

Nhập môn Công nghệ phần mềm

Tuần 2-3: Quy trình phần mềm



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Nội dung của slide được dịch và phát triển dựa vào bộ slide của Ian Sommerville

Nội dung



Mô hình quy trình phần mềm

Các hoạt động của quy trình

Thích nghi với sự thay đổi

Quy trình RUP

Nội dung



Mô hình quy trình phần mềm

Các hoạt động của quy trình

Thích nghi với sự thay đổi

Quy trình RUP



Quy trình phần mềm

- Quy trình phần mềm (software process) là một tập có cấu trúc các hoạt động cần thiết để phát triển một hệ thống phần mềm.
- Có nhiều quy trình phần mềm khác nhau. Tất cả đều bao gồm những hoạt động:
 - ▣ **Đặc tả:** Định nghĩa hệ thống làm gì;
 - ▣ **Thiết kế và cài đặt:** Định nghĩa tổ chức của hệ thống và cài đặt hệ thống;
 - ▣ **Kiểm định:** Kiểm tra rằng hệ thống đáp ứng được mong muốn của người dùng;
 - ▣ **Cải tiến:** thay đổi hệ thống để đáp ứng sự thay đổi yêu cầu người dùng.
- Mô hình quy trình phần mềm (software process model) là biểu diễn trừu tượng của một quy trình. Nó biểu diễn mô tả của quy trình từ một góc nhìn nào đó.



Mô tả quy trình phần mềm

- Khi mô tả về quy trình, ta thường nói về
 - ▣ các hoạt động trong những quy trình này. Ví dụ, đặc tả mô hình dữ liệu, thiết kế giao diện người dùng, ... ;
 - ▣ và thứ tự của các hoạt động này.
- Các mô tả quy trình có thể gồm:
 - ▣ Sản phẩm: kết quả đầu ra của một hoạt động;
 - ▣ Vai trò: phản ánh trách nhiệm của những người tham gia vào quy trình;
 - ▣ Điều kiện trước và điều kiện sau (Pre- and post-conditions): là những điều kiện phải đảm bảo trước và sau khi một hoạt động được thực hiện hay một sản phẩm được tạo ra.



Quy trình hoạch định sẵn và quy trình linh hoạt

- ☐ Các quy trình hoạch định sẵn (plan-driven process) là các quy trình mà trong đó tất cả các hoạt động được lên kế hoạch trước và tiến độ thực hiện được đánh giá dựa vào kế hoạch này.
- ☐ Trong các quy trình linh hoạt (agile process), kế hoạch được phát triển dần dần và dễ dàng thay đổi quy trình để đáp ứng sự thay đổi yêu cầu của khách hàng.
- ☐ Hầu hết các quy trình thực tế đều gồm những phần tử của cả hai phương pháp này.
- ☐ Không có quy trình phần mềm đúng hay sai!



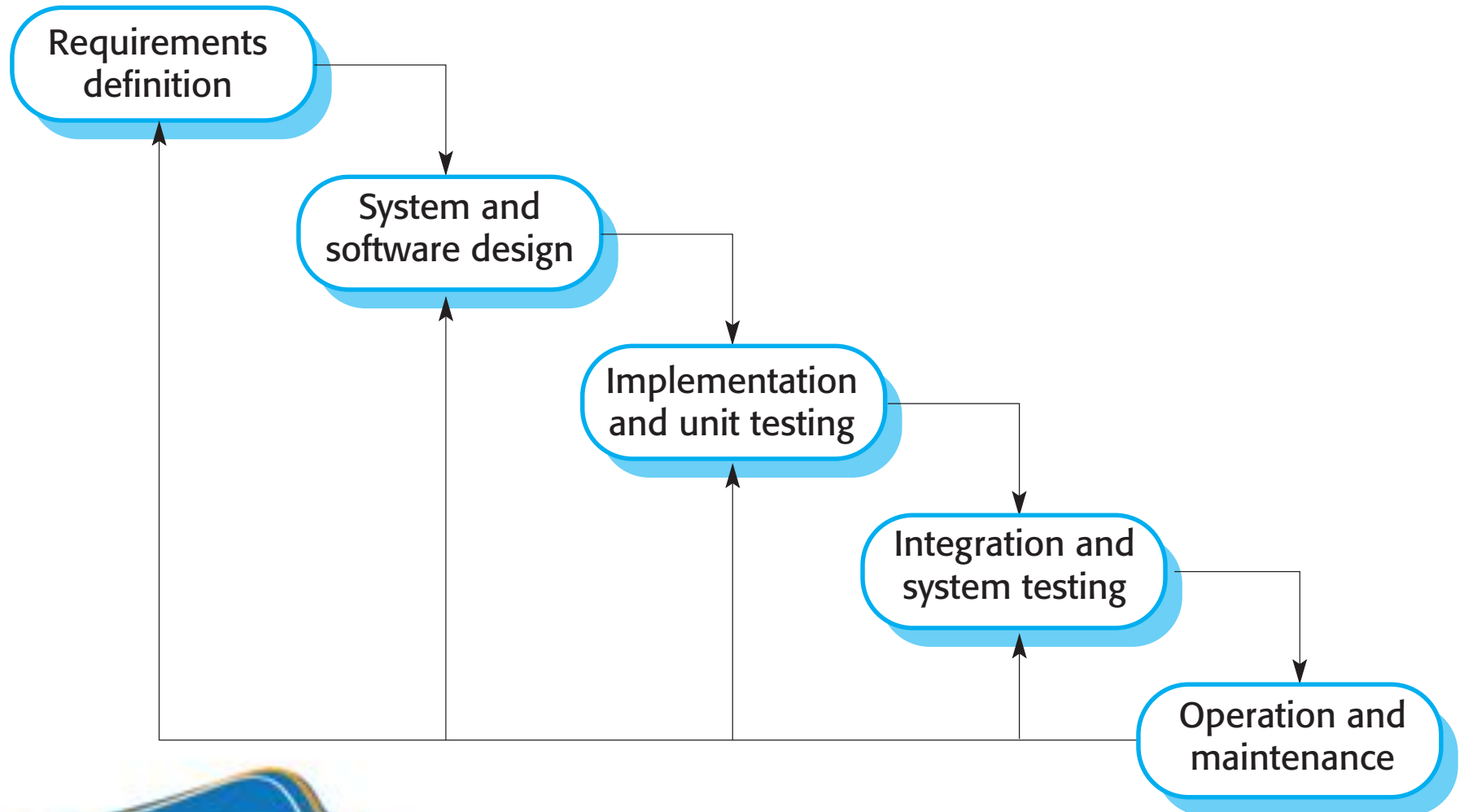
Các mô hình quy trình phần mềm

- Mô hình thác nước (waterfall model)
 - ▣ Mô hình hoạch định sẵn. Các pha đặc tả và phát triển phân biệt và tách rời nhau.
- Mô hình phát triển dần dần (incremental development)
 - ▣ Các pha đặc tả, phát triển và thẩm định đan xen nhau. Có thể là mô hình hoạch định sẵn, có thể là mô hình linh hoạt.
- CNPM theo hướng tái sử dụng (reuse-oriented software engineering)
 - ▣ Hệ thống được xây dựng từ những component có sẵn. Có thể là hoạch định sẵn, có thể là linh hoạt.
- Thực tế, những hệ thống lớn được phát triển bằng cách sử dụng quy trình tạo ra bằng cách kết hợp các phần tử từ các mô hình này.

Mô hình thác nước



Mô hình thác nước





Ưu điểm

- Quy trình rõ ràng → người quản lý dễ dàng theo dõi tiến độ công việc.
- Mô hình này được sử dụng trong các hệ thống lớn trong đó hệ thống được phát triển tại nhiều địa điểm khác nhau
 - ▣ Vì là quy trình hoạch định sẵn → giúp cho việc phối hợp trong công việc dễ dàng hơn.



