

Bảo trì phần mềm



Bảo trì phần mềm được ví như một tảng băng trôi...

Định nghĩa bảo trì phần mềm.

- **"Software maintenance** is the modification of a software product after delivery to correct faults to improve performance or other attributes or adapt the product to a changed environments" (IEEE)

Bảo trì phần mềm là sự thay đổi sản phẩm phần mềm sau khi được phân phối để sửa chữa các lỗi, nâng cao hiệu năng và thuộc tính khác hoặc làm cho sản phẩm thích ứng với sự thay đổi của môi trường.

- **"Software maintenance** is one of the activities in software engineering, and is the process of enhancing and optimizing deployed software (software release), as well as remedying defects ." (<http://en.wikipedia.org>).

Bảo trì phần mềm là một trong các hoạt động của công nghệ phần mềm, là quá trình gia tăng và phát triển lạc quan của phần mềm, được xem như là sửa chữa các thiếu sót.

Định nghĩa bảo trì phần mềm.(tt)

Hệ thống cũ



Hệ thống mới



Bảo trì

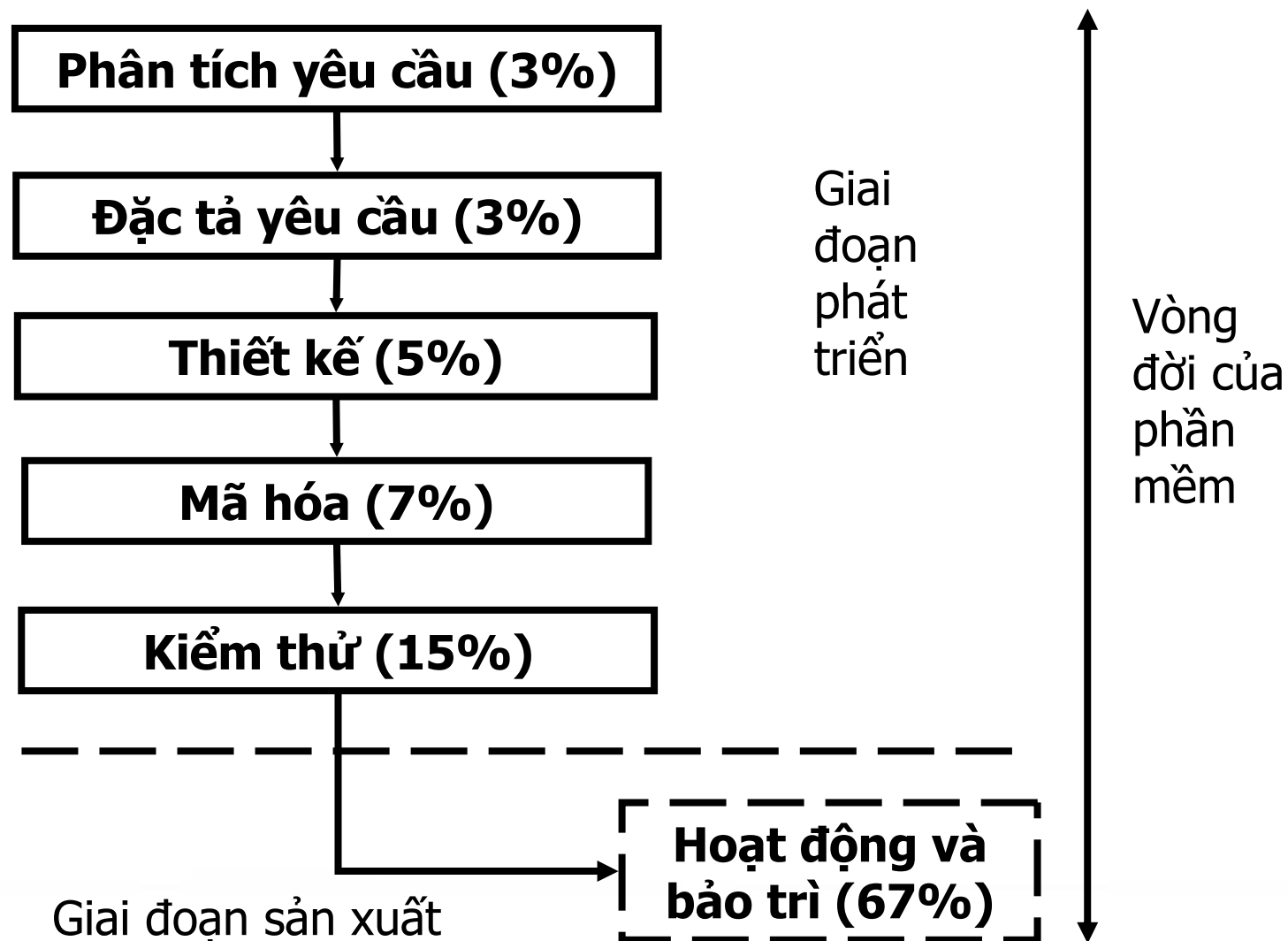
Tại sao phải bảo trì phần mềm?

- Bảo trì phần mềm tiêu tốn từ 50% đến 80% của vòng đời sản phẩm phần mềm.
- Bảo trì phần mềm là công việc thường xuyên của sản phẩm phần mềm.

Các chương trình máy tính luôn luôn thay đổi. Phải sửa lỗi, thêm các phần mở rộng và phải tối ưu hóa.

- Càng có nhiều phần mềm thì khối lượng của việc bảo trì càng tăng cao.
 - Hằng năm, bộ quốc phòng Mỹ (DoD – US Department of Defense) phải bảo trì hằng trăm hệ thống thông tin với hơn 1700 site với hơn 1.4 tỷ dòng lệnh. (Aiken94)
- Bảo trì phần mềm phức tạp hơn nhiều so với sửa lỗi.
- Phải duy trì trong thời gian dài.
 - Duy trì trong 10, 15, ..., 40 năm...

