

## CHƯƠNG 7

# BẢO TRÌ PHẦN MỀM VÀ QUẢN LÝ THAY ĐỔI PHẦN MỀM

Bảo trì là giai đoạn cuối cùng của một chu trình phát triển phần mềm. Các chương trình máy tính luôn thay đổi- phải mở rộng, sửa lỗi, tối ưu hoá,...và theo thống kê thì bảo trì chiếm đến 70% toàn bộ công sức bỏ ra cho một dự án phần mềm. Do vậy, bảo trì là một hoạt động phức tạp nhưng nó lại là vô cùng cần thiết trong chu trình sống của sản phẩm phần mềm để đảm bảo cho phần mềm phù hợp với người sử dụng.

Do nhu cầu phát triển của các hệ thống thông tin, rất hiếm hay không muốn nói là không thể có một hệ thống thông tin không có sự thay đổi trong suốt chu trình sống của nó. Để duy trì tính đúng đắn, trật tự trong giai đoạn bảo trì thì quản lý sự thay đổi phần mềm là một hoạt động cần thiết song song.

### 7.1. HOẠT ĐỘNG BẢO TRÌ PHẦN MỀM VÀ PHÂN LOẠI

Bảo trì phần mềm là phức tạp và chúng ta có thể chia hoạt động bảo trì ra làm bốn hoạt động như sau:

#### 1. *Bảo trì hiệu chỉnh*

Công việc bảo trì đầu tiên cần phải thực hiện là do việc kiểm tra chương trình không thể tránh được mọi lỗi ẩn chứa bên trong một hệ phần mềm lớn. Trong khi sử dụng bất kỳ một chương trình lớn nào, các lỗi sẽ được báo về lại cho người phát triển.

Bảo trì hiệu chỉnh chính là quá trình phân tích và hiệu chỉnh một hay nhiều lỗi của chương trình.

#### 2. *Bảo trì tiếp hợp*

Hoạt động thứ hai diễn ra bởi sự thay đổi thường xuyên môi trường. Những thế hệ phần cứng mới dường như được công bố theo chu trình 24 tháng một lần. Những hệ điều hành mới hay phiên bản mới của các hệ cũ đều đặn xuất hiện; thiết bị ngoại vi và các thành phần hệ thống khác liên tục được nâng cấp và thay đổi. Thời gian hữu dụng của một phần mềm ứng dụng mặt khác lại dễ dàng vượt qua thời hạn mười năm, lâu hơn môi trường hệ thống đã phát triển nó đầu tiên.

Bảo trì tiếp hợp là hoạt động sửa đổi phần mềm để thích ứng được với những thay đổi của môi trường.

### 3. Bảo trì hoàn thiện

Hoạt động thứ ba diễn ra khi một phần mềm đã được hoàn tất thành công. Hoạt động này chiếm hầu hết các công sức tiêu tốn cho việc bảo trì phần mềm. Lúc sử dụng, các yêu cầu về những khả năng mới, các thay đổi những chức năng đã có, và các mở rộng tổng quát được người dùng gửi đến.

Để thỏa mãn những yêu cầu phát triển của người sử dụng, ta tiến hành bảo trì hoàn thiện.

### 4. Bảo trì phòng ngừa

Bảo trì phòng ngừa là hoạt động bảo trì diễn ra khi phần mềm được thay đổi để cải thiện tính năng bảo trì hay độ tin cậy trong tương lai hoặc để cung cấp một nền tảng tốt hơn cho những mở rộng sau này.

Bảo trì phòng ngừa, hoạt động này vẫn còn nhiều xa lạ trong thế giới phần mềm hiện nay.

Các thuật ngữ dùng để mô tả ba hoạt động bảo trì đầu tiên là do Swanson đề xướng. Thuật ngữ thứ tư thường được dùng trong việc bảo trì phần cứng hay các hệ thống vật lý khác. Tuy nhiên cần chú ý rằng những điểm tương tự giữa bảo trì phần mềm và bảo trì phần cứng có thể gây nhầm lẫn. Phần mềm khác với phần cứng, không thể tận dụng được, vì vậy hoạt động bảo trì phần cứng chủ yếu - thay thế các bộ phận bị hỏng hóc hay gãy vỡ - không được kể đến.

Trong thực tế của hoạt động bảo trì, các nhiệm vụ được làm như một phần của bảo trì tiếp hợp và bảo trì hoàn thiện cũng giống như các nhiệm vụ cần làm trong giai đoạn phát triển của một chu trình phần mềm. Để tiếp hợp hay hoàn thiện, chúng ta đều phải xác định yêu cầu, thiết kế lại, tạo mã và kiểm tra phần mềm có được. Thông thường các nhiệm vụ đó đã được gọi là bảo trì rồi.