Nhược điểm

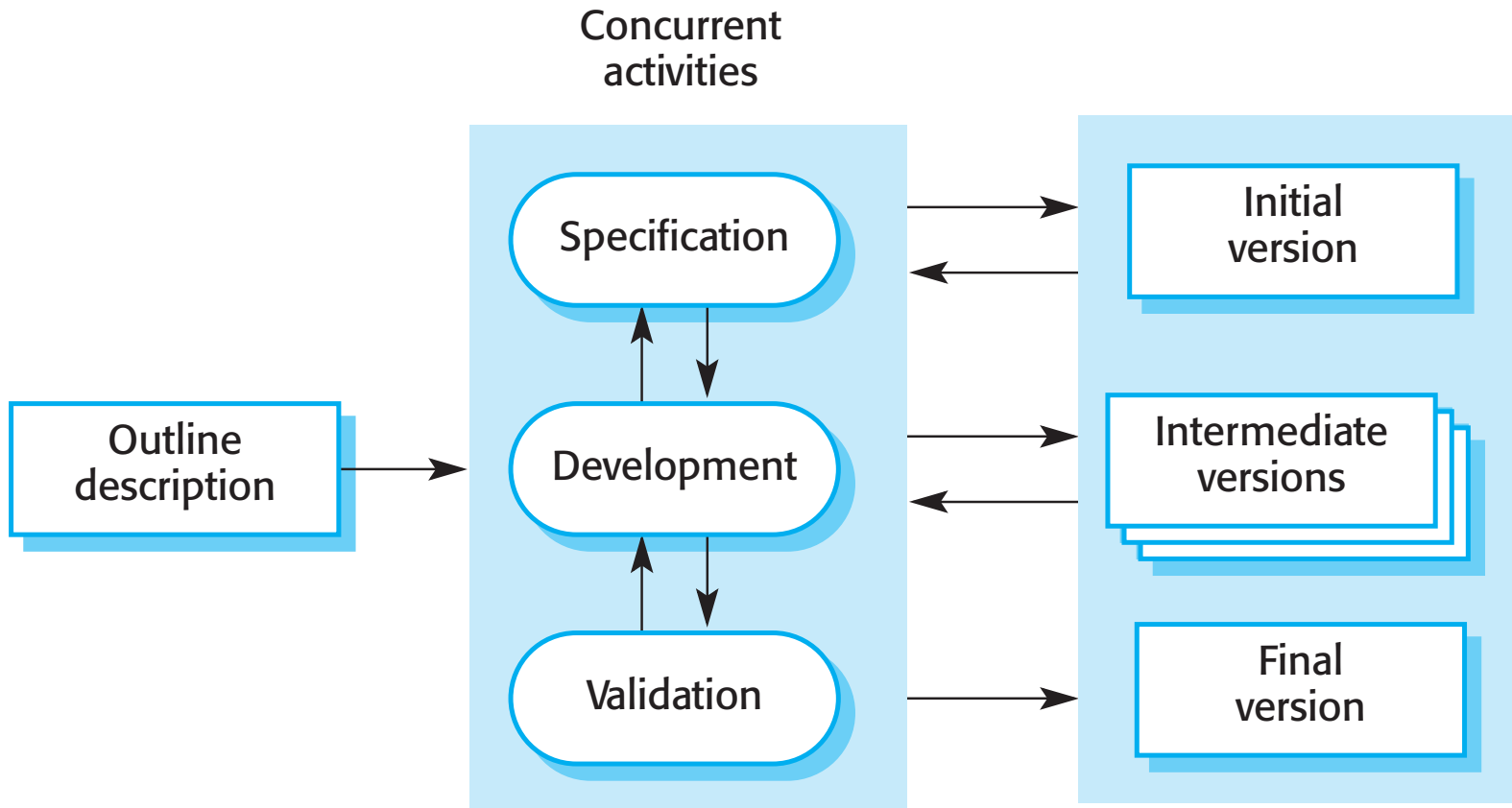
- Khó khăn trong việc thích nghi với sự thay đổi sau khi quy trình đã vào guồng.
 - ▣ Về nguyên tắc, pha này phải hoàn thành trước khi bắt đầu pha tiếp theo.
- Không linh động trong việc phân chia dự án thành những giai đoạn tách biệt → khó khăn trong việc đáp ứng sự thay đổi yêu cầu người dùng
 - ▣ Mô hình này chỉ hợp lý khi yêu cầu được hiểu rõ và ít thay đổi trong suốt quá trình phát triển;
 - ▣ Hệ thống thương mại thường có yêu cầu không ổn định → mô hình thác nước không phù hợp.



Biến thể của mô hình thác nước

- Phát triển hệ thống theo kiểu hình thức
 - ▣ Sử dụng mô hình toán học để đặc tả hệ thống.
 - ▣ Sử dụng các chuyển đổi toán học để chuyển các đặc tả thành chương trình chạy được → lý do thuyết phục để đảm bảo rằng một chương trình được phát sinh nhất quán với đặc tả đưa ra.
- Những quy trình phát triển hình thức (sử dụng B-method chẳng hạn) phù hợp với việc phát triển hệ thống có độ tin cậy cao, độ an toàn cao và tính bảo mật cao.
 - ▣ Ví dụ: đường metro 14 ở Paris, hệ thống tàu tự động ở sân bay CDG, Paris, etc.

Phát triển dần dần





Đặc điểm

- ☐ Dựa vào ý tưởng của việc phát triển phiên bản đầu tiên, lấy ý kiến khách hàng và cải tiến nó cho đến khi sản phẩm hoàn thiện.
- ☐ Các pha đặc tả, phát triển và thẩm định đan xen nhau.
- ☐ Lấy phản hồi nhanh từ khách hàng



Ưu điểm

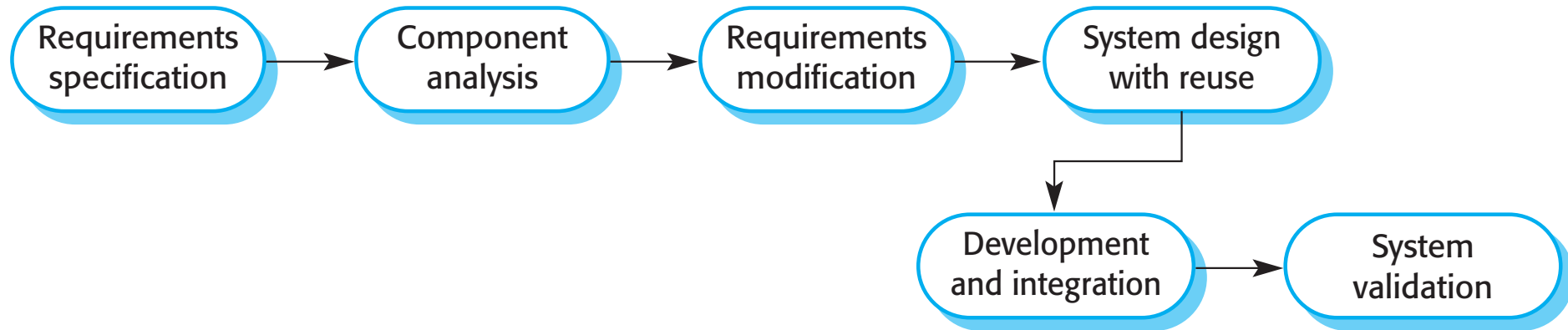
- ☐ Giảm được chi phí khi đáp ứng sự thay đổi yêu cầu của khách hàng
- ☐ Dễ dàng trong việc lấy phản hồi từ khách hàng.
- ☐ Phân phối và triển khai phần mềm đến khách hàng nhanh hơn.