Vị trí của bảo trì phần mềm



Bảo trì khác với phát triển

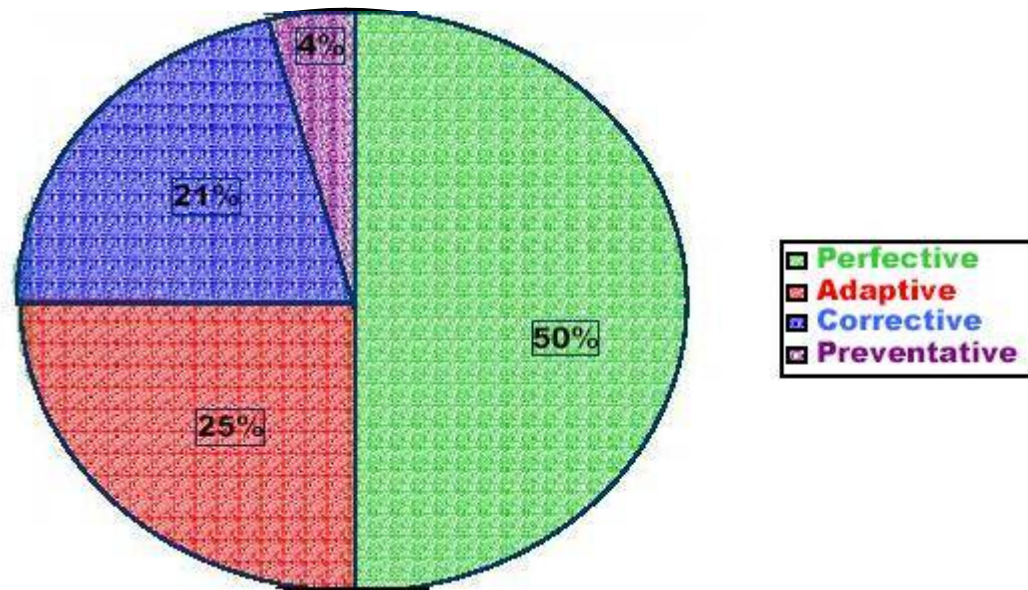
- Bảo trì thực hiện trên một hệ thống sẵn có.
 - ❖ Sự thay đổi phải phù hợp hoặc tương tích kiến trúc, thiết kế, mã hóa sẵn có.
- Khung thời gian thực hiện bảo trì ngắn hơn.
 - ❖ Phát triển tiêu tốn một hoặc nhiều năm.
 - ❖ Bảo trì tiêu tốn một giờ đến 6 tháng
- Dữ liệu kiểm tra có sẵn.
 - ❖ Phát triển tạo ra tất cả dữ liệu kiểm tra một cách không chọn lựa.
 - ❖ Bảo trì sử dụng các dữ liệu kiểm tra sẵn có, thực hiện kiểm tra hồi quy để tạo ra dữ liệu cho sự thay đổi.

Bảo trì kết thúc khi nào?

- Chi phí bảo trì quá cao.
- Tính tin cậy của hệ thống không thể chấp nhận được.
- Hệ thống khác làm việc nhanh hơn, tốt hơn và rẻ hơn.
- Chi phí của việc bảo trì phần mềm đủ lớn để thay thế phần cứng mới hơn và rẻ hơn.
- Các chức năng của hệ thống không còn hữu ích.

Các hoạt động bảo trì

- Bảo trì hiệu chỉnh (corrective maintenance).
- Bảo trì tiếp hợp (adaptive maintenance).
- Bảo trì hoàn thiện (perfective maintenance).
- Bảo trì phòng ngừa (preventive maintenance).



Bảo trì hiệu chỉnh

- Tập trung vào các lỗi thiếu sót trong quá trình phát triển.
- Đây là tiến trình ảnh hưởng trở lại.
 - ❖ Các thiếu sót thường được sửa chữa ngay lập tức hoặc trong tương lai gần.
- Đề cập đến hệ thống không làm việc như ý định ban đầu hay yêu cầu của đặc tả.
- Ví dụ:
 - ❖ Lỗi nằm sinh lớn hơn năm mất không được xử lý lúc phát triển chương trình.
 - ❖ Không thực hiện được việc cập nhập dữ liệu.

Bảo trì tiếp hợp

- Bao gồm tất cả các hoạt động liên quan đến các chức năng của phần mềm.
- Bao gồm tất cả các thay đổi để phần mềm phát triển đáp ứng được nhu cầu của người dùng và môi trường.
- Ví dụ:
 - ❖ Thay đổi mã số thuế.
 - ❖ Thay đổi mã số điện thoại từ 6 số sang 7 số.

Bảo trì hoàn thiện

- Bao gồm các hoạt động để hoàn thiện chất lượng của sản phẩm.
- Bao gồm việc mã hóa lại, tạo và cập nhật các tài liệu, cải tiến sự tin cậy và hiệu năng của phần mềm.
- Đây là hoạt động chính trong bảo trì phần mềm.
- Ví dụ:
 - ❖ Thêm chức năng mới vào sản phẩm.
 - ❖ Làm cho sản phẩm chạy nhanh hơn.

Bảo trì phòng ngừa

- Bao gồm các thay đổi để thích ứng tốt hơn với các môi trường mở rộng sau này.
- Hoạt động này thường được bỏ qua vì:
 - ❖ Tốn quá nhiều tài nguyên.
 - ❖ Không có cách thức để tính toán giá thành.
 - ❖ Nhiều tính năng có thể sẽ không bao giờ dùng đến.

