

Chapter 29

■ Bảo trì và Tái kỹ nghệ

Slide Set to accompany

Software Engineering: A Practitioner's Approach, 7/e

by Roger S. Pressman

Slides copyright © 1996, 2001, 2005, 2009 by Roger S. Pressman

For non-profit educational use only

May be reproduced ONLY for student use at the university level when used in conjunction with *Software Engineering: A Practitioner's Approach, 7/e*. Any other reproduction or use is prohibited without the express written permission of the author.

All copyright information MUST appear if these slides are posted on a website for student use.

Bảo trì phần mềm

- Phần mềm được phát hành cho người dùng cuối, và:
 - trong vài ngày, **báo cáo lỗi gửi lại** cho tổ chức kỹ thuật phần mềm.
 - trong vài tuần, một nhóm người sử dụng chỉ ra rằng phần mềm phải **được thay đổi để nó có thể phục vụ các nhu cầu đặc biệt** cho môi trường của họ.
 - trong vài tháng, một nhóm tập thể khác, những người không quan tâm đến phần mềm khi nó được phát hành, bây giờ nhận ra rằng nó có thể cung cấp cho họ lợi ích bất ngờ. Họ sẽ cần **một vài cải tiến** để làm cho nó hoạt động tốt cho họ.
 - Tất cả các công việc trên là bảo trì phần mềm.

Bảo trì

Sơ đồ quy trình

1

Sơ đồ qui trình bảo trì



2

Material requirement planning (MRP)