Nhược điểm

- ☐ Quy trình không rõ ràng.
- ☐ Cấu trúc hệ thống có xu hướng bị giảm đi vì những phần mới của hệ thống được thêm vào.



CNPM theo hướng tái sử dụng





Đặc điểm

- Dựa vào việc tái sử dụng một cách có hệ thống
 - ▣ Hệ thống được tích hợp từ những thành phần có sẵn hoặc từ các hệ thống COTS (Commercial-off-the-shelf).
- Các giai đoạn của quy trình
 - ▣ Phân tích component;
 - ▣ Bổ sung yêu cầu;
 - ▣ Thiết kế hệ thống với việc tái sử dụng;
 - ▣ Phát triển và tích hợp.
- Hiện nay, việc tái sử dụng là phương pháp chuẩn cho việc xây dựng nhiều hệ thống thương mại.



Các loại component

- Các dịch vụ Web (Web service): được phát triển theo các chuẩn dịch vụ và có sẵn để triệu gọi từ xa.
- Các tập các đối tượng được phát triển như một gói (package) tích hợp trong các framework như .NET hay J2EE.
- Những hệ thống phần mềm độc lập (COTS): được cấu hình để sử dụng cho một môi trường cụ thể.

Nội dung



Mô hình quy trình phần mềm

Các hoạt động của quy trình

Thích nghi với sự thay đổi

Quy trình RUP



Các hoạt động của quy trình

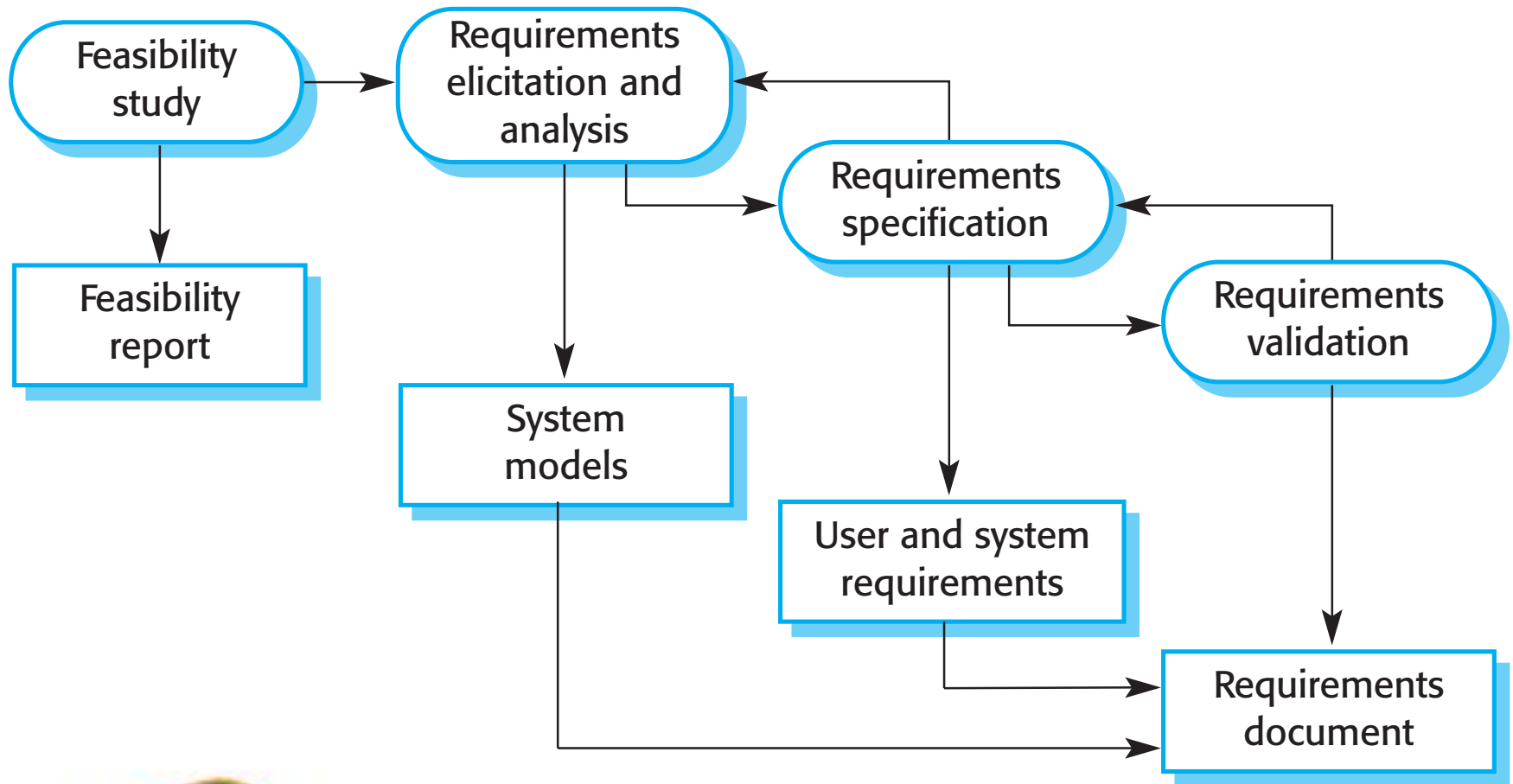
- 4 hoạt động quy trình cơ bản
 1. Đặc tả,
 2. Phát triển (thiết kế và cài đặt),
 3. Thẩm định và
 4. Cải tiến
- Các hoạt động này được tổ chức khác nhau trong các quy trình phát triển khác nhau
 - ▣ Trong mô hình thác nước: tổ chức tuần tự.
 - ▣ Trong mô hình phát triển dần dần: tổ chức đan xen.



1. Đặc tả phần mềm

- Là quy trình thiết lập danh sách các dịch vụ được yêu cầu và các ràng buộc đối với hoạt động của hệ thống và việc phát triển hệ thống.
- Quy trình công nghệ yêu cầu (requirements engineering process)
 - ▣ Nghiên cứu khả thi (Feasibility study)
 - ▣ Thu thập và phân tích yêu cầu (Requirements elicitation and analysis)
 - ▣ Đặc tả yêu cầu (Requirements specification)
 - ▣ Thẩm định yêu cầu (Requirements validation)