## 7.2. ĐẶC ĐIỂM CỦA BẢO TRÌ PHẦN MỀM

Bảo trì phần mềm cho tới gần đây vẫn còn là một giai đoạn bị coi nhẹ của chu trình phần mềm. Kiến thức về bảo trì có được rất ít khi so sánh với các giai đoạn hoạch định và phát triển. Có rất ít các số liệu nghiên cứu và chế tạo tập trung vào đề tài này, và rất ít các phương pháp kỹ thuật được đưa ra. Để hiểu được điểm bản chất của bảo trì, chúng ta sẽ xem xét các vấn đề từ ba góc độ khác nhau:

- Các hoạt động cần thiết để hoàn thành giai đoạn bảo trì và tính toàn vẹn của một cách tiếp cận theo công nghệ phần mềm đối với hiệu quả của những hoạt động đó, hay sự thiếu hụt nó.
- Chi phí kèm theo giai đoạn bảo trì.
- Các vấn đề thường gặp phải khi tiến hành bảo trì phần mềm.

### 7.2.1. Bảo trì có cấu trúc đối với bảo trì không cấu trúc.

Nếu thành phần có được duy nhất của một cấu hình phần mềm là mã nguồn, hoạt động bảo trì bắt đầu với việc đánh giá chi tiết mã nguồn thường là khá phức tạp

với những tài liệu nghèo nàn bên trong. Những đặc điểm tế nhị như cấu trúc phần mềm, các cấu trúc dữ liệu toàn cục, giao diện hệ thống, hoạt động và các ràng buộc thiết kế thường rất khó sáng tỏ và hay bị hiểu lầm. Các thay đổi lật vật thường xuyên làm cho các mã rất khó đánh giá. Các kiểm tra hồi quy (lặp lại các kiểm tra trước kia để đảm bảo rằng những thay đổi không tạo ra lỗi trong phần mềm đã hoạt động) là không thể thực hiện được bởi không hề có các bản lưu về các kiểm tra đó. Chúng ta đang tiến hành phép bảo trì không cấu trúc và đang phải trả giá (khi lãng phí công sức và gây tâm trạng thất vọng). Sự trả giá này luôn đi kèm với các phần mềm đã không được phát triển theo những phương pháp đúng đắn.

Nếu có một cấu hình phần mềm hoàn thiện, nhiệm vụ bảo trì bắt đầu bằng việc đánh giá các tài liệu thiết kế. Sau đó là xác định các đặc điểm thuộc cấu trúc quan trọng, các đặc điểm hoạt động và giao diện. Tính toàn vẹn của những sửa đổi và hiệu chỉnh cần thiết sẽ được đánh giá và một kế hoạch sửa đổi sẽ được thiết lập. Thiết kế được thay đổi (sử dụng những kỹ thuật phù hợp với những điều đã bàn luận ở các chương trước) rồi nhận xét đánh giá. Mã nguồn được phát triển, sau đó tiến hành các kiểm tra hồi quy sử dụng thông tin chứa trong phần "đặc tả kiểm tra" và rồi phần mềm lại được phát hành.

Các mô tả trên đây là phép bảo trì cấu trúc và được tiến hành như là kết quả của những ứng dụng trước đây trong khoa học về công nghệ phần mềm. Mặc dù sự có mặt của một cấu hình phần mềm không đảm bảo được các vấn đề bảo trì nảy sinh, nhưng khối lượng công việc đã được giảm bớt và chất lượng chung của những thay đổi và hiệu chỉnh đã được cải thiện.

### 7.2.2. Giá thành bảo trì

Theo thống kê, giá thành cho việc bảo trì các phần mềm tăng lên một cách đều đặn trong suốt 20 năm qua. Trong những năm 1970, bảo trì chiếm đến 35 đến 40 phần trăm kinh phí phần mềm dành cho tổ chức hệ thống thông tin. Tỷ lệ này đã nhảy tới con số 60 vào giữa những năm 1980. Và nhiều công ty đã chi 80% kinh phí cho việc bảo trì vào giữa những năm 1990.

Chi phí cho việc bảo trì là rõ ràng nhất. Tuy nhiên những chi phí khác khó thấy hơn có thể gây ra mối quan tâm lớn hơn:

- Một chi phí khó xác định của việc bảo trì phần mềm là các cơ hội phát triển bị bỏ qua vì các tài nguyên có sẵn đều dành cho nhiệm vụ bảo trì.
- Sự không hài lòng của người dùng khi các yêu cầu có vẻ hợp lý cho việc sửa chữa hay sửa đổi không được chú ý một cách hợp lý.
- Việc suy giảm chất lượng nói chung do lỗi tạo ra bởi sự thay đổi trong các phần mềm được bảo trì.
- Một yêu cầu bất chợt làm ngắt quãng quá trình phát triển của một nhân viên buộc anh ta tiến hành công việc bảo trì.