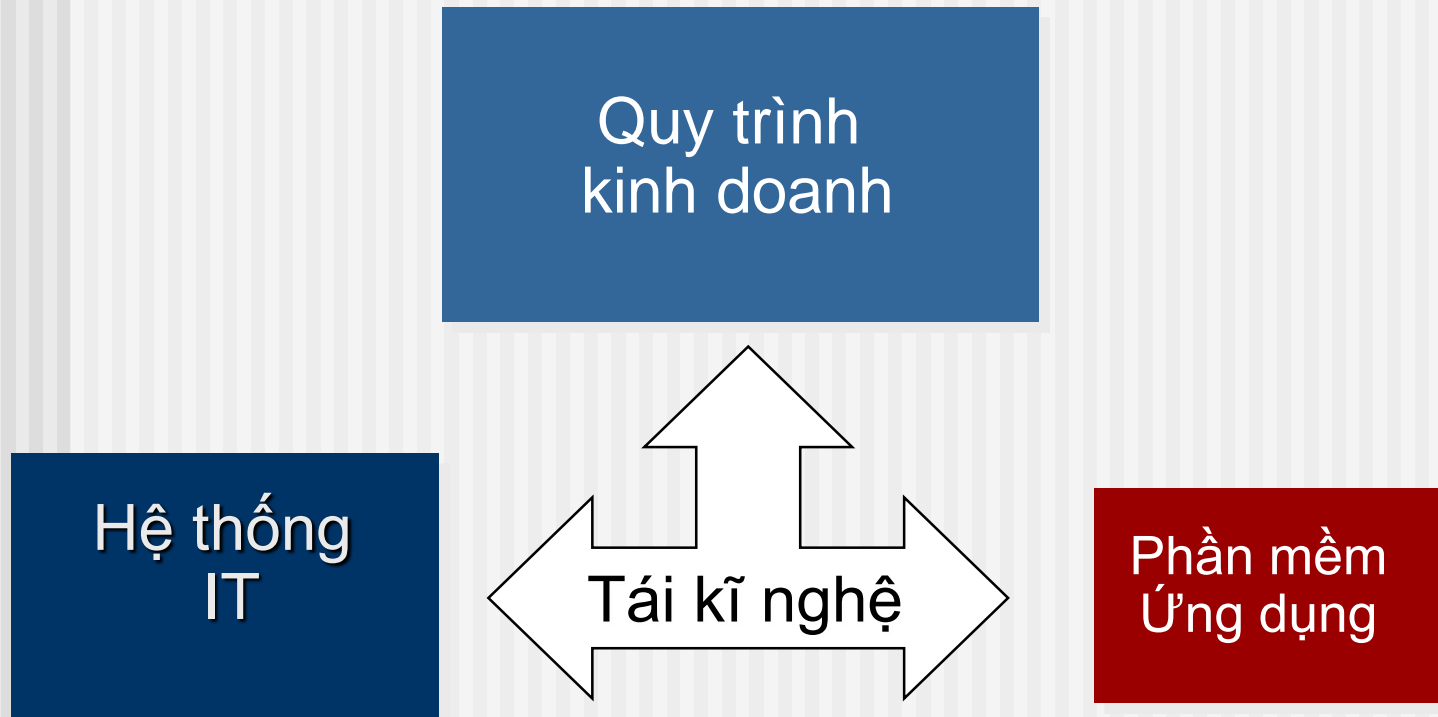
Bảo trì phần mềm

- Bảo trì phần mềm thể hiện tính mô đun có hiệu quả
- Nó sử dụng tốt tính chất của mẫu thiết kế là sự dễ hiểu.
- Nó được xây dựng bằng cách sử dụng các tiêu chuẩn lập trình rất tốt, dẫn đến mã nguồn có khả năng tự mô tả và dễ hiểu.
- Nó được trải qua rất nhiều kỹ thuật đảm bảo chất lượng mà có khả năng phát hiện ra các vấn đề bảo trì tiềm tàng trước khi phần mềm được phát hành.
- Nó được tạo ra bởi các kỹ sư phần mềm, người nhận ra rằng họ có thể không có mặt khi các thay đổi được thực hiện.
 - *Do đó, việc thiết kế và thực thi của phần mềm phải "hỗ trợ" người đang thay đổi nó.*

Khả năng hỗ trợ phần mềm

- "Khả năng hỗ trợ một hệ thống phần mềm phải trong cả vòng đời sản phẩm của chính nó."
 - Điều này ngụ ý rằng không những đem lại sự thoả mãn cho mọi nhu cầu cần thiết hoặc các yêu cầu, mà còn là sự cung cấp các thiết bị, cơ sở hạ tầng hỗ trợ, phần mềm phụ, tiện nghi, nhân lực, hoặc bất kỳ nguồn lực khác là cần thiết để duy trì hoạt động phần mềm và có khả năng đáp ứng chức năng của nó. "[SSO08]
- Phần mềm nên tạo điều kiện thuận lợi để giúp đỡ các nhân viên hỗ trợ khi gặp phải một lỗi trong môi trường hoạt động (và không phạm phải sai lầm, lỗi sẽ bị bắt gặp).
- Nhân viên hỗ trợ nên có quyền truy cập vào một cơ sở dữ liệu chứa các bản ghi của tất cả các lỗi đã được bắt gặp - đặc điểm của chúng, nguyên nhân và cách khắc phục.

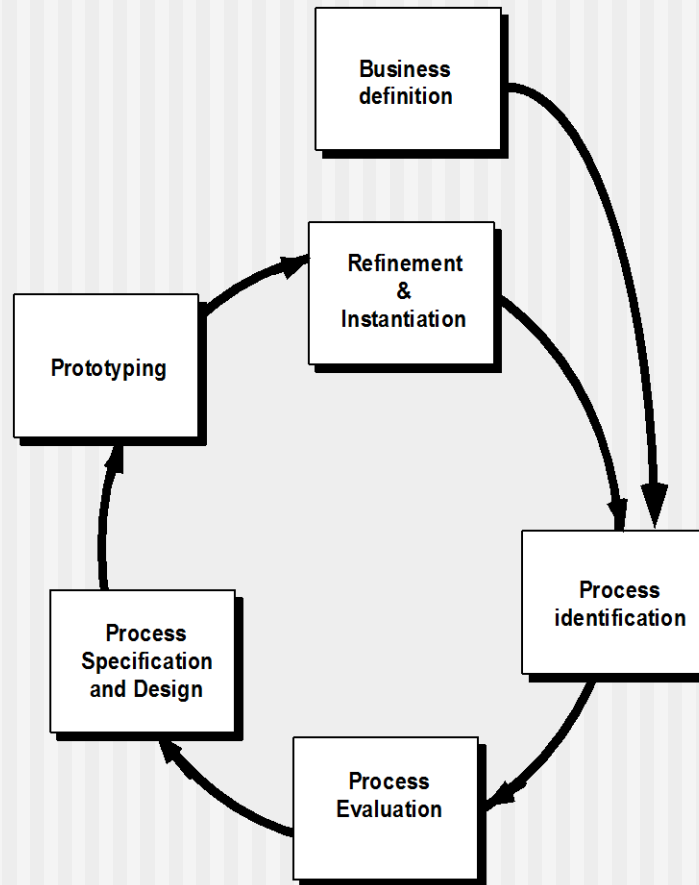
Tái kĩ nghệ



Tái kĩ nghệ quy trình kinh doanh (BPR)

