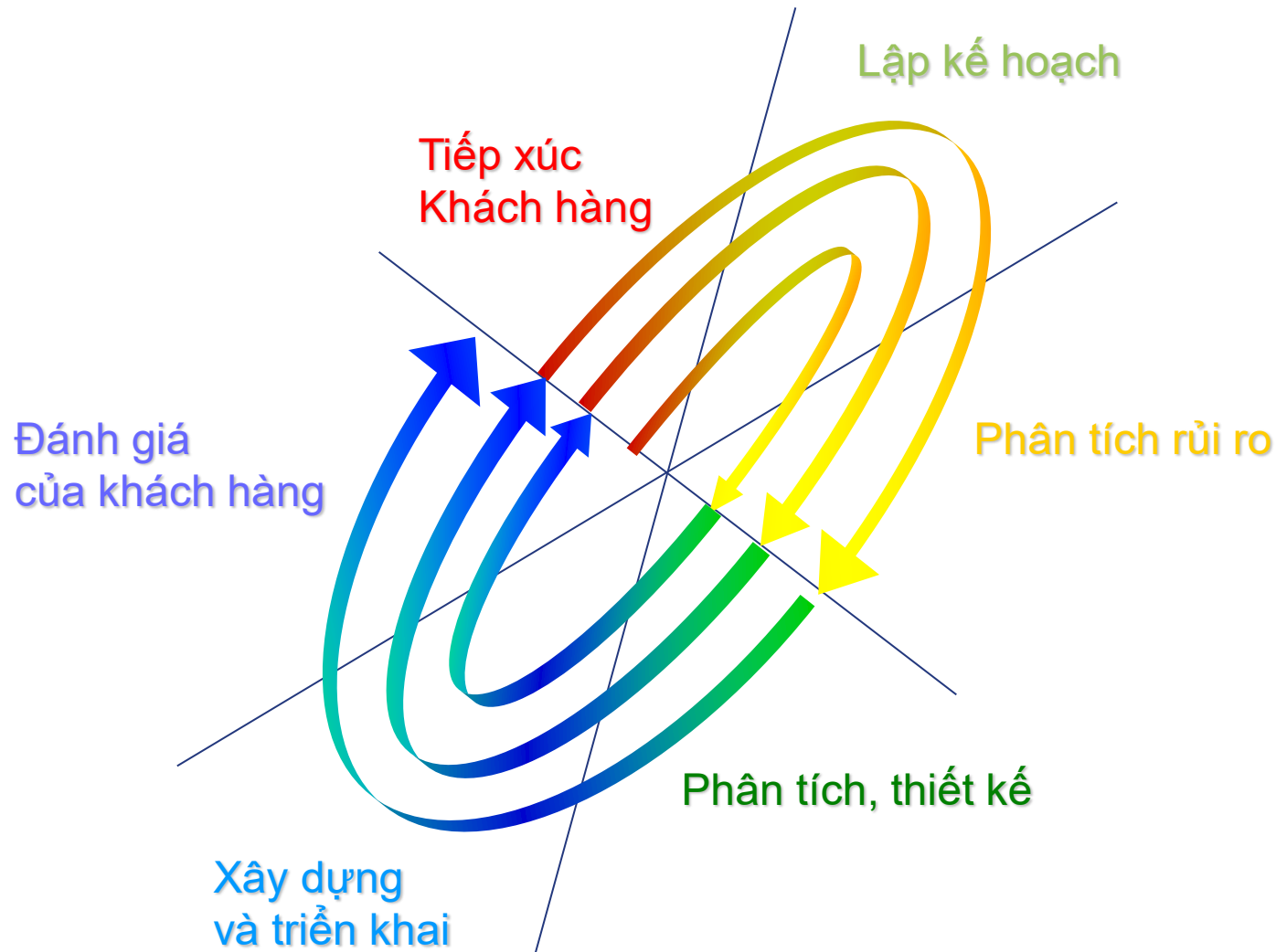
- Thời gian từ khi đề xuất dự án đến khi có sản phẩm cuối cùng thường rất dài (vài tháng -> vài năm)