Chi phí cuối cùng cho việc bảo trì là sự giảm sút kinh khủng về năng suất lao động (được đo theo số dòng lệnh -LOC- của một người trong một tháng hay số tiền chi phí cho dòng lệnh). Sự giảm sút này xuất hiện khi tiến hành bảo trì đối với các phần mềm cũ. Người ta đã ghi nhận sự giảm sút năng suất lao động theo tỷ lệ 40:1, có nghĩa

là chi phí cho việc phát triển trị giá \$25.00 trên một dòng lệnh sẽ có thể trị giá tới \$1000.00 cho việc bảo trì mỗi dòng lệnh.

Công sức cho việc bảo trì có thể được phân chia thành các thao tác làm việc: phân tích, ước lượng, thay đổi thiết kế, và sửa chữa chương trình nguồn và các thao tác lặp lại: việc cố gắng hiểu chương trình nguồn làm gì, cố gắng sáng tỏ cấu trúc dữ liệu, các thuộc tính giao diện, giới hạn của việc thực hiện,... Biểu thức dưới đây cung cấp một mô hình cho công việc bảo trì:

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

với M = toàn bộ các công việc cho việc bảo trì;

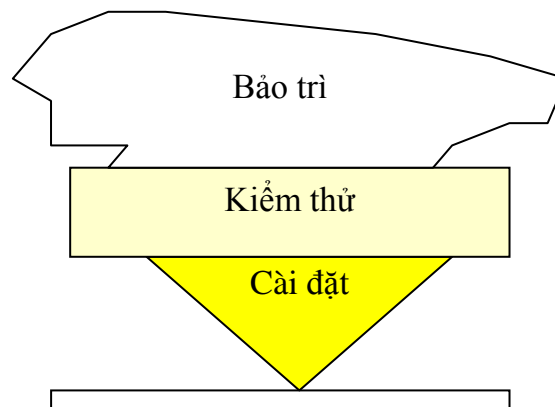
p = công việc làm (như miêu tả ở trên);

K = hằng số kinh nghiệm;

c = đánh giá mức 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.

Mô hình trên đây cho thấy công việc và giá thành có thể tăng lên theo cấp số mũ nếu một phương pháp phát triển phần mềm kém cỏi được sử dụng - tức là thiếu sót của công nghệ phần mềm, và nếu một người hay một nhóm dùng các phương pháp không có giá trị để bảo trì phần mềm. Chi phí cho bảo trì khi phần mềm được phát triển không đúng phương pháp được minh họa ở hình sau:



Chi phí của việc phát triển phần mềm không có phương pháp

### 7.2.3. Một số vấn đề khác

Hầu hết các vấn đề liên quan tới việc bảo trì phần mềm đều liên quan tới các sai lệch trong cách xây dựng và phát triển phần mềm. Sự thiếu sót trong việc điều khiển và tổ chức trong hai giai đoạn đầu tiên của một chu trình phần mềm gần như luôn luôn tạo ra các vấn đề giai đoạn cuối.

Nhiều vấn đề kinh điển có thể liên quan tới việc bảo trì phần mềm được trình bày dưới đây:

- Rất khó hoặc không thể theo dõi sự tiến hóa của phần mềm qua các phiên bản. Các thay đổi không được tư liệu hóa.
- Khó theo dõi được các quá trình xử lý được tạo bởi các phần mềm.
- Thường xuyên gặp rất nhiều khó khăn trong việc tìm hiểu chương trình của người khác viết. Những khó khăn này tăng lên khi số thành phần các cấu

hình của phần mềm giảm đi. Nếu chỉ có các chương trình nguồn không có tài liệu hướng dẫn thì không nên tìm hiểu phần mềm đó.

- Những người viết phần mềm thường không có mặt để giải thích. Chúng ta không thể trông cậy vào những giải thích cá nhân của các nhà phát triển phần mềm khi việc bảo trì được yêu cầu.
- Các tài liệu chính xác không có hay thiếu trầm trọng. Phải thừa nhận rằng phải có tài liệu về phần mềm là bước đầu tiên, nhưng tài liệu phải hiểu được và phù hợp với chương trình lại là chuyện khác.
- Phần lớn các phần mềm không thiết kế để thay đổi, trừ phi sử dụng phương pháp thiết kế dùng các khái niệm về phân tách chương trình thành các module độc lập. Việc thay đổi phần mềm sẽ rất khó khăn và dẫn đến xu hướng sai.
- Việc bảo trì phần mềm không được coi là một công việc dễ dàng mà công việc bảo trì phần mềm luôn liên quan tới các sai lệch ở mức độ cao.

### **7.3. CÔNG VIỆC BẢO TRÌ PHẦN MỀM VÀ MỘT SỐ HIỆU ỨNG LẪ**

#### **7.3.1. Khả năng bảo trì**

Khả năng bảo trì của phần mềm có thể coi như các khả năng hiểu, hiệu chỉnh, tiếp hợp hoặc có thể thêm vào khả năng phát triển. Khả năng bảo trì là chìa khóa để dẫn đến các phương pháp thiết kế xây dựng phần mềm.