- **Định nghĩa kinh doanh.** Mục tiêu kinh doanh được xác định trong bối cảnh bốn yếu tố chính: giảm chi phí, giảm thời gian, nâng cao chất lượng và phát triển và trao quyền nhân sự.
- **Xác định quy trình.** Các quy trình cốt yếu để đạt được các mục tiêu đã được vạch rõ trong định nghĩa kinh doanh sẽ được xác định.
- **Đánh giá quy trình.** Quy trình hiện tại sẽ được phân tích và đo đạc kỹ lưỡng.
- **Đặc tả và thiết kế quy trình.** Dựa trên những thông tin thu được từ ba hoạt động BPR đầu tiên, sơ đồ use-case được xây dựng cho từng quá trình sẽ được thiết kế lại
- **Chạy thử.** Quy trình kinh doanh được thiết kế lại phải được chạy thử nghiệm trước khi nó được tích hợp hoàn toàn vào thương mại.
- **Cải tiến and thực thể hóa.** Dựa trên thông tin phản hồi từ các mẫu thử nghiệm, quy trình kinh doanh được cải tiến và sau đó thực thể hóa trong một hệ thống kinh doanh.

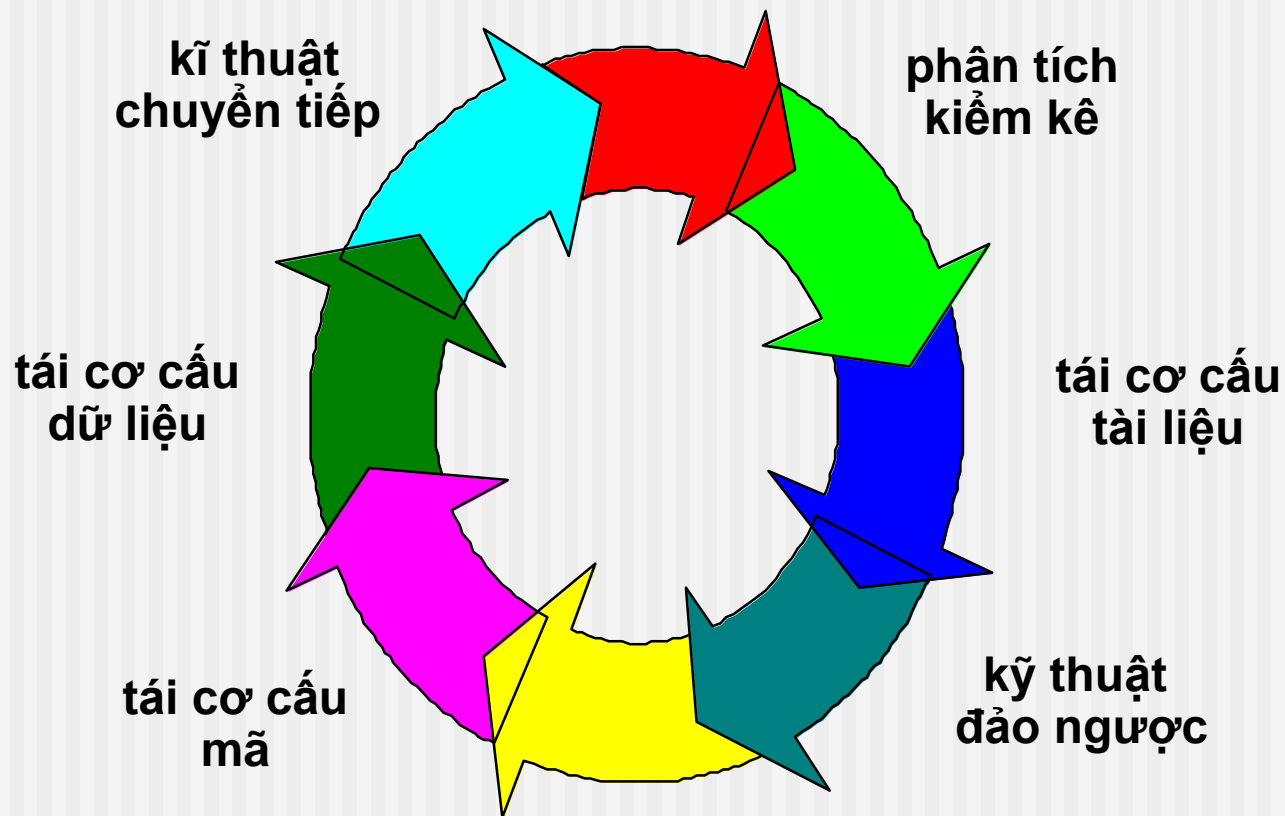
Tái kĩ nghệ quy trình kinh doanh



Nguyên tắc BPR

- Tổ chức theo kết quả, không phải nhiệm vụ.
- Có những người sử dụng đầu ra của quá trình thực hiện quá trình.
- Kết hợp chặt chẽ việc xử lý thông tin với công việc thực tế sản xuất ra thông tin thô.
- Đối xử với các nguồn tài nguyên địa lý phân tán như thể chúng được tập trung.
- Liên kết các hoạt động song song thay vì tích hợp các kết quả của chúng.
- Đặt điểm quyết định vào nơi công việc được thực hiện, xây dựng sự kiểm soát vào trong quá trình.
- Thu thập dữ liệu một lần, tại nguồn của nó.

Tái kĩ nghệ phần mềm



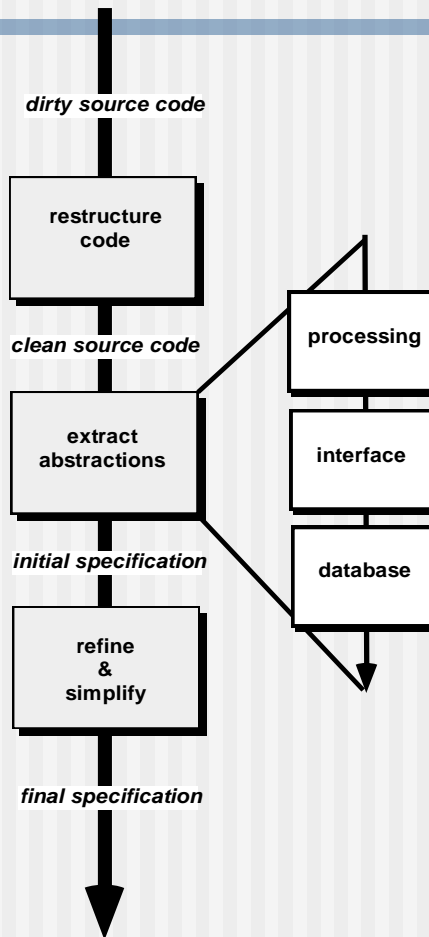
Phân tích kiểm kê