Parallel Development

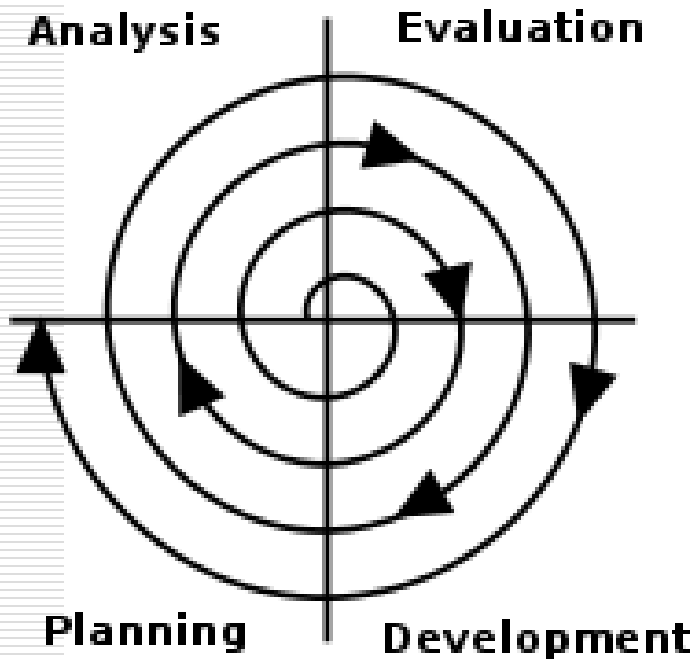


A Parallel Development-based Methodology

Quy trình xoắn ốc



Spiral



- ❖ Kết hợp giữa prototyping-in-stages và top-down and bottom-up
- ❖ Đánh giá và giảm thiểu rủi ro bằng cách tách nhỏ các phân đoạn thay vì quản lý trên toàn bộ dự án
- ❖ Mỗi vòng xoắn ốc đi qua:
 - Phân tích mục tiêu, lựa chọn thay thế.
 - Đánh giá các lựa chọn thay thế, xác định và giải quyết các rủi ro.
 - Phát triển từ các iteration
 - Kế hoạch được lập tiếp theo