Quy trình công nghệ yêu cầu



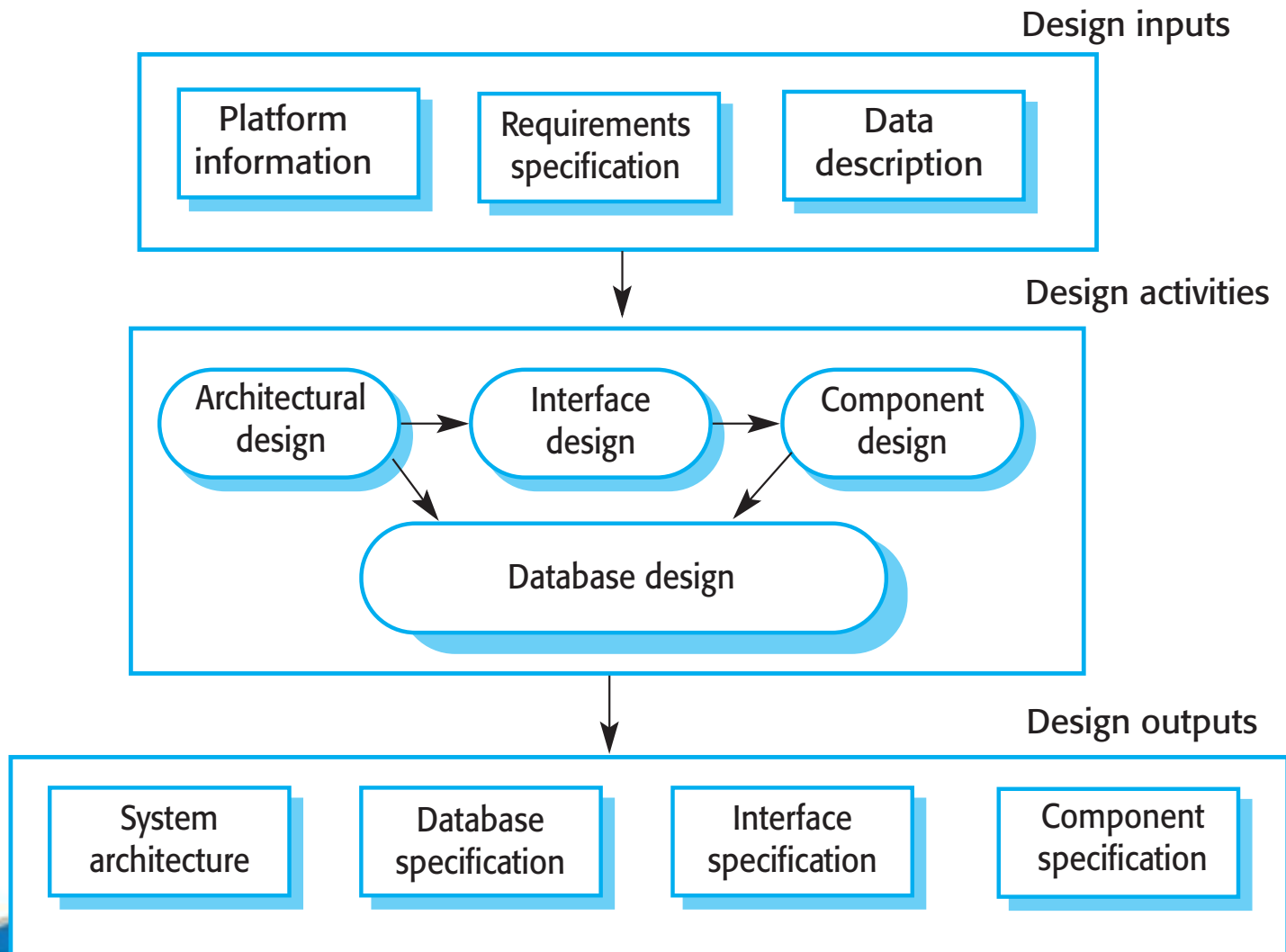


2. Thiết kế và cài đặt phần mềm

- ☐ Là quy trình chuyển đổi các đặc tả thành hệ thống thực thi được.
- ☐ Thiết kế phần mềm (software design)
 - ☐ Thiết kế một cấu trúc phần mềm để hiện thực hóa đặc tả;
- ☐ Cài đặt (implementation)
 - ☐ Dịch cấu trúc đó thành chương trình thực thi được.
- ☐ Các hoạt động của pha thiết kế và cài đặt thường liên quan đến nhau hoặc có thể đan xen nhau.



Một mô hình chung của quy trình thiết kế





Các hoạt động thiết kế

- Thiết kế kiến trúc (Architectural design)
 - ▣ Nhận diện cấu trúc toàn hệ thống, những component chính, mối quan hệ giữa các component và chúng được phân tán thế nào.
- Thiết kế giao diện (Interface design)
 - ▣ Định nghĩa giao diện giữa các component hệ thống.
- Thiết kế component (Component design)
 - ▣ Xem xét từng component hệ thống và thiết kế xem nó hoạt động thế nào.
- Thiết kế CSDL (Database design)
 - ▣ Thiết kế các cấu trúc dữ liệu hệ thống và cách chúng được biểu diễn trong CSDL.

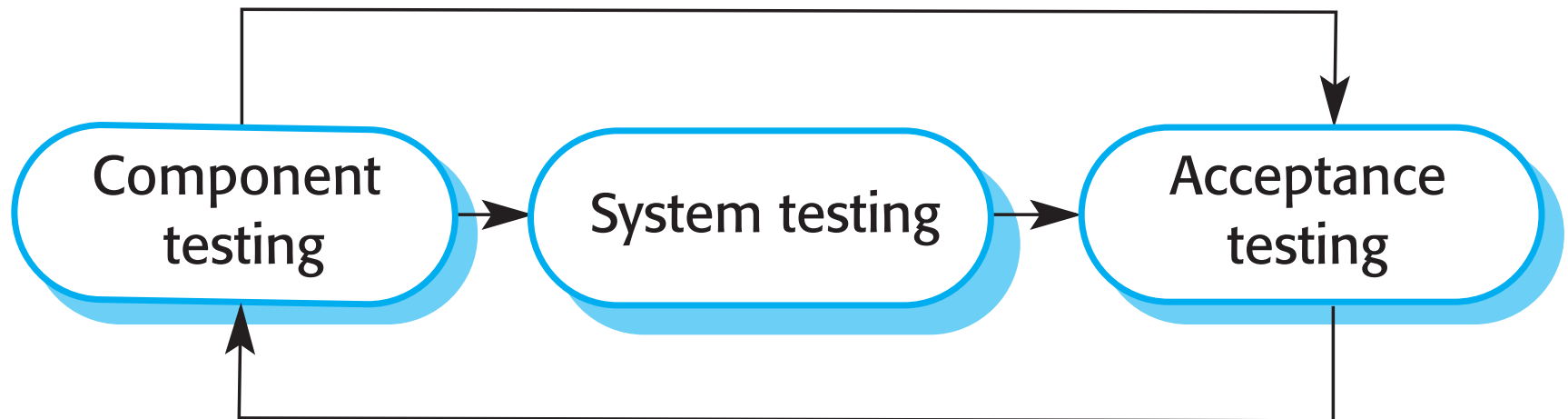


3. Thẩm định phần mềm

- Kiểm định (verification) và thẩm định (validation) (V & V) nhằm mục đích chỉ ra rằng
 - ▣ Một hệ thống tuân theo đặc tả của nó;
 - ▣ Thỏa mãn yêu cầu của khách hàng hệ thống.
- Bao gồm các quy trình kiểm tra, duyệt (review) và kiểm thử hệ thống (system testing).
- Kiểm thử hệ thống bao gồm việc chạy hệ thống sử dụng các test case được viết ra dựa vào đặc tả.
- Kiểm thử (Testing) là hoạt động V&V thường được sử dụng nhất.