##### **7.3.1.1. Yếu tố kiểm soát**

Khả năng bảo trì cơ bản bao gồm nhiều yếu tố. Sự thiếu cẩn trọng trong việc thiết kế, viết chương trình nguồn, kiểm tra có ảnh hưởng tiêu cực đến việc bảo trì có kết quả một phần mềm. Cấu hình yếu kém cũng có các tác động tương tự, thậm chí cả khi từng bước của kỹ thuật xây dựng phần mềm được xem xét một cách cẩn thận. Thêm vào đó nhiều yếu tố khác liên quan tới phương pháp phát triển phần mềm, như:

- Chất lượng hiệu quả của đội ngũ phần mềm.
- Cấu trúc của hệ thống dễ hiểu.
- Dễ dàng kiểm soát hệ thống.
- Dùng các ngôn ngữ lập trình chuẩn.
- Dùng các hệ điều hành chuẩn.
- Dùng các cấu trúc chuẩn tài liệu.
- Dùng được các tài liệu kiểm tra.
- Phương tiện gỡ rối đi kèm.
- Dùng được các máy tính tốt để thực hiện việc bảo trì.

Các yếu tố được đưa ra ở trên đã phản ánh những đặc điểm về phần cứng cũng như chương trình nguồn được dùng trong việc phát triển phần mềm. Những yếu tố khác chỉ ra sự cần thiết để có được phương pháp chuẩn, chương trình nguồn chuẩn. Có thể, yếu tố quan trọng nhất tác động tới khả năng bảo trì là kế hoạch cho khả năng bảo trì. Nếu coi phần mềm như là một hệ thống các thành phần sẽ phải chịu những thay đổi không tránh được, thì cơ hội tạo những phần mềm có khả năng bảo trì sẽ tăng thực sự.

### **7.3.1.2. Đánh giá định lượng**

Khả năng bảo trì, như chất lượng hay độ tin cậy là hết sức khó xác định.

Tuy nhiên chúng ta có thể đánh giá khả năng bảo trì gián tiếp bằng việc xem xét các thuộc tính của các công việc bảo trì có thể đánh giá được:

- 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 xác định sự thay đổi.
- Thời gian hiệu chỉnh (hay sửa đổi) thực sự.
- Thời gian chạy thử cục bộ.
- Thời gian chạy thử tổng thể.
- Thời gian tổng kết bảo trì.
- Tổng thời gian của công việc bảo trì.

Mỗi đánh giá trên thực tế là các dữ liệu đó có thể cung cấp cho người quản lý cùng với chỉ số về hiệu quả của công việc.

### **7.3.2. Các công việc bảo trì**

Những nhiệm vụ liên quan tới việc bảo trì gồm: các tổ chức bảo trì được thiết lập; các thủ tục ghi nhận và đánh giá được miêu tả; và một loạt thứ tự chuẩn của các bước cho mỗi yêu cầu bảo trì phải được định nghĩa. Thêm vào đó, một thủ tục lưu trữ các hồ sơ cho các hoạt động bảo trì được thiết lập và bản tổng kết những tiêu chuẩn đánh giá được vạch rõ.

#### **7.3.2.1. Cơ cấu bảo trì**

Mặc dù những tổ chức bảo trì chuẩn không cần được thiết lập, nhưng sự ủy thác trách nhiệm rất là cần thiết kể cả cho các tổ chức phát triển phần mềm nhỏ. Những yêu cầu bảo trì được chuyển qua cho người kiểm soát công việc bảo trì và từ đây chuyển tiếp yêu cầu tới người quản lý hệ thống để đánh giá. người quản lý hệ thống là thành viên của nhóm nhân viên kỹ thuật. Những nhân viên này có trách nhiệm về một phần nhỏ của chương trình sản phẩm. Khi mô tả yêu cầu được đánh giá, người được ủy quyền quản lý việc thay đổi phải quyết định hành động nào được thực hiện tiếp.

Cơ cấu được miêu tả ở trên phục vụ cho việc thiết lập phạm vi trách nhiệm đối với công việc bảo trì. Người kiểm soát và người ủy quyền quản lý việc thay đổi có thể là một người hay là một nhóm quản lý và chuyên gia kỹ thuật cao cấp.

#### **7.3.2.2. Báo cáo**

Tất cả các yêu cầu về việc bảo trì phần mềm cần được trình bày theo một tiêu chuẩn. Người phát triển phần mềm thường cung cấp một đơn yêu cầu bảo trì còn được gọi là báo cáo các lỗi phần mềm. Báo cáo này được người sử dụng điền vào khi yêu cầu công việc bảo trì. Nếu xuất hiện một lỗi, bản mô tả đầy đủ tình huống dẫn đến lỗi bao gồm dữ liệu, đoạn chương trình và các yêu cầu khác phải được điền đầy đủ vào

bản báo cáo. Nếu yêu cầu bảo trì là bảo trì tiếp hợp hay bảo trì hoàn thiện thì một yêu cầu chi tiết sẽ được thảo ra. Đơn yêu cầu bảo trì sẽ được người kiểm soát bảo trì và người quản lý hệ thống xem xét như phần trước đã nêu.