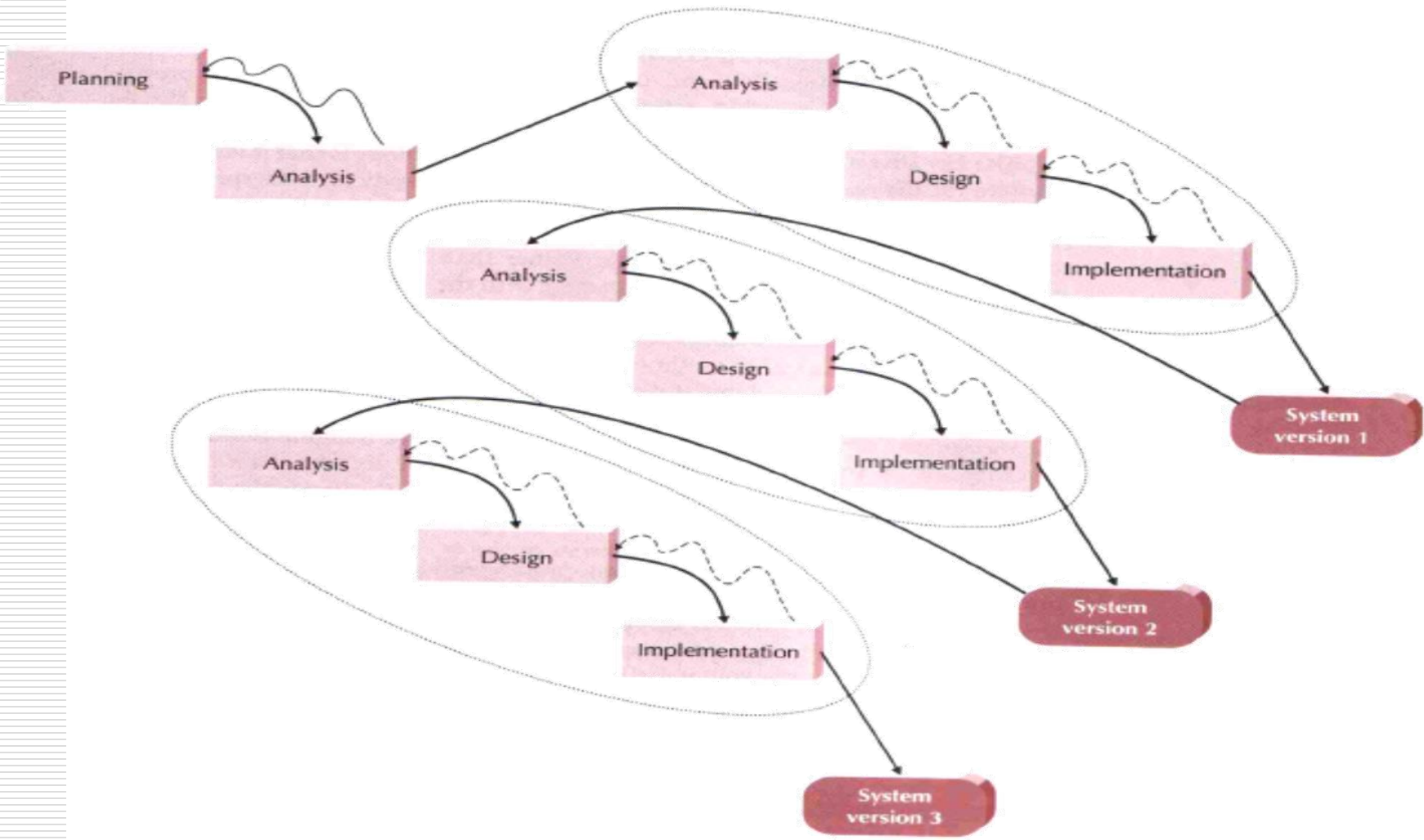
- ❖ Bắt đầu mỗi chu kỳ với: các bên liên quan xác định mục đích và kết quả cần thoả
- ❖ Kết thúc mỗi chu kỳ: xem xét và cam kết của các bên