Các giai đoạn kiểm thử phần mềm



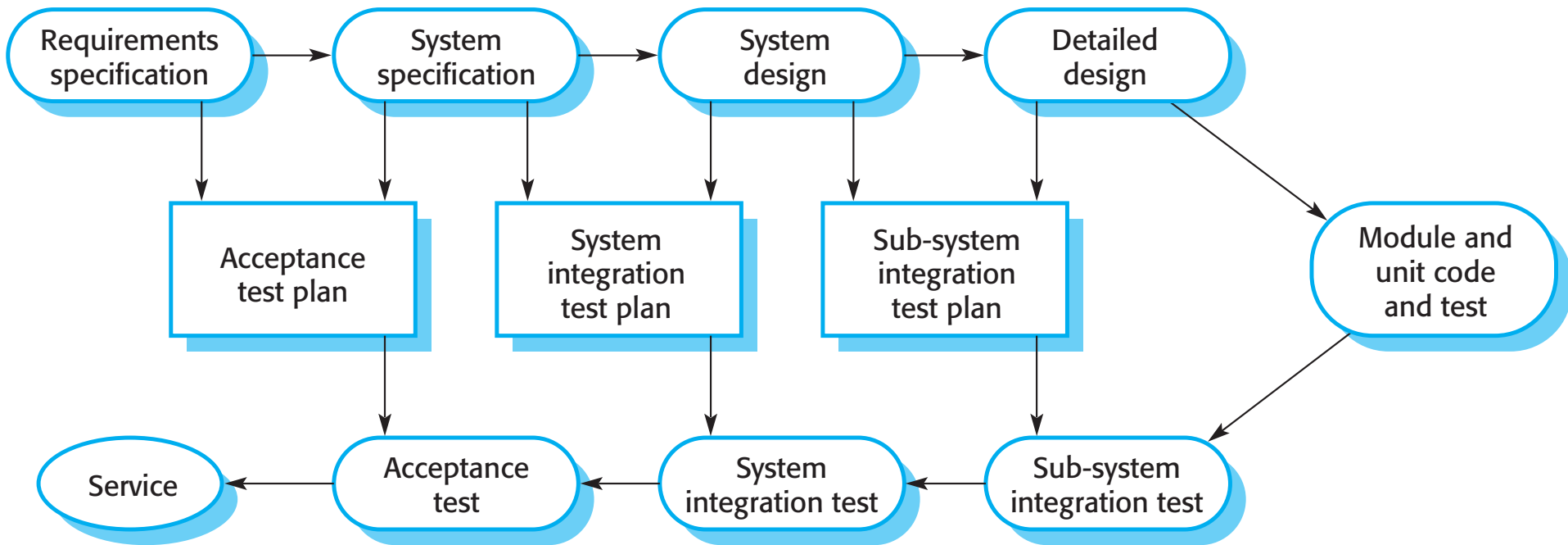


Các giai đoạn kiểm thử phần mềm

- Kiểm thử trong khi phát triển(Development hoặc component testing)
 - ▣ Các component được kiểm thử một cách độc lập;
 - ▣ Các component có thể là hàm hoặc đối tượng hoặc nhóm đồng nhất các thực thể.
- Kiểm thử hệ thống
 - ▣ Kiểm thử toàn bộ hệ thống.
- Kiểm thử người dùng
 - ▣ Kiểm thử với dữ liệu của khách hàng để kiểm tra rằng hệ thống đáp ứng được yêu cầu của khách hàng.



Các pha kiểm thử trong quy trình hoạch định sản

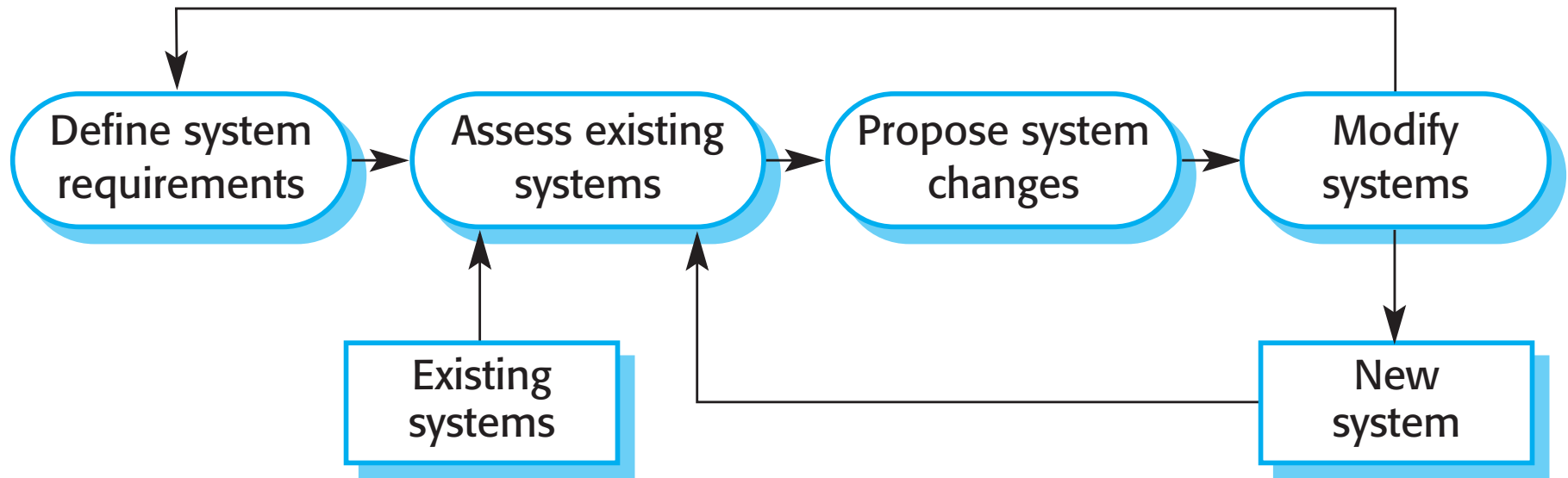




4. Cải tiến phần mềm

- ☐ Phần mềm vốn dĩ linh hoạt và có thể thay đổi.
- ☐ Yêu cầu thay đổi do hoạt động thương mại thay đổi → phần mềm hỗ trợ cũng phải cải tiến và thay đổi theo.
- ☐ Ranh giới giữa phát triển phần mềm và cải tiến phần mềm ngày càng mờ nhạt đi. Ngày càng ít phần mềm được phát triển hoàn toàn mới.

Cải tiến phần mềm



Nội dung



Mô hình quy trình phần mềm

Các hoạt động của quy trình

Thích nghi với sự thay đổi

Quy trình RUP



Thay đổi yêu cầu

- Sự thay đổi yêu cầu là điều hiển nhiên trong những dự án phần mềm lớn
 - ▣ Thay đổi trong hoạt động thương mại → thay đổi yêu cầu hoặc phát sinh những yêu cầu mới.
 - ▣ Công nghệ mới.
 - ▣ Thay đổi nền tảng → thay đổi ứng dụng.
- Thay đổi dẫn đến việc làm lại
 - ▣ Chi phí của việc thay đổi: chi phí làm lại và chi phí cài đặt tính năng mới.



Giảm thiểu chi phí làm lại

- Tránh thay đổi (change avoidance)
 - ▣ Trong quy trình phần mềm có chứa các hoạt động dự đoán trước những thay đổi có thể xảy ra.
 - ▣ Ví dụ: phát triển hệ thống nguyên bản(prototype system) để cho khách hàng xem những tính năng chính của hệ thống và tinh chỉnh yêu cầu.
- Chấp nhận thay đổi (change tolerance)
 - ▣ Quy trình được thiết kế sao cho thay đổi được thực hiện với chi phí khá thấp.
 - ▣ Thường sử dụng mô hình phân phối dần dần.



Nguyên bản phần mềm

- Nguyên bản (prototype) là phiên bản đầu tiên của một hệ thống được dùng để demo những khái niệm và thử các tùy chọn thiết kế, tìm giải pháp cho một vấn đề.
- Một nguyên bản có thể được sử dụng trong các trường hợp sau:
 - ▣ Trong quy trình công nghệ yêu cầu để giúp cho quá trình thu thập yêu cầu và thẩm định yêu cầu;
 - ▣ Trong quy trình thiết kế để tìm ra các giải pháp và phát triển thiết kế giao diện người dùng;
 - ▣ Trong quy trình kiểm thử để chạy các kiểm thử back-to-back.

