**Một số hình thức bảo trì phần mềm**

**Bảo trì mã chương trình xa lạ**

Các chương trình được gọi là mã chương trình xa lạ nếu:

• Không một thành viên này trong phòng kỹ thuật tiếp tục phát triển chương trình đó nữa,

• Không tiếp tục áp dụng lý thuyết phát triển, và vì vậy tồn tại vấn đề thiết kế nghéo nàn và ít tài liệu (theo tiêu chuẩn ngày nay), và

• Cấu trúc khối chưa được thiết kế theo tiêu chuẩn, và các khái niệm về thiết kế có cấu trúc chưa được áp dụng.

Yourdon đưa ra một số đề nghị hữu dụng cho người quản trị hệ thống phải bảo trì các chương trình xa lạ như sau:

• Nghiên cứu chương trình trước khi bạn bị đặt vào "chế độ khẩn cấp". Cố gắng thu nhận được càng nhiều thông tin cơ sở càng tốt...

• Cố gắng làm quen với toàn bộ các luồng điều khiển của chương trình; trước hết bỏ qua các chi tiết về mã chương trình. Sẽ rất có ích khi vẽ cho riêng bạn sơ đồ cấu trúc và sơ đồ luồng hoạt động mức cao, nếu chưa có bản nào tồn tại.

• Đánh giá tình hình hợp lý của tài liệu hiện có; bổ sung thêm các lời chú thích của bản thân bạn vào chương trình nguồn nếu bạn thấy cần thiết.

• Sử dụng tốt các danh sách chỉ dẫn tham khảo, các bảng ký hiệu, và các trợ giúp khác thông thường được chương trình dịch và/hoặc assembler cung cấp.

• Thực hiện sửa đổi chương trình với sự chú ý lớn nhất. Lưu ý tới kiểu và dạng của chương trình tại tất cả các chỗ có thể. Đánh dấu trên chương trình những lệnh bạn đã sửa.

• Đừng loại bỏ chương trình trừ khi bạn chắc chắn nó không được sử dụng nữa.

• Đừng cố sử dụng chung các biến tạm thời và vùng nhớ làm việc mà đã có sẵn trong chương trình. Thêm các biến của riêng bạn để tránh các rắc rối.

• Giữ các bản ghi chép chi tiết (vê hoạt động bảo trì và các kết quả).

• Tránh sự nóng vội vô lý ném chương trình cũ đi và viết lại nó.

• Thực hiện các kiểm tra lỗi.

**Công nghệ phản hồi và công nghệ tái sử dụng**

Công nghệ phản hồi -Reverse engineering**-**đối với phần mềm là đơn giản. Trong nhiều trường hợp, chương trình được tổ chức ngược không phải thuộc nhà cạnh tranh mà thuộc bản thân công ty. Bí mật cần khám phá là do không còn giữ được các đặc tả. Do đó tổ chức ngược đối với phần mềm là quá trình phân tích chương trình trong cố gắng biểu diễn lại các chương trình ở mức độ trừu tượng cao hơn mã nguồn. Tổ chức lại là quá trình khám phá thiết kế. Các công cụ của công nghệ phản hồi lấy ra dữ liệu, kiến trúc, các thông tin thiết kế thủ tục từ chương trình đã tồn tại.

Công nghệ tái sử dụng, Re-engineering, không đơn thuần phát hiện các thông tin thiết kế mà còn dùng các thông tin này để biến đổi hoặc tổ chức lại hệ thống đã tồn tại với mục đích cải thiện chất lượng. Trong nhiều trường hợp, phần mềm ứng dụng lại các chức năng của hệ thống tồn tại. Nhưng trong cùng thời điểm, nhà phát triển phần mềm cũng phải thêm các chức năng mới và/hoặc cải thiện các xử lý.

**Bảo trì phòng ngừa**

Bảo trì phòng ngừa các phần mềm máy tính là một vấn đề khá mới và còn đang được tranh cãi. Thay vì đợi cho đến khi nhận được yêu cầu bảo trì, các tổ chức phát triển hay bảo trì chọn một chương trình mà:

• Sẽ được sử dụng trong một số năm định trước;

• Hiện đang được sử dụng tốt,và

• Dễ bị thay đổi hoặc nâng cấp trong tương lai gần.

Thoạt đầu ý kiến đề nghị phát triển lại một chương trình lớn khi một phiên bản đang làm việc đã tồn tại ta thấy dường như quá phung phí. Nhưng chúng ta hãy xem xét các điểm sau:

• Chi phí để bảo trì một dòng mã lệnh có thể lớn hơn 20 tới 40 lần chi phí cho phát triển ban đầu dòng lệnh đó.

• Thiết kế lại cấu trúc của phần mềm, với sự sử dụng các khái niệm thiết kế hiện tại có thể làm cho việc bảo hành tương lai dễ dàng hơn.

• Bởi vì khuôn mẫu phần mềm đã tồn tại, năng suất phát triển chương trình chắc sẽ cao hơn mức trung bình nhiều.

• Người sử dụng bây giờ đã làm quen với phần mềm. Vì vậy, các đòi hỏi mới và hướng thay đổi có thể tìm ra dễ dàng hơn nhiều.

• Các công cụ CASE dành cho reverse engineering và re-engineering sẽ thực hiện tự động một số phần của công việc.

• Một cấu hình phần mềm sẽ tồn tại trên sự hoàn thành của bảo trì phòng ngừa.

Khi một tổ chức phát triển phần mềm bán phần mềm như là một sản phẩm, bảo trì phòng ngừa được xem như "phiên bản mới" của chương trình. Nhiều hãng phát triển phần mềm lớn có thể có từ 500 tới 2000 sản phẩm chương trình trong phạm vi

trách nhiệm của nó. Các chương trình như vậy có thể được xếp theo thứ tự ưu tiên và xem xét lại như các ứng cử cho bảo trì phòng ngừa.

**Chiến lược phần mềm thành phần**

Như đặc tính cổ điển của bảo hành phần cứng là tháo bỏ phần hỏng và thay thế bằng phụ tùng mới. Một khái niệm được gọi là nguyên mẫu phần mềm có thể dẫn tới việc phát triển các phụ tùng cho các chương trình.

Nguyên mẫu phần mềm là một quá trình mô hình hóa yêu cầu ngươi dùng trong một hay nhiều mức chi tiết, bao gồm cả các mô hình làm việc. Các tài nguyên của dự án được xếp đặt làm sao để sản xuất các phiên bản phần mềm được mô tả theo yêu cầu phải nhỏ đi. Phiên bản nguyên mẫu làm cho người dùng, người thiết kế và quản trị... có thể xem lại được phần mềm. Quá trình đó sẽ tiếp tục khi được đề nghị, với phiên bản đang chạy chuẩn bị sẵn sàng phát hành sau vài lần làm lại.

Nếu các mức nguyên mẫu khác nhau được phát triển, nó có thể có một bộ các phụ tùng phần mềm có thể được sử dụng khi nhận được yêu cầu bảo trì hiệu chỉnh. Ví dụ một module phân tích có thể được thiết kế và thực hiện theo hai cách khác nhau nhưng có cùng giao diện bên ngoài. Một phiên bản của module có được sử dụng trong phần mềm làm việc. Nếu module đó hỏng, một phụ tùng có thể được lắp thay ngay.

Mặc dù chiến lược phụ tùng thay thế cho phần mềm có vẻ khác thường một chút, nhưng không có bằng chứng gì nó tỏ ra tốn kém, khi chúng ta tính đến chi phí cho tất cả chu kỳ sống của phần mềm.