- xây dựng một bảng chứa tất cả các ứng dụng
- thiết lập danh sách các tiêu chuẩn, ví dụ:
 - tên của ứng dụng
 - năm nó được tạo ra
 - số các thay đổi quan trọng tạo đến nó
 - tổng số nỗ lực được áp dụng để tạo ra những thay đổi đó
 - thời gian của lần thay đổi quan trọng cuối cùng
 - nỗ lực được áp dụng để làm thay đổi cuối cùng
 - hệ thống (s) mà nó thuộc về
 - các ứng dụng mà nó kết nối, ...
- phân tích và ưu tiên để lựa chọn các ứng cử viên thích hợp cho việc tái kỹ nghệ

Tái cơ cấu tài liệu

- Tài liệu sơ sài là các thương hiệu của rất nhiều hệ thống kế thừa.
- Nhưng chúng ta có thể làm gì ? Có những lựa chọn nào?
- Những lựa chọn ...
 - *Việc tạo tài liệu là quá tốn thời gian.* Nếu hệ thống hoạt động tốt, chúng ta sẽ cứ để nguyên đó. Trong một số trường hợp, đây là phương pháp đúng.
 - *Tài liệu phải được cập nhật, nhưng chúng ta có nguồn lực hạn chế.* Chúng ta sẽ sử dụng phương pháp tiếp cận "chỉ làm tài liệu khi dụng tới". Có thể là không cần thiết để làm lại hoàn toàn tài liệu .
 - *Hệ thống là rất quan trọng trong công việc và phải được làm lại tài liệu 1 cách đầy đủ .* Ngay cả trong trường hợp này, phương pháp thông minh là cắt giảm tài liệu đến mức tối thiểu .

Kỹ thuật đảo ngược



Kỹ thuật đảo ngược

- Mã nguồn được phân tích nhờ sử dụng công cụ tái cơ cấu.
- Đoạn mã bị thiết kế kém được thiết kế lại
- Sự vi phạm trong việc xây dựng lập trình cấu trúc được ghi nhận và sau đó được tái cơ cấu (điều này có thể được thực hiện tự động)
- Các mã tái cơ cấu kết quả sẽ được đánh và kiểm tra để đảm bảo rằng không xảy ra bất thường
- Tài liệu mã bên trong được cập nhật.