Đơn yêu cầu bảo trì được thiết lập từ bên ngoài và được coi như một cơ sở để đề ra kế hoạch của công việc bảo trì. Ngoài ra trong nội bộ của cơ quan phần mềm một báo cáo thay đổi phần mềm cũng được tạo ra. Nó chỉ ra: các công sức đòi hỏi để thỏa mãn một đơn yêu cầu bảo trì, trạng thái ban đầu của yêu cầu sửa đổi, mức ưu tiên của yêu cầu, các dữ liệu cần cho việc sửa đổi,...

#### **7.3.2.3. Lưu giữ các hồ sơ**

Thường chúng ta không có đầy đủ các hồ sơ cho tất cả các giai đoạn trong chu kỳ sống của một phần mềm. Thêm nữa không có các hồ sơ về việc bảo trì phần mềm. Do đó chúng ta thường khó có thể tiến hành các công việc bảo trì có hiệu quả, không có khả năng xác định tính chất của các chương trình và khó xác định được giá bảo trì phần mềm.

Các thông tin cần phải lưu giữ trong hồ sơ bảo trì thường:

- Dấu hiệu nhận biết chương trình.
- Số lượng các câu lệnh trong chương trình nguồn.
- Số lượng các lệnh mã máy.
- Ngôn ngữ lập trình được sử dụng.
- Ngày cài đặt chương trình.
- Số các chương trình chạy từ khi cài đặt.
- Số các lỗi xử lý xảy ra.
- Mức và dấu hiệu thay đổi chương trình.
- Số các câu lệnh được thêm vào chương trình nguồn khi chương trình thay đổi.
- Số các câu lệnh được xóa khỏi chương trình nguồn khi chương trình thay đổi.
- Số giờ mỗi người sử dụng cho mỗi lần sửa đổi.
- Ngày thay đổi chương trình.
- Dấu hiệu của kỹ sư phần mềm.
- Dấu hiệu của đơn yêu cầu bảo trì.
- Kiểu bảo trì.
- Ngày bắt đầu và kết thúc bảo trì.
- Tổng số giờ của mỗi người dùng cho việc bảo trì.

#### **7.3.2.4. Xác định giá bảo trì**

Việc xác định giá trị bảo trì thường phức tạp do thiếu thông tin. Nếu tiến hành việc lưu giữ các hồ sơ có thể tiến hành một số cách đánh giá về việc thực hiện bảo trì. Theo các chuyên gia thì đánh giá về việc thực hiện bảo trì dựa vào:

- Số lượng trung bình các lỗi xử lý cho một lần chạy chương trình.
- Tổng số giờ của mỗi người dùng cho mỗi loại bảo trì.
- Số lượng trung bình các thay đổi theo chương trình, theo ngôn ngữ lập trình, theo kiểu bảo trì.

- Số giờ trung bình của mỗi người cho một dòng lệnh được thêm vào và xóa đi.
- Số giờ trung bình của mỗi người cho một ngôn ngữ lập trình.
- Thời gian trung bình cho việc bảo trì một đơn yêu cầu bảo trì.
- Tỷ lệ phần trăm của mỗi kiểu bảo trì.

### 7.3.3. Một số hiệu ứng lề của công việc bảo trì

Sửa đổi phần mềm là công việc nguy hiểm, ta thường gặp ba loại chính của hiệu ứng lề như sau:

#### 7.3.3.1. Hiệu ứng lề của việc thay đổi mã nguồn

Một thay đổi đơn giản tới một câu lệnh đơn cũng có thể đem lại một hậu quả thảm khốc. Mặc dù không phải các ảnh hưởng đều là tiêu cực, nhưng việc sửa lỗi luôn dẫn đến các vấn đề phức tạp.

Mặc dù tất cả các thay đổi mã lệnh chương trình đều có thể tạo ra lỗi, nhưng tập hợp các thay đổi sau có thể gây ra nhiều lỗi hơn.

- Một chương trình con bị xóa hay thay đổi.
- Một dòng nhãn bị xóa hay thay đổi.
- Một biến bị xóa hay thay đổi.
- Các thay đổi để tăng khả năng thực hiện.
- Việc mở và đóng file bị thay đổi.
- Các phép toán logic bị thay đổi.
- Việc thay đổi thiết kế chuyển thành các thay đổi lớn về chương trình.
- Các thay đổi ảnh hưởng đến việc chạy thử các trường hợp biên.

#### 7.3.3.2. Hiệu ứng lề của việc thay đổi dữ liệu

Trong quá trình bảo trì, việc sửa đổi thường được tiến hành đối với các phần tử riêng rẽ của cấu trúc dữ liệu. Khi dữ liệu thay đổi, việc thiết kế phần mềm sẽ không còn phù hợp với dữ liệu và lỗi có khả năng xảy ra.