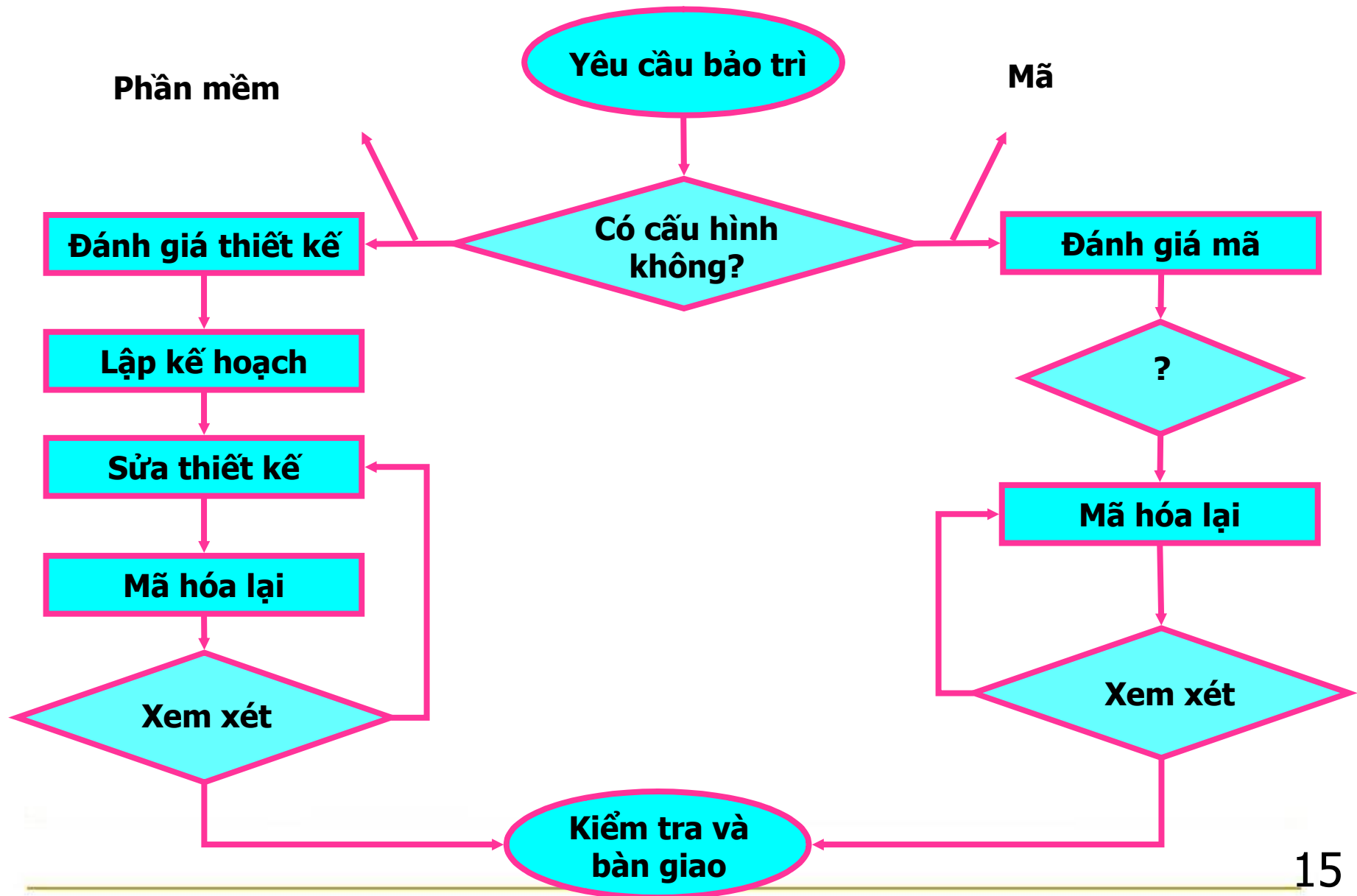
Hoạt động bảo trì bao gồm các công việc được nêu ở trên và được thực hiện đồng thời.

Đặc trưng của bảo trì

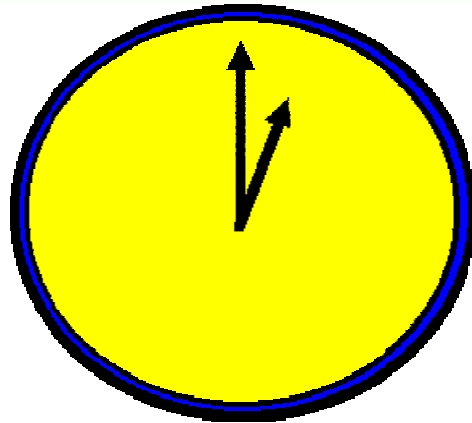
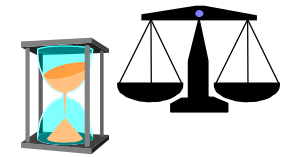
Để hiểu bản chất của bảo trì, chúng ta sẽ xem xét các vấn đề từ 3 góc độ khác nhau:

- Các hoạt động để hoàn thành giai đoạn bảo trì và cách thức tiếp cận phần mềm hiệu quả.
- Chi phí của giai đoạn bảo trì.
- Các vấn đề thường gặp trong khi tiến hành bảo trì.

Bảo trì có cấu trúc và không cấu trúc.



Chi phí của bảo trì

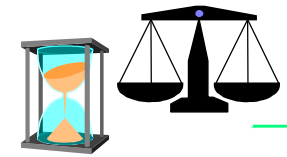


Trực tiếp

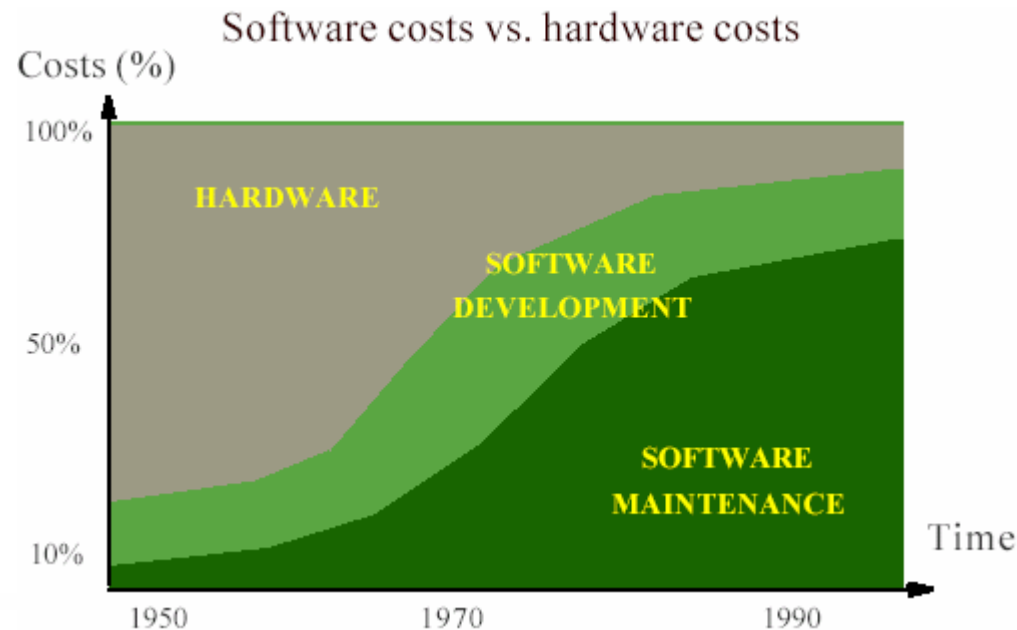


Gián tiếp

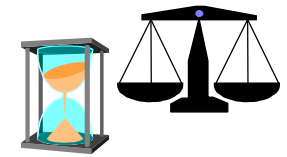
Chi phí trực tiếp



- Năng suất lao động giảm theo tỷ lệ 40:1
 - Chi phí phát triển 1 dòng lệnh là 25\$ thì chi phí để bảo trì có thể lên tới 1000\$.
- Chi phí bảo trì ngày càng chiếm tỷ lệ lớn