Tái cấu trúc dữ liệu

- Không giống như tái cấu trúc mã xảy ra ở mức độ trừu tượng tương đối thấp, cấu trúc dữ liệu là một hoạt động tái cấu trúc toàn diện
- Trong hầu hết các trường hợp, tái cấu trúc dữ liệu bắt đầu với hoạt động kỹ thuật đảo ngược.
 - Kiến trúc dữ liệu hiện tại sẽ được mở xẻ và các mô hình dữ liệu cần thiết được xác định (Chương 9).
 - Các đối tượng dữ liệu và các thuộc tính được xác định, và các cấu trúc dữ liệu hiện tại sẽ được đánh giá về chất lượng.
 - Khi cấu trúc dữ liệu yếu (ví dụ, CSDL kiểu flat-files hiện đang được sử dụng trong khi phương pháp CSDL quan hệ có khả năng đơn giản hóa việc xử lý một cách cực kì hiệu quả), dữ liệu sẽ được thiết kế lại.
- Bởi vì kiến trúc dữ liệu có một ảnh hưởng mạnh mẽ đến kiến trúc chương trình và các thuật toán liên quan tới nó, việc thay đổi dữ liệu sẽ luôn dẫn đến thay đổi kiến trúc hoặc tầng mã.

Kỹ thuật chuyển tiếp

1. Chi phí để duy trì một dòng trong mã nguồn có thể gấp 20 - 40 lần chi phí phát triển ban đầu của dòng đó.
2. Thiết kế lại của kiến trúc phần mềm (chương trình và / hoặc cấu trúc dữ liệu), sử dụng các khái niệm thiết kế hiện đại có thể tạo điều kiện bảo trì rất hiệu quả trong tương lai.
3. Bởi vì mẫu thử nghiệm của phần mềm đã tồn tại, năng suất phát triển sẽ cao hơn nhiều so với trung bình..
4. Người dùng bây giờ đã có kinh nghiệm đối với phần mềm. Vì vậy, yêu cầu và hướng thay đổi mới có thể được xác định một cách dễ dàng hơn.
5. Công cụ CASE cho việc tái cấu trúc sẽ tự động hoá một phần của công việc.
6. Một cấu hình phần mềm đầy đủ (tài liệu, chương trình và dữ liệu) sẽ có trong khi hoàn thành việc bảo trì phòng ngừa.

Kinh tế học của tái kỹ nghệ-I

- Mô hình phân tích chi phí/lợi ích cho việc tái cấu trúc đã được đề xuất bởi Sneed [Sne95]. Chín tham số được định nghĩa :
 - P_1 = chi phí bảo trì hiện tại hàng năm cho một ứng dụng.
 - P_2 = chi phí hoạt động hiện tại hàng năm cho một ứng dụng.
 - P_3 = giá trị kinh doanh hiện tại hàng năm của một ứng dụng.
 - P_4 = chi phí bảo trì hàng năm dự kiến sau khi tái cấu trúc.
 - P_5 = chi phí hoạt động dự kiến hàng năm sau khi tái cấu trúc.
 - P_6 = giá trị kinh doanh dự kiến hàng năm sau khi tái cấu trúc.
 - P_7 = chi phí tái cấu trúc ước tính.
 - P_8 = thời gian tái cấu trúc ước tính.
 - P_9 = hệ số rủi ro tái cấu trúc ($P_9 = 1.0$ là bình thường).
 - L = tuổi thọ dự kiến của hệ thống.

Kinh tế học của tái kỹ nghệ-II

- Các chi phí liên quan đến bảo trì liên tục của một ứng dụng (giả sử việc tái kỹ nghệ không được thực hiện) có thể được định nghĩa là:

$$C_{\text{maint}} = [P_3 - (P_1 + P_2)] \times L$$

- Các chi phí liên quan đến việc tái kỹ nghệ được xác định bằng công thức sau :

$$C_{\text{reeng}} = [P_6 - (P_4 + P_5) \times (L - P_8) - (P_7 \times P_9)]$$

- Sử dụng chi phí được trình bày trong các phương trình trên, lợi ích tổng thể của việc tái kỹ nghệ có thể được tính như sau:

$$\text{cost benefit} = C_{\text{reeng}} - C_{\text{maint}}$$