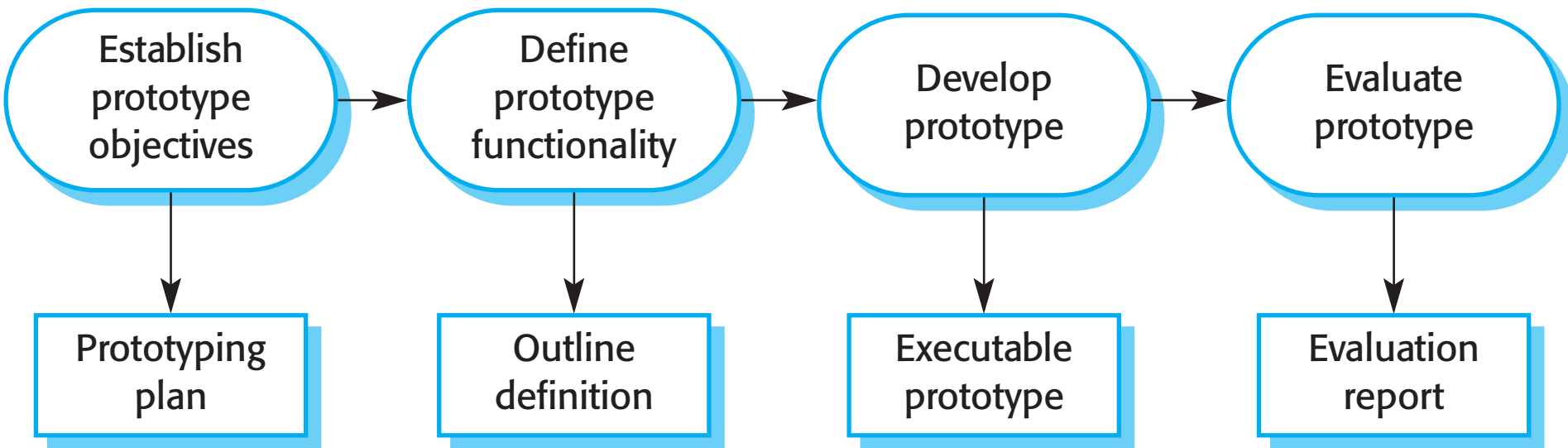


Lợi ích của nguyên bản

- ☐ Cải thiện khả năng sử dụng hệ thống.
- ☐ Thoả mãn tốt hơn nhu cầu thực của người dùng.
- ☐ Cải thiện chất lượng thiết kế.
- ☐ Cải thiện khả năng bảo trì hệ thống.
- ☐ Giảm bớt nỗ lực phát triển.



Quy trình phát triển nguyên bản





Phát triển nguyên bản

- Có thể dựa vào các công cụ và ngôn ngữ để phát triển nguyên bản.
- Có thể loại bỏ một số tính năng
 - ▣ Nguyên bản nên tập trung vào những tính năng chưa được hiểu rõ ràng;
 - ▣ Kiểm tra lỗi không nằm trong nguyên bản;
 - ▣ Tập trung vào các yêu cầu chức năng hơn là các yêu cầu phi chức năng.



Sử dụng nguyên bản

- Nguyên bản không phải là cái cơ bản để phát triển hệ thống → cần được loại bỏ. Do:
 - Khó có thể điều chỉnh hệ thống để đáp ứng được các yêu cầu phi chức năng;
 - Nguyên bản thường không được viết tài liệu;
 - Cấu trúc nguyên bản thường bị phá vỡ do bị thay đổi nhanh;
 - Nguyên bản có thể không đáp ứng được những tiêu chuẩn chất lượng về mặt tổ chức.



Chuyển giao dần dần

- Incremental delivery
- Thay vì phân phối hệ thống một lần, ta phát triển và phân phối hệ thống bằng cách chia ra thành từng phần nhỏ (increment). Mỗi phần giao cho khách hàng chứa một phần tính năng được yêu cầu.
- Những yêu cầu có độ ưu tiên cao nhất sẽ được đặt trong các phần đầu tiên.
- Trong quá trình phát triển, việc phân tích yêu cầu cho phần tiếp theo có thể được tiến hành nhưng thay đổi yêu cầu cho phần hiện tại không được chấp nhận.



Phát triển dần dần và chuyển giao dần dần

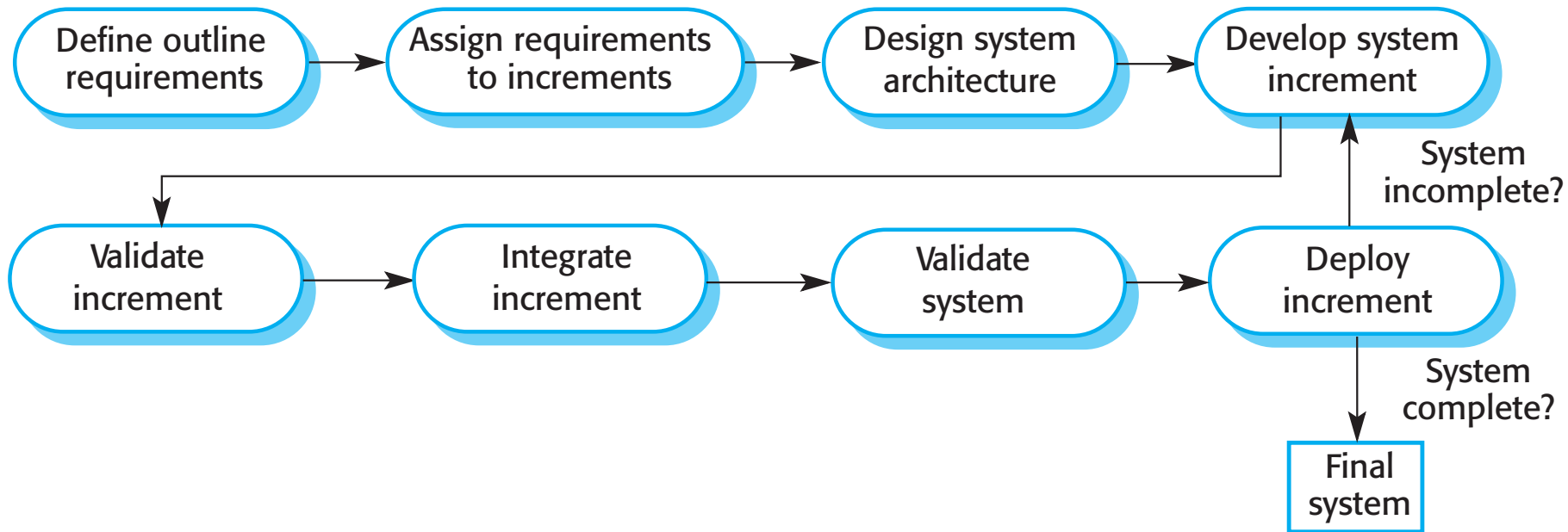
□ Phát triển dần dần

- Phát triển từng phần hệ thống và đánh giá mỗi phần trước khi tiến hành phát triển phần tiếp theo;
- Được sử dụng trong các phương pháp linh hoạt;
- Đánh giá được thực hiện bởi đại diện người sử dụng/khách hàng.

□ Chuyển giao dần dần

- Triển khai một phần để sử dụng cho người dùng cuối;
- Đánh giá thực tế hơn.

Chuyển giao dần dần





Ưu điểm của chuyển giao dần dần

- ☐ Khách hàng sớm được bàn giao sản phẩm (từng phần).
- ☐ Các phần đầu được xem như một nguyên bản để hỗ trợ cho việc xác định yêu cầu cho phần sau.
- ☐ Nguy cơ thất bại toàn hệ thống thấp.
- ☐ Duy trì được ưu điểm của phát triển từng phần → dễ thích nghi với sự thay đổi của hệ thống.
- ☐ Những dịch vụ hệ thống có độ ưu tiên cao nhất sẽ được kiểm thử nhiều nhất
 - ☒ Khách hàng ít gặp lỗi phần mềm ở những phần quan trọng của hệ thống.