RAD

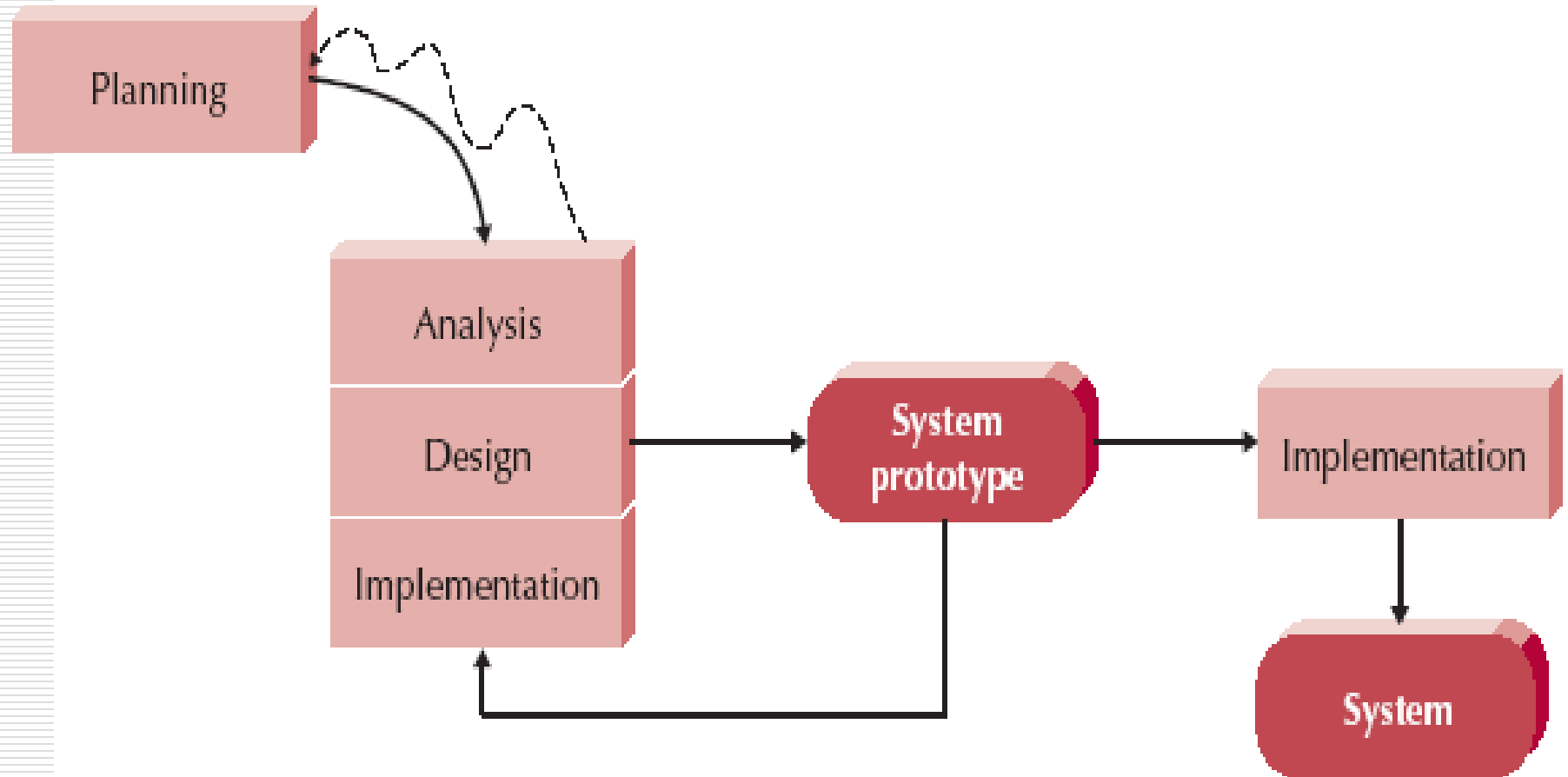
❖ Các nhân tố quan trọng:

- Công cụ CASE (Computer Aided Software Engineering)
- JAD (Joint Application Development)
- Ngôn ngữ lập trình thể hệ thứ tư/ visual
- Công cụ tạo mã