Hiệu ứng lề của dữ liệu xảy ra như là kết quả của việc thay đổi cấu trúc dữ liệu. Các thay đổi dữ liệu sau đây thường gây ra lỗi:

- Định nghĩa lại các hằng số cục bộ và hằng số địa phương.
- Định nghĩa lại cấu trúc bản ghi hay cấu trúc file.
- Tăng hoặc giảm kích thước một mảng.
- Thay đổi dữ liệu tổng thể.
- Định nghĩa lại các cờ điều khiển và các con trỏ.
- Xếp lại các tham số vào ra hay tham số của chương trình con.

Hiệu ứng lề dữ liệu có thể được hạn chế bằng tài liệu thiết kế mô tả cấu trúc dữ liệu và cung cấp một lời chỉ dẫn tham khảo đến từng phần tử dữ liệu, các bản ghi, các file và các cấu trúc khác của các module phần mềm.

### 7.3.3.3. *Hiệu ứng lề của việc thay đổi tài liệu*

Việc bảo trì thường tập trung vào cấu hình phần mềm và không tập trung riêng vào việc sửa đổi mã. Sự ảnh hưởng của tài liệu xảy ra khi thay đổi chương trình nguồn mà không thay đổi tài liệu thiết kế và tài liệu hướng dẫn sử dụng.

Bất cứ lúc nào có thay đổi về luồng dữ liệu, cấu trúc phần mềm, các thủ tục hay bất cứ cái gì có liên quan, tài liệu kỹ thuật phải được cập nhật. Tài liệu về thiết kế phần ánh không đúng trạng thái hiện tại của phần mềm có lẽ còn tồi tệ hơn không có tài liệu. Hiệu ứng lề xảy ra trong các lần bảo trì sau đó, khi việc nghiên cứu không kỹ các tài liệu kỹ thuật, dẫn tới sự đánh giá sai về các đặc tính của phần mềm. Đối với người sử dụng, phần mềm tốt chỉ khi có tài liệu hướng dẫn sử dụng chúng.

Các hiệu ứng lề trong tài liệu có thể được giảm về căn bản nếu toàn bộ cấu hình được xem xét trước khi phát hành phiên bản phần mềm tiếp sau. Thực tế một vài yêu cầu bảo hành có thể đòi hỏi không được thay đổi thiết kế của phần mềm hoặc mã chương trình, mà chỉ cần chỉ ra sự thiếu rõ ràng trong tài liệu của người sử dụng. Trong những trường hợp như vậy nỗ lực bảo trì tập trung vào tài liệu.

## 7.4. MỘT SỐ HÌNH THỨC BẢO TRÌ PHẦN MỀM

### 7.4.1. 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 nghèo 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 (về 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.

#### 7.4.2. 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.

#### 7.4.3. 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.

#### 7.4.4. 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 người 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.

### 7.5. QUẢN LÝ THAY ĐỔI PHẦN MỀM

Các ứng dụng thường xuyên phải thiết kế lại do sự phân công của một nhóm quản lý mới, dự án vượt quá ngân sách, ứng dụng chậm và có nhiều lỗi, và sự thiếu tin tưởng của chủ sử dụng về việc các kỹ sư phần mềm hiểu rõ các yêu cầu của mình,...

Các thay đổi có thể là các yêu cầu, thiết kế, chương trình, giao diện, phần cứng hoặc phần mềm phải mua. Phần lớn các thay đổi bắt nguồn từ bên trong tổ chức phát triển ứng dụng, nhưng cũng có thể được kích hoạt từ các tác nhân bên ngoài, ví dụ như thay đổi về luật.

Việc quản lý thay đổi ứng dụng giúp cho nhóm triển khai bỏ qua những ý thích chợt nảy ra của người sử dụng trong khi vẫn cho phép thực hiện các yêu cầu hợp lý.

#### 7.5.1. Các thủ tục quản lý thay đổi

Quản lý điều khiển thay đổi có hiệu lực từ khi sản phẩm đầu tiên được chấp nhận là hoàn thiện cho đến khi dự án kết thúc. Trước tiên, các sản phẩm công việc cơ sở được tạo lập để đưa vào quản lý. Một sản phẩm công việc cơ sở là một sản phẩm được coi là hoàn thiện và là cơ sở cho các công việc hiện tại khác của nhóm triển khai

dự án. Ví dụ như, một tài liệu cơ sở là bản quy định yêu cầu chức năng sau khi nó đã được chấp nhận bởi người sử dụng.