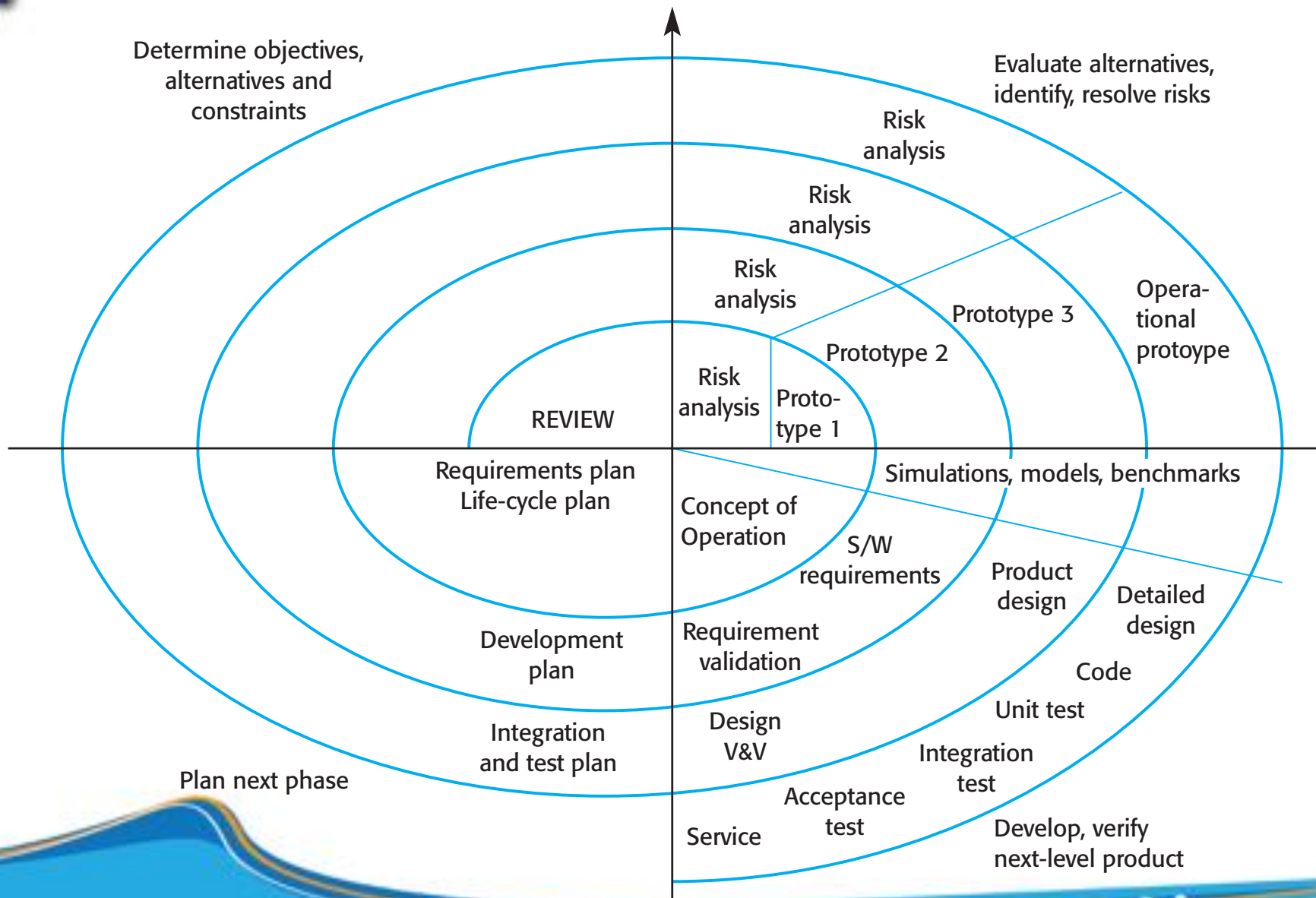


Mô hình xoắn ốc Boehm

- ☐ Quy trình được biểu diễn như một đường xoắn ốc hơn là một chuỗi tuần tự các hoạt động với các bước quay lui.
- ☐ Mỗi vòng lặp trong đường xoắn ốc biểu diễn một pha của quy trình.
- ☐ Không có những pha cố định như đặt tả hay thiết kế, các vòng lặp trong đường xoắn ốc được chọn theo nhu cầu.
- ☐ Rủi ro được đánh giá rõ ràng và được giải quyết trong suốt quy trình.



Mô hình quy trình xoắn ốc Boehm





Các phân khu (sector) của mô hình xoắn ốc

- Thiết lập mục tiêu
 - ▣ Xác định mục tiêu cụ thể của pha.
- Đánh giá và giảm thiểu rủi ro
 - ▣ Rủi ro được đánh giá và các hoạt động được tiến hành để giảm thiểu các rủi ro chính.
- Phát triển và thẩm định
 - ▣ Một mô hình phát triển cho hệ thống được chọn, đây có thể là một trong các mô hình tổng quát.
- Lập kế hoạch
 - ▣ Dự án được duyệt và pha tiếp theo trong đường xoắn ốc được lên kế hoạch.



Sử dụng mô hình xoắn ốc

□ Ưu điểm:

- Là phương pháp thực tế để phát triển các hệ thống phần mềm lớn.
- Giúp các kỹ sư hiểu rõ và tương tác tốt hơn với các nguy cơ tại mỗi mức tiến hóa.
- Cho phép áp dụng nguyên bản tại bất cứ giai đoạn tiến hóa nào.
- Giảm được các nguy cơ trước khi nó trở thành vấn đề của hệ thống.

□ Nhược điểm:

- Không phải là “thuốc chữa bách bệnh”.
- Khó khăn trong việc thuyết phục khách hàng rằng phương pháp này có thể điều khiển được.
- Cần chuyên gia đánh giá về nguy cơ và dựa vào chuyên gia để thành công. Nếu một nguy cơ lớn không được tìm ra và quản lý được, các vấn đề về hệ thống sẽ xảy ra.

Nội dung



Mô hình quy trình phần mềm

Các hoạt động của quy trình

Thích nghi với sự thay đổi

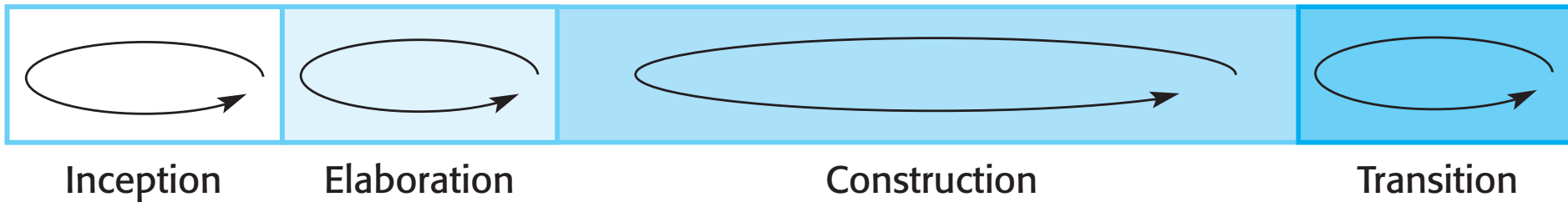
Quy trình RUP



Quy trình RUP

- The Rational Unified Process
- Đây là một quy trình tổng quát bắt nguồn từ UML và Unified Software Development Process.
- Kết hợp các khía cạnh của ba mô hình quy trình tổng quát.
- Thường mô tả ba góc độ:
 - ▣ Góc độ động (dynamic perspective): chỉ ra các pha theo thời gian;
 - ▣ Góc độ tĩnh (static perspective): chỉ ra các hoạt động của quy trình;
 - ▣ Góc độ về thực tiễn (practice perspective): đề xuất những kinh nghiệm thực tế.