Chi phí gián tiếp



- Các cơ hội phát triển bị bỏ qua vì các tài nguyên có sẵn đều dành cho bảo trì.
- Người dùng không hài lòng khi các yêu cầu được sửa đổi không được chú trọng .
- Các lỗi được tạo ra do các sự thay đổi của bảo trì làm giảm chất lượng của phần mềm.
- Làm ngắt quãng quá trình phát triển của nhân viên.

Công thức tính chi phí bảo trì

$$M = p + K * \exp(c-d)$$

- M : toàn bộ các công việc bảo trì.
- p : công việc làm(phân tích, thiết kế, mã hóa, kiểm thử và “wheel-spinning”).
- K : hằng số kinh nghiệm.
- c : đánh giá độ phức tạp được tính cho việc thiếu thiết kế về cấu trúc và dữ liệu.
- d : đánh giá mức độ hiểu biết về phần mềm.

Các vấn đề khác

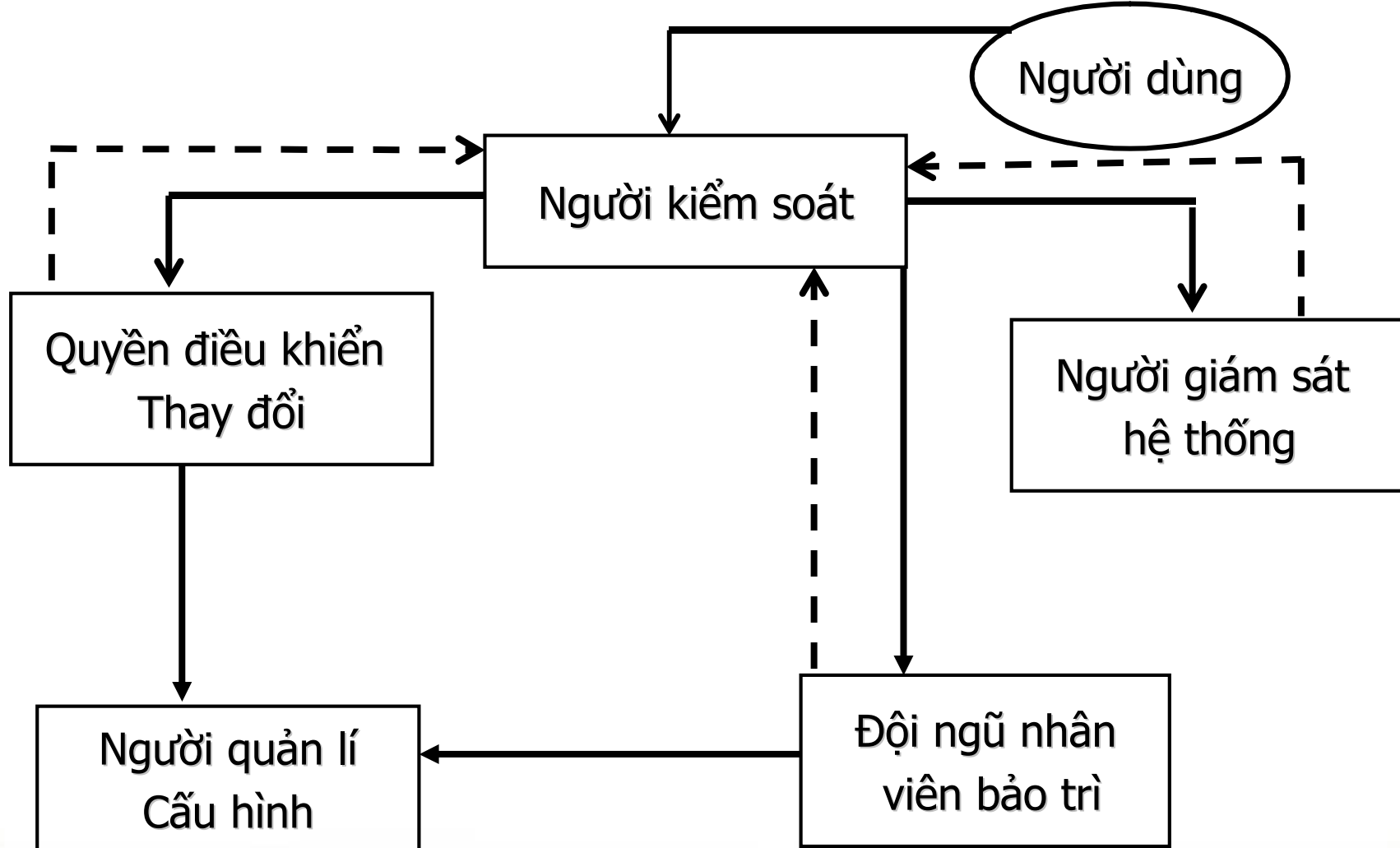
- Rất khó hoặc không thể theo dõi sự tiến hóa của phần mềm.
- Khó khăn trong việc tìm hiểu chương trình do người khác viết
- Các tài liệu không chính xác hoặc thiếu trầm trọng.
- Phần lớn các phần mềm không được thiết kế để thay đổi.
- Việc bảo trì được coi nhẹ quá mức.



Các công việc của bảo trì

- Tổ chức bảo trì được thiết lập.
- Mô tả các thủ tục báo cáo và đánh giá.
- Xác định trình tự cho các sự kiện đối với từng yêu cầu bảo trì.
- Thiết lập thủ tục cất giữ việc ghi lại đối với từng yêu cầu bảo trì.
- Xác định các tiêu chuẩn xét duyệt và đánh giá.