Dưới đây là ví dụ một quá trình của các thao tác yêu cầu thay đổi của một đặc tả chức năng:

- Tạo yêu cầu mở.
- Khai báo file tác động
- Phê chuẩn file về thời gian và chi phí do chủ, người sử dụng ký.
- Hoàn thiện danh sách và kiểm soát về thay đổi của người điều hành dự án.
- File tài liệu liên quan đến thay đổi. Nếu tài liệu hoặc chương trình bị thay đổi, thì xác định ngày và các mục cập nhật đã hoàn thiện. Nếu các thủ tục hoặc thử nghiệm bị thay đổi, xác định các ngày mà việc sửa đổi xảy ra.
- Mẫu yêu cầu đóng file được chủ/người sử dụng thông qua.
- Tóm tắt các ngày tháng, quá trình và chi phí.

Trước tiên, tài liệu cơ sở được giữ nguyên, sau đó thêm vào các yêu cầu thay đổi. Khi quy định chức năng được cập nhật để điều tiết thay đổi, nó được đóng băng lại và công việc lại tiếp tục. Ba yêu cầu trước có thể được thêm vào ứng dụng nếu chúng không làm thay đổi ứng dụng nhiều. Chúng cũng có thể bị bỏ qua cho đến sau khi ứng dụng đã được thực hiện.

Các thay đổi có thể phân loại theo một số cách.

- Thứ nhất, chúng được phân theo kiểu như loại bỏ lỗi, cải tiến thực hiện hoặc thay đổi chức năng.
- Thứ hai, thay đổi phân loại thành yêu cầu và lựa chọn.
- Thứ ba, phân theo độ ưu tiên như khẩn cấp, lệnh với một ngày kết thúc yêu cầu, lệnh với ngày bắt đầu yêu cầu hoặc ưu tiên thấp.
- Thông thường, kiểu loại bỏ lỗi là khẩn cấp theo yêu cầu, trong khi thay đổi chức năng là bảo dưỡng lệnh theo yêu cầu, và cải tiến thực hiện là lựa chọn và có thể không có ưu tiên.

Việc biết được loại yêu cầu thay đổi quyết định xem liệu nó có cần phải chịu điều khiển thay đổi hay không. Các thay đổi khẩn cấp thường phá vỡ thủ tục điều khiển thay đổi do các công việc thực hiện tuần tự nhưng chúng lại được tài liệu hoá sau khi thay đổi đã kết thúc. Tất cả các loại thay đổi khác đều phải tuân theo các điều khiển thay đổi. Ví dụ như thay đổi về yêu cầu chức năng có thể xảy ra bất cứ lúc nào, nhưng khi quy định yêu cầu chức năng được thông qua thì nó đóng băng cho đến khi ứng dụng hoạt động. Các thay đổi phải chịu sự điều khiển thay đổi: chúng được thêm vào danh sách yêu cầu thay đổi để xem xét trong tương lai trừ khi đó là một thiết kế khẩn cấp.

Một thủ tục điều khiển thay đổi yêu cầu người sử dụng phải đệ trình một lời yêu cầu thay đổi chính thức cho người điều hành dự án:

- Người sử dụng gửi cho người điều hành dự án và người chủ một mẫu yêu cầu thay đổi.

- Người điều hành dự án và kỹ sư phần mềm triển khai một khai báo tự động. Vào lúc đó, danh sách kiểm soát của người điều hành dự án được dùng để xác định tất cả các hoạt động và thay đổi công việc có liên quan tới yêu cầu.
- Yêu cầu thay đổi được thảo luận với chủ sử dụng để vạch ra các thay đổi về ưu tiên, tiến trình và chi phí.
- Thoả thuận được chính thức hoá và chủ sử dụng thông qua thay đổi về tiến trình và chi phí.
- Sử dụng khai báo tác động, ứng dụng và tất cả các tài liệu có liên quan được thay đổi.
- Thực hiện thay đổi: khi các nhiệm vụ hoàn thành, xoá nhiệm vụ trong danh sách kiểm soát của người điều hành dự án.
- Chủ sử dụng thông qua việc đóng yêu cầu và yêu cầu được đóng.
- Người điều hành dự án và kỹ sư phần mềm định nghĩa các tác động tiến trình và chi phí của thay đổi. Sau đó các thay đổi được bàn bạc với người sử dụng. Dựa trên thương lượng với người sử dụng, thay đổi được gán một ưu tiên hoạt động, và chi phí và tiến trình được thay đổi.

Yêu cầu, ngày dự định hoạt động, thay đổi tiến trình và tăng chi phí được thêm vào một file quá trình dự án. Các thay đổi có thể được quản lý bởi một nhân viên điều khiển thay đổi, là một người có nhiệm vụ bảo dưỡng quá trình dự án và các bản ghi điều khiển thay đổi, và hàng tháng in ra một bản báo cáo điều khiển thay đổi. Một file điều khiển thay đổi chứa tất cả các yêu cầu, thư từ và tài liệu về các thay đổi. Một yêu cầu thay đổi mới có thể được tạo ra khi yêu cầu được đưa ra và một số lượng thay đổi được gán. Yêu cầu thay đổi mới nằm trong file cho đến khi yêu cầu được hoàn thành, đóng và được báo cáo.