Các pha của RUP





Các pha của RUP

- Khởi động (Inception)
 - ▣ Thiết lập business case cho hệ thống.
- Phát triển (Elaboration)
 - ▣ Nghiên cứu các vấn đề và phát triển kiến trúc hệ thống.
- Xây dựng (Construction)
 - ▣ Thiết kế hệ thống, lập trình và kiểm thử.
- Chuyển tiếp (Transition)
 - ▣ Triển khai hệ thống.



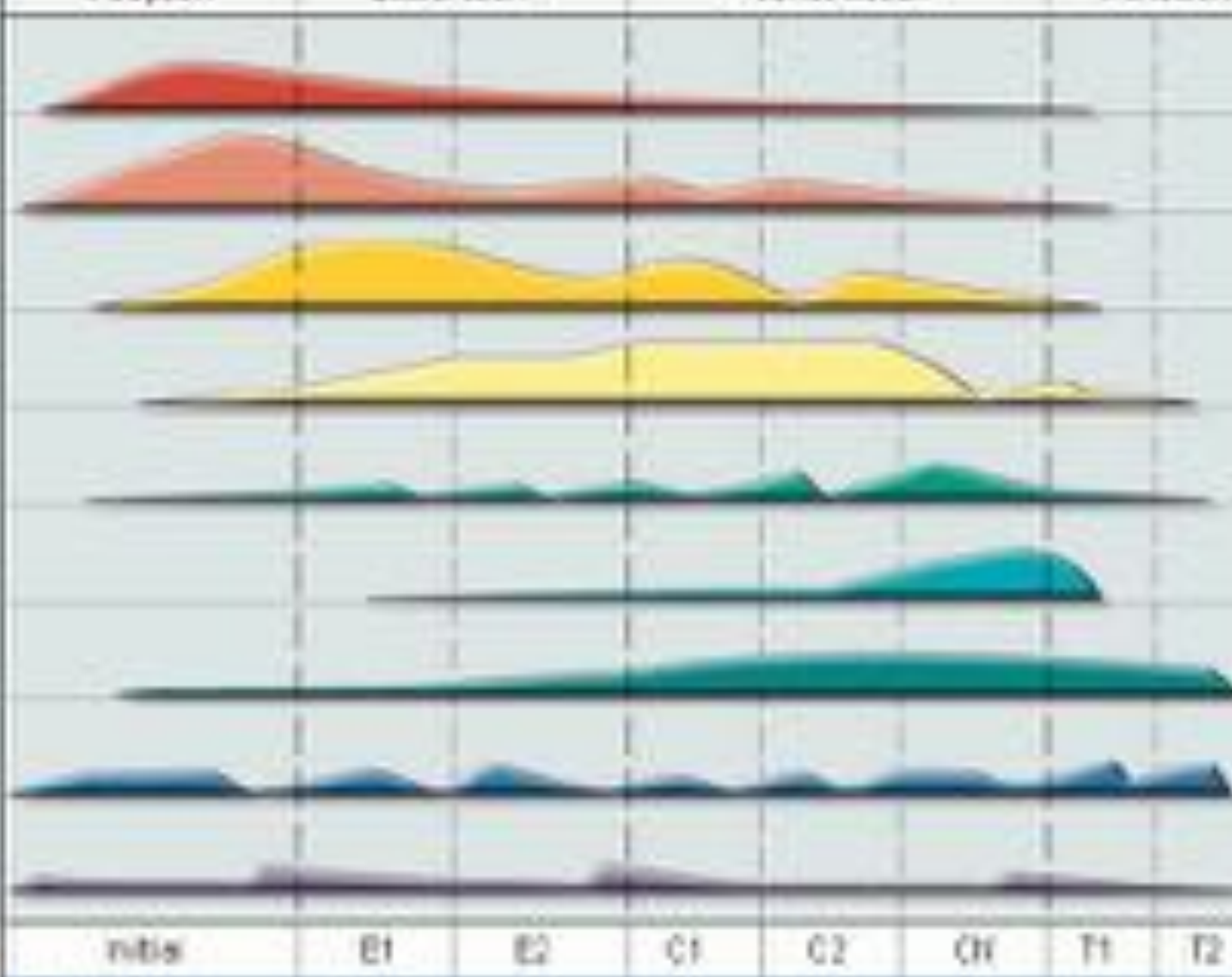
Vòng lặp trong RUP

- Lặp trong từng pha (In-phase iteration)
 - ▣ Mỗi pha được lặp lại với kết quả được phát triển tăng dần.
- Lặp qua các pha (Cross-phase iteration)
 - ▣ Việc lặp có thể được thực hiện qua toàn bộ các pha.



Phases

Inception Elaboration Construction Transition



Iterations

Các workflow tính trong RUP

Workflow	Description
Business modelling	The business processes are modelled using business use cases.
Requirements	Actors who interact with the system are identified and use cases are developed to model the system requirements.
Analysis and design	A design model is created and documented using architectural models, component models, object models and sequence models.
Implementation	The components in the system are implemented and structured into implementation sub-systems. Automatic code generation from design models helps accelerate this process.



Các workflow trong RUP

Workflow	Description
Testing	Testing is an iterative process that is carried out in conjunction with implementation. System testing follows the completion of the implementation.
Deployment	A product release is created, distributed to users and installed in their workplace.
Configuration and change management	This supporting workflow managed changes to the system.
Project management	This supporting workflow manages the system development.
Environment	This workflow is concerned with making appropriate software tools available to the software development team.



Fundamental best practices

- Phát triển phần mềm theo vòng lặp
 - ▣ Các phần được lên kế hoạch dựa vào độ ưu tiên của khách hàng và phân phối những phần có độ ưu tiên cao nhất trước.
- Quản lý yêu cầu
 - ▣ Viết tài liệu một cách rõ ràng cho các yêu cầu khách hàng và theo dõi sự thay đổi của những yêu cầu này.
- Sử dụng kiến trúc dựa vào component
 - ▣ Tổ chức hệ thống như một tập các component có thể tái sử dụng.



Fundamental best practices

- Mô hình hóa phần mềm một cách trực quan
 - ▣ Sử dụng các mô hình đồ họa UML để biểu diễn các góc nhìn tĩnh và động của phần mềm.
- Kiểm tra chất lượng phần mềm
 - ▣ Đảm bảo rằng phần mềm đáp ứng được các chuẩn chất lượng về mặt tổ chức.
- Điều khiển các thay đổi phần mềm
 - ▣ Quản lý những thay đổi phần mềm sử dụng những hệ thống quản lý thay đổi và các công cụ quản lý cấu hình.

Câu hỏi?



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



Bài tập tại lớp

- ☐ Thảo luận nhóm: Làm trên giấy
 - ☐ Tên nhóm, họ tên, mssv nhóm (sv có mặt)



- Với các hệ thống sau, hãy chọn quy trình phù hợp và nêu lý do vì sao nhóm bạn chọn quy trình đó:
1. A system to control anti-lock braking in a car
 2. A university accounting system that replaces an existing system
 3. An interactive travel planning system that helps users plan journeys with the lowest environmental impact
 4. Phát triển hệ thống e-banking trên thiết bị di động dựa vào hệ thống có sẵn: hệ thống web và ATM.
 5. Hệ thống điều khiển robot lau nhà.
 6. Mô tả sơ lược quy trình bạn sẽ sử dụng cho đề án của bạn. Nêu lý do vì sao quy trình đó lại phù hợp với nhóm của bạn và giải thích.