Tổ chức bảo trì



Bộ phận kiểm soát



- Tiếp nhận trực tiếp yêu cầu bảo trì từ người dùng.
- Các yêu cầu phải thông qua người kiểm soát để thống nhất quản lý công việc bảo trì.
- Cộng tác với bộ phận giám sát hệ thống và người có quyền điều khiển thay đổi để kiểm tra và đánh giá tính khả thi của việc bảo trì.
- Đưa chỉ thị bảo trì xuống đội ngũ nhân viên bảo trì.

Bộ phận giám sát hệ thống



- Là thành viên của bộ phận kỹ thuật.
- Am hiểu từng tập con nhỏ nhất của chương trình sản phẩm.
- Tiếp nhận và đánh giá từng yêu cầu bảo trì.
- Cộng tác với bộ phận kiểm soát bảo trì.

Quyền điều khiển thay đổi



- Cộng tác với người kiểm soát để kiểm tra tính hợp lệ của yêu cầu bảo trì.
- Quản lý những thay đổi và tài liệu về sự thay đổi (nội dung thay đổi, thời gian và chi phí cho thay đổi).
- Xác định hành động tiếp theo cần thực hiện.

Bộ phận quản lý cấu hình



- Quản lý cấu hình:
 - ❖ chương trình máy tính(cả chương trình gốc lẫn chương trình thực hiện được).
 - ❖ Tài liệu mô tả cho chương trình máy tính(nhắm vào cho cả người thực hành kỹ thuật lẫn người dùng).
 - ❖ Các cấu trúc dữ liệu.
- Cập nhật lại những thay đổi(dữ liệu và chương trình)từ bộ phận điều khiển thay đổi và nhân viên bảo trì.

Nhân viên bảo trì



- Thực hiện công việc bảo trì theo chỉ dẫn của bộ phận kiểm soát bảo trì.
- Báo cáo một cách cân đối đến bộ phận kiểm soát bảo trì và bộ phận quản lý cấu hình tiến triển của việc bảo trì, và công việc hoàn thành.

Mục đích thành lập tổ chức bảo trì

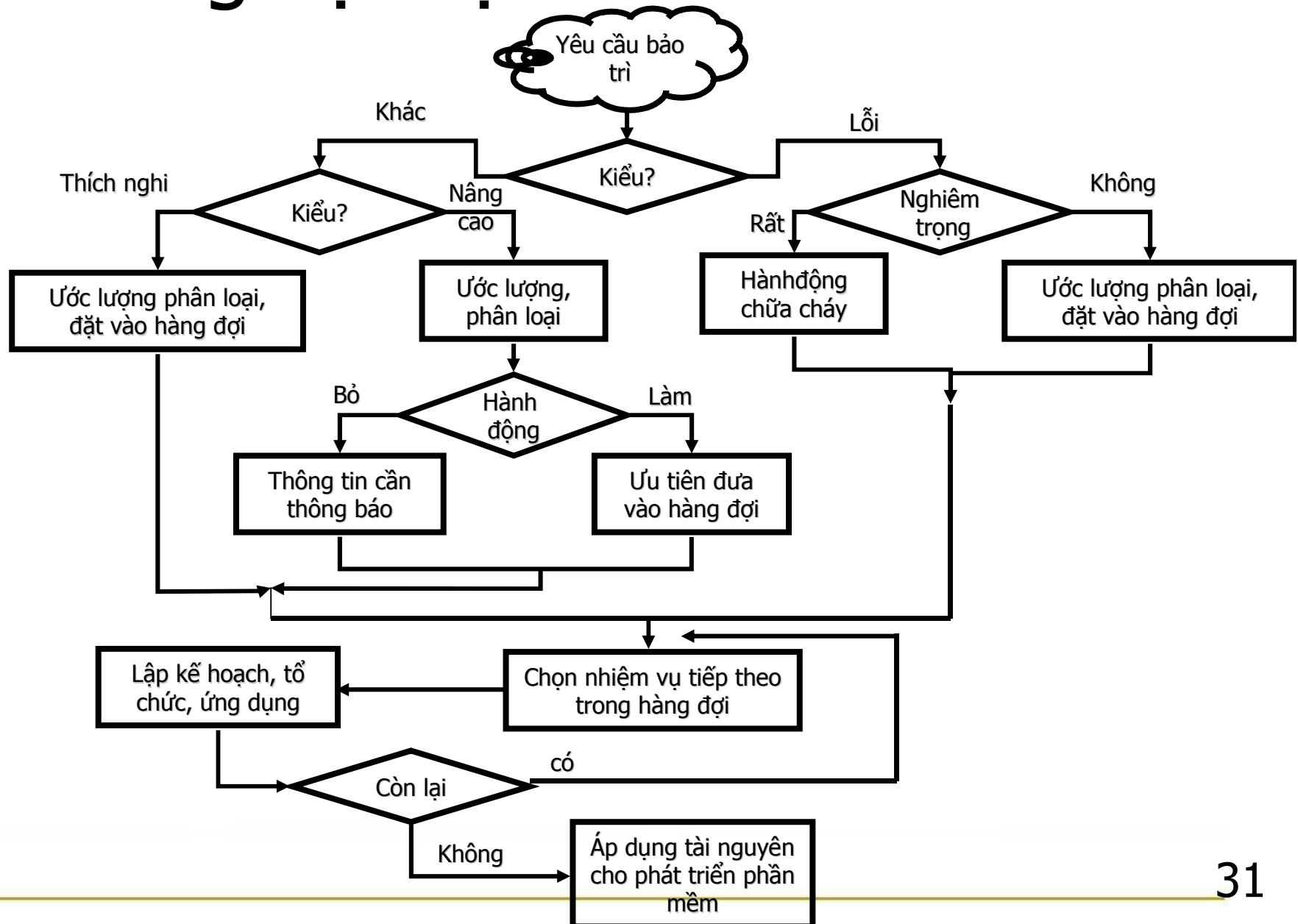
- Làm giảm bớt những lẩn lộn và cải tiến luồng hoạt động bảo trì.
- Việc xác định trách nhiệm của mỗi bộ phận, cá nhân làm tăng năng suất của mỗi người và kiểm chế cảm giác khó chịu khi một người bị kéo ra khỏi nỗ lực phát triển để tiến hành bảo trì.