Khi thay đổi được thực hiện, các mục có ảnh hưởng được cập nhật, bao gồm tư liệu tương ứng, mã,... Một danh sách kiểm soát của người điều hành dự án được dùng để loại bỏ các hoạt động đã được yêu cầu. Tài liệu mới được nhân viên điều khiển thay đổi sắp xếp và phân phối nó cho những người có quan tâm. Ngày hoàn thành thay đổi được đưa vào file điều khiển thay đổi. Thay đổi được xác định khi được đóng trong báo cáo tình trạng tới và yêu cầu mở được chuyển từ file điều khiển thay đổi sang.

Dựa trên tổ chức này, người điều hành hệ thống có thể theo dõi các yêu cầu thay đổi của dự án để nhận biết sự thành công trong nhóm các yêu cầu. Chi phí thay đổi chung của một năm thường được sử dụng như là một chỉ tiêu để chỉ ra xem ứng dụng đang có triển vọng hay cần vớt vớt hay cần công nghệ hoá lại. Trong những trường hợp này, cả chi phí và số lượng các yêu cầu thay đổi đều được theo dõi thông qua quá trình điều khiển thay đổi. Các báo cáo tổng kết bởi dự án thay đổi trong một thời kỳ nhất định, hoặc so sánh theo thời kỳ có thể được triển khai.

### **7.5.2. Ghi quyết định theo thời gian**

Khi bắt đầu một dự án, người điều hành dự án và kỹ sư phần mềm quyết định sử dụng các công cụ để lưu trữ quá trình quyết định. Có nghĩa là có thể dùng công cụ điện tử hoặc một phiên bản viết tay và các quyết định được duy trì dưới dạng văn bản. Với công cụ điện tử, các bản sao điện tử được lưu trữ. Với công cụ ghi tay, phiên bản cũ được cập nhật và đổi tên khi một tài liệu thay đổi. Ví dụ như, các quy định chức

năng của công ty ABC có thể được đặt tên là ABCFS-mmddyy, trong đó ABC là công ty, FS là viết tắt của quy định chức năng (Functional Specification) và mmddyy là ngày tháng. Phần ngày tháng của tên sẽ thay đổi do bất kỳ một thay đổi quan trọng nào của tài liệu. Thủ tục quản lý thay đổi sẽ được nói đến trong phần tiếp theo.

### 7.5.3. Quản lý thay đổi tài liệu

Các thay đổi tài liệu có thể được xác định bởi một bảng nội dung các thay đổi tại đầu mỗi tài liệu. Bảng nội dung các thay đổi bao gồm ngày hiệu lực, các phân bị ảnh hưởng của tài liệu và một tóm tắt về thay đổi. Mục đích của bảng nội dung các thay đổi là để tóm tắt tất cả các thay đổi cho người đọc.

Các thay đổi nên được đánh dấu đỏ trong văn bản để xác định được bộ phận thay đổi. Nếu nội dung cũ là quan trọng thì nó có thể được đưa vào trong chú ý, được ghi ngày tháng, và được dán nhãn là phiên bản trước. Cần nhớ rằng bạn cũng phải giữ tài liệu phiên bản cũ để dùng cho quá trình phát triển.

#### Câu hỏi

1. Chuyển giao phần mềm? Các hoạt động chính của chuyển giao phần mềm?
2. Bảo trì phần mềm là gì?
3. Các hoạt động nào trong bảo trì phần mềm là chính yếu?
4. Các khó khăn của hoạt động bảo trì?
5. Một số hình thức bảo trì phần mềm?



## TÀI LIỆU THAM KHẢO

1. Ngô Trung Việt, Kỹ nghệ phần mềm - bản dịch, Nhà xuất bản Giáo dục, 1999.
2. Đoàn Văn Ban, Phân tích thiết kế và lập trình hướng đối tượng, Trung tâm Khoa học Tự nhiên và Công nghệ Quốc gia, 1996.
3. Lê Đức Trung, Công nghệ phần mềm, Nhà xuất bản Khoa học và Kỹ thuật, 2002.
4. Khoa Công nghệ Thông tin - Đại học Khoa học Tự nhiên Hà Nội, Bài giảng về nhập môn công trình học phần mềm, Hà Nội, 1997.
5. Lê Đức Trung, Những viên ngọc trong kỹ thuật lập trình - bản dịch, Trung tâm Tin học và Điện tử Phương Đông, 1992 .
6. Roger S. Pressman Ph.D, Software engineering a practitioner's - 5<sup>th</sup>, McGraw-Hill book Co.-Singapore, 2001.
7. Yourdon, E. Software reuse, Application development strategies, Vol. 6, No. 12, December 1994, pp.1-16.
8. Sommerville I., Software engineering - 4<sup>th</sup>, Addison Wesley, 1995.
9. Eldon Wig, Industry Software engineering course - VCIT project, Intellitec Consulting Inc., Hà Nội 10/1999.