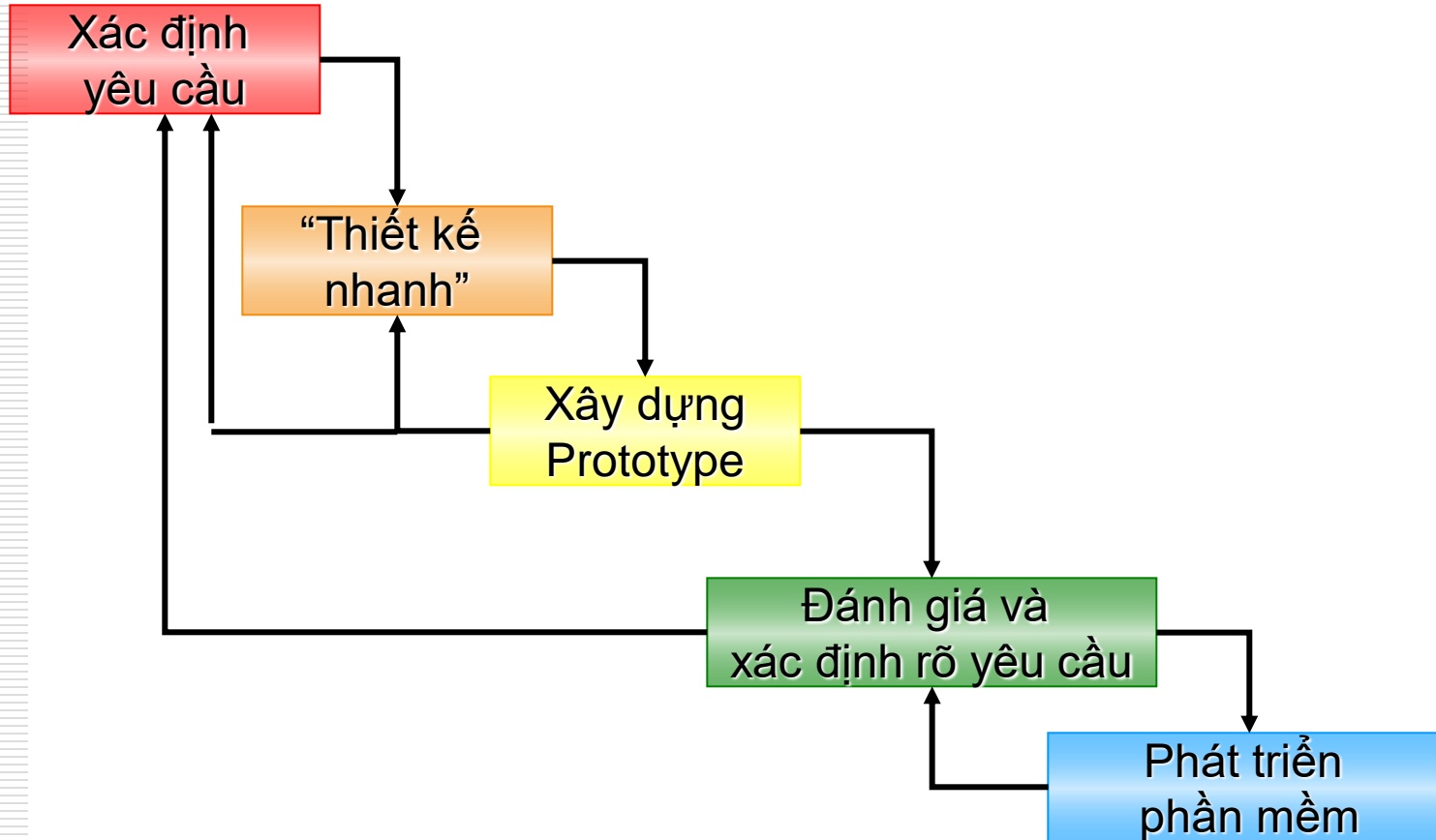
Phương pháp phát triển theo pha



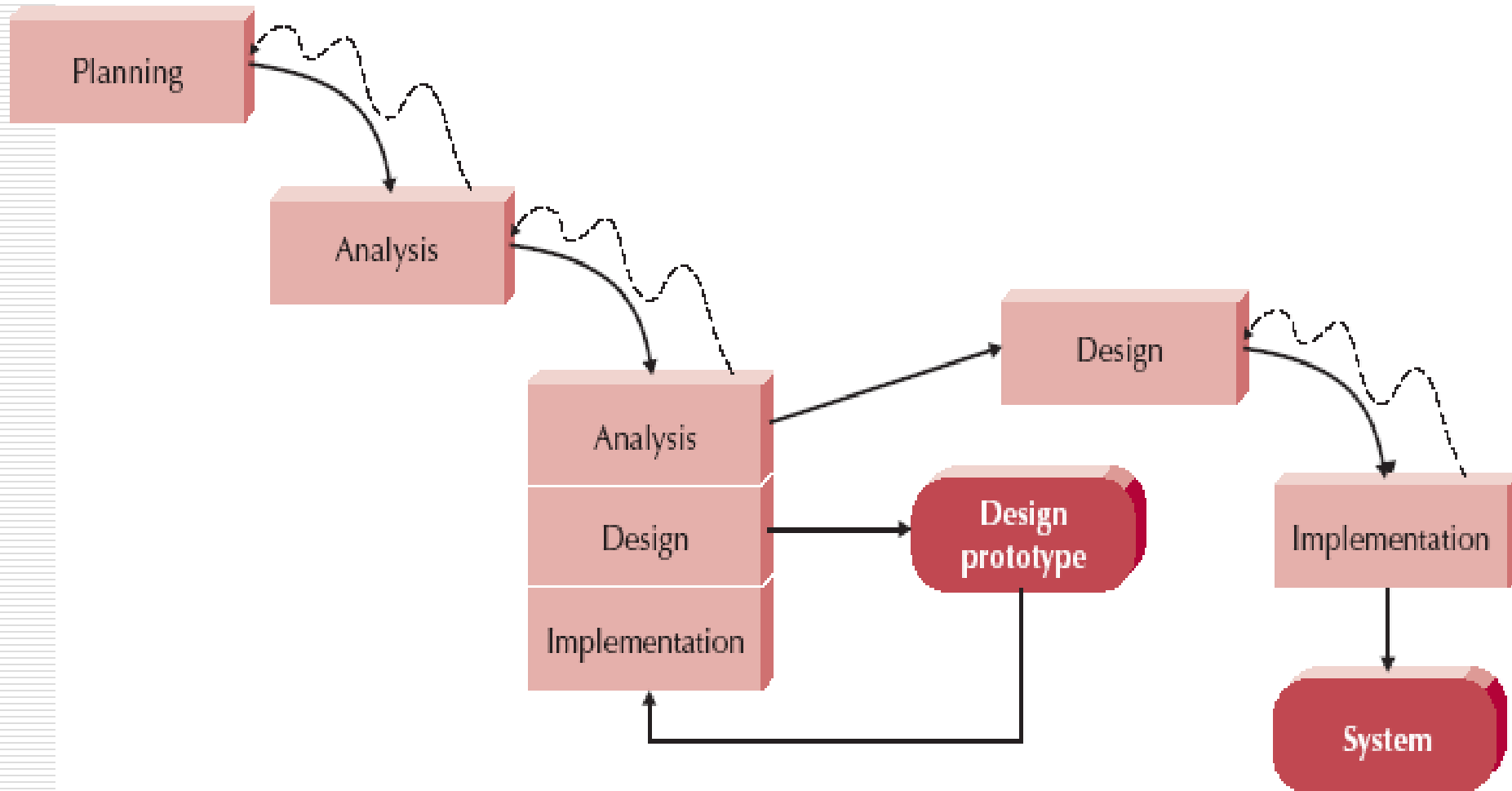
Software prototyping



Quy trình Prototype



Phương pháp xây dựng nguyên mẫu loại bỏ



Lựa chọn phương pháp phù hợp

❖ Tiêu chí:

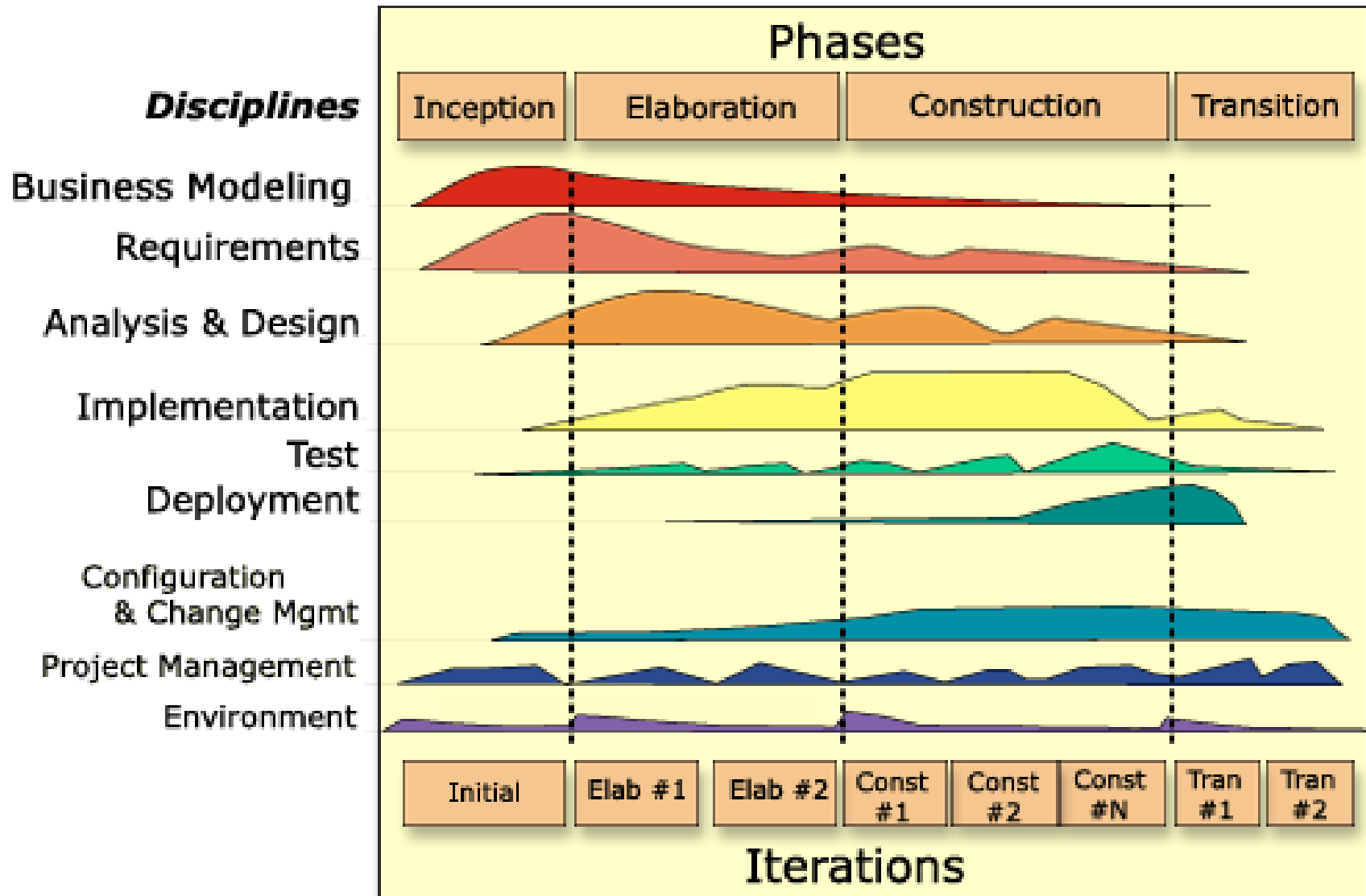
- Độ rõ ràng, đầy đủ của các yêu cầu của người sử dụng
- Khả năng, mức độ thành thạo về công nghệ
- Độ phức tạp của hệ thống
- Độ tin cậy của hệ thống
- Quỹ thời gian

Lựa chọn phương pháp phù hợp

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

Criteria for Selectiong a Methodology

Rational Unified Process - RUP



Ví dụ

Yêu cầu về căn nhà



Ý tưởng về căn nhà



Phân tích & thiết kế



Xây dựng



Căn nhà xây xong



Chuyển giao sử dụng và bảo trì

Câu hỏi