Báo cáo

- Mẫu yêu cầu bảo trì (MRF) hay còn gọi là báo cáo vấn đề phần mềm, nội dung do người có yêu cầu bảo trì điền vào bao gồm những mô tả đầy đủ về lỗi phát sinh.
- Báo cáo thay đổi phần mềm (SCR) do tổ chức phần mềm xây dựng để chỉ ra:
 - ❖ Độ lớn của nỗ lực cho việc thoã mãn MRF.
 - ❖ Báo cáo những thay đổi cần có.
 - ❖ Số ưu tiên của yêu cầu.
 - ❖ Dữ liệu sau sự kiện về việc thay đổi.

Luồng sự kiện



Lưu trữ bản ghi

Tại sao phải lưu trữ bản ghi?

- Thẩm định tính hiệu quả của kỹ thuật bảo trì.
- Xác định chất lượng của những chương trình được thay đổi sau khi bảo trì.
- Xác định được việc bảo trì là thực sự tốn kém hay không?

Lưu trữ bản ghi(tt)

Việc lưu trữ bản ghi dựa trên những cân nhắc sau:

- Xác định chương trình.
- Số các câu lệnh trong chương trình gốc.
- Số các lệnh mã máy.
- Ngôn ngữ lập trình được dùng.
- Ngày thiết đặt chương trình.
- Số chương trình chạy lần thiết đặt đó.
- Số lần hỏng hóc xử lí liên quan tới mục trên.
- Mức độ và việc định danh sự thay đổi chương trình.
- Số câu lệnh gốc được thêm vào bởi việc thay đổi chương trình.

Lưu trữ bản ghi(tt)

- Số câu lệnh gốc bị xoá đi bởi việc thay đổi chương trình.
- Số người-giờ dành cho việc thay đổi.
- Ngày thay đổi chương trình.
- Định danh kỹ sư phần mềm.
- Định danh MRF.
- Kiểu bảo trì.
- Ngày bắt đầu và kết thúc bảo trì.
- Số tích lũy về người-giờ dành cho bảo trì.
- Ích lợi rõ ràng liên kết với việc bảo trì được thực hiện.

Lưu trữ bản ghi(tt)

Công thức sau đây được gợi ý như là bộ dự báo về số người-tháng, E.maint ,cần chi phí cho việc bảo trì hàng năm:

$$E.maint = ACT \times 2.4 \times KLOC^{1.05}.$$

Với ACT là lưu lượng thay đổi hàng năm :

$$ACT = \frac{KLOC \text{ cho hệ thống đang được bảo hành}}{CI}$$

Với CI là số lệnh chương trình gốc bị thay đổi hay thêm vào trong một năm bảo trì.

Ước lượng

Danh sách tóm tắt các độ đo tiềm tàng có thể được áp dụng để ước lượng nhiệm vụ bảo trì:

- Số trung bình những sai hỏng xử lý trên việc chạy chương trình.
- Toàn bộ số người-giờ dành cho từng phạm trù bảo trì.
- Số trung bình những thay đổi chương trình được thực hiện theo chương trình, theo ngôn ngữ, theo kiểu bảo trì.
- Số trung bình người-giờ trên câu lệnh chương trình gốc cần thêm vào hay bỏ đi do việc bảo trì.
- Số người -giờ trung bình cần dành ra cho ngôn ngữ.
- Thời gian quay vòng trung bình cho một MRF.
- Số phần trăm các yêu cầu bảo trì theo kiểu.

Tính bảo trì được

- **Khái niệm:** *Khả năng bảo trì (tính bảo trì được) có thể coi là các khả năng hiểu, hiệu chỉnh, tiếp hợp hoặc có thể thêm vào khả năng phát triển, nâng cấp trong tương lai, là chìa khoá dẫn đến các phương pháp thiết kế xây dựng phần mềm.*

Các nhân tố kiểm soát

Khả năng bảo trì chịu ảnh hưởng từ nhiều yếu tố. Kopetz định nghĩa một số nhân tố liên quan đến môi trường phát triển:

- Chất lượng hiệu quả của đội ngũ phần mềm.
- Cấu trúc hệ thống dễ hiểu.
- Dễ kiểm soát hệ thống.
- Dùng ngôn ngữ lập trình chuẩn hoá.
- Dùng hệ điều hành chuẩn.
- Cấu trúc tài liệu đã chuẩn hoá.
- Dùng được các trường hợp kiểm thử.
- Tiện nghi gỡ rối bên trong
- Dùng được các máy tính tốt để tiến hành bảo trì.

Cách đo định lượng

Để thẩm định được khả năng bảo trì phải thông qua một số thuộc tính của hoạt động bảo trì có thể đo được như sau:

- Thời gian nhận biết vấn đề.
- Thời gian trễ do các công việc hành chính.
- Thời gian lựa chọn công cụ bảo trì.
- Thời gian phân tích vấn đề.
- Thời gian đặc tả thay đổi.
- Thời gian hiệu chỉnh thực sự.
- Thời gian kiểm thử cục bộ.
- Thời gian kiểm thử tổng thể.
- Thời gian xét duyệt bảo trì.
- Tổng thời gian của công việc bảo trì.

Hiệu ứng phụ của bảo trì.



*...nhưng tất cả những
điều tôi đã làm là
thay đổi câu lệnh này*

Hiệu ứng phụ là bao hàm các lỗi và các kết quả không mong muốn xuất hiện do việc sửa đổi trong quá trình bảo trì.



Hiệu ứng phụ của việc thay đổi mã nguồn

- Một chương trình con bị xoá hay thay đổi
- Một nhãn câu lệnh bị xoá hay thay đổi
- Một biến bị xoá hay thay đổi.
- Những thay đổi để tăng khả năng thực hiện.
- Việc mở hay đóng tệp bị thay đổi
- Các phép toán logic bị thay đổi
- Những thay đổi thiết kế chuyển thành thay đổi về chương trình.
- Thay đổi ảnh hưởng đến việc chạy thử các trường hợp biên.

Hiệu ứng phụ của việc thay đổi dữ liệu

Khi bảo trì, việc sửa đổi thường được tiến hành cho từng phần tử dữ liệu riêng của một cấu trúc dữ liệu. Khi đó thiết kế phần mềm không còn khớp với dữ liệu nữa và lỗi có thể xuất hiện.

Những thay đổi dữ liệu thường gây ra:

- Định nghĩa lại các hằng cục bộ hay toàn cục.
- Định nghĩa lại cấu trúc bản ghi hay tệp.
- Tăng hay giảm kích thước mảng.
- Sửa đổi dữ liệu toàn cục.
- Định nghĩa lại các cờ hay con trỏ điều khiển .
- Bố trí lại các đối số vào ra hay tham số của chương trình con.

Hiệu ứng phụ của việc thay đổi tài liệu

Là những thay đổi trong chương trình gốc mà không được phản ánh trong tài liệu thiết kế hay tài liệu cho người dùng.

- Bất cứ một thay đổi nào đều phải được cập nhật vào tài liệu kỹ thuật.
- Tài liệu mà phản ánh không đúng trạng thái hiện tại của hệ thống thì còn tồi tệ hơn không có tài liệu.
- Một số bảo trì không đòi hỏi phải thay đổi thiết kế hay chương trình mà chỉ cần chỉ ra sự thiếu rõ ràng trong tài liệu người dùng.

Các hình thức bảo trì tiên tiến

- Bảo trì “chương trình xa lạ”
- Kỹ nghệ ngược và tái kỹ nghệ
 - ❖ Khái niệm.
 - ❖ Bảo trì phòng ngừa.
 - ❖ Các yếu tố của kỹ nghệ ngược.
 - ❖ Kỹ thuật tái cấu trúc cho việc Tái kỹ nghệ.

Bảo trì “chương trình xa lạ”

- Tại sao gọi là “chương trình xa lạ” ?
 - ❖ Không còn nhân viên kĩ thuật nào tiếp tục làm việc phát triển chương trình đó nữa.
 - ❖ Không tiếp tục áp dụng lý thuyết phát triển, nên gây ra thiết kế kiến trúc và dữ liệu nghèo nàn
 - ❖ Tài liệu không đầy đủ và việc ghi lại những thay đổi trong quá khứ thì rất sơ sài.

Bảo trì “chương trình xa lạ”

- Vậy phải làm gì với những “chương trình xa lạ” ?
 - ❖ Nghiên cứu chương trình trước khi bạn bị đặt vào “chế độ khẩn cấp”. Cố gắng có được nhiều thông tin cơ sở càng tốt...
 - ❖ Cố gắng làm quen với luồng điều khiển toàn bộ của chương trình, bỏ qua các chi tiết về mã chương trình. Sẽ rất có ích nếu bạn tự vẽ ra được biểu đồ cấu trúc và sơ đồ khối mức cao, nếu chưa có bản nào tồn tại.
 - ❖ Ước lượng tính hợp lí của tài liệu hiện có, bổ sung thêm vào lời giải thích của bạn trong bản in chương trình nguồn nếu bạn nghĩ chúng có ích.
 - ❖ Sử dụng tốt các danh sách chỉ dẫn tham khảo, các bảng kí hiệu và các trợ giúp khác, thông thường nó được trình biên dịch hay hợp dịch cung cấp.

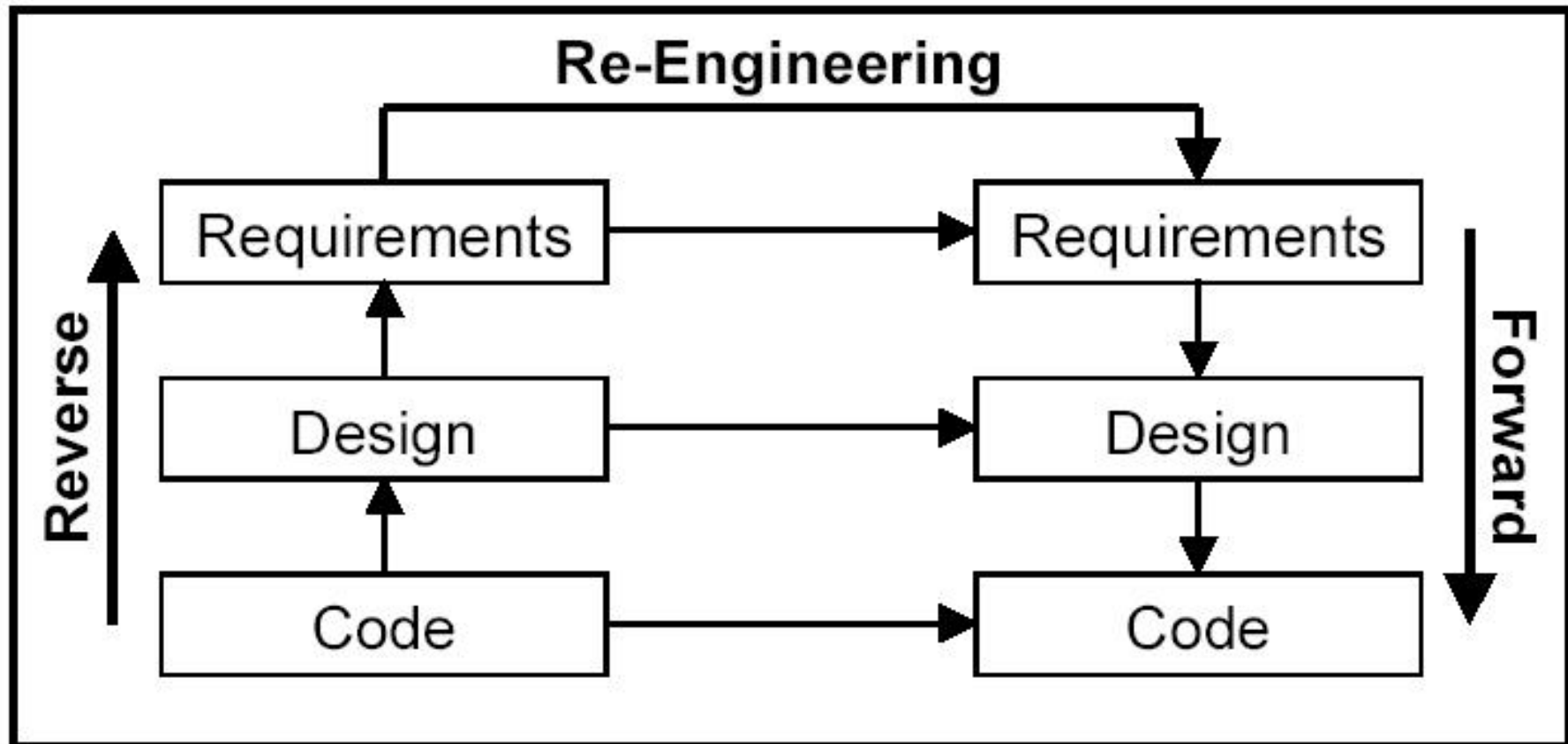
Bảo trì “chương trình xa lạ”(tt)

- ❖ Thực hiện thay đổi chương trình với sự thận trọng lớn nhất. Lưu ý tới kiểu và định dạng của chương trình nếu có thể được. Hãy chỉ ra trên bản in những lệnh nào bạn đã thay đổi.
- ❖ Đừng hủy bỏ mã lệnh trừ phi bạn chắc chắn là nó không được dùng tới
- ❖ Đừng cố dùng chung biến tạm thời và bộ nhớ làm việc đã có trong chương trình. Hãy thêm vào các biến của riêng bạn để tránh rắc rối
- ❖ Hãy giữ các bản ghi chi tiết (về hoạt động và kết quả bảo trì)
- ❖ Tránh sự nóng vội vô lý ném chương trình cũ đi và viết lại nó
- ❖ Thực hiện các kiểm tra lỗi.

Kĩ nghệ ngược và Tái kĩ nghệ

- Kĩ nghệ ngược(re-engineering) là một tiến trình dò lại thiết kế
- Tái kĩ nghệ (cải tiến hay tái chế - reverse engineering) là quá trình dò lại thông tin thiết kế từ phần mềm hiện có và dùng thông tin đó để thay đổi hay thiết kế lại hệ thống hiện có trong cố gắng cải tiến chất lượng toàn bộ của nó.
- Tại sao việc Kĩ nghệ ngược và Tái kĩ nghệ trong mọi công ty lớn (và một số công ty nhỏ) lại không hiện thực?

Kỹ nghệ ngược và Tái kỹ nghệ(tt)



Kĩ nghệ ngược và Tái kĩ nghệ(tt)

- ❖ Một số chương trình này hiếm khi được dùng và không thể thay đổi.
- ❖ Các công cụ kĩ nghệ ngược và Tái kĩ nghệ vẫn còn trong giai đoạn non trẻ.
- ❖ Những công cụ này có thể thực hiện việc kĩ nghệ ngược và Tái kĩ nghệ chỉ cho một lớp giới hạn những ứng dụng.
- ❖ Chi phí (theo nỗ lực và tiền) sẽ bị hạn chế.

Các yếu tố của kỹ nghệ ngược

- Mức trừu tượng: là độ tinh vi của thông tin thiết kế có thể được trích ra từ chương trình gốc.

Tiến trình kỹ nghệ ngược có khả năng:

- ❖ Suy dẫn ra biểu diễn thiết kế thủ tục (mức độ thấp của trừu tượng)
- ❖ Thông tin chương trình và cấu trúc dữ liệu (mức độ trừu tượng cao hơn)
- ❖ Mô hình luồng điều khiển và dữ liệu (mức độ trừu tượng tương đối cao)
- ❖ Mô hình thực thể kết hợp (mức độ trừu tượng cao).

Các yếu tố của kĩ nghệ ngược(tt)

- Tính đầy đủ: là mức độ chi tiết được đưa ra ở mức trừu tượng. Nó làm cải thiện theo tỉ lệ trực tiếp khối lượng phân tích mà con người phải thực hiện trong kĩ nghệ ngược.
- Tính tương tác: được là mức độ con người được “tích hợp” với các công cụ tự động hóa để tạo ra tiến trình kĩ nghệ ngược hiệu quả.
- Tính định hướng: tiến trình kĩ nghệ ngược là một chiều.

Kỹ nghệ tái cấu trúc cho việc tái kỹ nghệ

- Tái kỹ nghệ: tổ hợp các tính năng trích ra thiết kế và phân tích của kỹ nghệ ngược với khả năng tái cấu trúc cho logic, kiến trúc và dữ liệu chương trình
- Tái cấu trúc: được thực hiện để đưa ra một thiết kế sẽ tạo ra một chương trình mới có cùng chức năng nhưng chất lượng cao hơn chương trình gốc.

Bảo trì phòng ngừa

➤ Các tùy chọn:

- ❖ Chúng ta có thể vật lộn với những sửa đổi rồi lại sửa đổi, “đánh nhau” với thiết kế không tưởng minh và chương trình gốc để thực hiện những thay đổi cần thiết.
- ❖ Chúng ta có thể cố gắng hiểu cách làm việc bên trong rộng hơn của chương trình trong nỗ lực tiến hành những thay đổi hiệu quả hơn
- ❖ Chúng ta có thể thiết kế lại, lập trình lại và kiểm thử những phần của phần mềm cần tới sự thay đổi, áp dụng cách tiếp cận kỹ nghệ phần mềm cho tất cả các đoạn được sửa.
- ❖ Chúng ta có thể thiết kế lại hoàn toàn, lập trình lại và kiểm thử chương trình, dùng các công cụ CASE (các công cụ Kỹ nghệ ngược và Tái kỹ nghệ) để trợ giúp chúng ta trong việc hiểu thiết kế hiện tại.

Bảo trì phòng ngừa(tt)

- ❖ Cách tiếp cận bảo trì phòng ngừa Miller tiên phong nêu ra dưới tiêu đề “Khớp ngược có cấu trúc”.
- ❖ Chi phí để bảo trì một dòng chương trình gốc có thể gấp từ 20 đến 40 lần chi phí phát triển ban đầu cho dòng đó

Bảo trì phòng ngừa

- ❖ Việc thiết kế lại kiến trúc phần mềm (chương trình và/hoặc cấu trúc dữ liệu), bằng cách dùng các khái niệm thiết kế hiện đại, có thể làm thuận tiện rất lớn cho việc bảo trì trong tương lai
- ❖ Vì bản mẫu của phần mềm đã có nên hiệu suất phát triển sẽ cao hơn trung bình rất nhiều
- ❖ Người dùng bây giờ có kinh nghiệm với phần mềm. Do đó các yêu cầu mới và hướng thay đổi có thể biết chắc
- ❖ Các công cụ CASE cho kỹ nghệ ngược và tái kỹ nghệ sẽ tự động hóa một phần công việc
- ❖ Một cấu hình phần mềm sẽ tồn tại theo việc hoàn chỉnh bảo trì phòng ngừa.